Turk J Elec Eng & Comp Sci
() : –
© TÜBİTAK
doi:10.3906/elk-

Privacy preserving hybrid recommender system based on deep learning

Sangeetha SELVARAJ [1*], Sudha Sadasivam GANGADHARAN[2]
[1]Department of Information Technology, PSG College of Technology, Coimbatore, Tamil Nadu, India
ORCID: https://orcid.org/0000-0003-2513-5502
[2]Department of Computer Science and Engineering, PSG College of Technology, Coimbatore, Tamil Nadu, India
ORCID: https://orcid.org/0000-0001-6845-1869

Received: .201 • **Accepted/Published Online:** .201 • **Final Version:** ..201

Abstract: Deep Learning Models are widely being used to provide relevant recommendations in hybrid recommender systems. These hybrid systems combine the advantages of both content based and collaborative filtering approaches. However, these learning systems hamper the user privacy and disclose sensitive information. This paper proposes a privacy preserving deep learning based hybrid recommender system. In hybrid deep neural network, user's side information such as age, location, occupation, zip code along with user rating is embedded and provided as input. These embedding's pose a severe threat to individual privacy. In order to eliminate this breach of privacy, we have proposed a private embedding scheme that protects user privacy while ensuring that the nonlinear latent factors are also learnt. In this paper, we address the privacy in hybrid system using Differential Privacy, a rigorous mathematical privacy mechanism in statistical and machine learning systems. In the reduced feature set, the proposed adaptive perturbation mechanism is used to achieve higher accuracy. The performance is evaluated using three datasets with Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Mean Squared Error (MSE), R squared, Precision and Recall. These evaluation metrics are compared with varying values of privacy parameter ϵ. The Experimental results show that the proposed solution provides high user privacy with reasonable accuracy than the existing system. As the engine is generic it can be used on any recommendation framework.

Key words: Differential Privacy, Adaptive Perturbation, Private Hybrid Recommender, Embedding Perturbation, Deep Neural Network, Laplace Noise, Randomized Response

1. Introduction

Recommendation systems facilitate users to choose from a wide range of items by providing suggestions on relevant items based on the user's interests. Furthermore, recommendations are useful to people as it helps them to choose from a variety of items available with the service provider. As they provide relevant suggestions to customers, these systems are crucial in ecommerce based industries. Hence state of art research focuses on designing and implementing optimized algorithms to provide personalized user recommendations. Adding more information about the user, results in good recommendations. But the results are generated at the cost of user's privacy. Hence, the objective of the research is to propose an optimal recommendation algorithm that protects the privacy of individual and to provide relevant recommendations.

Recommender systems are broadly classified into three categories: Content based, Collaborative and Hybrid method. Content based mechanism uses both the user profile and the product information to offer recommendations. The term 'content' signifies the attribute of the product that is liked by the user which can

*Correspondence: vns.sangeetha@gmail.com

be obtained from the tags, keywords, or side information associated with a product. Collaborative mechanism uses information from user profiles to calculate similarities between users or similarity between items and provide recommendations based on it. Matrix Factorization which is a type of collaborative filtering constructs latent features for users and items. These features are learnt from the past ratings provided by the user. Based on the learned features, predictions for unrated products can be made and the product with topmost prediction rate is recommended to the user. Hybrid method is a combination of content based and collaborative filtering techniques. To improve efficiency without losing the advantage of the above two mechanisms, a hybrid approach offers greater synergy, as compared to the individual recommender system.

These traditional linear models can effectively memorize sparse feature interactions using cross product feature transformation, whereas deep neural networks can generalize interactions through low dimensional embedding to previously unseen features. These networks therefore, have a clear understanding of the user and item, so it delivers exceptional results. Adoption of deep learning techniques in tabular or structured data resulted in a huge improvement in performance. But this adoption also created few shortcomings. For example, deep neural networks make use of a huge amount of user data to make decisions, that apparently pose a threat to an individual's privacy.

From a privacy perspective, recommendation systems are broadly classified into trusted recommender system and untrusted recommender system [1]. In the trusted recommender, the system is trusted by the user, and they send original raw data. In addition to this, a private recommendation algorithm is run by the trusted recommender to produce the results. Whereas, in an untrusted recommender system, users are confined from sending the original raw data and adds noise in the user rating. Further recommendations are made with usual nonprivate algorithms. Such trusted and untrusted recommender systems are called Global differential privacy (GDP) and Local differential privacy (LDP) respectively. In this paper the development of a privacy-preserving algorithm for trusted recommender system is discussed.

Major contributions of the paper are as follows:

1. Analyze and experimentally evaluate the differentially private hybrid deep neural network.
2. Improve the performance using adaptive perturbation, which provides varying perturbation for outlier features and common features.
3. Avoid huge noise addition by employing dimensionality reduction techniques.
4. Combine randomized response with Laplace noise addition for improved accuracy.
5. Experimentally compare the recommendation results generated by the differentially private hybrid system with other non-private baseline algorithms.

The rest of this research paper is organized as follows:- Section 2 includes a literature review of the existing privacy mechanisms. Section 3 outlines the background of differential privacy and the nonprivate deep neural network architecture used in the paper. Section 4 details the proposed private deep learning algorithm along with the theoretical proof for utility and privacy. Section 5 summarizes the experimental results with three datasets namely Movielens100K, Book-Crossing, FilmTrust and result comparison with other baseline nonprivate algorithms. Section 6 deals with the conclusion.

2. Related Work:

A brief survey of the privacy preserving mechanisms such as anonymity, perturbation, and suppression based methods was reviewed by Sangeetha and Sudha Sadasivam [2]. A study by Zhang et al. [3] broadly classify privacy preservation in collaborative filtering into secure multiparty communication, homomorphic encryption, and differential privacy in recommender system. Several researchers [4–6] use differential privacy in distributed multiparty computation. Our work is primarily focused on the usage of the differential privacy mechanism in recommender system.

This paper focuses on introducing differential privacy in the DL model-based hybrid recommendation system. Hence, the literature survey includes sections on privacy in recommenders, and privacy in DL systems.

2.1. Privacy in model and content based Recommender System

Narayanan and Shmatikov [7] performed a statistical de-anonymization of large sparse datasets. The authors launched a deanonymization attack on the anonymized Netflix dataset. They proved that an adversary with little information about the subscriber can easily provide the identity of a particular person in the database including the person's entire movie watching history. The attack was demonstrated using the Internet Movie Database (IMDB) as the source of background knowledge. Even though the researchers used the movie dataset, the attack is generic and applicable to any recommendation system like healthcare, e-commerce, etc. Later researchers provided solutions for deanonymization attacks in the recommender system and our literature review presents an overview of solutions to the attack.

A novel application of differential privacy in recommendation was introduced by McSherry and Mironov [8]. The authors perturbed the rating matrix with differentially private noise addition and proved that it is feasible to design a private recommendation system without losing accuracy. The authors also concluded that loss in accuracy decreases as more data becomes available. Hence, differential privacy based solutions are more suitable for problems involving big data. In this scenario recommendation system is a practical big data framework that is popular and used by industries like Amazon, Netflix, MovieLens, etc. Thus, a privacy preserving algorithm designed using differential privacy based approach offers realistic solution to the issue of big data privacy.

Friedman et al. [9] proposed a generic framework to apply differential privacy to matrix factorization. The privacy framework proposed by Friedman et al. categorizes privacy-preserving algorithms into input perturbation, gradient perturbation, and output perturbation. In input perturbation, the rating matrix is perturbed with Laplace noise. While in the gradient perturbation two private algorithms based on Alternating Least Square (ALS) and Stochastic Gradient Descent (SGD) are proposed, in the output perturbation the nonprivate Matrix Factorization algorithms are executed and the resulting latent factors are perturbed. The authors also compared the results and concluded that input perturbation produced better results.

It can be observed from the literature that the existing techniques cannot prevent the inference of users from the output of neural network model and are orthogonal to the techniques discussed in this paper.

2.2. Privacy in Deep Learning

Shokri et al. [10] demonstrated that machine learning models leak the information about the individual data on which the model is trained. Such attacks are called inference attacks. Shokri et al. trained the model with commercial machine learning as a service and proved that the model is vulnerable to membership inference attacks. Fredrikson et al. [11] performs a model inversion attack that recovers images from the facial recognition system. Fredrikson et al. demonstrated that the inversion attack only requires black-box access to the trained model. Abadi et al. [12] proposed a deep learning-based privacy-preserving algorithm with differential privacy. The primary focus of the work is to design a private algorithm for the classification task. The differentially private noise is added to the nonconvex optimization technique called stochastic gradient descent to protect the user from model inversion attack and inference attack. Though model inversion attack is demonstrated on image classification, the same attack is possible in any trained model. These attacks indicate the need for robust model training. Hence our work protects users from such attacks on trained model through perturbation of the user embedding's. The differential privacy used in proposed private algorithm design provides a mathematical guarantee against the attacks. Another benefit of our approach is that it takes advantage of offline perturbation and can converge faster than gradient perturbation.

Other than differential privacy few cryptography based solutions are available to protect model inversion attack. Hence crytography based solutions are surveyed further. CryptoNet [13] is a combination of homomorphic encryption and neural network. The data is shared by secret encryption and predictions are obtained from the pretrained neural network model. The core part of CryptoNet is that it is able to make such predictions using the encrypted data and returns the predictions in encrypted form. The encrypted classification results thus obtained can be decrypted only by the corresponding sender device and the predictions are used by the sender. Ma et al. [14] created a privacy-preserving ensemble-based classification algorithm for face recognition based on secret sharing and edge computing. The features are collaboratively learned from encrypted face images using two edge servers. Ma et al. [15] proposed privacy-preserving Long Short Term Memory (LSTM) based neural network for smart Internet of Things (IoT) devices. Secret sharing is used for secure communication of voice information from IoT devices. The features are extracted from encrypted audio information using edge devices. These Cryptography based mechanisms demand higher computational costs. Our proposed solution avoids this overhead by using differentially private perturbations. Also, Homomorphic encryption with machine learning assumes that a pretrained model is available [13]. Hence secret sharing is primarily used in the inference stage and it cannot be used for secured model training. Our work focuses on creating a private model through private training. Differential privacy is more suitable for our problem approach. Usage of differential privacy during model training controls the amount of information leaked from an individual record in a dataset.

It should be highlighted that differential privacy is a powerful mechanism that protects the privacy of users in a dataset with a strong privacy guarantee. As inferred from the literature there are several works using differential privacy in recommendation systems using matrix factorization and classification tasks in deep learning. Most of the privacy works in deep learning [12, 16, 18, 19] address the classification task. But the privacy model in classification setting cannot be directly used for recommender systems where each and every user rating and presence or absence of rating is a threat to user privacy. Recently deep learning-based mechanisms are gaining popularity and are extensively used for improved accuracy in recommender system. As observed in most of the deep learning-based recommender systems [20–23] accuracy can be improved by leveraging user-item rating matrix and side information. Hence this improved accuracy comes at the cost of user's privacy and our goal is to develop a novel deep learning-based privacy-preserving hybrid recommendation system. The proposed work is first of its kind with a private hybrid algorithm. Section 3 elaborates on the background of the work with the definition of differential privacy and explains the proposed nonprivate hybrid recommender.

3. Preliminaries

3.1. Differential Privacy

Differential privacy was originally proposed for privacy preserving statistical data release by DWork [24, 25]. It provides a mathematical guarantee for private data release. However, the original differential privacy was later used in numerous other applications in industries [26–28] and academic research [8, 9, 29].

Definition1: A randomized algorithm M satisfies ϵ - differential privacy if for any two neighboring databases

D and $D^{'}$ any measurable subset [25],

$$Pr[M(x) \in S] \leq exp(\epsilon) \times Pr[M(y) \in S] + \delta \tag{1}$$

Where the probability is over randomness of ϵ . If $\delta = 0$ we say that M is $\epsilon -$ differentially private. The privacy parameter ϵ controls privacy and accuracy tradeoff. The neighboring databases D and $D^{'}$ differ in one record in the rating matrix.

Definition2: The Laplace distribution (centered at 0) with scale b is the distribution with probability density function [25]:

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

Definition3:(Sequential Composition[1])suppose a set of privacy mechanisms $M = M_1, ...M_m$ are sequentially performed on a dataset, and each M_i provides, ϵ_i privacy guarantee, M will provide $(\sum_{i=1}^{m} \epsilon_i)$ - differential privacy. The set of randomized mechanisms are performed sequentially on a dataset, and the final privacy guarantee is determined by the summation of total privacy budgets. In our work Algorithm 1 uses Laplace noise addition (Definition 2) along with sequential composition (Definition 3).

3.2. Deep Learning-based Hybrid Recommender system

Hybrid recommendation algorithms are more significant and produces outstanding results when compared to non hybrid solutions. Such improvement is achieved with the combination of rating and side information used in the algorithm. Two key issues in recommendation system are cold start problem and accuracy improvement. A cold start problem occurs when sufficient information about a user or item is not available. Recent hybrid recommendation algorithms based on deep learning addresses these issues and produces outstanding results. But these results comes at the cost of user privacy. The existing works does not address privacy in hybrid algorithms. Hence we extend the recommender proposed by Kiran et al.[20]. Kiran et al. devised a novel hybrid deep learning-based recommender that uses side information and their primary focus was to improve the accuracy. Further, we investigate and propose a privacy-preserving hybrid algorithm.

The deep neural network consists of multiple layers between the input and output layer [30]. Each layer consists of multiple simple processing units called neurons. Neurons in each layer are connected with every other neuron in the previous layer. Every connection is associated with a weight that is randomly initialized and it is improved through multiple epochs. These weights are modified based on the optimization function. The input user id and item id are embedded and concatenated along with the side information (Table 1). Each hidden layer computes a linear function which is input to LeakyReLU (Rectified Linear Unit) function, followed by DropOut. LeakyReLU is a popular activation function used in the deep neural network that overcomes the "dying ReLU" issue. The ReLU returns the value provided as input directly for positive input and returns a 0.0 for negative inputs. The major drawback of ReLU is "dying ReLU" which occurs when a set of nodes output an activation value of 0.0 forever in the training process. "dying ReLU" is solved by LeakyReLU by permitting small negative values. Dropout is a regularization mechanism used in the neural network. During the neural network training, randomly chosen neurons are dropped out by this mechanism to avoid overfitting. The number of neurons in each hidden layer and the activation function used in each layer are all hyperparameters that can be tuned. Each hidden layer employs batch normalization. Batch normalization is used to normalize the values in the hidden layer which in turn ensures a faster convergence. The output from the previous activation layer

Table 1: Side information about users and items in different datasets

Dataset	User or Item	Side Information
MovieLens 100K	Item	Tag, Title, Genre of Movies
	User	Not Available
Film Trust	Item	Not Available
	User	Trust ratings on the User on other users and vice versa
Book-Crossing	Item	Year of Publication. Publisher, Book Title
	User	Age, Location, Author Name

is normalized by subtracting batch mean and dividing by batch standard deviation. The final layer is a fully-connected layer with 1 node, which is input to the sigmoid activation function that predicts the ratings in the range 0 - 5.

In a deep hybrid algorithm, the traditional representation of user id and item id is replaced by embedding. Embedding identifies correlation among data and enables the deep learning model to extract more features from user and item id when compared to one-hot encoding. An embedding is a representation of categorical value as a vector in N-dimensional space. Embeddings are lookup matrices of size K, where K is the number of embeddings. For each user, an array of size K_u is returned, and the user embedding matrix collected from all the users is denoted by E_{user}. For each item, an array of size K_I is returned, and the item embedding matrix collected from all the users is denoted by E_{item}. K_u and K_I are the hyperparameters and it is tuned by the analyst. These hyperparameters used in embedding are more suitable for a scalable big data environment whereas, one hot encoded matrices requires a column updation for every change in the item addition or removal. These user E_{user} and item E_{item} features are similar to latent factors in collaborative filtering based matrix factorization algorithms. But the latent factors in collaborative filtering only capture linear features whereas embeddings are capable of capturing both linear and nonlinear user and item factors.

As stated earlier, a hybrid recommender is a combination of content-based and collaborative filtering approaches. In our hybrid algorithm, the embedded user E_{user} and item E_{item} features are similar to the collaborative filtering approach. Further content-based features are added to make the deep learning model a hybrid model. Table 1 indicates the side information of various datasets used in the hybrid model training.

In Section 4 the proposed differentially private algorithm is described. The section also elaborates the usage of differentially privacy on the deep learning architecture.

4. System Overview

This section presents the system overview and the proposed Global differentially private trusted recommender algorithm.

The Global differential private algorithm performs Laplace noise addition along with sequential composition. In this setting, the recommender system is trusted and the user sends original rating information to the server. However, privacy has to be ensured during model training to prevent model inversion attack [11] and inference attack [10]. An algorithm designed using differential privacy is resistant against these attacks. The proposed algorithm is ϵ differentially private and the privacy analysis is proved in Section 4.2.

Differential privacy is already used [12] as resistance against these attacks. Hence our algorithm uses Differentially Private perturbation to preserve user privacy and to protect from these attacks. Existing algorithms add noise to the optimization functions like Stochastic Gradient Descent and Alternating Least Square [9, 12].

But our new approach perturbs the input user embedding and bias used by the deep learning algorithm. In a hybrid recommendation system, user embedding and bias added to the weights in the neural network convey user-specific information, and perturbing these values preserves user privacy.

For training any deep neural network the categorical features in the dataset cannot be used directly and it requires some preprocessing. As a preprocessing step, the categorical features present in the dataset have to be converted to numerical form and given as input to the neural network. A naive approach is to convert the categorical text information to numerical form using one hot encoding. For eg: If there are 10000 unique words present in the dataset then one-hot encoding constructs matrix with a slot for each word. Further, as the name one-hot encoding suggests if a word is used by an item the value for the corresponding index in the matrix is made '1', and the remaining indices values are made '0'. But one hot encoding produces a highly sparse input features with very few non zero values. Such sparse input features need more weights in a neural network with a large amount of data and higher computation. Another major issue with one-hot encoding is that it is not capable of capturing the semantic relationship between the features.

As a solution to the issues in one-hot encoding, embeddings are used. It transforms the large sparse features into lower-dimensional space. The obtained lower-dimensional features are capable of identifying and preserving the semantic relationship between the categorical items. The embedding to our neural network is a combination of user rating and side information like genre, tag, etc. These semantic relationships help in improved model accuracy, using personal information about user behavior. Such personal information is sensitive and its usage in the model thwarts user privacy. So the embedding is perturbed to ensure privacy. For training a network model, it is not mandatory to use bias information. If a model is trained without bias, it provides only a generic recommendation. It is used in the model to comprehend the user and the item better. Thus the bias added for a user is also sensitive and hence it is also perturbed to enhance privacy.

Initially, the collaborative filtering based model is used to train the original input, from which the embeddings are obtained. These embeddings, thus obtained contain user and item embeddings, along with user and item biases. From these values our approach perturbs only the user embedding and bias values, using Algorithm 1. However, the user embedding obtained has a higher dimension, and this increased dimensionality results in more noise addition which degrades the model accuracy. Hence, the dimensionality of obtained user embeddings are reduced using Principal Component Analysis (PCA) and user features are obtained. In the reduced feature set users with similar tastes are close to each other and the users whose characteristics are not similar to others are far away.

The features that are close to each other are similar to the K anonymization mechanism and user privacy is retained for the points that are next to each other. K anonymization is a privacy preservation mechanism proposed by Latanya Sweeney [31]. Sweeny defines K-anonymity as the privacy requirement for publishing microdata that requires each equivalent class to contain atleast K records.

But, for the outlier features the privacy is easily violated. So, the outlier user information that is at a considerable distance from other features is more prone to attacks and these features are identified initially. Further, more noise is added to the outliers and less random noise is added to other features. The noise is obtained from the differentially private Laplace mechanism. The user bias information is also perturbed using Laplace mechanism. The principal user features are converted back to dense embeddings before deep neural network training. The overall flow of our proposed methodology is depicted in Figure 1.

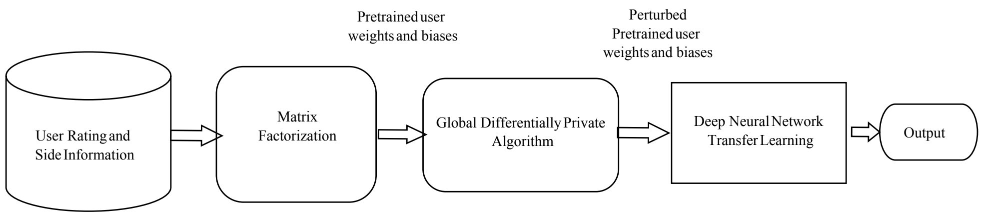


Figure 1: Overview Flow Diagram

4.1. Proposed Private Recommender

In the proposed system the user's privacy is protected by adding a Differentially Private noise to the embedding and bias values. The user, item embedding are generated from user id, user side information, movie id, and movie side information respectively. Such embedding is not suitable for private model training hence perturbed embedding is obtained using Algorithm 1. The deep learning model depicted in Figure 2 is trained with the perturbed embeddings and biases using transfer learning mechanism. Transfer Learning is a key advancement in deep learning that supports model training with pretrained weights. The perturbed pretrained embeddings bring in randomness and prevents the model from memorizing user-specific information. The fc in the figure indicates fully connected layer. Totally three fully connected deep neural networks are used for transfer learning.

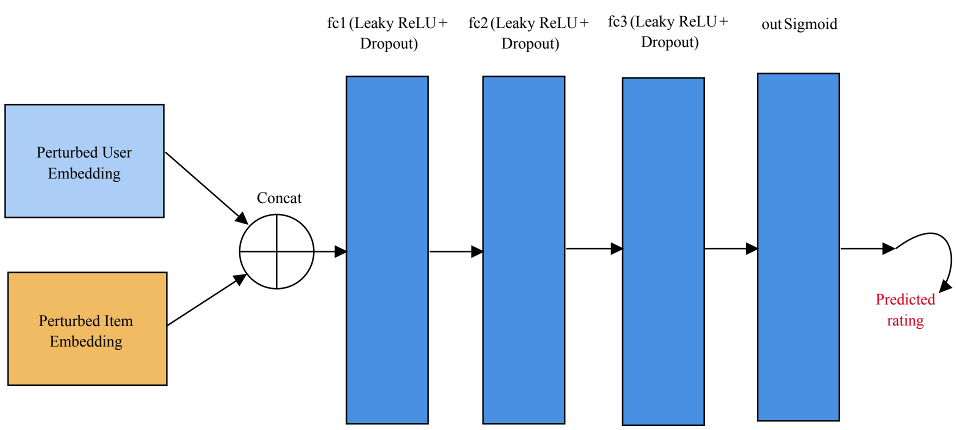


Figure 2: Proposed Private Hybrid Recommendation System

Most of the existing works [9, 12, 29] perform noise addition during model training stage which is split among multiple epochs and noise addition increases with epochs. Offline noise addition in our work ensures a reduction in noise compared to the existing algorithms. Algorithm1 is executed offline and the steps are briefed in the following paragraphs.

Before the execution of the proposed algorithm, the embeddings with 'd' dimensions are obtained from collaborative filtering based matrix factorization. A naïve approach is to add 'd' dimensional noise to the embeddings. However adding 'd' dimensional noise $E'_{userij} = E_{userij} + Lap\left(\frac{d*\Delta S}{\epsilon}\right)$ introduces a huge amount of noise and completely degrades the utility of data. Hence the proposed solution uses Principal Component Analysis (PCA) and a reduced dimension 'g' is obtained. The PCA algorithm produces a reduced feature set

Algorithm 1 Global Differentially Private Algorithm

Input: ϵ_1 and ϵ_2 - Privacy parameter
$\Delta S_1, \Delta S_2$ - Sensitivity
$E^{(n\times d)}$ - Embedding for 'n' users with dimension 'd'
$b^i, i \in 1, 2, ...n$ - User bias parameter of the model for 'n' users
Output:
$E^{'(n\times d)}$, $b^{'i}, i \in 1, 2, ...n$ - Perturbed user Embedding and bias

1. Apply PCA on $E^{(n\times d)}$ and obtain the reduced feature set $f^{(n\times g)}$
2. Identify the outlier in $f^{(n\times g)}$ by applying z score normalization
3. Store the outlier user id and feature id in O
4. **for** $i = 1, 2, ...n$ do
5. **for** $j = 1, 2, ...g$ do
6. if i,j in O
7. $f'_{ij} = f_{ij} + Lap\left(\frac{g * \Delta S_1}{\epsilon_1}\right)$
8. Clamp f'_{ij} to $[f_{ij_{min}}, f_{ij_{max}}]$
9. else
10. v = randomized response()
11. if v == True
12. Retain the True Value $f'_{ij} = f_{ij}$
13. else
14. $f'_{ij} = f_{ij} + Lap\left(\frac{g * \Delta S_2}{\epsilon_1}\right)$
15. Clamp f'_{ij} to $[f_{ij_{min}}, f_{ij_{max}}]$
16. **end for**
17. $b'_i = b_i + Lap\left(\frac{\Delta S_3}{\epsilon_2}\right)$
18. **end for**
19. Convert $f^{'(n\times g)}$ into dense embeddings $W^{'(n\times d)}$

function randomized response()

1. Initialize $v_1 = 1$
2. $v'_1 = \begin{cases} 1 & with\ probability\ \frac{1}{2}p \\ 0 & with\ probability\ \frac{1}{2}p \\ v_1 & with\ probability\ 1-p \end{cases}$
3. $if v'_1 == 1$
4. Return True
5. else
6. Return False

$f^{(n\times g)}$ in step2. Further, a Laplace noise addition is performed on the reduced dimension 'g' which eventually increases recommendation accuracy. The algorithm uses adaptive noise addition that adds larger noise to outlier and minimal random noise to remaining features. The outliers in the embedding signify the users who have unique features and these users completely deviate from other users and they are more prone to attacks. Therefore, our algorithm initially identifies the outliers present in the feature set in step 3 using z score normalization using Equation 3.

$$z = \left(\frac{x - \mu}{\sigma}\right) \tag{3}$$

Where, μ signifies mean and σ denotes standard deviation. After outlier identification more noise is added to outliers with sensitivity $\Delta s_1 = f_{j_{max}} - f_{j_{min}}$ and ϵ_1 in step 7. To avoid excess noise addition the feature is clamped (Step 15) as follows

$$f_{ij}^{'} = \begin{cases} f_{ij_{min}} & if\ f_{ij}^{'} < f_{ij_{min}} \\ f_{ij_{max}} & if\ f_{ij}^{'} > f_{ij_{max}} \\ f_{ij} & otherwise \end{cases} \tag{4}$$

Hence, steps 3 to 8 identify the outliers and adds more noise to the outlier features. The remaining features are perturbed with minimal noise by a randomized response mechanism. In step 10 randomized response is called, which is a traditional coin flip used in differential privacy. The coin flip is made with probability 0.75 returning the true answer and 0.25 returning the perturbed answer. The minimal perturbation is made in step 14 with $\Delta s_2 = 1$ and ϵ_1 . The coin flip and less sensitivity are chosen to minimize the randomness introduced in the model and to improve the utility. Hence our perturbation mechanism is an adaptive mechanism.

In step 17 user bias is perturbed with sensitivity $\Delta s_3 = b_{imax} - b_{imin}$ and epsilon of ϵ_2. Throughout the algorithm, the perturbation is applied in two steps. Therefore, based on the composability property (Definition3) of differential privacy the epsilon is divided into 0.75 for ϵ_1 and 0.25 for ϵ_2. This division is user-defined and it is made in-line with the existing algorithms [8], where the composability partition is made based on the importance of the feature.

Since the proposed Global Differentially Private Algorithm (Algorithm 1) is generic it can be extended and applied to any Deep Learning network. Such network can perform classification or regression, but the private algorithm requires perturbed embedding as input.

4.2. Privacy and Utility Analysis

In this section we present a theoretical analysis for privacy and utility of the proposed Global Differentially Private Algorithm.

4.2.1. Privacy Analysis

The proposed algorithm contains one private operation that is embedding perturbation. In this section, we analyze the privacy guarantee of the embedding perturbation. Along with embedding the bias values are also perturbed whose privacy is guaranteed by the Composition of differential privacy [17]. The composition undertakes the privacy guarantee for a sequence of differentially private computation.

Theorem 1 *Algorithm 1 satisfies ϵ - differential privacy.*

Proof Suppose two Datasets D and D' differ in the ratings of one user. Let 'f' be the reduced feature set.

$$\begin{aligned} \frac{\Pr(\mathbf{R}(D))}{\Pr(\mathbf{R}(D'))} &= \frac{\prod_{j=1}^{g}(\Pr(f_j(D) + Lap(\frac{\Delta S}{\epsilon}) = R)}{\prod_{j=1}^{g}(\Pr(f_j(D') + Lap(\frac{\Delta S}{\epsilon}) = R)} \leq e^{\epsilon} \\ &= \frac{\prod_{j=1}^{g} \exp\left(\frac{-\|R - f_j(D)\|_1 \epsilon}{\Delta S}\right)}{\prod_{j=1}^{g} \exp\left(\frac{-\|R - f_j(D')\|_1 \epsilon}{\Delta S}\right)} \\ &= \prod_{j=1}^{g} \exp\left(\frac{\epsilon}{\Delta S}(\|R - f_j(D)\|_1 - \|R - f_j(D')\|_1)\right) \end{aligned}$$

$$= \prod_{j=1}^{g} \exp\left(\frac{\epsilon}{\Delta S}(\|f_j(D) - f_j(D'))\|_1)\right) \le e^{\epsilon}$$

□

Lemma 1 *Composition [17], $M = m_1, m_2, ...m_n$ if each m_i provides ϵ' privacy guarantee, the sequence of M will provide $n * \epsilon'$ differential privacy.*

Proof In the reduced feature set we add independent Laplace noise to the features.

$$f_{ij}^{'} = f_{ij} + \frac{g * \Delta S_1}{\epsilon_1}$$

further noise is added to bias

$$b_{ij}^{'} = b_{ij} + \frac{\Delta S_3}{\epsilon_2}$$

when combining both operations the proposed method preserves ϵ-differential privacy by applying the composition lemma. □

4.2.2. Utility Analysis

In order to protect data privacy noise is added to the low dimensional embeddings. A naive solution adds noise to all the embeddings which degrades the performance of the algorithm. This is due to the fact that the magnitude of noise added is directly proportional to the performance of the algorithm.

Theorem 2 *For a given privacy parameter ϵ, Algorithm1 adds less noise compared to the naive solution.*

Proof In algorithm1 the reduced feature set has $n * g$ elements. To each element noise drawn from $Lap\left(\frac{\Delta S}{\epsilon}\right)$ is added. The magnitude of noise $M1 = O(\frac{n*g*\Delta S^2}{\epsilon^2})$.

In naive solution noise is added to $n * m$ elements. To each element noise drawn from $Lap\left(\frac{\Delta S}{\epsilon}\right)$ is added. The magnitude of noise $M2 = O(\frac{n*m*\Delta S^2}{\epsilon^2})$.

$$M1 = O(\frac{n * g * \Delta S^2}{\epsilon^2}) < O(\frac{n * m * \Delta S^2}{\epsilon^2}) = M2$$

where g<<m. Hence we observe that M1 < M2. That is the Algorithm.1 adds less noise than naive approach. □

5. Experimental Evaluation

5.1. Dataset

The experiments are conducted using three datasets MovieLens(100k), Book-Crossing, and FilmTrust. The properties of the dataset are briefed in Table 2.

5.2. Computing Environment

The experiments are implemented in python 3.7.3 by leveraging the libraries like scikit-learn 0.20.3, pandas 0.24.2, and numpy 1.16.2. The computing environment with Nvidia GPU and Linux operating system with 12GB RAM are used. PyTorch 1.1.0 library was used for training the deep learning models.

Table 2: Datasets

Dataset	Users	Items	Ratings	Sparsity	Scale
Book-Crossing	278,858	271,379	1,149,780	99.99%	[0-10]
ML100K	610	9,742	100,836	93.7%	[1-5]
FilmTrust	1,508	2,071	35,497	98.86%	[1-5]

5.3. Evaluation Criteria

The algorithm is evaluated with varying privacy parameter ϵ and six evaluation metrics Root Mean Squared Error (RMSE), Mean Squared Error (MSE), Mean Absolute Error (MAE), R-squared, Precision@N, and Recall@N values are used. The accuracy metrics are calculated with eight runs. The mathematical definitions of the evaluation metrics namely MSE, RMSE, and MAE are in the Equations 5,6 and 7. For these equations, p_{ij} is a matrix with a cell value $p_{ij} = 1$ only if the user i rated item j and 0 otherwise. In Equations 5,6 and 7 r_{ij}^{actual} denotes the rating provided by user i to item j and $r_{ij}^{predicted}$ is the rating predicted by the model. For a user u, Precision@N, and Recall@N are computed by $\frac{|\mathrm{Rel}_u \cap \mathrm{Rec}_u|}{|\mathrm{Rec}_u|}$ and $\frac{|\mathrm{Rel}_u \cap \mathrm{Rec}_u|}{|\mathrm{Rel}_u|}$, respectively. Where Rec_u denote a set of N items recommended to u, and Rel_u denote a set of items considered relevant.

$$RMSE = \sqrt{\sum_{i,j}^{m,n} p_{ij} \left(r_{ij}^{actual} - r_{ij}^{predicted} \right)^2} \tag{5}$$

$$MSE = \sum_{i,j}^{m,n} p_{ij} \left(r_{ij}^{actual} - r_{ij}^{predicted} \right)^2 \tag{6}$$

$$MAE = \sum_{i,j}^{m,n} p_{ij} \left| r_{ij}^{actual} - r_{ij}^{predicted} \right| \tag{7}$$

5.4. Results

The nonprivate Deep Neural Network (DNNRec) Recommendation algorithm results are chosen as the baseline and the private recommendation results are computed for varying values of epsilon ϵ. In Table 3,4, and 5 the boldfaced results signify that the accuracy is close to the nonprivate results.

Choosing a value for ϵ is an open question and the works done so far [9, 12, 24, 28] record the results with various epsilon values of ϵ ranging from 0.1 (very low) to 40 (very high). As per the literature our experimental results tabulate the ϵ value of 0.1 to 40. Where 0.1 denotes high privacy and 40 denotes low privacy.

Table 3: Performance of Global Differentially Private Algorithm on Movielens 100K Dataset

Measure	MSE (lower is better)	RMSE (lower is better)	MAE (lower is better)	R-Squared (higher is better)	Precision @10 (higher is better)	Recall @10 (higher is better)
non private	0.747	0.864	0.666	0.338	0.907	0.783
ϵ =0.1	0.883	0.939	0.739	0.199	0.903	0.780
ϵ =0.5	0.874	0.935	0.734	0.207	0.904	0.782
ϵ =1	0.898	0.947	0.741	0.186	0.901	0.782

ϵ =5	0.878	0.937	0.734	0.203	0.903	0.780
ϵ =10	0.854	0.924	0.723	0.225	0.901	0.778
ϵ =15	0.826	0.909	0.711	0.250	0.906	0.780
ϵ =20	0.813	0.901	0.701	0.262	0.904	0.778
ϵ =25	0.805	0.897	0.699	0.270	0.905	0.779
ϵ =30	**0.789**	**0.888**	**0.696**	**0.284**	**0.906**	**0.781**
ϵ =35	**0.780**	**0.883**	**0.685**	**0.292**	**0.905**	**0.780**
ϵ =40	**0.787**	**0.887**	**0.689**	**0.286**	**0.905**	**0.782**

As observed in Table 3,4 MovieLens 100K, and Film Trust performance match the nonprivate algorithm with ϵ = 25 to 40. On the other hand the Book-Crossing dataset Table 4 match the nonprivate results for all the values of ϵ. As expected the performance improves with lower privacy and vice versa. The findings clearly indicate that the private algorithm is practical and provides higher accuracy even for highly sparse datasets. As observed from Table 2, Book-Crossing have the highest sparsity of 99% and it produces outstanding results. Also, it can be observed from Table 2 that Book-Crossing has more users with a count of 278,858 and it is inferred from the results that the differentially private algorithm is more suitable for datasets with a large number of users. Hence, we conclude that our algorithm is more suitable for the realistic recommendation which is highly sparse and consists of a large number of users. The experimental results confirm that the proposed algorithm is on par with the baseline concerning less error, good recommendation quality, and high coverage rate. The error rate is measured with MSE, RMSE, MAE and R-Squared. Recommendation quality and coverage rate are evaluated with Precision @ 10 and Recall @ 10 respectively.

Table 4: Performance of Global Differentially Private Algorithm on Film Trust Dataset

Measure	MSE (lower is better)	RMSE (lower is better)	MAE (lower is better)	R-Squared (higher is better)	Precision @10 (higher is better)	Recall @10 (higher is better)
non private	0.649	0.805	0.626	0.225	0.843	0.922
ϵ =0.1	0.798	0.893	0.705	0.075	0.812	0.988
ϵ =0.5	0.800	0.894	0.709	0.073	0.811	0.982
ϵ =1	0.792	0.890	0.708	0.082	0.810	0.980
ϵ =5	0.802	0.895	0.707	0.071	0.811	0.984
ϵ =10	0.769	0.877	0.696	0.109	0.814	0.978
ϵ =15	0.759	0.871	0.689	0.120	0.823	0.948
ϵ =20	0.745	0.863	0.681	0.137	0.829	0.951
ϵ =25	0.725	0.851	0.676	0.159	0.825	0.956
ϵ =30	**0.720**	**0.848**	**0.666**	**0.166**	**0.835**	**0.932**
ϵ =35	**0.711**	**0.843**	**0.663**	**0.176**	**0.833**	**0.942**
ϵ =40	**0.711**	**0.843**	**0.665**	**0.176**	**0.833**	**0.949**

Table 5: Performance of Global Differentially Private Algorithm on Book-Crossing Dataset

Measure	MSE (lower is better)	RMSE (lower is better)	MAE (lower is better)	R-Squared (higher is better)	Precision @10 (higher is better)	Recall @10 (higher is better)
non private	2.758	1.661	1.280	0.169	0.988	0.988
ϵ =0.1	**2.710**	**1.646**	**1.256**	**0.180**	**0.988**	**0.988**
ϵ =0.5	**2.957**	**1.719**	**1.295**	**0.106**	**0.988**	**0.988**
ϵ =1	**2.779**	**1.667**	**1.267**	**0.159**	**0.988**	**0.987**
ϵ =5	**2.803**	**1.674**	**1.278**	**0.152**	**0.988**	**0.988**
ϵ =10	**2.849**	**1.688**	**1.294**	**0.138**	**0.988**	**0.987**
ϵ =15	**2.760**	**1.661**	**1.262**	**0.165**	**0.988**	**0.988**
ϵ =20	**2.748**	**1.657**	**1.270**	**0.169**	**0.989**	**0.986**
ϵ =25	**2.755**	**1.659**	**1.267**	**0.167**	**0.989**	**0.986**
ϵ =30	**2.839**	**1.684**	**1.310**	**0.141**	**0.988**	**0.987**
ϵ =35	**2.864**	**1.692**	**1.288**	**0.134**	**0.988**	**0.988**
ϵ =40	**2.803**	**1.674**	**1.298**	**0.152**	**0.988**	**0.987**

5.5. Accuracy Comparison to other nonprivate collaborative filtering approaches

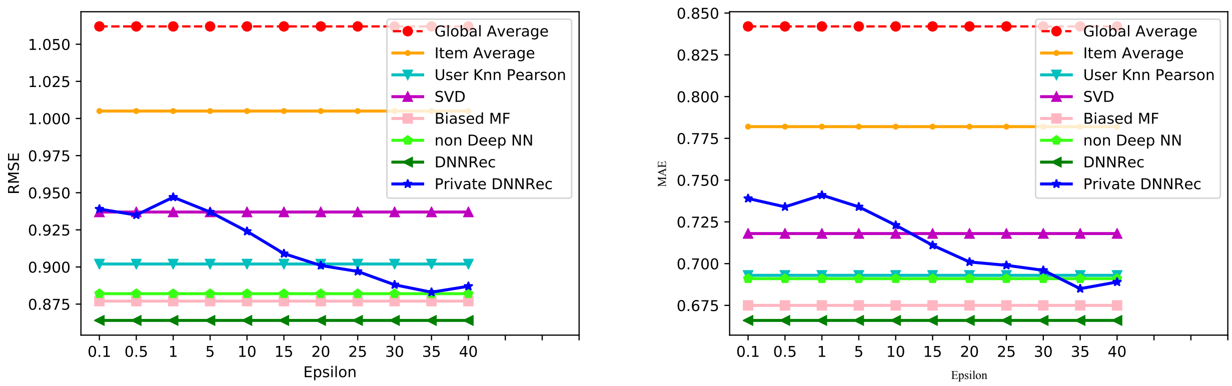
In this section, we have compared the RMSE, MAE of our private algorithm with other nonprivate algorithms. The proposed deep learning-based private approach produces outstanding results. Table 6 highlights the baseline, nonprivate algorithms used for comparison.

The accuracy measures are compared with the baseline algorithms and graphs are plotted accordingly in this section. The results that are better than the baseline indicate high utility. We also plot the non-private Matrix Factorization, non-Deep Neural Network, and Deep Neural Network algorithms which produce outstanding results. For private algorithms, such exceptional results are difficult to achieve. Therefore we chose all these three algorithms as the upper bound and plot them along with our results for brevity.

Table 6: Summary of nonprivate baseline RMSE and MAE

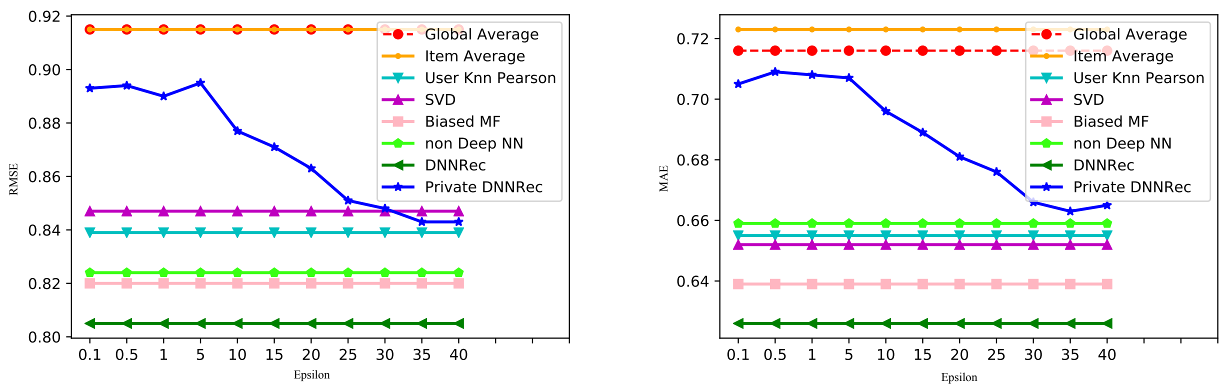
	ML 100K		Film Trust		Book Crossing	
Baseline	RMSE	MAE	RMSE	MAE	RMSE	MAE
Global Average	1.062	0.842	0.915	0.716	1.859	1.509
Item Average	1.005	0.782	0.915	0.723	1.952	1.537
User KNN Pearson	0.902	0.693	0.839	0.655	1.851	1.429
SVD	0.937	0.718	0.814	0.629	1.932	1.537

The experimental results indicate that the baseline global and item average is achieved by all the datasets. However, attaining the performance of the remaining algorithms varies from one dataset to the other and is explained separately for every dataset. Figure 3a RMSE comparison indicates that beyond $\epsilon = 5$ the performance of the private algorithm is better than SVD and it is better than user KNN Pearson beyond $\epsilon =$ 20. Hence the baseline RMSE accuracy is achieved for Movielens 100 K dataset but the upper bound is not achieved. Figure 3b MAE comparison indicates that beyond $\epsilon = 11$ the performance of the private algorithm is better than SVD and it is better than user KNN Pearson. The upper bound non-Deep Neural Network performance is achieved beyond $\epsilon = 30$. So the baseline MAE accuracy is achieved for Movielens 100 K dataset and one upper bound algorithm result is achieved.



(a) RMSE (b) MAE

Figure 3: Accuracy of Movielens100K



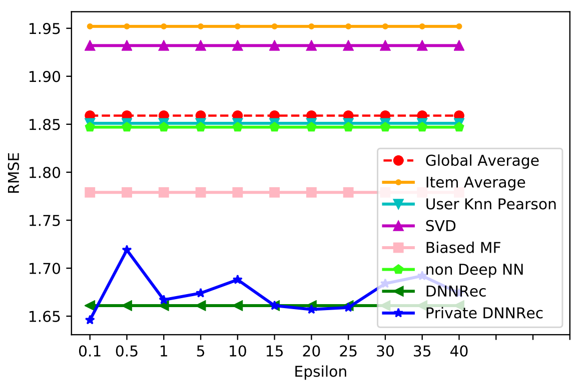
(a) RMSE (b) MAE

Figure 4: Accuracy of Film Trust

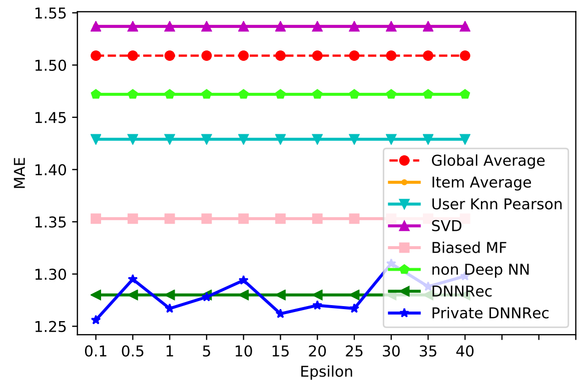
Figure 4a RMSE comparison indicates that beyond $\epsilon = 30$ the performance of the private algorithm is better than SVD and it is not able to achieve the user KNN Pearson results. Thus, the Film Trust private algorithm is better than three baseline RMSE results but the upper bound results are not attained. In Figure 4b two baseline, MAE accuracy results are achieved and upper bound algorithm results are not achieved by the Film Trust dataset.

Figure 5a and Figure 5b indicate that our algorithm produces extraordinary performance for the Book-Crossing dataset and satisfies all the baseline accuracy. In upper bound, it shows good performance for all algorithms except Deep Neural Network.

To the best of our knowledge, none of the existing private algorithms attain the performance of state-of-the-art collaborative filtering algorithms like SVD and KNN. Our exhaustive experimental results confirm that our private algorithm outperforms SVD and KNN for a few values of epsilon. We further observe that it is possible to attain most of the upper bound accuracies.



(a) RMSE



(b) MAE

Figure 5: Accuracy of Book Crossing

5.6. Accuracy Comparison to other private Differential Privacy based approaches

In Fig. 6 , other differentially privacy based algorithm [32] is compared with proposed algorithm. [32] proposed Personalized Differential Privacy (PDP-PMF) based probabilistic matrix factorization and compares the results with Differential Privacy (DP-PMF). Fig. 6 clearly indicates that the proposed algorithms outperforms the existing private algorithms. Section 6 concludes our work.

6. Conclusion

In this paper, a privacy-preserving hybrid deep learning algorithm based on differential privacy is proposed. The accuracy is enhanced using adaptive perturbation where large noise is added to outliers in the data and a minimal random noise addition is performed on all other features.

The contributions of this paper are fivefold. First, the experimental results of the proposed private deep learning algorithm prove that it is feasible to achieve the benefits of deep learning with reasonable privacy. Second, the proposed solution is based on differential privacy which provides a mathematical guarantee for protecting individual privacy. The implementation is compared with varying values of epsilon with very high privacy of $\epsilon = 0.1$ to low privacy of $\epsilon = 40$. The accuracy comparison with other algorithms indicates that the proposed noisy private algorithm outperforms the nonnoisy baseline approaches in terms of accuracy. Our findings also indicate that attaining upper bound is feasible for few datasets with deep learning approach. Third, since our noise addition is not performed during model training it does not require additional model convergence time like the existing algorithms. Fourth, the noise addition is performed on the embeddings in a compressed form obtained from PCA. This mechanism avoids excess noise addition and also ensures accuracy. Fifth, multiple epochs can be performed without incremental noise as the proposed algorithm uses perturbed pretrained weights with transfer learning. Transfer learning is a key advancement in deep learning that runs the model with pretrained weights. The private deep learning approach is generic, so it can be applied to any big data recommendation engine.

References

[1] Tianqing Z, Gang L, Wanlei Z, Philip S Y. Differential Privacy and Applications. NY, USA: Springer International Publishing, 2017.

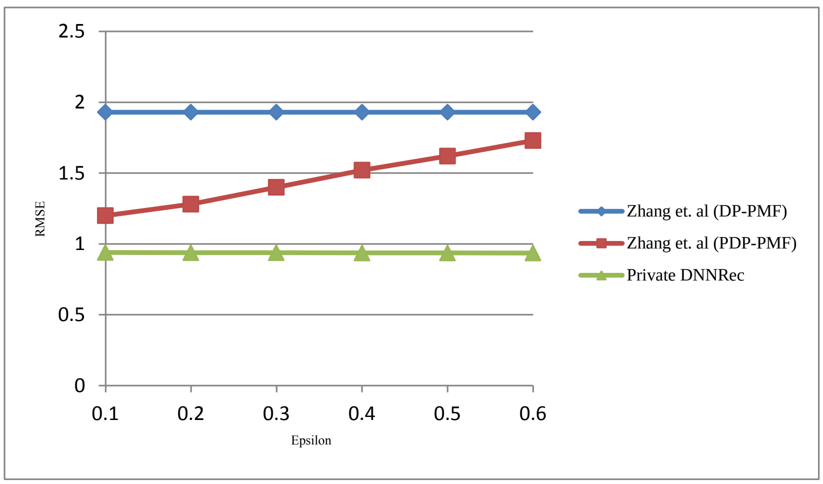


Figure 6: Comparison to other Differential Privacy based Algorithm

[2] Sangeetha S, Sudhasadasivam G. Privacy of Big Data: A Review. In: Dehghantanha A, Choo KK (editors). Handbook of Big Data and IoT Security, NY, USA: Springer Cham, 2019, pp. 5–23.

[3] Zhang D, Chen X, Wang D, Shi J. A survey on collaborative deep learning and privacy- preserving. In: IEEE Third International Conference on Data Science in Cyberspace, Guangzhou, China; 2018. pp. 652–658.

[4] Hamm J, Cao P, Belkin M. Learning Privately from Multiparty Data. In: Proceedings of 33rd International

Conference on Machine Learning, NY, USA; 2016. pp. 555 - 563.

[5] Arun R, Shivani A. A Differentially Private Stochastic Gradient Descent Algorithm for Multiparty Classification. In: Proceedings of Machine Learning Research, La Palma, Canary Islands, Spain; 2012, pp. 933-941.

[6] Pathak MA, Rane S, Raj B. Multiparty Differential Privacy via Aggregation of Locally Trained Classifiers. In: Proceedings of the 23rd International Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada; 2010, pp. 1876–1884.

[7] Narayanan A, Shmatikov V. Robust De-anonymization of Large Sparse Datasets. In: IEEE Symposium on Security and Privacy (sp 2008), Oakland, CA; 2008, pp. 111-125.

[8] Mironov I, McSherry F. Differentially private recommender systems: Building privacy into the Netflix prize contenders. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France; 2009, pp. 627–636.

[9] Friedman A, Berkovsky S, Kaafar MA. A differential privacy framework for matrix factorization recommender systems. User Modeling and User-Adapted Interaction 2016; 26(5): 425–458. doi: 10.1007/s11257-016-9177-7.

[10] Shokri R, Stronati M, Song C, Shmatikov V. Membership Inference Attacks Against Machine Learning Models, In: IEEE Symposium on Security and Privacy (SP), San Jose, CA; 2017, pp. 3-18.

[11] Fredrikson M, Jha S, Ristenpart T. Model Inversion Attacks That Exploit Confidence Information and Basic Countermeasures. In: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, Colorado, USA; 2015, pp. 1322–1333.

[12] Abadi M, Chu A, Goodfellow I, McMahan HB, Mironov I et al. Deep Learning with Differential Privacy. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria; 2016, pp. 308–318.

[13] Dowlin N, Gilad-Bachrach R, Laine K, Lauter K, Naehrig M et al. Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy. In: Proceedings of the 33rd International Conference on International Conference on Machine Learning, NY, USA; 2016, pp. 201–210.

[14] Ma Z, Liu Y, Liu X, Ma J, Ren K. Lightweight privacy-preserving ensemble classification for face recognition. IEEE Internet of Things Journal 2019; 6 (3): 5778–5790. doi: 10.1109/JIOT.2019.2905555.

[15] Ma Z, Liu Y, Liu X, Ma J, Li F. Privacy-preserving outsourced speech recognition for smart iot devices. IEEE Internet of Things Journal 2019; 6 (5):8406–8420.

[16] McMahan HB, Ramage D, Talwar K, Zhang Li. Learning differentially private recurrent language models. In: 6th International Conference on Learning Representations; Vancouver, BC, Canada ; 2018. pp. 1–14.

[17] McSherry F, Talwar K. Mechanism Design via differential privacy. In: Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Society, Washington, DC, USA; 2007. pp. 94-103.

[18] Phan N, Wang Y, Wu X, Dou D. Differential Privacy Preservation for Deep Auto-Encoders: An Application of Human Behavior Prediction. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, Arizona, 2016; pp.1309–1316.

[19] Phan N, Wu X, Hu H, Dou D. Adaptive Laplace Mechanism : Differential Privacy Preservation in Deep Learning. In: Proceedings - IEEE International Conference on Data Mining ICDM; New Orleans, USA; 2017. pp. 385-394.

[20] Kiran R, Pradeep K, Bharat B. DNNRec: A novel deep learning based hybrid recommender system. Expert Systems with Applications 2020; 144. doi: 10.1016/j.eswa.2019.113054.

[21] Sahoo A, Pradhan C, Barik R, Dubey H. DeepReco : Deep Learning Based Health Recommender System Using Collaborative Filtering. Computation 2019; 7: 25. doi: 10.3390/computation7020025.

[22] Covington P, Adams J, Sargin E. Deep neural networks for youtube recommendations. In: Proceedings of the 10th ACM Conference on Recommender Systems, Boston, Massachusetts, USA, 2016; pp. 191–198.

[23] Wei J, He J, C K, Zhou Y, Tang Z . Collaborative filtering and deep learning based recommendation system for cold start items, Expert Systems With Applications 2016, 69: 29-39, doi: 10.1016/j.eswa.2016.09.040.

[24] Dwork C. Differential Privacy. In: Bugliesi M., Preneel B., Sassone V., Wegener I. (eds). Automata, Languages and Programming, ICALP 2006. Lecture Notes in Computer Science. Heidelberg, Berlin: Springer, 2006, pp. 1-12.

[25] Dwork C, Roth A. The algorithmic foundations of differential privacy. Hanover, MA, USA: Now Publishers Incorporation, 2014.

[26] Fanti G, Pihur V, Erlingsson Ú. Building a RAPPOR with the Unknown: Privacy-Preserving Learning of Associations and Data Dictionaries. Proceedings on Privacy Enhancing Technologies 2016; (3): 41–61.

[27] Erlingsson Ú, Pihur V, Korolova A. RAPPOR: Randomized Aggregatable Privacy-Preserving Ordinal Response. In: Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security - CCS '14, Scottsdale Arizona, USA; 2014, pp. 1054 – 1067.

[28] Shin H, Kim S, Shin J, Xiao X. Privacy Enhanced Matrix Factorization for Rec- ommendation with Local Differential Privacy. IEEE Transactions on Knowledge and Data Engineering 2018; 30 (9): 1770–1782, doi:10.1109/TKDE.2018.2805356.

[29] Berlioz A, Friedman A, Kaafar MA, Boreli R, Berkovsky S. Applying differential privacy to matrix factorization. In: Proceedings of the 9th ACM Conference on Recommender Systems; Vienna, Austria; 2015. pp. 107–114.

[30] Goodfellow I, Bengio Y, Courville A. Deep Learning. USA: The MIT Press, 2016.

[31] Sweeney L. k-anonymity: A model for protecting privacy. International Journal of Uncertainty, Fuzziness and Knowlege-Based Systems 2002; 10 (5): 557 – 570, doi: 10.1142/S0218488502001648.

[32] Shun Zhang, Laixiang Liu, Zhili Chen, Hong Zhong. Probabilistic matrix factorization with personalized differential privacy.Knowledge-Based Systems 2019; 183: 104864, doi: https://doi.org/10.1016/j.knosys.2019.07.035.