IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 21 July 2024, accepted 4 August 2024, date of publication 7 August 2024, date of current version 16 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3439668

RESEARCH ARTICLE

Transformer-Based Federated Learning Models for Recommendation Systems

M. SUJAYKUMAR REDDY, HEMANTH KARNATI, AND L. MOHANA SUNDARI
School of Computer Science and Engineering, Vellore Institute of Technology, Vellore 632014, India
Corresponding author: L. Mohana Sundari (mohanasundari.l@vit.ac.in)

ABSTRACT In today's data-driven environment, safeguarding user privacy is a top priority, particularly in machine learning applications. Our study introduces an innovative approach that combines the privacy-preserving attributes of federated learning with the advanced capabilities of transformer-based models, specifically tailored for recommendation systems. Federated learning emerges as a decentralized alternative to traditional machine learning, enhancing both user privacy and data security. Our research employs two distinct transformer models: BERT (Bidirectional Encoder Representations from Transformers) and BST (Behavior Sequence Transformer), within a federated learning context. The models performance is analyzed using the Amazon Customer Review and movielens-1m datasets. The empirical results are compelling: the federated BERT model achieves a notable 87% and 76% accuracy in the global model for 2 different datasets. Similarly, the federated BST model demonstrates a performance with an mean absolute error of 0.8. This research not only highlights the effectiveness of federated learning in boosting model accuracy but also emphasizes its crucial role in preserving user privacy. Our findings illustrate that integrating federated learning can lead to enhanced performance in recommendation systems without sacrificing data privacy. Consequently, this research marks a significant step forward in developing more effective, privacy-conscious machine learning solutions, contributing to the broader field of ethical and responsible AI.

INDEX TERMS Federated learning, recommendation systems, information privacy, data security, transformers, deep learning.

I. INTRODUCTION

In the evolving digital ecosystem, recommendation systems have become integral in shaping user experiences across diverse platforms, from e-commerce to social networking. These systems are instrumental in tailoring user interactions, as observed by Kim and Chen [1]. They underline the substantial growth in this field, particularly noting advancements in methodologies like collaborative filtering, which have adapted to modern digital trends including social media. Yet, the effectiveness of these recommendation systems is often contingent upon the analysis of substantial user data. This reliance raises significant privacy and data security concerns [2]. Traditional models, primarily based on centralized data collection, pose notable privacy risks despite their efficiency. Such concerns are increasingly relevant amidst evolving data protection regulations and growing public demand for enhanced data privacy. In response, federated learning presents a viable solution. As a distributed machine learning approach, it facilitates learning from decentralized data sources without necessitating direct data sharing, thereby aligning with the surging need for privacy-preserving techniques in machine learning [3]. Applying federated learning to recommendation systems could effectively mitigate the privacy challenges inherent in traditional methodologies.

Transformers, such as BERT (Bidirectional Encoder Representations from Transformers) and BST (Behavior Sequence Transformer), have established new performance standards in machine learning. These models which are typically used in centralized environments, are known for their exceptional efficacy but often rely on centralized data processing. Adapting these transformer models to a federated learning framework represents an innovative shift, with the potential to transform the landscape of recommendation

The associate editor coordinating the review of this manuscript and approving it for publication was Rahim Rahmani.

systems [4]. Despite significant progress in both federated learning and transformer technologies, a gap persists in the research landscape: the exploration of transformer models within a federated learning context, particularly in the domain of recommendation systems. Our study seeks to fill this gap, examining the performance of transformer-based models in both federated and traditional settings. The focal point of this research is to assess the capability of these models to maintain high accuracy in recommendation systems while enhancing user privacy.

The paper's structure is as follows: In Section II, a review of previous research in Federated Learning in Recommendation Systems and its various applications is presented. Section III offers a high-level overview of the general methodology employed in this study, including an analysis of dataset statistics and also provides detailed descriptions of the algorithms and models used in constructing the Recommendation Systems. Finally, Section IV discusses the results obtained by the models.

II. RELATED WORKS

Recommendation systems have evolved significantly since their inception, transitioning from traditional methods to incorporating advanced techniques like deep learning. This evolution is crucial for understanding the context in which transformer-based federated learning models operate within recommendation systems.

A. TRADITIONAL RECOMMENDATION SYSTEMS

Traditional recommendation systems have been classified mainly into collaborative filtering, content-based filtering, and hybrid approaches. Collaborative filtering relies on the assumption that users who agreed in the past tend to agree again in the future. In contrast, content-based filtering recommends items similar to those a user liked in the past, based on item features. Hybrid systems combine these approaches to leverage their respective strengths while mitigating their weaknesses [6].

Several studies have refined these traditional methods. For instance, user-based and item-based collaborative filtering have been extensively optimized to improve scalability and accuracy. Despite their success, traditional methods struggle with privacy issues and limited ability to capture complex user behavior patterns.

B. EVOLUTION TO DEEP LEARNING

The advent of deep learning has brought significant advancements in recommendation systems. Neural networks, with their ability to model complex non-linear relationships, have been increasingly applied to enhance the performance of these systems. This shift is evident in the growing body of literature exploring various deep learning architectures for recommendation purposes [5].

Li et al. [5] provide a comprehensive survey of the recent developments in recommender systems, highlighting the integration of deep learning methods, especially in personalized and group-based systems. Their survey categorizes personalized recommendation systems into collaborative filtering, content-based, knowledge-based, and hybrid systems, noting the increased popularity of deep learning-enhanced methods.

Furthermore, Dong et al. [6] reviews the history of web recommender systems, emphasizing the transformation brought about by the integration of deep learning techniques. Their work traces the progression from early recommendation models to the current state, where deep learning plays a pivotal role. Challenges remain in computational complexity, the need for large datasets, and integrating deep learning with user privacy considerations.

C. CONTEXT-AWARE RECOMMENDER SYSTEMS

Context-aware recommender systems (CARS) represent a significant step forward, incorporating contextual information to refine recommendations. Occhioni et al. [7] discusses the introduction of context analysis techniques in recommender systems, especially highlighting their application in the cultural heritage field. This approach aligns with the current trend towards more personalized and situation-specific recommendation systems, a domain where deep learning, including transformer-based models, shows great promise. Recent studies have successfully integrated contextual data to improve the relevance of recommendations in various domains. Issues such as the complexity of context integration, scalability, and maintaining user privacy remain.

D. FEDERATED LEARNING IN RECOMMENDATION SYSTEMS

The concept of Federated Learning (FL) has emerged as a significant advancement in the realm of machine learning, particularly in the context of decentralized data and privacy preservation.

McMahan et al. [8] introduced the concept of Federated Learning as a method for learning deep networks from decentralized data. They proposed an approach termed FedAvg, which involves aggregating locally-computed updates to learn a shared model while keeping the training data on the mobile devices. This method addresses the privacy concerns and communication efficiency in decentralized settings, laying the groundwork for subsequent research in FL.

Li et al. [9] conducted a comprehensive review of applications in Federated Learning, highlighting the evolution of FL in addressing challenges such as data silos and sensitivity. They explored various optimization paths and identified significant applications of FL in industrial engineering and computer science. Their work provides a foundational understanding of FL's role in diverse fields, including its potential in recommendation systems.

Javeed [10] discussed the integration of FL in Personalized Recommendation Systems (PRS), particularly focusing on security and privacy challenges in next-generation Consumer Electronics (CE). They emphasized how FL enhances data privacy and security in PRS by sharing local recommender parameters while keeping the training data localized.

Jie et al. [11] proposed a federated recommendation system based on historical parameter clustering to address the challenges of non-independent and identically distributed data in FL. Their approach involves a weighted average of historical and global parameters, enhancing the recommendation system's accuracy.

Muhammad et al. [12] introduced FedFast, an innovative technique for accelerating the training of federated recommender systems. By employing a novel sampling technique and active aggregation strategy, FedFast improves the convergence speed and accuracy of federated recommendation models, demonstrating its effectiveness across various benchmark datasets.

Studies have shown FL's effectiveness in collaborative filtering, matrix factorization, and personalized recommendation systems while addressing data privacy. Handling non-IID data, communication overhead, and ensuring model convergence are ongoing challenges.

E. TRANSFORMERS IN RECOMMENDATION SYSTEM

Transformers have revolutionized various domains of machine learning, especially natural language processing (NLP) and recommendation systems. This section reviews key studies that explore the architecture and applications of transformers, providing a background relevant to their integration into federated learning for recommendation systems.

The transformer model, introduced by Vaswani et al. [13], is a deep learning model that primarily utilizes the self-attention mechanism. This mechanism allows the model to weigh the significance of different parts of the input data, which is particularly beneficial in understanding the context in language models.

Devlin et al. [14] introduced Bidirectional Encoder Representations from Transformers (BERT), a groundbreaking model in NLP. BERT's novelty lies in its ability to pretrain deep bidirectional representations from unlabeled text, considering both left and right context. This model has significantly advanced the performance in various NLP tasks, including question answering and language inference, demonstrating the versatility and power of transformers in language understanding.

The application of transformers in recommendation systems has been explored in several studies. For instance, Yang et al. [15] proposed a semantic and explainable research-related recommendation system utilizing BERT and LDA models. Their system effectively embeds and classifies research literature, enhancing the relevance and contextual matching in academic recommendations.

Chen et al. [16] applied the transformer model to improve e-commerce recommendations. Their model, Behavior Sequence Transformer (BST), captures sequential signals in users' behavior sequences, significantly enhancing the Click-Through-Rate (CTR) in Alibaba's recommendation system. This study demonstrates the effectiveness of transformers in capturing complex user behavior patterns for more accurate recommendations.

The reviewed literature underscores the transformative impact of transformer models in NLP and their promising applications in recommendation systems. These studies provide a solid foundation for exploring the integration of transformer-based models in federated learning environments, as addressed in our research.

F. FEDERATED LEARNING WITH TRANSFORMERS FOR RECOMMENDATION SYSTEMS

The integration of federated learning with transformer models has been a recent focus in the machine learning community, particularly for applications in recommendation systems. This section reviews key studies that demonstrate the merging of these two technologies and discusses their potential benefits and challenges.

Li et al. [17] introduced FedTP, a novel framework that integrates transformers with federated learning, focusing on personalized self-attention for each client. They identified that federated averaging algorithms could negatively impact self-attention in non-IID (independent and identically distributed) data scenarios. To overcome this, FedTP employs a hypernetwork on the server, which outputs personalized projection matrices for self-attention layers, thereby generating client-specific queries, keys, and values. This approach not only improves the performance in non-IID scenarios but also enhances the model's robustness and scalability. However, they acknowledge that the complexity of transformer-based methods can lead to slower training speeds, posing a challenge for efficiency.

Wei et al. [18] presented KG-FedTrans4Rec, a model that combines knowledge graphs with transformer-based sequential recommendation systems in a federated learning context. This model leverages Graph Convolutional Networks (GCNs) and two-stream self-attention strategies to enhance recommendation accuracy while preserving user privacy. Their findings indicate superior performance compared to various models, although they note a trade-off between enhanced privacy and accuracy. This study highlights the potential of integrating knowledge graphs and transformers within the federated learning framework for recommendation systems.

The integration of federated learning with transformer models offers significant advantages, particularly in terms of privacy preservation and personalized recommendation capabilities. Transformers ability to handle complex data patterns complements federated learning's distributed nature, allowing for more efficient and effective recommendation systems. However, challenges such as the trade-off between privacy and accuracy, as well as the increased computational and communication overhead, are crucial considerations in this integration.

These studies showed that combining transformers with federated learning can significantly enhance recommendation

accuracy while preserving privacy. Issues such as the trade-off between privacy and accuracy, increased computational and communication overhead remain critical challenges. The reviewed literature provides insights into the innovative applications of transformers in federated learning environments, particularly in recommendation systems. These studies form a solid foundation for exploring the development of efficient, private, and accurate recommendation systems using transformer-based federated learning models.

G. OUR CONTRIBUTION

Our study addresses these gaps by:

- Integrating federated learning with BERT and BST models to create a novel, privacy-preserving recommendation system.
- Demonstrating significant performance improvements using the Amazon Customer Review datasets, with federated BERT achieving 87% and 76% accuracy in the global model and Movielens dataset for federated BST achieving MAE of 0.8.
- Providing empirical evidence that federated learning can enhance model performance while maintaining user privacy, advancing the current state-of-the-art in recommendation systems.

III. PROPOSED METHODOLOGY

This paper introduces two recommendation systems that leverage FL. The proposed models are Transformers and Feed-Forward Networks, to enhance the efficiency and effectiveness of the recommendation process. The Movie Recommendation system uses two types of model architectures, the BST and Feed-Forward models, as its global and local models. In order to train our global model, FL is used to aggregate small client models that are trained on non-independent and non-identically distributed (non-iid) data. The weights of these models are then transferred to the server for averaging the weights. For product recommendation, we use the BERT and Feed-Forward Neural Network models, leveraging their strength in text classification.

The Federated Averaging Adam Algorithm in both of our applications, which has been proven to be robust in various studies [22], [23]. This ensures optimal performance and precision for the recommendation systems. To develop a Movie Recommendation System using Federated Learning (FL), the methodology illustrated in Figure 1 is adopted. This methodology involves three primary entities: Users, Ratings, and Movies. The dataset is organized by partitioning the movie IDs into sequences. To control the number of sequences generated for each user, sequence length and step size parameters are utilized.

The dataset consists of three key tables: User, Ratings, and Movies. The Users table encompasses user-id, gender, age group, and occupation, while the Movies table includes movie-id, title, and genre. The Ratings table contains user-id, movie-id, ratings, and timestamps. To prepare the data for modeling, sequences of movie IDs and their corresponding ratings are generated. The output is then reorganized to ensure that each sequence is represented as a separate record in the DataFrame, linking user characteristics with the rating data to capture the nuances of user preferences. Non-iid datasets are created for each client, enabling local training. The Federated Averaging Algorithm is employed to aggregate insights from these client models into the global model, optimizing the Movie Recommendation System for diverse user preferences and characteristics.

Figure 2 presents our Federated Learning-based Product Recommendation System. This system serves as a predictive filter, displaying items that users are likely to purchase, thereby enhancing the user experience with personalized suggestions. Leveraging Federated Learning, our system draws insights from diverse models, refining recommendations while prioritizing user privacy. These algorithms significantly contribute to a positive user experience by tailoring product suggestions to individual preferences. Our product recommendation system relies on text data extracted from reviews of each product, along with a corresponding rating system which assigns a score out of five. The system addresses the issue of class imbalance by assigning a value of one to ratings equal to or above three, while lower ratings are assigned a value of zero. This approach enables the system to provide more accurate recommendations based on user feedback.

A. DATASET PREPARATION

Movielens 1m dataset [20] is used for Movie Recommendation system and Amazon review dataset [19] for the Product Recommendation. The moveielens dataset consists of 3 other tables Users, ratings, Movies which are joined using the userID resulting in a table of total 6040 unique users and highest number of ratings given by the user-4169 with 1157 ratings out of 1 million. Table 2 presents a summary of the dataset, offering an overview of various movie-related attributes. The dataset spans movies from 1939 onwards, encompassing a diverse range of 18 genres, with the distribution of these genres across different years visually represented in Figure 3. Table 1 provides a comprehensive overview of the data generation following the sequencing process. For each user, a sequence is crafted, encompassing movie IDs and their corresponding ratings. Age is categorized into seven distinct groups, while gender is denoted as Male (M) or Female (F). Furthermore, occupation is classified into 21 diverse categories resulting in the length of 498623 rows and 6 columns.

Following the data creation process, each row in the dataset represents a user's interactions, encapsulating the movies they have viewed and their corresponding ratings. Notably, the final movie and its associated rating are isolated from the remaining sequence, serving as the prediction target for the model. To facilitate efficient learning, the processed data is organized into batches.For Transformer-based models, input features must be encoded as embedding vectors. To this

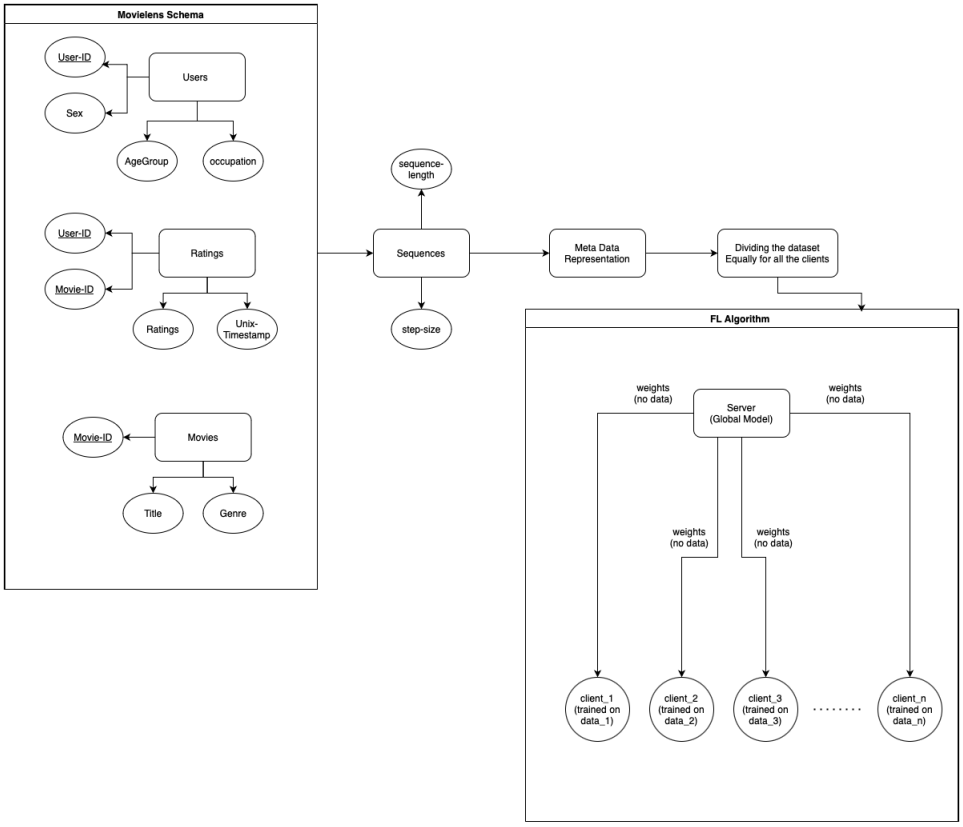


FIGURE 1. Proposed methodology of movie recommendation model by using movielens dataset.

TABLE 1. Sample table from the Movielens dataset, which has undergone sequencing and metadata creation.

user_id	sequence_movie_ids	sequence_ratings	sex	age_group	occupation
user_1	movie_3186,movie_1721,movie_1270,movie_1022	4.0,4.0,5.0,5.0	F	group_1	occupation_10
user_1	movie_1270,movie_1022,movie_2340,movie_1836	5.0,5.0,3.0,5.0	F	group_1	occupation_10
user_1	movie_2340,movie_1836,movie_3408,movie_1207	3.0,5.0,4.0,4.0	F	group_1	occupation_10
user_1	movie_3408,movie_1207,movie_2804,movie_260	4.0,4.0,5.0,4.0	F	group_1	occupation_10
user_1	movie_2804,movie_260,movie_720,movie_1193	5.0,4.0,3.0,5.0	F	group_1	occupation_10
...	...	...	...	...	...

TABLE 2. Summary of the Movielens dataset.

Name	Statistics
Number of users	6040
Number of movies	4052
Number of reviews	1000209
Click/browse	6976551
Review	354016
Favorites	72604
Like	244740
Total user actions	7647911
Dataset sparsity	0.958

end, the StringLookup Layer [24] is applied to convert string representations to integer equivalents. The embedding dimension is set to the square root of the vocabulary size ($\sqrt{|V|}$) [16]. This choice strikes a balance between representation capacity and model complexity. Insufficient embedding dimensions may limit the layer's representational capabilities, while excessively large dimensions introduce unnecessary complexity and computational overhead. Similarly, movie features, target movies, and sequence movie IDs are encoded. Subsequently, a single embedding vector for user features is created.

TABLE 3. Summary of the amazon software and sports product dataset.

Name	Software Statistics	Sports Statistics
Total Reviews	12805	2610134
1-Rating	1500	98020
2-Rating	719	91986
3-Rating	1598	192521
4-Rating	3016	450050
5-Rating	5972	1777557

The Product Recommendation System utilizes two subsets of comprehensive datasets, Software and Sports and Outdoors 5-core datasets. This framework operates within the context

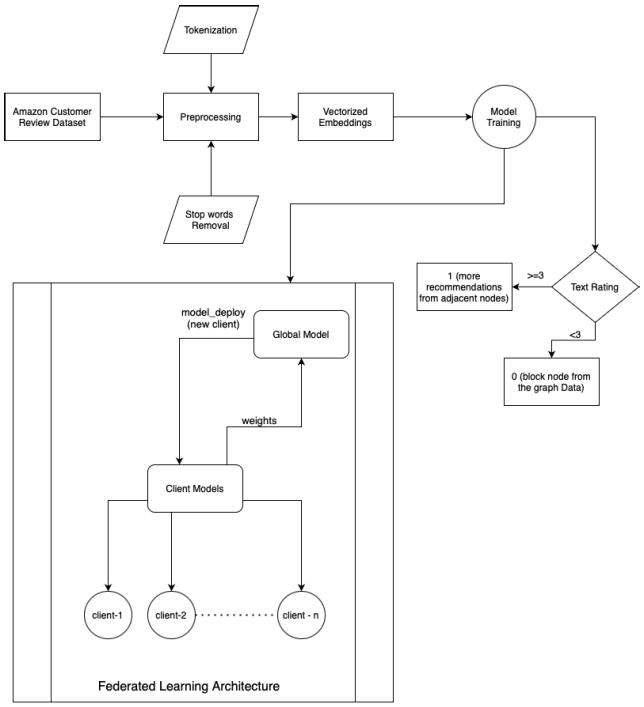


FIGURE 2. Proposed methodology of product recommendation model by using amazon review dataset.

TABLE 4. Sample of the software product recommendation dataset.

ReviewText	Ratings
Microsoft Office is still the standard all the...	1
Been using Office for over twenty years. Still...	0
Always liked Office Products. Expensive for a...	1
I have been using MS Office professional for d...	1
This is really a great buy. It breaks down to...	1
...	...

TABLE 5. Sample of the sports and outdoors product recommendation dataset.

ReviewText	Ratings
What a spectacular tutu! Very slimming.	1
What the heck? Is this a tutu for nuns? I know...	0
Exactly what we were looking for!	1
I used this skirt for a Halloween costume and ...	1
This is thick enough that you can't see through...	1
...	...

of an E-commerce chatbot, enabling users to provide reviews about products while ensuring data decentralization and is employed when a user inputs a prompt/text to a chatbot. The algorithm classifies the input as either positive or negative. If positive, the chatbot recommends products that are closely associated (i.e., neighboring nodes) in the graph database. Conversely, if negative, it suggests products that are more distant from the input node. The dataset contains two subsets, one with 12,805 reviews and the other with 2,610,134 reviews (using only 40,000 samples to evaluate the model's performance for a larger sample, the ratings distribution remains the same as in the comprehensive dataset). These subsets are derived from a comprehensive dataset containing reviews where all users and items have a minimum of 5 reviews. Table 4 and 5 gives the sample dataset which is used for modelling. Our analysis focuses exclusively on reviews from Verified Users, excluding those from non-verified users. Table 3 provides a detailed overview of the datasets employed in our study.

In the analysis, a word cloud (Figure 4) is presented to visualize the most frequent words in ratings categorized as 1 (greater than or equal to 3) and 0 (less than 3). To transform text into vector embeddings, The BERT Tokenizer from Hugging Face [25] is applied. This tokenizer converts text sequences into input tokens, facilitating the creation of fixed-size input sequences. Padding is applied for shorter sequences, while longer sequences are truncated. The tokenizer assigns a unique ID to each token in the vocabulary and generates an attention mask, indicating which parts of the input consist of actual tokens and which parts are padding.

B. FEDERATED LEARNING

In the realm of Federated Learning (FL), the Adaptive Federated Optimization (FedAvgOpt) algorithm stands out as a prevalent approach for training ML models across a multitude of decentralized devices. This algorithm involves a series of iterative communication rounds between a central server and numerous local devices, commonly referred to as clients. At its core, FedAvgOpt aims to address an optimization problem by harnessing the Optimizers to Clients technique. The foundation of this framework comes from the work by Reddi et al. [27]. In their study, they introduced the integration of Yogi, AdaGrad, and Adam optimizers into the federated learning process to train a global model. The results they obtained were exceptionally promising, demonstrating the potential of this approach. Building upon this foundational research, our work delves deeper into the framework of federated optimization utilizing server and client optimizers with an array of algorithms, encompassing BERT and Feed-Forward Neural Networks. By leveraging this framework, we meticulously design novel adaptive federated optimization applications for movie recommendation systems and product recommendation. The intricate parameters employed in the construction of these models are meticulously outlined in Table 3. To our knowledge, our proposed methods represent the first applications of adaptive server optimization in the context of FL. Through extensive experimentation, we have comprehensively evaluated the performance of these methods across a diverse set of benchmark datasets.

TABLE 6. Federated adam averaging parameters.

Parameters
Number of Clients
Number of Rounds
Batch Size
Epochs per Round
Learning Rate

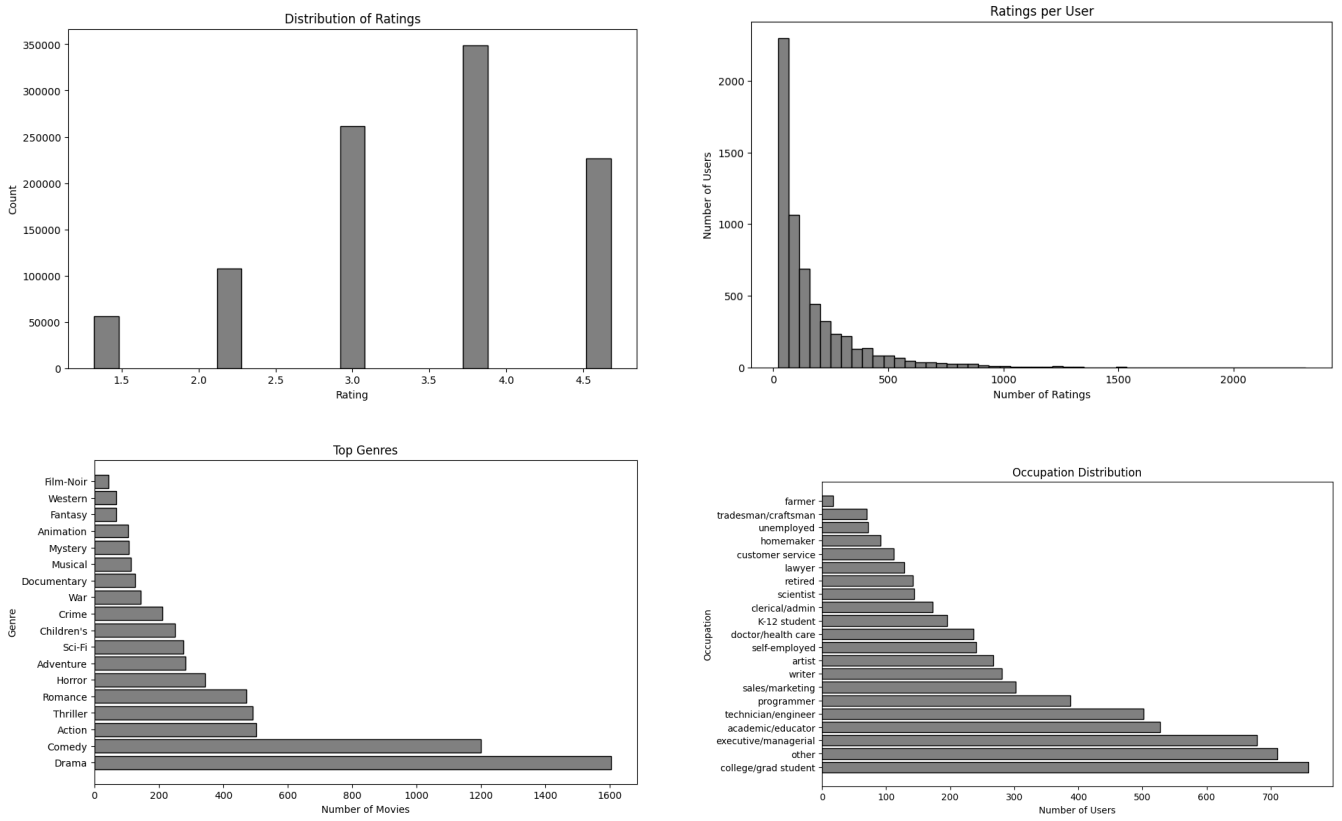
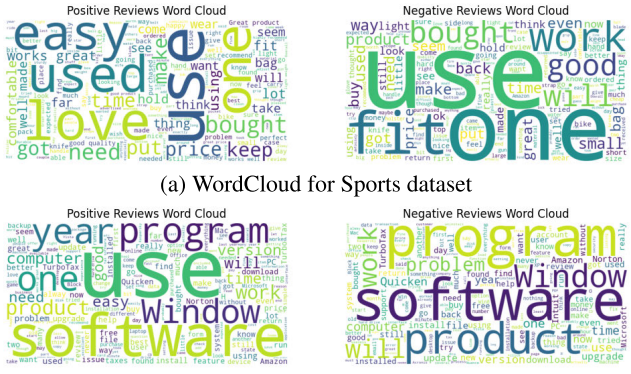


FIGURE 3. Data analysis plots for movielens dataset (ratings, users and movies).



(a) WordCloud for Sports dataset

(b) WordCloud for Software dataset

FIGURE 4. Word cloud for product recommendation datasets.

1) FEED-FORWARD

In the context of movie recommendation systems, a deep learning architecture is employed, consisting of two dense linear layers following the user, ratings, and movie embedding layers. This architecture incorporates additional features such as occupation, age, and gender to enhance prediction accuracy. The user embedding layer captures the user's preferences and characteristics, while the movie embedding layer encodes the attributes of each movie. The ratings layer represents the user's ratings for previously watched movies. The subsequent dense linear layers leverage these embedded representations to learn complex relationships between users, movies, and their associated features, enabling the model to make personalized movie recommendations tailored to each user's preferences and demographics.

In contrast, for product recommendations, a similar architecture with dense feed-forward layers is utilized, with differences in the input layer based on the nature of product data. Instead of user, movie, and rating embeddings, a text-based input layer is employed. This layer processes product descriptions or reviews using TF-IDF vectorization, converting them into numerical vectors. The subsequent dense linear layers then operate on these vectors to extract meaningful features and make product recommendations.

2) BERT

The Bidirectional Encoder Representations from Transformers (BERT) model, introduced by Devlin et al. [14], has revolutionized the field of Natural Language Processing (NLP). BERT's innovative approach to capturing contextualized word representations within sentences has propelled it to the forefront of NLP tasks. At its core, BERT employs self-attention layers [13] to establish intricate relationships between words, enabling it to derive rich semantic understanding from text. As a transformer-based model, BERT operates solely as an encoder, utilizing token,

segment, and position embeddings to encode input sequences. This encoding process is crucial for downstream tasks such as question answering and natural language inference, where understanding the context of words is paramount. Notably, the versatility of BERT extends beyond NLP, with applications in text processing, genomics, and a wide range of classification tasks, as highlighted by Mohammed and Ali [21].

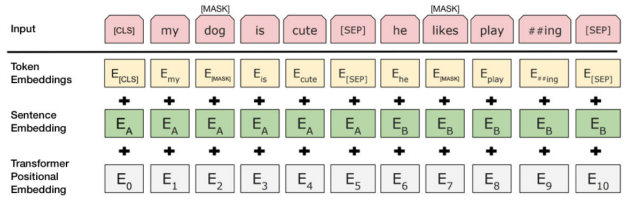


FIGURE 5. BERT input representation.

Despite the widespread adoption of BERT, there is a paucity of research exploring the integration of FL with transformer-based models. FL offers a compelling paradigm for collaborative learning across multiple devices or organizations while preserving data privacy. Our proposed system, named FedBERT, can harness the collective knowledge from diverse data sources without compromising sensitive information. By combining the strengths of FL and BERT, we can unlock new possibilities for NLP tasks and Large Language Models (LLMs), particularly in scenarios where data is distributed across multiple parties and privacy is a primary concern.

During the process of fine-tuning BERT, we rely on the Hugging Face Transformers package [26] to customize it for product recommendation. The first task, known as "Masked LM," involves randomly masking a certain percentage of input tokens and then predicting those tokens. This allows BERT to form a deep bidirectional representation. The second task, "Next Sentence Prediction (NSP)," focuses on predicting whether a given sentence follows another sentence or not. The BERT-base model is made up of a 12-layer Transformer encoder with 12 self-attention heads and a hidden size of 768. It can process input sequences with a maximum length of 512 tokens, generating sequence representations. Each sequence consists of one or two segments, with the first token being [CLS] for classification and [SEP] for segment separation. When it comes to text classification tasks, the final hidden state (h) of the first token ([CLS]) is used to represent the entire sequence. Additionally, a softmax classifier is added to BERT for label prediction.

3) BST

In the realm of recommendation systems, the Behavioral Sequence Transformer (BST) model, introduced by Chen et al. [16], stands out as a robust and effective solution. It leverages the power of Transformers to capture the intricate sequential signals present in users' behavioral sequences. The primary objective of BST is to enhance recommendation accuracy and relevance by delving into nuanced patterns within these sequences, ultimately elevating the overall performance of recommendation systems. We adopt the BST model as a key component in the evaluation of our Federated Learning-based Movie Recommendation System.

Our system architecture, as shown in Figure 3, is made up of three main layers. The Embedding layers are where we utilize 5 feature embeddings (User, Movie, Sex, Age-Group, Occupation), which are specified in the creation of metadata. These layers play a vital role in reconfiguring data features that are extracted from user-item interactions. They amalgamate the features into a unified input vector, which is crucial for processing the data effectively. Subsequently, the Transformer Layer for encoders employs self-attention mechanisms to effectively combine signals derived from users' prior interactions. Additionally, an MLP (Multi-Layer Perceptron) with three fully connected layers is integrated to enhance the understanding of interactions among dense features, a standard practice in industrial recommendation systems.

In the design of our model, categorical user features are encoded using embeddings layers with the embedding dimension set as the square root of the vocabulary size. These feature embeddings are then concatenated into a single input tensor. Movies in the sequence, including the target movie, undergo encoding using embeddings layers with the dimension being the square root of the total movie count. Movie representations are further enriched by concatenating multi-hot genre vectors with their embeddings, and the combined vectors are processed using non-linear Dense layer. Temporal information is incorporated through positional embeddings for each movie, multiplied by their corresponding ratings. The embedding of the target movie is concatenated with the sequence embeddings to create a tensor suitable for the transformer's attention layer.

C. OPTIMIZERS AND EVALUATION METRICS

To comprehensively evaluate the performance of the proposed methodology, which leverages the BST model for movie recommendations and BERT for text classification, specific evaluation metrics have been devised for both prediction and classification tasks. The Adam optimizer (Adaptive Moment Estimation) is employed to enhance the effectiveness of the models. This optimizer combines the advantages of RMSProp (Root Mean Square Propagation) and AdaGrad (Adaptive Gradient Algorithm), making it a popular choice for training deep neural networks. The Adam optimizer has demonstrated effectiveness in a wide range of deep learning tasks and is often the optimizer of choice, as shown in equation (1). To address the FL optimization problem, it is essential to minimize the gap between the actual (target) values and the predicted values. To determine the performance of regression models, we commonly use the Mean Squared Error (MSE) metric, which can be expressed mathematically as equation (2). For BST, we make use of the loss function, while for BERT, we use the Binary

Cross Entropy Loss (BCELoss). To assess the performance of our model, we employ two evaluation metrics: Accuracy and Mean Absolute Error (MAE). Accuracy measures the proportion of correct predictions, while MAE quantifies the average magnitude of errors in predictions. The mathematical formulations of Accuracy and MAE are presented in equations (3) and (4), respectively. By utilizing these evaluation metrics, we can comprehensively evaluate the performance of our model in both prediction and classification tasks. This allows us to gain insights into the effectiveness of our methodology and identify areas for potential improvement.

$$w_{t+1} = w_t - \alpha \cdot \frac{m_t}{\sqrt{v_t} + \epsilon} - \alpha \cdot \text{weight\_decay} \cdot w_t \quad (1)$$

where w_{t+1} represents the updated parameter value, w_t is the current parameter value, α is the learning rate, set to 1×10^{-5}, m_t is the moving average of the gradient, $\sqrt{v_t}$ is the square root of the moving average of the squared gradient, ϵ is a small constant used to prevent division by zero, weight_decay is the weight decay coefficient, introducing L2 regularization.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \quad (2)$$

where, n is the number of data points, y_i represents the true target value for the i-th data point and y_hat represents the predicted value for the i-th data point.

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} \quad (3)$$

Accuracy is a metric used to measure the overall correctness of the model's predictions, with higher accuracy indicating better performance

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \quad (4)$$

The MAE measures the average absolute difference between the predicted values $\hat{y}_i$ and the actual values y_i across all data points in your dataset.

IV. RESULTS AND DISCUSSION

This section presents the experimental results and discuss the findings of our study on Federated Learning Recommendation Systems using Transformers.Adam optimizer is utilized in both experiments. For the FedAvg algorithm [27], we evenly split the dataset among participating clients and trained each local model using the Adam optimizer. Subsequently, we aggregated the weights contributed by each client and saved them onto our global model. Through federated training, distinct client models are trained independently without sharing their local data. The results indicate that the FL Recommendation system which uses Transformer model outperforms baseline models in terms in accuracy. This approach effectively addresses the concerns for traditional Feed Forward models without comprising user data privacy.

The training pipeline starts by loading the dataset and dividing it into 75% for the federated training process and 25% for testing the global model after weight aggregation. A single round of weights aggregation in both experiments, ensuring an even distribution of the dataset among the specified number of clients, as defined by the hyper-parameter. This choice aims to prevent potential over-fitting, which can occur when using the same dataset in multiple consecutive rounds of training. Such an approach helps maintain a balanced performance and generalization of the model.

During the training of each client model, we evenly split the dataset to create decentralized data (non-iid), ensuring no sharing of data between the global model on the server and the client model on the user's device. The client model is constructed with the same architecture as the global model to facilitate proper weight aggregation and avoid potential system intrusion detection issues. Fed-Average-Opt [27] introduces a new research gap where 2 optimizers are used in client and server side which produces different outcomes. To avoid this problem, this paper utilizes the same Adam as the optimizer during the local model or client training. Once the local model architecture is set up, the decentralized data is further divided into an 80:20 ratio. The 80% portion is utilized to train the local model, while the remaining 20% is reserved to assess each client model's accuracy in prediction or classification. After the training process, we save the model weights, which are then aggregated with weights from other client models and transferred to the global model. Table 13 provides an overview of the fixed hyper-parameters utilized in this paper.

A. PRODUCT RECOMMENDATION SYSTEM

The Amazon Product datasets, particularly in the Software and Sports categories, exhibited inherent challenges associated with class imbalance. To address the intricacies of data heterogeneity within the FL framework, sampling techniques were deliberately avoided. However, to optimize resource utilization and to decrease load on user devices, the epochs and learning rate were maintained at 10 and 0.01, respectively. Given the binary nature of the classification task, the Binary Cross Entropy Loss function was uniformly applied across all models.

Tables 7 and 8 show how well the BERT and Feed Forward models did with the Software dataset. Tables 9 and 10 do the same for the Sports and Outdoors dataset. One thing to notice is that the Feed Forward model tends to favor class 1 (the majority class), which points to a bias problem. On the other hand, the Transformer BERT model performed better, beating the feed-forward neural network in this situation. The global software dataset demonstrated a notable enhancement in accuracy, achieving 87% accuracy with the BERT model compared to the comparatively inferior performance of the feed-forward model at 50%. Likewise, in the Sports and Outdoors dataset, BERT exhibited a significant accuracy improvement at 76%, surpassing the feed-forward model's accuracy of 43%. These findings underscore BERT's effectiveness in

TABLE 7. Results report for federated BERT amazon software dataset.

Clients s.no	Class 0 Precision	Class 0 Recall	Class 0 F1-Score	Class 1 Precision	Class 1 Recall	Class 1 F1-Score
1	0.00	0.00	0.00	0.98	1.00	0.99
2	0.02	0.33	0.04	0.67	0.07	0.13
3	0.04	0.50	0.08	0.82	0.16	0.27
4	0.33	0.08	0.13	0.81	0.96	0.88
5	0.33	0.14	0.20	0.89	0.96	0.93
6	0.00	0.00	0.00	0.79	0.84	0.81
7	0.00	0.00	0.00	0.84	0.90	0.87
8	0.10	1.00	0.18	1.00	0.34	0.51
9	0.00	0.00	0.00	0.83	1.00	0.91
10	0.14	0.40	0.21	0.81	0.52	0.63

TABLE 8. Results report for federated feed-forward neural network amazon software dataset.

Clients s.no	Class 0 Precision	Class 0 Recall	Class 0 F1-Score	Class 1 Precision	Class 1 Recall	Class 1 F1-Score
1	0.0	0.0	0.0	0.38	1.0	0.56
2	0.0	0.0	0.0	0.62	1.0	0.76
3	0.0	0.0	0.0	0.42	1.0	0.59
4	0.0	0.0	0.0	0.38	1.0	0.56
5	0.0	0.0	0.0	0.52	1.0	0.68
6	0.0	0.0	0.0	0.60	1.0	0.75
7	0.0	0.0	0.0	0.52	1.0	0.68
8	0.0	0.0	0.0	0.36	1.0	0.53
9	0.0	0.0	0.0	0.56	1.0	0.72
10	0.0	0.0	0.0	0.40	1.0	0.57

TABLE 9. Results report for federated BERT amazon sports dataset.

Clients s.no	Class 0 Precision	Class 0 Recall	Class 0 F1-Score	Class 1 Precision	Class 1 Recall	Class 1 F1-Score
1	0.09	0.66	0.15	0.96	0.51	0.67
2	0.18	0.44	0.25	0.96	0.86	0.90
3	0.10	0.90	0.17	0.98	0.36	0.53
4	0.10	0.51	0.16	0.96	0.70	0.81
5	0.07	0.55	0.12	0.96	0.60	0.74
6	0.08	0.32	0.13	0.94	0.76	0.84
7	0.09	0.48	0.15	0.96	0.72	0.83
8	0.08	0.33	0.14	0.96	0.81	0.88
9	0.07	0.81	0.13	0.97	0.40	0.57
10	0.11	0.37	0.17	0.96	0.85	0.90

TABLE 10. Results report for federated feed-forward neural network amazon sports dataset.

Clients s.no	Class 0 Precision	Class 0 Recall	Class 0 F1-Score	Class 1 Precision	Class 1 Recall	Class 1 F1-Score
1	0.0	0.0	0.0	0.38	1.0	0.56
2	0.0	0.0	0.0	0.62	1.0	0.76
3	0.0	0.0	0.0	0.42	1.0	0.59
4	0.0	0.0	0.0	0.38	1.0	0.56
5	0.0	0.0	0.0	0.52	1.0	0.68
6	0.0	0.0	0.0	0.60	1.0	0.75
7	0.0	0.0	0.0	0.52	1.0	0.68
8	0.0	0.0	0.0	0.36	1.0	0.53
9	0.0	0.0	0.0	0.56	1.0	0.72
10	0.0	0.0	0.0	0.40	1.0	0.57

addressing class imbalance and mitigating bias towards minority classes, leading to a substantial improvement in classification accuracy. This heterogeneity stems from the unique characteristics and distributions of data collected from diverse sources, leading to distinct patterns of learning and convergence. The decentralized nature of the training process allows each client to leverage its local data, resulting in models that are specifically tailored to their respective data distributions.

B. MOVIE RECOMMENDATION SYSTEM

In our second experiment, we customized the Movielens 1M dataset to implement a Federated Transformer model for a movie recommendation system. The choice of regression evaluation metrics in this model aligns with the inherent nature of the movie recommendation task. The primary objective in this task is to accurately predict continuous values, specifically user ratings. Regression metrics, such as Mean Squared Error (MSE) and Mean Absolute Error

TABLE 11. Regression prediction metrics for for federated feed-forward neural network using Movielens-1m dataset.

Clients s.no	MeanAbsoluteError	MeanAbsolutePercentageError	R2Score	MeanSquaredError	RootMeanSquaredError
1	0.861271	34.193677	0.121443	1.110641	1.053869
2	0.844557	33.979844	0.128833	1.083926	1.041118
3	0.838203	34.173737	0.130822	1.078016	1.038275
4	0.843004	34.873277	0.123515	1.094560	1.046212
5	0.837860	35.227289	0.118945	1.099636	1.048635
6	0.856534	33.510565	0.120854	1.095998	1.046899
7	0.850350	33.967662	0.122866	1.093216	1.045570
8	0.844025	34.068765	0.130675	1.085979	1.042103
9	0.858443	33.897860	0.112239	1.105887	1.051612
10	0.845914	34.077291	0.127015	1.088638	1.043378

TABLE 12. Regression prediction metrics for federated BST network using Movielens-1m dataset.

Clients s.no	MeanAbsoluteError	MeanAbsolutePercentageError	R2Score	MeanSquaredError	RootMeanSquaredError
1	0.858673	35.574612	0.167635	1.123865	1.060125
2	0.877021	31.635834	0.062347	1.123744	1.060068
3	0.838705	31.783789	0.110575	1.055081	1.027172
4	0.857418	32.303276	0.086534	1.102805	1.050145
5	0.858730	35.612671	0.126721	1.146216	1.070615
6	0.854401	32.466656	0.097632	1.108166	1.052694
7	0.872625	32.748936	0.072957	1.138216	1.066872
8	0.866132	31.976746	0.074363	1.106160	1.051741
9	0.850114	33.112431	0.112652	1.111786	1.054413
10	0.841249	32.768108	0.108808	1.091704	1.044846

TABLE 13. Federated adam averaging hyper-parameters.

Parameters	Values
Number of Clients	10
Number of Rounds	1
Batch Size	256
Epochs per Round (local models)	10
Learning Rate	0.01

(MAE), offer a more meaningful and interpretable evaluation of the model's performance in this context compared to accuracy. Unlike classification tasks where accuracy measures the correctness of predictions for discrete classes, the regression metrics used in this system assess how well the model predicts the exact numerical values of user ratings. Movie ratings are inherently continuous and ordinal, making regression metrics a more suitable choice for quantifying the prediction accuracy. Therefore, the decision to employ regression metrics in this movie recommendation system is rooted in the task's nature and the need for precise evaluation of the model's ability to predict user ratings accurately. The BST global model exhibited a Mean Absolute Error (MAE) of 0.8, while the Feed-Forward model displayed a slightly higher MAE of 0.865.

Tables 11 and 12 display the regression metrics for each client's evaluation. From these tables, we can see that the error rates differ among clients. This indicates that, in addition to the known issue of system heterogeneity in FL, the specific model architecture also plays a role in performance variation. The varying error rates suggest that certain clients perform better than others. This difference is not only due to the expected variations in federated systems but is also influenced by the choices made in designing the transformer model. While system heterogeneity is a recognized challenge in FL, the impact of the model's design on individual client performance adds an extra layer of complexity. It implies that the effectiveness of the chosen transformer architecture differs across clients, affecting the overall performance of the FL system. These findings in this paper highlight the need to consider both system differences and model design when working with FL. Striking the right balance between adapting to inherent system variations and tailoring the model to diverse client characteristics is crucial for achieving optimal performance.

V. CONCLUSION

This work opens up new avenues for research in Federated Learning (FL), particularly in the realm of adaptive optimization. The adaptive federated optimization framework developed here is expected to be a valuable tool for the FL community, facilitating the creation of more efficient and effective FL algorithms. Our research has commenced with the deployment of transformers in limited-scale applications. For Federated Learning to thrive in real-time applications, we must confront the challenges of Data Heterogeneity and Communication Overhead. Data Heterogeneity necessitates robust methods for harmonizing diverse data sources, while Communication Overhead requires streamlined protocols and reduced information exchange. In the future, our research will concentrate on the practical deployment of Federated models within real-time systems. This initiative will provide a comprehensive and clear perspective on the security aspects of Machine Learning systems as well as increased personalization.

REFERENCES

[1] M. C. Kim and C. Chen, "A scientometric review of emerging trends and new developments in recommendation systems," *Scientometrics*, vol. 104, no. 1, pp. 239–263, Jul. 2015.

[2] I. Mazeh and E. Shmueli, "A personal data store approach for recommender systems: Enhancing privacy without sacrificing accuracy," *Exp. Syst. Appl.*, vol. 139, Jan. 2020, Art. no. 112858.

[3] M. Ammad-Ud-Din, E. Ivannikova, S. A. Khan, W. Oyomno, Q. Fu, K. E. Tan, and A. Flanagan, "Federated collaborative filtering for privacy-preserving personalized recommendation system," 2019, *arXiv:1901.09888*.

[4] S. Latifi, D. Jannach, and A. Ferraro, "Sequential recommendation: A study on transformers, nearest neighbors and sampled metrics," *Inf. Sci.*, vol. 609, pp. 660–678, Sep. 2022.

[5] Y. Li, K. Liu, R. Satapathy, S. Wang, and E. Cambria, "Recent developments in recommender systems: A survey," 2023, *arXiv:2306.12680*.

[6] Z. Dong, Z. Wang, J. Xu, R. Tang, and J. Wen, "A brief history of recommender systems," 2022, *arXiv:2209.01860*.

[7] D. Occhioni, A. Ferrato, C. Limongelli, M. Mezzini, G. Sansonetti, and A. Micarelli, "Eyeing the visitor's gaze for artwork recommendation," in *Proc. 31st ACM Conf. User Model., Adaptation Personalization*, Jun. 2023, pp. 374–378.

[8] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. 20th Int. Conf. Artif. Intell. Statist.*, 2017, pp. 1273–1282.

[9] L. Li, Y. Fan, M. Tse, and K. Y. Lin, "A review of applications in federated learning," *Comput. Ind. Eng.*, vol. 149, Nov. 2020, Art. no. 106854.

[10] D. Javeed, "Quantum-empowered federated learning and 6G wireless networks for IoT security: Concept, challenges and future directions," *Quantum*, vol. 96, 2023.

[11] Z. Jie, S. Chen, J. Lai, M. Arif, and Z. He, "Personalized federated recommendation system with historical parameter clustering," *J. Ambient Intell. Humanized Comput.*, vol. 14, no. 8, pp. 10555–10565, Aug. 2023.

[12] K. Muhammad, Q. Wang, D. O'Reilly-Morgan, E. Tragos, B. Smyth, N. Hurley, J. Geraci, and A. Lawlor, "FedFast: Going beyond average for faster training of federated recommender systems," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2020, pp. 1234–1242.

[13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.

[14] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.

[15] N. Yang, J. Jo, M. Jeon, W. Kim, and J. Kang, "Semantic and explainable research-related recommendation system based on semi-supervised methodology using BERT and LDA models," *Exp. Syst. Appl.*, vol. 190, Mar. 2022, Art. no. 116209.

[16] Q. Chen, H. Zhao, W. Li, P. Huang, and W. Ou, "Behavior sequence transformer for e-commerce recommendation in Alibaba," in *Proc. 1st Int. Workshop Deep Learn. Pract. High-Dimensional Sparse Data*, Aug. 2019, pp. 1–4.

[17] H. Li, Z. Cai, J. Wang, J. Tang, W. Ding, C.-T. Lin, and Y. Shi, "FedTP: Federated learning by transformer personalization," *IEEE Trans. Neural Netw. Learn. Syst.*, 2023.

[18] S. Wei, S. Meng, Q. Li, X. Zhou, L. Qi, and X. Xu, "Edge-enabled federated sequential recommendation with knowledge-aware transformer," *Future Gener. Comput. Syst.*, vol. 148, pp. 610–622, Nov. 2023.

[19] J. McAuley and J. Leskovec, "Hidden factors and hidden topics: Understanding rating dimensions with review text," in *Proc. 7th ACM Conf. Recommender Syst.*, Oct. 2013, pp. 165–172.

[20] F. M. Harper and J. A. Konstan. *MovieLens 1M Dataset*. Accessed: Nov. 2023. [Online]. Available: https://grouplens.org/datasets/movielens/1m/

[21] A. H. Mohammed and A. H. Ali, "Survey of BERT (bidirectional encoder representation transformer) types," *J. Phys., Conf.*, vol. 1963, no. 1, Jul. 2021, Art. no. 012173.

[22] A. Nilsson, S. Smith, G. Ulm, E. Gustavsson, and M. Jirstrand, "A performance evaluation of federated learning algorithms," in *Proc. 2nd Workshop Distrib. Infrastructures Deep Learn.*, Dec. 2018, pp. 1–8.

[23] S. P. Karimireddy, M. Jaggi, S. Kale, M. Mohri, S. Reddi, S. U. Stich, and A. T. Suresh, "Breaking the centralized barrier for cross-device federated learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 28663–28676.

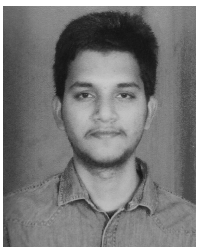
[24] *TensorFlow Core Documentation*. Accessed: Nov. 2023. [Online]. Available: https://www.tensorflow.org/api_docs/python/tf/keras/layers/StringLookup

[25] Hugging Face Documentation. *BERT Tokenizer*. Accessed: Nov. 2023. [Online]. Available: https://huggingface.co/docs/transformers/main_classes/tokenizer

[26] *Hugging Face Transformers Documentation: BERT*. [Online]. Available: https://huggingface.co/docs/transformers/model_doc/bert

[27] S. Reddi, Z. Charles, M. Zaheer, Z. Garrett, K. Rush, J. Konečný, S. Kumar, and H. B. McMahan, "Adaptive federated optimization," 2020, *arXiv:2003.00295*.

M. SUJAYKUMAR REDDY is currently pursuing the B.Tech degree in computer science engineering with Vellore Institute of Technology, Vellore, India. His expertise includes working with transformative technologies such as transformers, reinforcement learning, and physics based neural networks. These techniques find application in interdisciplinary areas, specifically within neuro and behavioral artificial intelligence, and robotics. His research interests include trustworthy artificial intelligence (AI) and machine learning (ML).



HEMANTH KARNATI is currently pursuing the B.Tech. degree in computer science engineering with Vellore Institute of Technology, Vellore. He specializes in integrating the IoT with machine learning to develop intelligent systems that enhance efficiency and decision-making. With a strong passion for deep learning and improving data privacy, he applies his knowledge to create web applications that utilize AI and ML algorithms to address real-world challenges. His innovative methods not only showcase his technical skills but also reflect his commitment to contributing significantly to the tech industry through education and innovation.

L. MOHANA SUNDARI received the degree in electronics and communication engineering from Vellore Engineering College, University of Madras, Tamil Nadu, in 2003, the M.E. degree in applied electronics from Anna University, Chennai, in 2009, and the Ph.D. degree from the Department of Information and Communication, Anna University, in 2022. She is currently an Assistant Professor (Senior Grade) with the School of Computer Science and Engineering, Vellore Institute of Technology, Vellore. She has a teaching experience of more than 17 years. She has published papers in various international journals and conferences. She also published two books on *Antennas* and *Satellite Communication*. Her current research interests include artificial intelligence, image processing, and networking and communication. She is a Lifetime Member of IEI.