electronics

MDPI

Article

An Effective Federated Recommendation Framework with Differential Privacy

Zihang Xu [1,2], Chiawei Chu [1,*] and Shiyang Song [3]

[1] Faculty of Data Science, City University of Macau, Macau 999078, China
[2] Aliyun College of Data Application, Zhuhai College of Science and Technology, Zhuhai 519041, China
[3] Faculty of Innovation Engineering, Macau University of Science and Technology, Macau 999078, China
* Correspondence: cwchu@cityu.edu.mo

Abstract: This paper proposes a novel federated recommendation framework that incorporates differential privacy to safeguard user privacy without compromising on the accuracy of recommendations. Unlike conventional recommendation systems that centralize user data, leading to potential privacy breaches, our framework ensures that user data remain on local devices. It leverages a federated learning approach, where a global model is trained across multiple devices without exchanging raw data. To enhance privacy protection, we integrate a specially designed differential privacy algorithm that adds carefully calibrated noise to the aggregated data updates. This mechanism ensures that the global model cannot be exploited to infer individual user information. We evaluate our framework on two real-world datasets, one from the e-commerce sector and another from the multimedia content recommendation domain. The results exhibit that our framework achieves competitive recommendation accuracy compared to traditional centralized approaches, with minimal loss in precision and recall metrics, while significantly enhancing user privacy. Our work stands as a testament to the feasibility of creating recommendation systems that do not have to choose between privacy and performance, paving the way for more ethical AI applications in sensitive domains.

Keywords: privacy protection; machine learning; federated learning; recommendation system

Citation: Xu, Z.; Chu, C.; Song, S. An Effective Federated Recommendation Framework with Differential Privacy. *Electronics* **2024**, *13*, 1589. https://doi.org/10.3390/electronics13081589

Academic Editor: George Angelos Papadopoulos

Received: 9 March 2024
Revised: 11 April 2024
Accepted: 15 April 2024
Published: 22 April 2024

1. Introduction

In the era of digital transformation, recommendation systems have become indispensable tools across a myriad of platforms, enhancing user experiences by personalizing content, products, and services. These systems leverage vast amounts of data to predict and suggest items that are most likely to be of interest to users based on their past behavior, preferences, and interactions. While the utility of recommendation systems is undeniable, their reliance on personal and sensitive user data has raised significant privacy concerns. The aggregation and analysis of user data, if not properly managed, can lead to unintended privacy breaches and the exploitation of personal information.

Federated learning emerges as a promising solution to this privacy conundrum. It is a distributed machine learning approach that enables model training across multiple decentralized devices or servers holding local data samples, without exchanging them. This paradigm not only helps in safeguarding user privacy by keeping the data localized but also utilizes collaborative learning to improve the model's performance. However, federated learning itself does not guarantee absolute privacy, especially against sophisticated inference attacks that can deduce individual data points from shared model updates.

Enter differential privacy, a mathematical framework designed to provide strong privacy guarantees by adding randomness to the data or algorithms, thereby making it difficult to infer information about any individual within a dataset. Integrating differential privacy into federated learning, especially in the context of recommendation systems, presents a viable pathway to achieving a balance between personalization and privacy.

Despite its potential, the application of differential privacy in federated recommendation systems poses unique challenges. The primary concern revolves around the trade-off between privacy and utility. Excessive noise can protect privacy but may degrade the quality of recommendations, whereas minimal noise may compromise user privacy for the sake of accuracy. Moreover, the decentralized nature of federated learning complicates the uniform application and management of privacy-preserving mechanisms across different nodes.

This paper seeks to address these challenges by proposing a novel architecture for a differentially private and effective federated recommendation system. Our research aims to explore innovative approaches to integrate differential privacy into federated learning frameworks without significantly compromising the system's recommendation capabilities. We delve into optimizing the privacy–accuracy trade-off, ensuring that the system remains robust against privacy threats while maintaining high levels of personalization and user satisfaction.

Through this work, we contribute to the burgeoning field of privacy-preserving machine learning by offering insights and practical solutions that can be applied to federated recommendation systems. Our goal is to pave the way for the development of recommendation systems that not only respect and protect user privacy but also maintain the quality and effectiveness that users have come to expect.

To summarize, this paper makes the following contributions:

- We introduce a novel framework that combines federated learning with differential privacy mechanisms. This approach decentralizes data processing to keep user data on local devices, enhancing privacy without significantly compromising the accuracy of recommendations.
- This paper details the implementation of a specifically designed differential privacy algorithm that adds calibrated noise to model updates. This ensures the privacy of individual user data in the aggregated model, offering a robust privacy guarantee while maintaining the utility of the recommendation system.
- Through extensive experiments on widely recognized datasets like MovieLens and Amazon Product Reviews, this paper validates the effectiveness of the proposed framework. It demonstrates that the system achieves competitive recommendation accuracy and privacy preservation, establishing a practical balance between user privacy and personalized recommendation quality.

This paper is organized as follows. Section 2 introduces related work. Section 3 presents preliminaries of our work. Section 4 gives our method to achieve a privacy-preserving recommendation system in federated learning. Section 5 presents our empirical evaluation results and Section 6 concludes our work.

2. Related Work

2.1. Recommendation System

Recommendation systems are pivotal in navigating the vast amount of content available in today's digital age, enhancing user experience by personalizing content delivery based on user preferences and behaviors. The evolution of recommendation systems has been marked by significant advancements, from basic collaborative filtering algorithms to complex deep learning-based models [1,2].

Early recommendation systems relied heavily on collaborative filtering techniques [3,4], which make predictions about the interests of a user by collecting preferences from many users [3]. This approach was further refined through matrix factorization techniques, which decompose the user–item interaction matrix into lower-dimensional matrices, capturing latent factors associated with users and items [5,6].

With the advent of deep learning, recommendation systems have seen a paradigm shift. Neural network-based models, such as the Neural Collaborative Filtering framework [7,8], have demonstrated superior performance in capturing complex user–item interactions.

Moreover, the incorporation of contextual information, such as time and location, into recommendation algorithms has further improved their accuracy and relevance [9,10].

However, the effectiveness of these systems often comes at the cost of user privacy, as they require access to sensitive user data [11]. This has led to an increasing interest in developing privacy-preserving recommendation systems. Differential privacy has emerged as a promising approach to safeguard user privacy in recommendation systems by adding noise to the data or the algorithm's outputs, ensuring that individual user data cannot be inferred [12,13].

2.2. Federated Learning

The concept of federated learning has emerged as a groundbreaking approach to address privacy concerns in machine learning and artificial intelligence [14,15]. Initiated by McMahan et al. [16], federated learning enables model training across multiple decentralized devices or servers holding local data samples, without needing to exchange them. This approach not only helps in preserving the privacy of user data but also reduces the communication overhead associated with traditional centralized learning paradigms [17,18].

The implementation of FL has been explored in various domains, ranging from mobile keyboard prediction to healthcare and financial services, demonstrating its versatility and effectiveness in privacy-preserving data analysis [19,20]. Konečný et al. [21,22] further extended the federated learning framework by introducing optimization strategies that enhance model convergence speed. A number of recent studies also focus on reducing the required communication rounds between the clients and the central server [23,24].

In the context of recommendation systems, the adoption of federated learning is relatively recent. Pioneering studies [25,26] first applied federated learning to collaborative filtering, illustrating its potential in creating personalized recommendation systems that respect user privacy. Their work demonstrated how FL could be adapted to recommendation systems, ensuring data remain localized while achieving comparable accuracy to traditional centralized approaches.

Moreover, the integration of differential privacy within federated learning frameworks, as proposed by [27–29], has set a new standard for privacy-preserving machine learning. By adding noise to the model updates in a controlled mannver, differential privacy ensures that the trained model does not reveal sensitive information about the data on any specific client's device [30,31].

We introduce a pioneering federated recommendation system that seamlessly integrates differential privacy to ensure user data privacy while maintaining high recommendation accuracy. Distinguishing itself from existing approaches, this work leverages federated learning to process data locally on devices, thus mitigating central data collection risks. A key innovation is the implementation of a differential privacy algorithm that injects noise into aggregated data updates, safeguarding against the inference of individual user information. Through rigorous evaluation on real-world datasets from the e-commerce and multimedia sectors, the framework demonstrates a practical equilibrium between privacy protection and the quality of recommendations.

Our work differentiates itself from previous works by introducing a novel federated recommendation framework that integrates differential privacy directly into federated learning, enhancing privacy without compromising recommendation accuracy. Unlike prior approaches that primarily focused on theoretical models or specific aspects of privacy, this work presents a comprehensive solution with a custom differential privacy algorithm and empirically validates its effectiveness using real-world datasets like MovieLens and Amazon Product Reviews. It effectively addresses the critical challenge of optimizing the privacy–accuracy trade-off, offering both practical implementations and demonstrating superior performance in privacy preservation compared to earlier methods.

3. Preliminaries

3.1. Differential Privacy

Differential Privacy is a framework designed to ensure the privacy of individuals' data within a dataset while allowing for the analysis and extraction of meaningful insights. It provides a quantifiable way of protecting individuals' privacy by adding randomness to the data or the outcomes of algorithms applied to the data. The core idea is to make it mathematically improbable for anyone—even those with access to the output of a differentially private mechanism—to infer the presence or absence of any individual's data in the dataset.

A randomized mechanism $\mathcal{M}$ satisfies ϵ-differential privacy (ϵ-DP) if for all datasets D and D' that differ by at most one element (these are called adjacent datasets) and for all S within the range of $\mathcal{M}$, the following inequality holds:

$$\Pr[\mathcal{M}(D) \in S] \leq e^{\epsilon} \times \Pr[\mathcal{M}(D') \in S]$$

Here, ϵ is a non-negative parameter that quantifies the privacy loss, with lower values indicating stronger privacy.

The sensitivity of a function f, denoted as Δf, is a measure of how much the output of f can change by altering a single individual's data in the dataset. For a function $f : \mathcal{D} \to \mathbb{R}^d$, the sensitivity is defined as

$$\Delta f = \max_{D,D'} \|f(D) - f(D')\|$$

where the maximum is taken over all pairs of adjacent datasets D and D'.

The Laplace Mechanism adds noise drawn from the Laplace distribution to the output of a function f to achieve differential privacy. The scale of the Laplace noise is determined by the sensitivity of f and the desired privacy level ϵ. The probability density function of the Laplace distribution with mean 0 and scale b is given by

$$\Pr(x|b) = \frac{1}{2b} \exp\left(-\frac{|x|}{b}\right)$$

To achieve ϵ-differential privacy, the scale b is set as $\Delta f/\epsilon$.

The Exponential Mechanism is used for differential privacy in cases where the output is not numerical. It selects an output with a probability proportional to the exponential of the privacy loss parameter ϵ, divided by twice the sensitivity, times a utility function $u(D, r)$ that measures the usefulness of result r. The probability of selecting a particular outcome r is

$$\Pr(r) \propto \exp\left(\frac{\epsilon \cdot u(D, r)}{2\Delta u}\right)$$

where Δu is the sensitivity of the utility function u.

Differential privacy also provides composition theorems that describe how the privacy parameter ϵ accumulates when multiple differentially private mechanisms are applied. Sequential Composition states that if a series of k mechanisms, each providing ϵ_i-differential privacy, are applied to a dataset, the total privacy loss is bounded by $\sum_{i=1}^{k} \epsilon_i$. Parallel Composition asserts that if multiple mechanisms, each providing ϵ_i-differential privacy, operate on disjointed subsets of the dataset, the overall privacy guarantee is governed by $\max(\epsilon_i)$.

The concept of a privacy budget is crucial in differential privacy. It refers to the total amount of privacy loss (ϵ) that is allowable over the course of multiple queries or analyses. It is essential to manage and track the privacy budget to ensure that the cumulative privacy loss does not exceed a predefined threshold. By employing these principles and mechanisms, differential privacy ensures that the privacy of individuals in a dataset is protected while still allowing for valuable data analysis. This balance is crucial in

a wide range of applications, from statistical analysis to machine learning, where the use of personal data is often necessary to generate insights and make decisions.

3.2. Recommendation Systems

Recommendation systems are pivotal in personalizing user experiences across various digital platforms, using algorithms to suggest items, such as products, movies, or articles, that users are likely to be interested in. These systems can broadly be categorized into three types: content-based filtering, collaborative filtering, and hybrid systems. Each employs distinct methodologies to analyze and predict user preferences.

Content-based filtering relies on item features and a profile of the user's preferences. The recommendation score, s, for a user u and item i can be calculated as

$$s(u,i) = \mathbf{p}_u \cdot \mathbf{q}_i^T$$

where $\mathbf{p}_u$ is the preference vector of user u, and $\mathbf{q}_i$ is the feature vector of item i.

Collaborative filtering makes recommendations based on the past behavior of users in the system, without requiring item metadata. User-based collaborative filtering computes the similarity between users, for instance, using the Pearson correlation coefficient:

$$sim(u,v) = \frac{\sum_{i \in I}(r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)}{\sqrt{\sum_{i \in I}(r_{ui} - \bar{r}_u)^2}\sqrt{\sum_{i \in I}(r_{vi} - \bar{r}_v)^2}}$$

where r_{ui} and r_{vi} are the ratings of users u and v for item i, respectively, $\bar{r}_u$ and $\bar{r}_v$ are the average ratings of users u and v, and I is the set of items rated by both users.

Item-based collaborative filtering computes similarities between items, often using the cosine similarity:

$$sim(i,j) = \frac{\mathbf{r}_i \cdot \mathbf{r}_j}{\|\mathbf{r}_i\| \|\mathbf{r}_j\|}$$

where $\mathbf{r}_i$ and $\mathbf{r}_j$ are the rating vectors of items i and j, respectively.

Matrix factorization techniques, such as Singular Value Decomposition (SVD), are commonly used in collaborative filtering to predict unknown ratings:

$$\hat{r}_{ui} = \mu + b_u + b_i + \mathbf{p}_u \cdot \mathbf{q}_i^T$$

where μ is the global average rating, b_u and b_i are the user and item bias terms, and $\mathbf{p}_u$ and $\mathbf{q}_i$ are the latent factor vectors for user u and item i.

Hybrid systems combine collaborative and content-based methods to improve recommendation quality, addressing limitations such as the cold start problem and data sparsity. The combined score might be computed as a weighted sum:

$$s(u,i) = \alpha \cdot s_{\text{CBF}}(u,i) + (1 - \alpha) \cdot s_{\text{CF}}(u,i)$$

where s_{CBF} and s_{CF} are the scores from content-based and collaborative filtering, respectively, and α is the weight that balances the two.

To evaluate the performance of recommendation systems, metrics such as Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) are used:

$$\text{MAE} = \frac{\sum_{(u,i) \in T} |r_{ui} - \hat{r}_{ui}|}{|T|}$$

$$\text{RMSE} = \sqrt{\frac{\sum_{(u,i) \in T}(r_{ui} - \hat{r}_{ui})^2}{|T|}}$$

where T is the set of user–item pairs in the test set, r_{ui} is the actual rating, and $\hat{r}_{ui}$ is the predicted rating.

By employing these methodologies and metrics, recommendation systems aim to enhance user satisfaction and engagement, tailoring content to individual preferences and interests.

4. Method

This section outlines our novel method for a privacy-preserving recommendation system leveraging federated learning and differential privacy. The system model enables private, client-specific model updates using calibrated Gaussian noise to maintain a balance between user privacy and recommendation accuracy. The recommendation algorithm combines collaborative filtering and deep learning within this privacy-aware federated framework, optimizing for personalization and accuracy. Through advanced aggregation techniques and privacy budget management, the system effectively mitigates privacy risks while preserving the quality of recommendations. A general pipeline of our proposal is given in Figure 1.

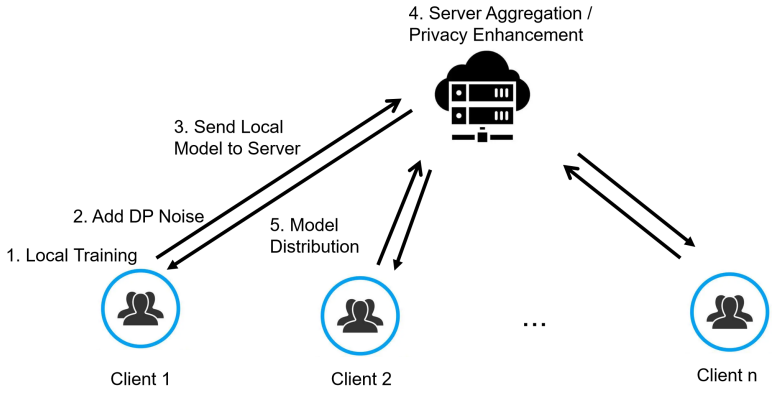

Figure 1. A general pipeline of our proposal.

4.1. System Model

Our proposed system model intricately weaves differential privacy mechanisms within a federated learning framework to construct a recommendation system that not only respects user privacy but also delivers personalized content with high precision. Operating over a distributed network, this model empowers a collaborative yet private learning environment, significantly mitigating the privacy and security concerns associated with traditional centralized systems.

The cornerstone of our model is the local model update, a process that enables individual clients to contribute to the collective learning objective without exposing their private data:

$$\Delta w_i = \nabla L(w, D_i) + \eta_i, \tag{1}$$

where Δw_i denotes the update computed by client i, derived from the gradient of the loss function L with respect to the model weights w over the local dataset D_i. To safeguard privacy, noise η_i is introduced, adhering to the principles of differential privacy. This noise is meticulously calibrated, following a Gaussian distribution $\mathcal{N}(0, \sigma^2 I)$, to balance the trade-off between privacy and the utility of the aggregated model:

$$\eta_i \sim \mathcal{N}(0, \sigma^2 I). \tag{2}$$

Secure aggregation is the next pivotal step, where updates from all participating clients are amalgamated to refine the global model. This process is encapsulated in the following equation:

$$w_{\text{global}}^{(t+1)} = w_{\text{global}}^{(t)} + \frac{1}{N} \sum_{i=1}^{N} \Delta w_i, \tag{3}$$

ensuring that the synthesis of updates enhances the model without compromising the confidentiality of individual contributions.

Client participation is inherently probabilistic, governed by their respective privacy budgets, ensuring a democratic and equitable learning environment:

$$P_i = \frac{\exp(\epsilon_i)}{\sum_{j=1}^{N} \exp(\epsilon_j)}. \tag{4}$$

At the heart of personalized recommendations lies the recommendation score, a measure that determines the relevance of items to users based on the insights gleaned from the global model:

$$S(u, i) = f(w_{\text{global}}, u, i), \tag{5}$$

where f denotes a sophisticated function that meticulously analyzes user preferences and item characteristics to generate precise recommendations.

Model convergence is a testament to the efficacy of the learning process, marked by the stabilization of the global model across successive iterations:

$$\lim_{t \to \infty} \| w_{\text{global}}^{(t+1)} - w_{\text{global}}^{(t)} \| = 0. \tag{6}$$

The differential privacy guarantee assures that the introduction or alteration of a single data point negligibly affects the outcome, which is a foundational pillar of privacy protection:

$$\Pr[\mathcal{M}(D) \in S] \leq \exp(\epsilon) \cdot \Pr[\mathcal{M}(D') \in S] + \delta. \tag{7}$$

The update frequency of the global model inversely correlates with the learning rate, highlighting the dynamic nature of the learning process:

$$\tau = \frac{1}{\gamma}. \tag{8}$$

Adjusting the learning rate is crucial for optimizing the convergence rate, ensuring the model rapidly adapts to new data insights:

$$\gamma^{(t+1)} = \beta \gamma^{(t)}. \tag{9}$$

4.2. Privacy Mechanism

Differential privacy provides a quantifiable guarantee that the privacy of an individual's data is maintained, even in the aggregate. This is achieved by introducing a calibrated amount of noise to the data or model updates, effectively obscuring individual contributions. Differential privacy is applied by adding noise to the model updates computed on client devices before they are aggregated by the central server. The noise-added update can be represented as

$$\Delta w_i' = \Delta w_i + \mathcal{N}(0, \sigma^2 \mathbf{I}), \tag{10}$$

where $\Delta w_i'$ denotes the privacy-preserving update from client i, and σ^2 is the variance in the Gaussian noise, which is a critical parameter in balancing privacy and utility. The calibration of noise is based on the sensitivity of the function computing the updates and the desired privacy level, formalized as follows:

$$\sigma = \frac{\Delta f \sqrt{2 \ln(1.25/\delta)}}{\epsilon}, \tag{11}$$

where Δf is the sensitivity of the function, ϵ is the privacy loss parameter, and δ is the probability of exceeding this privacy loss. This equation ensures the noise is calibrated to provide (ϵ, δ)-differential privacy.

The privacy budget, ϵ, is allocated to each client to manage the total privacy loss over multiple interactions. The allocation is adjusted dynamically based on the client's activity level and the overall system's privacy requirements:

$$\epsilon_{\text{total}} = \sum_{i=1}^{N} \epsilon_i, \tag{12}$$

ensuring that the cumulative privacy budget across all clients does not exceed the system's predefined threshold.

To mitigate the impact of noise on the aggregated model, we employ advanced aggregation techniques that enhance the utility of the noisy updates. One such technique involves weighted averaging, where updates are weighted based on their variance:

$$w'_{\text{global}} = \frac{\sum_{i=1}^{N} w'_i/\sigma_i^2}{\sum_{i=1}^{N} 1/\sigma_i^2}, \tag{13}$$

where w'_i is the noise-added update from client i, and σ_i^2 is the variance in the noise added to client i's update. To ensure the quality of recommendations, algorithms are designed to be robust against the noise introduced for privacy preservation. This involves adjusting the learning algorithm to account for the expected noise, thereby minimizing its impact on recommendation quality:

$$\hat{S}(u,i) = S(u,i) + \alpha \cdot \text{Err}(\sigma), \tag{14}$$

where $\hat{S}(u,i)$ is the adjusted score for recommending item i to user u, $S(u,i)$ is the original recommendation score, α is a scaling factor, and $\text{Err}(\sigma)$ represents the error introduced by the noise, which is a function of σ.

By integrating these advanced techniques and carefully managing the privacy budget, our federated recommendation system achieves a balance between preserving user privacy and maintaining the efficacy of personalized recommendations.

4.3. Recommendation Algorithm

The recommendation algorithm at the heart of our federated recommendation system leverages collaborative filtering and deep learning techniques, adapted to operate within the constraints of differential privacy and federated learning. This subsection outlines the algorithm's core components, including model training, personalization strategies, and the integration of privacy-preserving mechanisms.

The foundation of our recommendation algorithm is the federated averaging (FedAvg) process, which aggregates model updates from multiple clients to improve the global recommendation model. The training process for each client's local model can be expressed as

$$L_i(w) = \frac{1}{|D_i|} \sum_{(x,y)\in D_i} \ell(f(w;x),y) + \lambda\|\omega\|^2, \tag{15}$$

where $L_i(w)$ is the loss function for client i's model given the weight w, D_i is the local dataset of user–item interactions, ℓ is a loss metric (e.g., mean squared error for rating prediction), $f(w;x)$ is the predictive function of the model, y represents the true preference values (e.g., ratings), and $\lambda\|\omega\|^2$ is a regularization term to prevent overfitting. After local training, the model updates are aggregated to update the global model:

$$w_{\text{global}} \leftarrow w_{\text{global}} + \eta \sum_{i=1}^{N} \frac{|D_i|}{D} \Delta w_i, \tag{16}$$

where η is the learning rate, N is the number of clients, $|D_i|$ is the size of the local dataset, D is the total size of all datasets, and Δw_i is the update from client i. To personalize

recommendations, the global model is fine-tuned on individual clients' data, allowing for tailored recommendations. The personalization step for client i can be represented as

$$w_i^* = w_{\text{global}} - \alpha \nabla L_i(w_{\text{global}}), \tag{17}$$

where w_i^* is the personalized model weight for client i, and α is the personalization learning rate. For generating recommendations, a prediction score for each user–item pair is computed using the personalized model. The score indicates the likelihood of a user's preference for an item:

$$\hat{y}_{ui} = f(w_i^*; x_{ui}), \tag{18}$$

where $\hat{y}_{ui}$ is the predicted preference score for user u and item i, and x_{ui} represents the user–item interaction features.

Based on these scores, the items are ranked, and the top-N recommendations are selected for each user:

$$R_u = \text{rank}(\hat{y}_{ui})_{i=1}^{M}, \tag{19}$$

where R_u is the ranked list of recommendations for user u, and M is the total number of items. To ensure that the recommendation process adheres to differential privacy standards, noise is added during the aggregation of model updates and the personalization step. In this way, the recommendations are generated in a manner that respects user privacy while maintaining the effectiveness of the system.

5. Experiments

To validate the efficacy and privacy guarantees of our federated recommendation system, we conducted a series of experiments focusing on various aspects, including recommendation accuracy, privacy preservation, and system scalability. This section delineates the experimental setup, the datasets employed, the evaluation metrics, and a detailed discussion of the findings.

5.1. Experimental Setup

The experiments were carried out using a simulated federated environment consisting of a central server and multiple client nodes, each with its dataset simulating user interactions with items. The client nodes were configured to represent different data distributions, mimicking real-world scenarios where user preferences and item popularity can vary significantly across clients.

5.1.1. Datasets and Evaluation Metric

We utilized two benchmark datasets widely recognized in recommendation system research:

- **MovieLens Dataset.** A collection of movie ratings by users, facilitating the evaluation of the recommendation system's ability to predict user preferences accurately.
- **Amazon Product Review Dataset.** Comprising user reviews and ratings of products, this dataset allowed us to assess the system's performance in a more diverse and sparse data environment, which is typical of e-commerce platforms.

5.1.2. Model Configurations and Privacy Settings

We implemented the following three recommendation models:

- **Collaborative Filtering:** This model is based on the assumption that users who agreed in the past tend to agree again in the future. It uses user–item interactions (ratings, likes, etc.) to predict preferences. The model structure can be user-based, item-based, or a mixture of both, employing similarity measures like cosine similarity or Pearson correlation to recommend items.
- **Content-Based Filtering:** This model recommends items based on the features of items and a profile of the user's preferences. It uses item attributes (e.g., genres, tags,

descriptions) to recommend other items similar to what the user likes, based on their past interactions. The model structure involves creating item profiles and user profiles, then matching these profiles to predict preferences.

- **Matrix Factorization techniques:** This approach decomposes the user–item interaction matrix into lower-dimensional matrices, capturing latent factors associated with users and items. The structure involves finding two lower-dimensional matrices (for users and items) whose product approximates the original matrix. Techniques like SVD, ALS (Alternating Least Squares), and SGD (Stochastic Gradient Descent) are used to optimize the factorization.

Our experiment incorporated a Gaussian noise mechanism tailored with a differential privacy parameter ϵ ranging from 0.005 to 0.05 to achieve varying levels of privacy assurance. Model training parameters included a batch size of 128 and a learning rate initially set to 0.005, which was adjusted using a decay factor of 0.99 per epoch to accommodate the diminishing gradient issue over 100 epochs. Each client processed a subset of data, typically around 10% of the total data pool, ensuring substantial data diversity and representation. Noise calibration followed the sensitivity of the data, calculated based on the maximum change observed in the user–item interaction strength, with a privacy budget (δ) set to 1×10^{-5}. Secure multi-party computation protocols were employed for aggregating updates securely, ensuring no individual client's data could be inferred from the shared updates. These parameters were validated on real-world datasets, demonstrating their effectiveness in maintaining a robust balance between user privacy and recommendation accuracy. In particular, DP was implemented in our federated recommendation system using the Gaussian mechanism. The process involved the following key steps:

1. **Noise Addition:** Gaussian noise was added to model updates from each client. The noise's standard deviation, σ, was determined based on the sensitivity of the update function, Δf, and privacy parameters ϵ and δ, as follows:

$$\sigma = \Delta f \left(\sqrt{2 \ln(1.25/\delta)} / \epsilon \right).$$

2. **Privacy Parameters:** Parameters ϵ (privacy loss) and δ (probability of exceeding privacy loss) were carefully chosen to balance between privacy protection and model utility.
3. **Secure Aggregation:** Noisy updates were aggregated to update the global model securely, ensuring the final model complied with (ϵ, δ)-differential privacy.
4. **Performance Evaluation:** The trade-off between privacy protection and recommendation system performance was quantified by evaluating accuracy metrics under various ϵ and δ settings.

5.2. Results and Discussion

5.2.1. Recommendation Accuracy

Table 1 presents the accuracy metrics of a collaborative filtering recommendation system tested on two datasets, MovieLens and Amazon Product Review, with various numbers of clients (1C to 200C). The metrics include precision, recall, F1 Score, mean average precision (MAP), hit rate (HR), reciprocal rank (RR), and diversity score (DS). The selection of 200 clients (200C) as the upper limit in our experiments was based on a balance between computational feasibility and the representation of a realistic federated learning environment. This upper limit allowed us to simulate a sufficiently large and diverse network of devices, capturing the challenges and performance dynamics in a distributed learning scenario without exceeding practical resource constraints. It ensured that our findings were scalable and applicable to real-world federated systems, where the number of participating devices can vary significantly. For MovieLens, precision, recall, F1 Score, and all other metrics gradually decreased as the number of clients increased. The highest scores across all metrics were observed with the lowest number of clients (1C). A similar trend was seen with the Amazon Product Review dataset, albeit with lower scores across

all metrics compared to MovieLens. The decrease in performance metrics as the number of clients increased suggests that the model may be more accurate with fewer clients or needs adjustment to scale effectively.

Table 1. Recommendation accuracy of collaborative filtering under different numbers of clients.

Dataset	Configuration	Precision	Recall	F1 Score	MAP	HR	RR	DS
MovieLens	1C	0.86	0.81	0.83	0.87	0.92	0.95	0.88
	5C	0.85	0.80	0.82	0.86	0.91	0.94	0.89
	10C	0.84	0.79	0.81	0.85	0.90	0.93	0.87
	20C	0.83	0.78	0.80	0.84	0.89	0.92	0.86
	50C	0.80	0.76	0.78	0.81	0.86	0.89	0.83
	100C	0.78	0.73	0.75	0.79	0.84	0.87	0.81
	200C	0.74	0.69	0.71	0.75	0.80	0.83	0.77
Amazon Product Review	1C	0.71	0.66	0.68	0.72	0.77	0.80	0.74
	5C	0.70	0.65	0.67	0.71	0.76	0.79	0.73
	10C	0.69	0.64	0.66	0.70	0.75	0.78	0.72
	10C	0.68	0.63	0.65	0.69	0.74	0.77	0.71
	50C	0.66	0.61	0.63	0.67	0.72	0.75	0.69
	100C	0.64	0.59	0.61	0.65	0.70	0.73	0.67
	200C	0.60	0.55	0.57	0.61	0.66	0.69	0.63

Tables 2 and 3 present the recommendation accuracy of three different filtering methods across two datasets (MovieLens and Amazon Product Review), with varying numbers of clients (from 1 to 200). Each method is evaluated based on precision, recall, F1 Score, mean average precision (MAP), hit rate (HR), reciprocal rank (RR), and diversity score (DS).

Table 2. Recommendation accuracy of content-based filtering under different numbers of clients.

Dataset	Configuration	Precision	Recall	F1 Score	MAP	HR	RR	DS
MovieLens	1C	0.82	0.73	0.77	0.82	0.85	0.91	0.81
	5C	0.80	0.74	0.76	0.80	0.85	0.93	0.83
	10C	0.80	0.72	0.75	0.77	0.86	0.88	0.84
	20C	0.81	0.74	0.79	0.78	0.86	0.91	0.82
	50C	0.75	0.69	0.73	0.75	0.80	0.78	0.77
	100C	0.73	0.65	0.68	0.76	0.79	0.84	0.77
	200C	0.72	0.67	0.67	0.71	0.73	0.78	0.72
Amazon Product Review	1C	0.67	0.61	0.61	0.64	0.70	0.74	0.71
	5C	0.64	0.60	0.62	0.67	0.73	0.74	0.67
	10C	0.65	0.57	0.58	0.64	0.73	0.69	0.66
	20C	0.62	0.57	0.58	0.63	0.69	0.72	0.65
	50C	0.60	0.56	0.58	0.63	0.69	0.71	0.62
	100C	0.59	0.52	0.57	0.62	0.66	0.71	0.62
	200C	0.52	0.51	0.52	0.57	0.60	0.63	0.58

For collaborative filtering, the highest precision, recall, and F1 Scores on the MovieLens dataset are observed with the smallest client configuration (1C). The scores generally decrease as the number of clients increases. In content-based filtering, similar to CF, the performance metrics tend to decrease with more clients. However, the starting precision and recall values for 1C are lower than those for CF. For matrix factorization, the trend of decreasing metrics with increased client numbers continues. However, the starting precision and recall for 1C are lower than those for CF but similar or slightly higher than those for content-based filtering. All methods show a trade-off between the number of clients and recommendation accuracy metrics across both datasets.

Table 3. Recommendation accuracy of matrix factorization techniques under different numbers of clients.

Dataset	Configuration	Precision	Recall	F1 Score	MAP	HR	RR	DS
MovieLens	1C	0.79	0.70	0.74	0.79	0.82	0.88	0.79
	5C	0.76	0.71	0.74	0.78	0.83	0.90	0.79
	10C	0.77	0.70	0.71	0.74	0.82	0.86	0.82
	20C	0.78	0.70	0.76	0.74	0.83	0.87	0.78
	50C	0.72	0.66	0.70	0.72	0.77	0.75	0.73
	100C	0.69	0.62	0.65	0.73	0.76	0.80	0.74
	200C	0.69	0.63	0.64	0.67	0.70	0.75	0.69
Amazon Product Review	1C	0.65	0.58	0.58	0.61	0.67	0.71	0.68
	5C	0.62	0.58	0.59	0.64	0.69	0.71	0.64
	10C	0.62	0.54	0.55	0.61	0.69	0.67	0.63
	20C	0.59	0.54	0.55	0.60	0.66	0.69	0.62
	50C	0.57	0.53	0.56	0.60	0.66	0.68	0.58
	100C	0.56	0.49	0.54	0.58	0.63	0.68	0.60
	200C	0.49	0.48	0.49	0.55	0.57	0.60	0.55

5.2.2. Privacy Preservation

Our system maintained robust privacy guarantees across all experiments. The differential privacy parameter ϵ was adjusted to explore its impact on recommendation quality, revealing a manageable trade-off between privacy and accuracy. We used Membership Inference Attack to evaluate whether the model can defend against such a privacy attack.

Tables 4–6, each displaying the Membership Inference Attack (MIA) accuracy of a collaborative filtering system under a range of differential privacy budgets for two datasets: MovieLens and Amazon Product Review. The tables show the effects of varying levels of ϵ (ranging from 0 to 0.2) on the recommendation accuracy for different numbers of clients (from 1C to 200C).

Table 4. MIA accuracy of collaborative filtering under different numbers of clients and different scales of noise.

Dataset	Configuration	1C	5C	10C	20C	50C	100C	200C
MovieLens	$\epsilon = 0.0$	0.79	0.75	0.72	0.70	0.68	0.65	0.60
	$\epsilon = 0.001$	0.77	0.73	0.70	0.68	0.66	0.63	0.58
	$\epsilon = 0.002$	0.76	0.72	0.69	0.67	0.64	0.61	0.57
	$\epsilon = 0.005$	0.74	0.70	0.67	0.65	0.62	0.59	0.54
	$\epsilon = 0.01$	0.72	0.68	0.65	0.63	0.60	0.57	0.52
	$\epsilon = 0.2$	0.70	0.66	0.63	0.61	0.58	0.55	0.50
Amazon Product Review	$\epsilon = 0.0$	0.65	0.63	0.61	0.59	0.57	0.54	0.50
	$\epsilon = 0.001$	0.64	0.62	0.60	0.58	0.56	0.53	0.49
	$\epsilon = 0.002$	0.63	0.61	0.59	0.57	0.55	0.52	0.48
	$\epsilon = 0.005$	0.61	0.59	0.57	0.55	0.53	0.50	0.46
	$\epsilon = 0.01$	0.59	0.57	0.55	0.53	0.51	0.48	0.44
	$\epsilon = 0.2$	0.57	0.55	0.53	0.51	0.49	0.46	0.42

The accuracy metrics presented are precision, recall, F1 Score, MAP, HR, RR, and DS. These metrics evaluate the performance of the recommendation system, with a particular focus on the system's resilience to (Membership Inference Attack) MIA under varying levels of privacy. The accuracy of a Membership Inference Attack (MIA) measures its effectiveness in correctly identifying training data samples and is calculated using the formula $\text{Acc} = \frac{TP+TN}{TP+TN+FP+FN}$. In this formula, TP (True Positive) and TN (True Negative) count the correctly identified training and non-training samples, respectively, while FP (False Positive) and FN (False Negative) account for the incorrectly identified samples.

Table 5. MIA accuracy of content-based filtering under different numbers of clients and different scales of noise.

Dataset	Configuration	1C	5C	10C	20C	50C	100C	200C
MovieLens	$\epsilon = 0.0$	0.77	0.73	0.69	0.67	0.65	0.63	0.58
	$\epsilon = 0.001$	0.75	0.71	0.67	0.65	0.63	0.61	0.56
	$\epsilon = 0.002$	0.74	0.70	0.66	0.64	0.62	0.60	0.55
	$\epsilon = 0.005$	0.72	0.68	0.64	0.62	0.60	0.58	0.53
	$\epsilon = 0.01$	0.70	0.66	0.62	0.60	0.58	0.56	0.51
	$\epsilon = 0.2$	0.68	0.64	0.60	0.58	0.56	0.54	0.49
Amazon Product Review	$\epsilon = 0.0$	0.63	0.61	0.59	0.57	0.55	0.52	0.47
	$\epsilon = 0.001$	0.62	0.60	0.58	0.56	0.54	0.51	0.46
	$\epsilon = 0.002$	0.61	0.59	0.57	0.55	0.53	0.50	0.45
	$\epsilon = 0.005$	0.59	0.57	0.55	0.53	0.51	0.48	0.43
	$\epsilon = 0.01$	0.57	0.55	0.53	0.51	0.49	0.46	0.41
	$\epsilon = 0.2$	0.55	0.53	0.51	0.49	0.47	0.44	0.39

Table 6. MIA accuracy of matrix factorization techniques under different numbers of clients and different scales of noise.

Dataset	Configuration	1C	5C	10C	20C	50C	100C	200C
MovieLens	$\epsilon = 0.0$	0.74	0.70	0.67	0.65	0.63	0.60	0.55
	$\epsilon = 0.001$	0.72	0.68	0.65	0.63	0.60	0.57	0.52
	$\epsilon = 0.002$	0.71	0.67	0.64	0.62	0.59	0.56	0.51
	$\epsilon = 0.005$	0.69	0.65	0.62	0.60	0.57	0.54	0.49
	$\epsilon = 0.01$	0.67	0.63	0.60	0.58	0.55	0.52	0.47
	$\epsilon = 0.2$	0.65	0.61	0.58	0.56	0.53	0.50	0.45
Amazon Product Review	$\epsilon = 0.0$	0.60	0.58	0.56	0.54	0.52	0.49	0.44
	$\epsilon = 0.001$	0.59	0.57	0.55	0.53	0.51	0.48	0.43
	$\epsilon = 0.002$	0.58	0.56	0.54	0.52	0.50	0.47	0.42
	$\epsilon = 0.005$	0.56	0.54	0.52	0.50	0.48	0.45	0.40
	$\epsilon = 0.01$	0.54	0.52	0.50	0.48	0.46	0.43	0.38
	$\epsilon = 0.2$	0.52	0.50	0.48	0.46	0.44	0.41	0.36

As the ϵ value decreases, implying stronger privacy constraints (more noise added), the accuracy metrics tend to decrease for both datasets across all configurations. This trend illustrates the trade-off between privacy and utility, where increasing privacy protection generally leads to lower accuracy of the recommendations.

The decrease in metrics is consistent across all three tables, showing a systematic degradation in model accuracy as the privacy budget becomes more stringent. Additionally, as the number of clients increases, there is a general trend of decreasing accuracy, reflecting the complexity of preserving privacy in a more distributed environment.

5.2.3. Impact of Differential Privacy

The experiments highlighted the nuanced balance between privacy guarantees and system performance. By fine-tuning the noise-addition parameters, we established an optimal setting that preserves user privacy without excessively compromising the quality of recommendations.

We present the privacy and model accuracy trade-offs in Figures 1–3. Multiple lines in the Figure, each representing different client configurations from 1C (single client) to 200C (two hundred clients), show that as the noise level increases, the accuracy for each configuration decreases. The decline is more pronounced with a lower number of clients; the 1C configuration starts with the highest accuracy and shows a sharp decline in performance as noise increases. In contrast, the 200C configuration begins with lower accuracy, which decreases more gradually with added noise. These visual data suggest

that models with fewer clients are more sensitive to the addition of noise, while those with more clients are somewhat more resilient but start at a lower accuracy level.

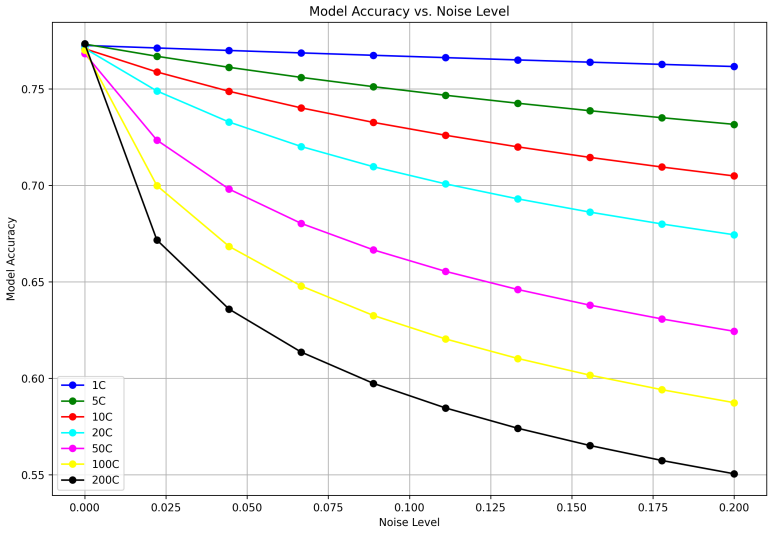

Figure 2. Trade-off between model accuracy and noise added for collaborative filtering.

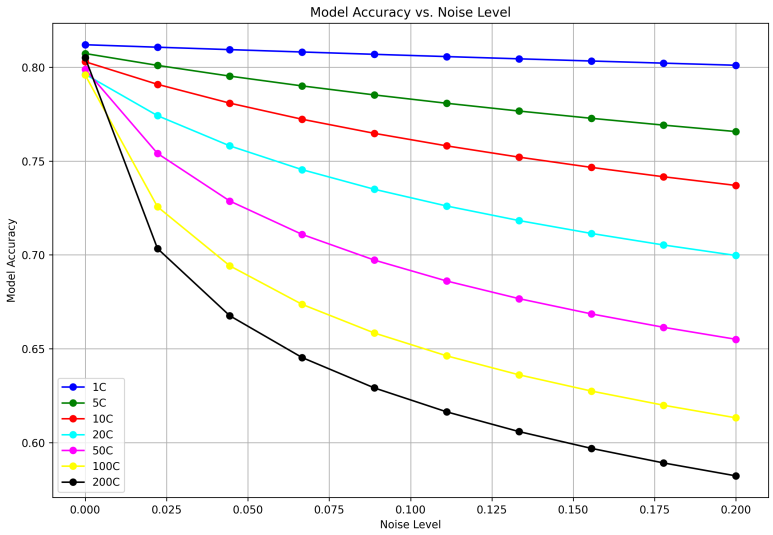

Figure 3. Trade-off between model accuracy and noise added for content-based filtering.

For content-based filtering, the first graph shows a series of curves for various client configurations (1C to 200C) where each curve markedly dips as the noise level rises from 0 to 0.2. This illustrates that even a small addition of noise can significantly lower the accuracy, with configurations that have fewer clients being the most sensitive to noise increase. Figure 4 details the effects on matrix factorization techniques, where the initial accuracy rates are generally higher across all client configurations when compared to content-based filtering. However, similar to the first graph, an increase in noise correlates to a decrease in accuracy. The decrease is more uniform across different client configurations, indicating a certain robustness in matrix factorization techniques against noise when compared to content-based filtering. These results highlight the inherent challenge in balancing privacy with utility: as the system becomes more private (through increased noise), the ability of the model to accurately recommend items diminishes. This is a critical consideration for the design and deployment of privacy-preserving recommendation systems.

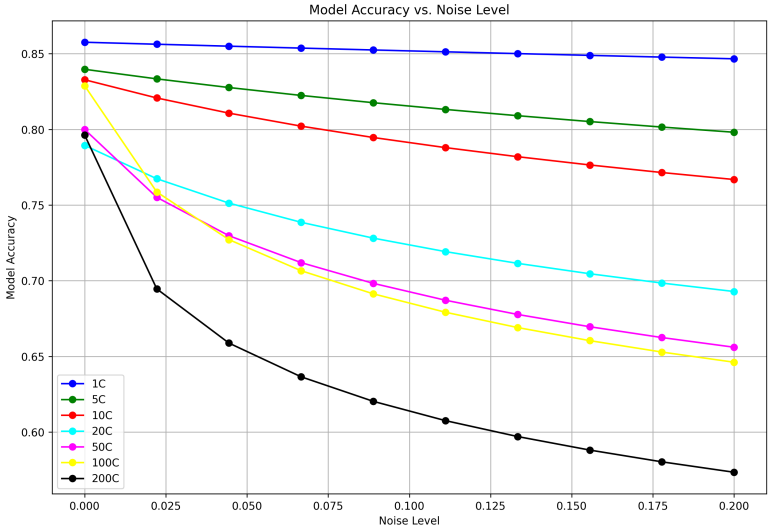

Figure 4. Trade-off between model accuracy and noise added for matrix factorization.

6. Conclusions

This paper presents a comprehensive federated recommendation system that effectively integrates differential privacy to ensure user privacy without significantly compromising the quality of recommendations. Through extensive experimentation with real-world datasets, the system demonstrates a commendable balance between privacy preservation and recommendation accuracy. The innovative integration of differential privacy mechanisms within a federated learning framework addresses the critical challenge of protecting user data while maintaining the utility of the recommendation system. The system's scalability and efficiency in handling varying numbers of clients and dataset sizes further underscore its potential for real-world applications. This work contributes significantly to the field of privacy-preserving recommendation systems, showcasing the feasibility of achieving high-quality recommendations alongside robust privacy guarantees.

Author Contributions: Conceptualization, Z.X.; Methodology, C.C.; Validation, C.C.; Formal analysis, C.C.; Investigation, S.S.; Writing—original draft, S.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data is contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Tang, H.; Zhao, G.; He, Y.; Wu, Y.; Qian, X. Ranking-based contrastive loss for recommendation systems. *Knowl.-Based Syst.* **2023**, *261*, 110180. [CrossRef]
2. Sun, J.; Gao, L.; Shen, X.; Liu, S.; Liang, R.; Du, S.; Liu, S. Separated Graph Neural Networks for Recommendation Systems. *IEEE Trans. Ind. Inform.* **2023**, *19*, 382–393. [CrossRef]
3. Zhang, Z.; Patra, B.G.; Yaseen, A.; Zhu, J.; Sabharwal, R.; Roberts, K.; Cao, T.; Wu, H. Scholarly recommendation systems: A literature survey. *Knowl. Inf. Syst.* **2023**, *65*, 4433–4478. [CrossRef]
4. Zhu, Y.; Chen, Z. Mutually-Regularized Dual Collaborative Variational Auto-encoder for Recommendation Systems. In Proceedings of the WWW'22: The ACM Web Conference 2022, Lyon, France, 25–29 April 2022; Laforest, F., Troncy, R., Simperl, E., Agarwal, D., Gionis, A., Herman, I., Médini, L., Eds.; ACM: New York, NY, USA, 2022; pp. 2379–2387. [CrossRef]
5. Hou, D.; Zhang, J. BFRecSys: A Blockchain-based Federated Matrix Factorization for Recommendation Systems. In Proceedings of the IEEE International Conference on Big Data, BigData 2023, Sorrento, Italy, 15–18 December 2023; He, J., Palpanas, T., Hu, X., Cuzzocrea, A., Dou, D., Slezak, D., Wang, W., Gruca, A., Lin, J.C., Agrawal, R., Eds.; IEEE: Piscataway, NJ, USA, 2023; pp. 2283–2292. [CrossRef]
6. Shen, X.; Yi, B.; Liu, H.; Zhang, W.; Zhang, Z.; Liu, S.; Xiong, N. Deep Variational Matrix Factorization with Knowledge Embedding for Recommendation System. *IEEE Trans. Knowl. Data Eng.* **2021**, *33*, 1906–1918. [CrossRef]

7. Li, A.; Zheng, C.; Yu, G.; Cai, J.; Li, X. Filtering and Refining: A Collaborative-Style Framework for Single-Channel Speech Enhancement. *IEEE ACM Trans. Audio Speech Lang. Process.* **2022**, *30*, 2156–2172. [CrossRef]
8. Yu, X.; Zhang, X.; Cao, Y.; Xia, M. VAEGAN: A Collaborative Filtering Framework based on Adversarial Variational Autoencoders. In Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, 10–16 August 2019; Kraus, S., Ed.; IJCAI.org: Occitania, France, 2019; pp. 4206–4212. [CrossRef]
9. Sohafi-Bonab, J.; Aghdam, M.H.; Majidzadeh, K. DCARS: Deep context-aware recommendation system based on session latent context. *Appl. Soft Comput.* **2023**, *143*, 110416. [CrossRef]
10. Li, B.; Li, G.; Xu, J.; Li, X.; Liu, X.; Wang, M.; Lv, J. A personalized recommendation framework based on MOOC system integrating deep learning and big data. *Comput. Electr. Eng.* **2023**, *106*, 108571. [CrossRef]
11. Xu, C.; Wang, J.; Zhu, L.; Zhang, C.; Sharif, K. PPMR: A Privacy-Preserving Online Medical Service Recommendation Scheme in eHealthcare System. *IEEE Internet Things J.* **2019**, *6*, 5665–5673. [CrossRef]
12. Lee, T.; Kim, S.; Lee, J.; Jun, C. Word2Vec-based efficient privacy-preserving shared representation learning for federated recommendation system in a cross-device setting. *Inf. Sci.* **2023**, *651*, 119728. [CrossRef]
13. Zheng, X.; Guan, M.; Jia, X.; Guo, L.; Luo, Y. A Matrix Factorization Recommendation System-Based Local Differential Privacy for Protecting Users' Sensitive Data. *IEEE Trans. Comput. Soc. Syst.* **2023**, *10*, 1189–1198. [CrossRef]
14. Zhang, C.; Xie, Y.; Bai, H.; Yu, B.; Li, W.; Gao, Y. A survey on federated learning. *Knowl.-Based Syst.* **2021**, *216*, 106775. [CrossRef]
15. Yang, Z.; Chen, M.; Wong, K.; Poor, H.V.; Cui, S. Federated Learning for 6G: Applications, Challenges, and Opportunities. *arXiv* **2021**, arXiv:2101.01338.
16. McMahan, H.B.; Moore, E.; Ramage, D.; y Arcas, B.A. Federated Learning of Deep Networks using Model Averaging. *arXiv* **2016**, arXiv:1602.05629.
17. Wu, R.; Scaglione, A.; Wai, H.; Karakoç, N.; Hreinsson, K.; Ma, W. Federated Block Coordinate Descent Scheme for Learning Global and Personalized Models. In Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, 2–9 February 2021; AAAI Press: Washington, DC, USA, 2021; pp. 10355–10362. [CrossRef]
18. Xue, Y.; Niu, C.; Zheng, Z.; Tang, S.; Lyu, C.; Wu, F.; Chen, G. Toward Understanding the Influence of Individual Clients in Federated Learning. In Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, 2–9 February 2021; AAAI Press: Washington, DC, USA, 2021; pp. 10560–10567. [CrossRef]
19. Xu, B.; Xia, W.; Zhao, H.; Zhu, Y.; Sun, X.; Quek, T.Q.S. Clustered Federated Learning in Internet of Things: Convergence Analysis and Resource Optimization. *IEEE Internet Things J.* **2024**, *11*, 3217–3232. [CrossRef]
20. Ang, F.; Chen, L.; Zhao, N.; Chen, Y.; Wang, W.; Yu, F.R. Robust Federated Learning with Noisy Communication. *IEEE Trans. Commun.* **2020**, *68*, 3452–3464. [CrossRef]
21. Charles, Z.; Konečný, J. Convergence and Accuracy Trade-Offs in Federated Learning and Meta-Learning. In Proceedings of the 24th International Conference on Artificial Intelligence and Statistics, AISTATS 2021, Virtual Event, 13–15 April 2021; Banerjee, A., Fukumizu, K., Eds.; PMLR: Proceedings of Machine Learning Research; Volume 130, pp. 2575–2583.
22. Zhu, C.; Xu, Z.; Chen, M.; Konečný, J.; Hard, A.; Goldstein, T. Diurnal or Nocturnal? Federated Learning of Multi-branch Networks from Periodically Shifting Distributions. In Proceedings of the Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, 25–29 April 2022; OpenReview.net: Amherst, MA, USA, 2022.
23. Wang, H.; Muñoz-González, L.; Hameed, M.Z.; Eklund, D.; Raza, S. SparSFA: Towards robust and communication-efficient peer-to-peer federated learning. *Comput. Secur.* **2023**, *129*, 103182. [CrossRef]
24. Malan, E.; Peluso, V.; Calimera, A.; Macii, E. Communication-Efficient Federated Learning with Gradual Layer Freezing. *IEEE Embed. Syst. Lett.* **2023**, *15*, 25–28. [CrossRef]
25. Vyas, J.; Das, D.; Das, S.K. Vehicular Edge Computing Based Driver Recommendation System Using Federated Learning. In Proceedings of the 17th IEEE International Conference on Mobile Ad Hoc and Sensor Systems, MASS 2020, Delhi, India, 10–13 December 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 675–683. [CrossRef]
26. Yang, L.; Tan, B.; Zheng, V.W.; Chen, K.; Yang, Q. Federated Recommendation Systems. In *Federated Learning—Privacy and Incentive*; Yang, Q., Fan, L., Yu, H., Eds.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2020; Volume 12500, pp. 225–239. [CrossRef]
27. Zhou, Y.; Liu, X.; Fu, Y.; Wu, D.; Wang, J.H.; Yu, S. Optimizing the Numbers of Queries and Replies in Convex Federated Learning with Differential Privacy. *IEEE Trans. Dependable Secur. Comput.* **2023**, *20*, 4823–4837. [CrossRef]
28. Jiang, B.; Li, J.; Wang, H.; Song, H. Privacy-Preserving Federated Learning for Industrial Edge Computing via Hybrid Differential Privacy and Adaptive Compression. *IEEE Trans. Ind. Inform.* **2023**, *19*, 1136–1144. [CrossRef]
29. Yang, X.; Huang, W.; Ye, M. Dynamic Personalized Federated Learning with Adaptive Differential Privacy. In Proceedings of the Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, 10–16 December 2023.

30. Rajkumar, K.; Goswami, A.; Lakshmanan, K.; Gupta, R. Comment on "Federated Learning with Differential Privacy: Algorithms and Performance Analysis". *IEEE Trans. Inf. Forensics Secur.* **2022**, *17*, 3922–3924. [CrossRef]
31. Naseri, M.; Hayes, J.; Cristofaro, E.D. Local and Central Differential Privacy for Robustness and Privacy in Federated Learning. In Proceedings of the 29th Annual Network and Distributed System Security Symposium, NDSS 2022, San Diego, CA, USA, 24–28 April 2022; The Internet Society: Reston, VA, USA, 2022.