Comparative Analysis of Classic and Federated Learning-Based Neuronal Recommender Systems on the Yelp Open Dataset

Samuel Pariente
CentraleSupélec

Final Report

Abstract

This project investigates the comparative performance of Neural Matrix Factorization (NeuMF) and Federated Neural Matrix Factorization (FedNeuMF) using the Yelp dataset, with a focus on three critical aspects: metrics, cost, and carbon emissions. Our findings reveal that NeuMF outperforms FedNeuMF in terms of key metrics such as Mean Average Precision (MAP), Recall, Precision, and Normalized Discounted Cumulative Gain (NDCG), offering insights into the trade-offs between centralized and federated approaches in recommender systems. In terms of operational costs, we compare the time and financial expenditure associated with training NeuMF on an AWS server against the federated approach, which involves multiple mobile devices and API communications. Lastly, the study contrasts the carbon emissions resulting from a centralized AWS server with those from the distributed mobile devices used in FedNeuMF. This comparison sheds light on the environmental impact of each approach, providing a comprehensive understanding of the sustainability implications in the deployment of these recommender systems.

1. Introduction/Motivation

The realm of recommender systems has seen significant advancements with the advent of sophisticated machine learning techniques, particularly in the context of large-scale online platforms. Neural Matrix Factorization (NeuMF), an amalgamation of matrix factorization and neural network architectures, has emerged as a potent tool for enhancing recommendation quality. However, with growing concerns over data privacy and the need for decentralized data processing, Federated Neural Matrix Factorization (FedNeuMF) has gained attention as an alternative that leverages federated learning principles.

The motivation for this project stems from the increasing reliance on recommender systems in various online services, notably in platforms like Yelp, where personalized recommendations hold substantial value for user experience and business growth. Despite their utility, traditional centralized recommender systems, such as NeuMF, pose challenges in terms of data privacy and centralized data processing. In contrast, FedNeuMF offers a more privacy-preserving and distributed approach. However, the trade-offs between these two methodologies, particularly in performance metrics, operational costs, and environmental impacts, are not well-explored.

This project is driven by the necessity to understand these trade-offs in a real-world scenario. By comparing NeuMF and FedNeuMF on the Yelp dataset, we aim to elucidate the implications of adopting a federated approach over a centralized one, in terms of recommendation quality, computational efficiency, cost-effectiveness, and environmental sustainability. The insights gained from this study are expected to contribute significantly to the domain of recommender systems, offering guidance for future implementations that balance performance with privacy, cost, and ecological considerations.

2. Problem Definition

The primary objective of this research is to conduct a comparative analysis of two distinct approaches to recommender systems: Neural Matrix Factorization (NeuMF) and Federated Neural Matrix Factorization (FedNeuMF), particularly in the context of the Yelp dataset. This comparison is centered around three main axes: performance metrics, operational cost, and carbon emissions. The study aims to quantify and understand the trade-offs involved when choosing between these two methodologies for practical applications.

2.1. Performance Metrics

The first axis of comparison involves key performance metrics such as Mean Average Precision (MAP), Recall, Precision, and Normalized Discounted Cumulative Gain (NDCG). These metrics are critical for assessing the effectiveness of recommender systems. The hypothesis is that while NeuMF may offer superior performance in these metrics due to its centralized learning paradigm, FedNeuMF might present limitations due to its distributed nature.

2.2. Operational Cost

The second comparison criterion is the operational cost, which includes the time and financial resources required for training the models. For NeuMF, this involves the computation cost on an AWS server, whereas for FedNeuMF, it encompasses the cost associated with the federated learning process, including multiple mobile devices and API communications. This comparison seeks to establish a cost-benefit analysis of both approaches.

2.3. Carbon Emissions

Finally, the study evaluates the environmental impact of both methodologies by comparing the carbon emissions from a centralized AWS server (used in NeuMF) and the distributed mobile devices (used in FedNeuMF). This aspect is crucial in understanding the sustainability of these recommender systems, especially in the context of increasing environmental concerns related to technological advancements.

Through this comparative analysis, the research aims to provide a comprehensive understanding of the implications of adopting either a centralized or federated approach in the design and implementation of recommender systems, with a particular focus on the Yelp dataset.

3. Related Work

In this section, we first discuss algorithms used for recommendation including matrix factorization and neural collaborative filtering. We then introduce proposed methods from the state-of-the art for privacy-preserving recommenders based on federated learning. Finally we talk about related work of carbon emission study of federated learning.

3.1. Recommender systems

Recommender systems are widely used to offer a personalized experience for users. The most popular techniques are Martix Factorization (MF) and Neural Collaborative Filtering (NCF).

3.1.1 Matrix Factorization

Given the rating matrix $R = (r_{ui})_{1\leq u\leq m, 1\leq i\leq n}$, matrix factorization models [9] aim to learn the user latent factors matrix $X \in \mathbb{R}^{n\times d}$ and the the item latent factors matrix $V \in \mathbb{R}^{m\times d}$ so that the matrix R is factorized to: $R \approx X \cdot V^T$. The missed values $\hat{r}_{ui}$ that represent the rating information given by u to i will be predicted as the dot product of: $\hat{r}_{ui} = x_u \cdot v_i^T$ where x_u is the user latent factor vector and v_i is the item latent factor vector. The computing process of X and V can be done by solving the following regularized least squares minimization:

$$\mathcal{L} = \min_{p^*, q^*} \sum_{(u,i)\in\mathcal{D}} (r_{ui} - \hat{r}_{ui})^2 + \lambda\|p_u\|^2 + \mu\|q_i\|^2 \quad (1)$$

where λ and μ are small positive values to rescale the penalizer. Stochastic gradient descent iteratively updates X and V with the following equations [9]:

$$x_u^t = x_u^{t-1} + 2\gamma \sum_i [v_i(r_{ui} - x_u^{t-1} \cdot v_i^T) + 2\lambda x_u^{t-1}] \quad (2)$$

$$v_i^t = v_i^{t-1} + 2\gamma \sum_i [v_i(r_{ui} - x_u^{t-1} \cdot v_i^T) + 2\lambda v_i^{t-1}] \quad (3)$$

The number of iterations relies on the stopping criteria. A typical criteria is to set a small threshold ϵ, such that the training stops when the two gradients of X and V (or one of them) are smaller than ϵ.

3.1.2 Neural Collaborative Filtering

The MF model estimates an interaction r_{ui} as the inner product of the latent vectors. However, [5] argued that the product vector is inefficient in formulating users' similarity and showed that this limitation can be overcome by learning the interaction function using deep neural networks. So, they presented a novel framework for CF based a neural architecture to model the user-item interactions. First, they presented a generalized MF (GMF) which is the neural implementation of MF by using embedding layers to obtain the latent user–item vectors. Then, the latent vectors are fed into a linear layer, which outputs the predicted score using the sigmoid function. The second proposed model is a multilayer perceptron (MLP) model, where the user and item latent vectors are concatenated into a single vector and the output is then fed to the hidden layers. The third model consist of combining the linear GMF model and the nonlinear MLP model to construct the NeuMF model that offer higher recommendation quality and faster convergence. In this model, the GMF outputs the product of latent vectors and the MLP feeds the concatenation of the latent vectors into the deep neural network. The two outputs are concatenated in the last hidden layer, where the value of r_{ui} is predicted.

3.2. Privacy-preserving recommenders

Building privacy-preserving models receives great attention over the last years because of the need to protect users private data from being shared with service providers. Several works [2, 6] rely on applying privacy-preserving techniques such as differential privacy or homomorphic encryption before transmitting the users data to the central server.

Even though this techniques ensure a certain level of privacy protection, they lead to a loss of predictive power of the learned latent feature vectors and the data is still shared with service providers. Federated Learning (FL) [7] has been applied to recommender systems to allow to train the central model on privacy-sensitive data distributed across users' devices.

3.2.1 Federated learning for privacy preservation

Federated learning [7, 8, 11] is a privacy-preserving distributed learning scheme proposed by Google, which enables participants to collaboratively train a machine learning model by sharing intermediate parameters (e.g., model parameters, gradients) to the central server instead of their real data, so that the self-owned data of each client does not leave his local device with guarantees at some level privacy of his local raw data. In a system where N users $\{u_1, u_2, ..., u_N\}$ having local private data $\{d_1, d_2, ..., d_N\}$ stored in their local devices collaborate to learn a central model w. First, the server initiate a global model and share it with all users. Each of the users receive the global model and train his local model w_u on his local data d_u, then he send only the intermediate updated parameters with the server. After that, the server aggregate all the received parameters to update the global model using usually the FedAvg algorithm [7, 8, 11] which consists of taking the weighted average of the received $\{w_1, w_2, ..., w_n\}$. This typically cycle represented in figure 1 is repeated until the model converges.

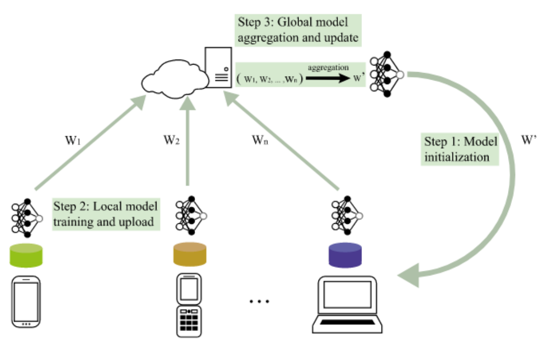

Figure 1. A schematic diagram of federated learning[20] .

3.2.2 Federated Recommenders (FedRec)

Combining federated learning with recommendation systems becomes a promising solution for privacy-preserving recommendation systems. FedRec is a distributed architecture used for privacy-preserving recommendation system. the goal is that the performance of recommendation model trained in FedRec should be closed to the performance of the recommendation model trained in the data centralized setting, which can be formalized as:

$$|V_{FED} - V_{SUM}| < \sigma \tag{4}$$

where V_{FED} is the recommendation model performance in FedRec , V_{CEN} is the recommendation model performance in traditional recommendation systems for centralized data storage, and σ is a small positive numbers [15].

The federated matrix factorization (FedMF) was initially proposed by [1] where the user factor vectors x_u are stored and updated locally on the clients devices, however the item factor vectors are updated locally and uploaded to the central server that holds the global item model for aggregation. Because sharing items vectors in clear format can still be source of inference attacks where an honest but curious server can still get some data about the user interactions [4], other works such as [10, 3, 19, 12] based on the model of [1] proposed a privacy-preserving FedMF models where the shared items vectors are secured by either local differential privacy (LDP), Secure multiparty Computation (SMC) or Homomorphic encryption (HE). MF model based FedRec can simply and effectively capture user tastes with the interaction and rating information between users and items. However, plus MF limitations such as sparsity and cold-start problems, embedding models like MF are not adapted for FL with FedAvg [14].

Other works focused on exploiting the power of neural architectures to model the user-item interactions and proposed deep learning based FedRec. [14] proposed the Federated NCF (FedNCF) framework which the federated adaptations of the popular NCF approach [5]. Their approach is founded on three federated neural models: Federated GMF, Federated NCF and Federated NeuMF. For each of the models, only the item embeddings and the neural architecture weights are shared with the server. To address the problem of inference attacks, they incorporate a privacy-preserving secure aggregation method that satisfies the security requirements against an honest but curious entity. [18] proposed an Efficient-FedRec: Efficient Federated Learning Framework for Privacy-Preserving News Recommendation. The Efficient-FedRec consists of a large news model maintained in the server and a light-weight user model shared on both server and clients, where news representations and user model are communicated between server and clients. For more privacy, they use secure aggregation which mainly based on secure multi-party computation (SMC). [17] proposed a federated graph neural network (FedGNN) framework based on GNN. In FedGNN, the clients locally train GNN models and update the user/item embeddings from their local subgraph, then send the perturbed gradients of GNN model and item embedding to the central server for aggregation. Indeed, the proposed approaches achieve a good recommendation quality and guarantee that the user data is processed locally and never shared, however in the most of the time they neglect the real scenario where models are really built in users mobile phones where computation and communication costs are limited and user experience is affected.

These works motivate us to provide a foundation for the current study, which aims to compare traditional and federated learning-based recommendation systems in term of industrial application, specially in the tourism industry.

3.2.3 Carbon Emission of Federated Learning

A critical aspect of evaluating federated learning, especially in the context of its environmental impact, is the assessment of carbon emissions. The paper [13] provides an insightful analysis of the carbon footprint associated with federated learning. According to this study, the distributed nature of federated learning, which involves computations on numerous edge devices, can lead to a significant amount of carbon emissions. This is attributed to the energy consumption of these devices during the model training process, and the additional network-related energy expenditure required for the communication between the edge devices and the central server.

The paper highlights that while federated learning offers privacy and security benefits by keeping the data on the user's device, this comes at the cost of increased energy consumption due to the need for continuous communication and data exchange over the network. Moreover, the varied energy efficiency of different devices participating in the federated learning process can further contribute to the overall carbon footprint. As such, the study underscores the importance of developing energy-efficient federated learning algorithms and strategies for minimizing network communication to reduce the environmental impact of these systems.

This perspective on federated learning is particularly relevant to our study, as it adds another dimension to the comparison between traditional centralized methods and federated approaches. Understanding the environmental implications is crucial in making informed decisions about the deployment of these technologies, especially in the context of growing concerns about climate change and sustainability.

In an attempt to quantify the carbon emissions of federated learning systems, the paper [13] introduces two key formulas to estimate the energy consumption for federated learning (E_{FL}) and centralized training (E_{center}):

$$E_{FL} = C_{rate} \cdot [T(e, N, R) + C(e, N, R)], \tag{5}$$

$$E_{center} = C_{rate} \cdot T_{center}, \tag{6}$$

where C_{rate} represents the carbon emission rate, $T(e, N, R)$ is the total training time, $C(e, N, R)$ denotes the communication cost, all in the context of federated learning with e epochs, N clients, and R rounds. T_{center} is the total training time for centralized training. These formulas provide a framework for calculating and comparing the carbon footprint of federated and centralized training approaches.

4. Methodology

This section outlines the methods employed in the study, including data preparation, implementation details of the recommendation algorithms, cost computation strategies, and the framework for estimating carbon emissions.

4.1. Dataset

The dataset employed for this study is the Yelp Dataset. It comprises user interactions with various items, capturing the essence of real-world use cases in recommender systems. The dataset is characterized by a high degree of sparsity, which is typical of user-item interaction data in many recommender systems. The sparsity indicates that only a small fraction of the potential user-item interactions are known, which presents a significant challenge for any recommendation algorithm.

The statistics of the evaluation dataset are presented in Table 1, illustrating the number of interactions, users, items, and the overall sparsity of the dataset.

Dataset	#Interaction	#User	#Item	Sparsity
Yelp Dataset	6,990,280	1,987,929	150,346	99.99%

Table 1. Evaluation dataset statistics

Additionally, the dataset exhibits significant heterogeneity in terms of the quantity of data contributed by individual users. This aspect is captured in Table 2, which summarizes the minimum, maximum, average, standard deviation, and variance of the number of interactions per user.

Dataset	#Min	#Max	Avg.	St. dev.	Variance
Yelp Dataset	5	7,673	46.49	124.51	15,504

Table 2. Data quantity heterogeneity

Before feeding the data into the recommendation algorithms, a preprocessing step was conducted to clean the data and format it appropriately for the input requirements of the models. This preprocessing included filtering out noise, handling missing values, and normalizing the data where necessary.

The dataset's characteristics, such as sparsity and heterogeneity, pose unique challenges that the recommender system must address, such as the cold start problem and the long-tail distribution of user interactions. These challenges are critical factors in the evaluation of the recommendation algorithms' performance.

4.2. NeuMF and FedNeuMF Implementation

Neural Matrix Factorization (NeuMF) combines the linearity of Matrix Factorization (MF) and the non-linearity of Multi-Layer Perceptron (MLP) to enhance the recommendation process. This hybrid approach captures the complex interactions between users and items more effectively than either approach alone.

4.2.1 NeuMF

The architecture of the NeuMF model, as shown in Figure 2, consists of two pathways that process user and item vectors: one using GMF to perform element-wise multiplication, and another using MLP to capture non-linear interactions. These pathways are then concatenated to form a feature vector that is processed by a final prediction layer to estimate user-item interactions.

4.2.2 FedNeuMF

In the federated setting, the FedNeuMF model adapts NeuMF to accommodate decentralized data residing on users' devices. As depicted in Figure 3, the federated learning process involves a sequence of operations where a central server coordinates the model updates. The server selects a subset of clients to perform local computations and then aggregates these local updates to refine the global model parameters.

4.3. Cost Analysis

The cost of implementing a federated learning algorithm in the AWS cloud environment hinges upon several key components:

4.3.1 Virtual Machine (VM) Allocation Costs

Virtual Machine pricing depends primarily on the VM's specifications and the cloud region it's deployed in. In the AWS ecosystem, for instance, the VM type t2.medium, which is equipped with 2 vCPUs and 4 GB RAM, incurs a charge of $0.0464 per hour when deployed in the North Virginia (us-east-1 region). AWS also provides a multitude of VM choices, each with its own set of specifications, benefits, and associated costs. Hence, it's crucial to select a VM type that aligns with the requirements of the federated learning algorithm to ensure cost-effectiveness without compromising on performance.

4.3.2 Communication Costs

Data transfer and communication form an integral part of federated learning, and it's vital to factor in these costs for an accurate estimate. In AWS, transferring data to the Internet is priced at $0.09 per GB for the first 10 TB/month. It's noteworthy that these costs might vary depending on the type of data transfer (inbound or outbound) and the regions involved. Furthermore, transferring data between AWS services or regions might have its own pricing structure.

4.3.3 Storage Costs

Data storage is another major expense. Amazon S3, AWS's scalable storage service, levies charges based on the amount of data stored and the operations performed on it. In the North Virginia (us-east-1 region), users are billed $0.023 for every GB of data stored. Additionally, operations on this data, such as PUT, GET, and LIST requests, incur charges. For instance, $0.005 is charged for every 1000 operations in the North Virginia region. It's imperative to monitor and manage stored data efficiently to control storage costs, especially when dealing with large datasets.

The holistic formula to derive the cost of running a federated learning algorithm on AWS can thus be expressed as:

$$\text{Total Cost} = \text{VM Costs} + \text{Communication Costs} + \text{Storage Costs}$$

4.4. Estimation of Carbon Impact

This section elaborates on the estimation of the carbon impact of Federated Learning (FL). The carbon impact is ascertained by evaluating the energy consumption during the training and communication phases, and subsequently converting this energy consumption into CO_2e emissions.

4.4.1 Energy Consumption during Training

The energy consumption during the training phase is calculated by monitoring the power usage of both the GPU and CPU while the training is underway. This measurement is taken for each individual client, incorporating the power usage of both the GPU and CPU. The total energy consumption during training, denoted as $T_{FL}(e, N, R)$, is calculated considering N as the total number of clients in the pool, each with a hardware power of e, and R as the total number of rounds in the Federated Learning (FL) setup. The equation for this calculation is:

$$T_{FL}(e, N, R) = \sum_{j=1}^{R} \sum_{i=1}^{N} 1_{Clt_{i,j}} \cdot t_i \cdot e_{client,i} \tag{7}$$

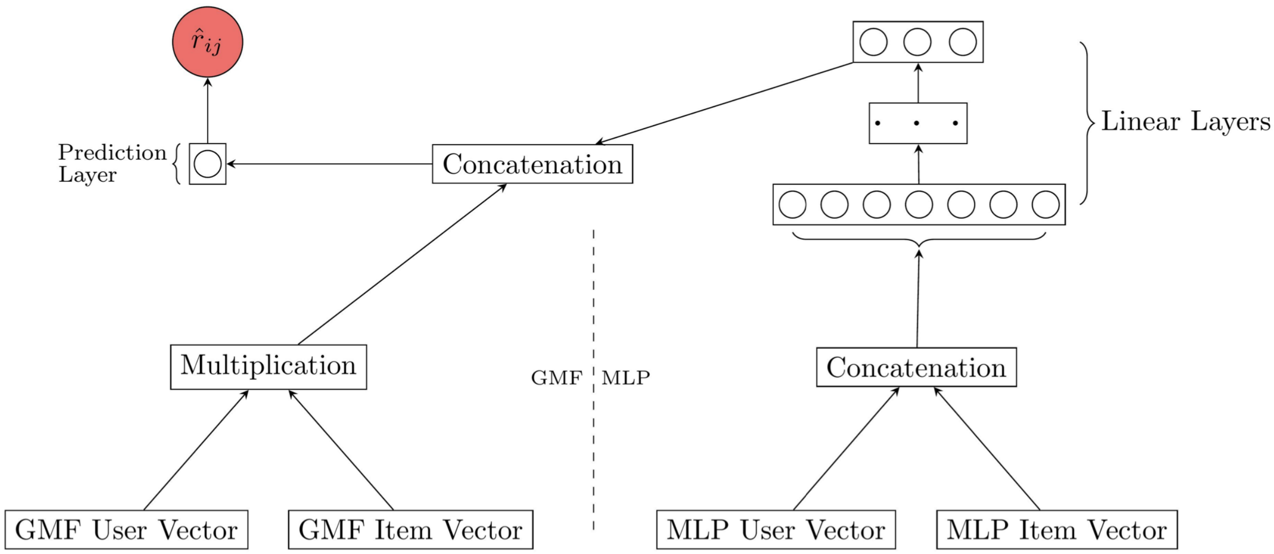

Figure 2. Architecture overview of the NeuMF model [16].

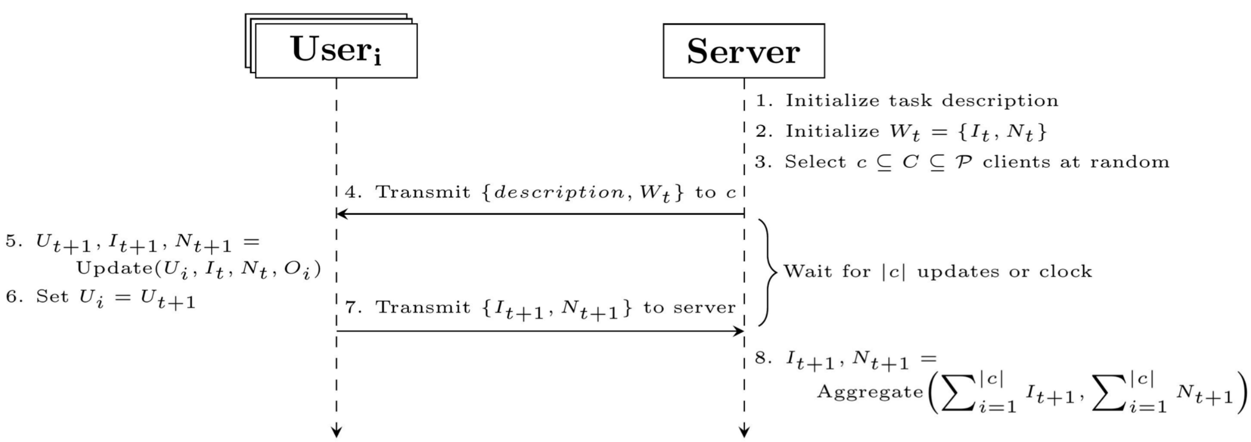

Figure 3. Order of operations and parameter exchange between the coordination server and clients in an aggregation round [16].

In this equation, $1_{Clt_{i,j}}$ is an indicator function that equals 1 if client i is selected for training in round j, t_i is the wall clock time per round, and $e_{clt,i}$ is the power usage of client i.

For a centralized training setup, the energy consumption is represented as:

$$T_{center} = PUE \cdot (t \cdot e_{center}), \tag{8}$$

In this equation, e_{center} signifies the combined power usage of both the GPUs and CPUs. The variable t represents the total training time. The product of t and e_{center} is then multiplied by the Power Usage Effectiveness (PUE), a metric that describes the efficiency of a data center's energy use, to determine the total energy consumption.

The communication cost formula is given as:

$$C(e,N,R) = \sum_{j=1}^{R}\sum_{i=1}^{N} 1_{Clt_{i,j}} \cdot S \cdot \left(\frac{1}{D} + \frac{1}{U}\right) \cdot (e_r + e_{idle,i}) \tag{9}$$

In this formula, R is the number of communication rounds, N is the number of clients, $1_{Clt_{i,j}}$ is an indicator function that equals 1 if client i is selected in round j and 0 otherwise, S is the size of the model in Mb, D and U are the download and upload speeds expressed in Mbps respectively, e_r is the power usage of the router, and $e_{idle,i}$ is the power usage of the idle hardware of the clients.

4.4.2 Conversion to CO_2e emissions

The energy consumption is then converted to CO2e emissions based on geographical locations. The total amount of CO2e emitted in kilograms for FL (EFL) and centralized training ($Ecenter$) obtained from the above formula are:

$$E_{FL} = c_{rate} \cdot [T(e,N,R) + C(e,N,R)], \tag{10}$$

$$E_{center} = c_{rate} \cdot T_{center}, \tag{11}$$

where c_{rate} is the conversion rate factor. The c_{rate} will depend on the physical location of the hardware where the training takes place, and it is possible that c_{rate} is not unique across the FL settings as clients can be scattered around the globe.

4.4.3 Quantification of Carbon Footprint

The quantification of the carbon footprint takes into account both the energy consumption during training and communication. These factors are influenced by the geographical location of the hardware, the efficiency of the hardware, the hyper-parameters used during training, and the strategies employed in Federated Learning (FL).

Quantifying the carbon footprint of FL can be challenging compared to centralized training due to the inherent complexities involved in FL. These complexities may include data heterogeneity, the geographical distribution of clients, and system heterogeneity.

In summary, the carbon impact of FL can be estimated by evaluating the energy consumption during training and communication phases, and subsequently converting this energy consumption into CO_2e emissions. This conversion is based on geographical locations. However, it is important to note that accurately determining the carbon impact can be difficult without specific context due to the inherent complexities in how FL is implemented.

5. Evaluation

The evaluation of the federated learning approach using the FedNeuMF model encompasses various aspects, including performance metrics, cost analysis, and carbon emissions. We detail these evaluations using the Yelp dataset as our benchmark.

Parameter	Value
Model	FedNeuMF
Number of Users	200
Number of Iterations	400
Duration	28.8 hours
Weights Size	750KB
VM Price	$0.05/hour
Communication Cost	$0.09/GB
Storage Cost	$0.023/GB
Interaction Cost	$0.005 for every 1000 interactions

5.1. Metrics

The performance of the recommendation models was measured using several standard metrics in the field:

- **MAP@10 (Mean Average Precision at 10)**: Reflects the precision of the top ten recommendations, averaged across all users. The formula for MAP@10 is given by:

$$MAP@10 = \frac{1}{U}\sum_{u=1}^{U}\frac{1}{\min(m_u,10)}\sum_{k=1}^{10} P(u,k)\cdot rel(u,k) \tag{12}$$

where U is the number of users, m_u is the number of relevant items for user u, $P(u,k)$ is the precision at cut-off k in the user's list of recommendations, and $rel(u,k)$ is an indicator function equaling 1 if the item at rank k is relevant for user u, and 0 otherwise.

- **NDCG@10 (Normalized Discounted Cumulative Gain at 10)**: Assesses the ranking quality by considering the positions of the relevant items. The NDCG@10 is calculated as:

$$NDCG@10 = \frac{1}{U}\sum_{u=1}^{U}\frac{DCG@10(u)}{IDCG@10(u)} \tag{13}$$

where $DCG@10(u)$ is the discounted cumulative gain at rank 10 for user u, and $IDCG@10(u)$ is the ideal discounted cumulative gain at rank 10 for user u.

- **Precision@10**: Evaluates the proportion of recommended items in the top ten that are relevant. It is defined as:

$$Precision@10 = \frac{1}{U}\sum_{u=1}^{U}\frac{1}{10}\sum_{k=1}^{10} rel(u,k) \tag{14}$$

- **Recall@10**: Assesses the proportion of relevant items that are found in the top ten recommendations. The formula for Recall@10 is:

$$Recall@10 = \frac{1}{U} \sum_{u=1}^{U} \frac{\sum_{k=1}^{10} rel(u,k)}{m_u} \quad (15)$$

The NeuMF model outperformed FedNeuMF across all metrics 3, which may suggest that while federated learning adds a layer of privacy, it could also introduce a performance trade-off.

5.2. Cost Analysis

The cost analysis provides insight into the financial feasibility of employing the federated learning model in a cloud computing environment. The total cost encompasses VM allocation, model download from the server, storage, and interactions, summing up to $7.35 for the FedNeuMF model.

Cost Component	Cost
VM Allocation	$1.44
Downloading Model from Server	$4.39
Storage (Weights and Model)	$1.12
Interactions	$0.40
Total Cost	**$7.35**

Table 4. Cost Analysis

5.3. Estimation of Carbon Impact

The estimation of carbon impact involves calculating the energy consumption for both the training and communication phases of the FedNeuMF model, and then converting this energy consumption into CO_2e emissions. The following tables 5 6 provide an overview of power consumption and the resulting CO_2e emissions for a federated learning setup.

Training Strategy	PUE	c_{rate}	CO_2e
Centralized NeuMF	1.55	0.316	3.30
FedNeuMF	x	0.428	55.64

Table 6. CO_2e emissions

6. Conclusions

The comprehensive evaluation of the FedNeuMF model using the Yelp dataset has provided valuable insights into the performance, cost, and environmental impact of federated learning approaches. Our analysis demonstrates that while federated learning, as instantiated in the FedNeuMF model, offers enhanced privacy features, it does present trade-offs in terms of recommendation accuracy and efficiency when compared to the centralized NeuMF model.

The cost analysis and carbon impact estimation have underlined the financial and environmental considerations that come into play when deploying machine learning models in cloud-based environments. The total cost of operation and the associated CO_2e emissions are non-trivial factors that must be factored into the decision-making process for organizations aiming to utilize federated learning systems.

We conclude that the balance between model performance, operational cost, and carbon footprint is a delicate one. Our findings suggest that while FedNeuMF may not match the performance metrics of NeuMF, it remains a viable option for industries that prioritize user privacy and are willing to navigate the complexities of federated systems.

For future implementations, we recommend:

- Emphasizing the development of federated learning models that can achieve a closer parity with centralized models in terms of performance metrics without significantly increasing costs or carbon emissions.
- Exploring cost reduction strategies, particularly in the communication and storage aspects of federated learning, which are the primary contributors to the overall cost.
- Prioritizing the development and adoption of energy-efficient training protocols that could mitigate the carbon footprint of federated learning systems, thereby supporting the environmental sustainability goals of the organization.

Ultimately, the trade-offs between performance, cost, and environmental considerations should guide the choice of federated learning models. As federated learning technology matures, we anticipate improvements that will address these challenges, leading to more robust, cost-effective, and eco-friendly recommender systems.

7. References

References

[1] Muhammad Ammad-ud din, Elena Ivannikova, Suleiman A. Khan, Were Oyomno, Qiang Fu, Kuan Eeik Tan, and Adrian Flanagan. Federated Collaborative Filtering for Privacy-Preserving Personalized Recommendation System, January 2019. arXiv:1901.09888 [cs, stat]. 3

[2] Arnaud Berlioz, Arik Friedman, Mohamed Ali Kaafar, Roksana Boreli, and Shlomo Berkovsky. Applying differential privacy to matrix factorization. In *Proceedings of the 9th ACM Conference on Recommender Systems*, pages 107–114, 2015. 2

[3] Di Chai, Leye Wang, Kai Chen, and Qiang Yang. Secure federated matrix factorization. *CoRR*, abs/1906.05108, 2019. 3

Method	MAP@10	NDCG@10	Precision@10	Recall@10
FedNeuMF	0.69	0.15	0.12	0.62
NeuMF	0.71	0.70	0.13	0.71

Table 3. Results

Training Strategy	HW	Power Usage (W)	Time per Epoch	Num. Rounds	Energy per device	Total Energy
Centralized NeuMF	t2.large	14.2	4.5	400	7.1	7.1
FedNeuMF	mobile	4.5	1.3	400	0.65	130

Table 5. Power Consumption

[4] Di Chai, Leye Wang, Kai Chen, and Qiang Yang. Secure Federated Matrix Factorization. *IEEE Intelligent Systems*, 36(5):11–20, September 2021. arXiv:1906.05108 [cs]. 3

[5] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural Collaborative Filtering, August 2017. arXiv:1708.05031 [cs]. 2, 3

[6] Jinsu Kim, Dongyoung Koo, Yuna Kim, Hyunsoo Yoon, Junbum Shin, and Sungwook Kim. Efficient privacy-preserving matrix factorization for recommendation via fully homomorphic encryption. *ACM Transactions on Privacy and Security (TOPS)*, 21(4):1–30, 2018. 2

[7] Jakub Konečný, H. Brendan McMahan, Daniel Ramage, and Peter Richtárik. Federated Optimization: Distributed Machine Learning for On-Device Intelligence, October 2016. arXiv:1610.02527 [cs]. 3

[8] Jakub Konečný, H. Brendan McMahan, Felix X. Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated Learning: Strategies for Improving Communication Efficiency, October 2017. arXiv:1610.05492 [cs]. 3

[9] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix Factorization Techniques for Recommender Systems. *Computer*, 42(8):30–37, August 2009. Conference Name: Computer. 2

[10] Guanyu Lin, Feng Liang, Weike Pan, and Zhong Ming. FedRec: Federated Recommendation With Explicit Feedback. *IEEE Intelligent Systems*, 36(5):21–30, September 2021. 3

[11] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data, February 2017. arXiv:1602.05629 [cs]. 3

[12] Lorenzo Minto, Moritz Haller, Benjamin Livshits, and Hamed Haddadi. Stronger privacy for federated collaborative filtering with implicit feedback. In *Proceedings of the 15th ACM Conference on Recommender Systems*, RecSys '21, page 342–350, New York, NY, USA, 2021. Association for Computing Machinery. 3

[13] Javier Fernandez-Marques Pedro P. B. GusmãoYan Gao Daniel J. Beutel Taner Topal Akhil Mathur Nicholas D. Lane nchi Qiu, Titouan Parcollet. A first look into the carbon footprint of federated learning. 2023. 4

[14] Vasileios Perifanis and Pavlos S. Efraimidis. Federated Neural Collaborative Filtering. *Knowledge-Based Systems*, 242:108441, April 2022. 3

[15] Zehua Sun, Yonghui Xu, Yong Liu, Wei He, Yali Jiang, Fangzhao Wu, and Lizhen Cui. A survey on federated recommendation systems. 2023. 3

[16] Pavlos S. Efraimidisa Vasileios Perifanisa. Federated neural collaborative filtering. 2022. 6

[17] Chuhan Wu, Fangzhao Wu, Lingjuan Lyu, Tao Qi, Yongfeng Huang, and Xing Xie. A federated graph neural network framework for privacy-preserving personalization. *Nature Communications*, 13(1), jun 2022. 3

[18] Jingwei Yi, Fangzhao Wu, Chuhan Wu, Ruixuan Liu, Guangzhong Sun, and Xing Xie. Efficient-fedrec: Efficient federated learning framework for privacy-preserving news recommendation. 2021. 3

[19] Senci Ying. Shared MF: A privacy-preserving recommendation system, August 2020. arXiv:2008.07759 [cs, stat]. 3

[20] Chen Zhang, Yu Xie, Hang Bai, Bin Yu, Weihong Li, and Yuan Gao. A survey on federated learning. *Knowledge-Based Systems*, 216:106775, March 2021. 3