

The Study of Orthogonal Gradient Descent in Unlearning Algorithms

Xu He^a

Computer Science and Technology, Chinese University of Hongkong, Shenzhen, Guangdong, China

Keywords: Federated Learning, Unlearning, Orthogonal Gradient Descent.

Abstract: In today's world, where smartphones and laptops are commonly used for communication, Federated Learning (FL) has become a key method for training models while keeping data private. This method lets data stay on the user's device instead of being shared. However, when users decide to leave FL, removing their data from the model can be difficult and requires a lot of resources, as it usually involves retraining the model. This paper investigates the application of Orthogonal Gradient Descent (OGD) and Steepest Descent in federated learning to enhance data removal and comply with the 'right to be forgotten' under GDPR. Employing OGD minimizes residual data impact, while Steepest Descent facilitates rapid gradient reduction, tested against algorithms like FedRecovery. Despite increasing computational demands by up to 10%, this approach significantly boosts unlearning efficiency and retains model performance, proving viable for stringent unlearning requirements. The study underscores OGD's potential and limitations, such as sensitivity to learning rate changes and its ineffectiveness when tasks greatly deviate, emphasizing the need for further research to optimize these methods in practical federated learning scenarios.

1 INTRODUCTION


In today's digital age, nearly everyone uses electronic devices like phones, computers, and laptops to communicate, entertain, or work from home. This prevalence has made federated learning (FL) a viable model training method. FL is designed to enhance data privacy while accommodating heterogeneous data. It operates by keeping data on individual devices, training models locally, and then sending the updated models to a central server (Li et al., 2020). Although this method avoids direct data sharing, it still raises important information security concerns about "right to be forgotten" (RTBF). Rooted in European legislation such as the GDPR, RTBF empowers individuals to take control of their data. When participants opt out of federated learning, it should be possible to do so by removing their own data from the training model (Liu et al., 2020).

One method to achieve the elimination of personal data from a federated learning model is through retraining, which involves having the remaining participants retrain the model. However, this approach presents significant challenges, as it not

only incurs substantial computational and communication costs but also increases the burden on the remaining participants. This added workload can diminish participants' willingness to continue training, ultimately rendering federated learning less feasible in practical applications. But in the past few years, an approach called unlearning has been gaining traction. Consequently, the concept of "unlearning" has emerged as a valuable solution to address these challenges, gaining increased research interest and practical significance.

In existing work, gradient ascent is a simple and effective method. Currently, several algorithms, such as FedEraser and FedRecovery, have been developed for federated learning. FedEraser, for instance, was tested on four real-world datasets to evaluate its effectiveness (Liu et al., 2021), while FedRecovery focuses on "recovering from poisoning attacks" (Cao et al., 2023). These algorithms use historical gradient information to modify the model and remove specific data. FedEraser, for example, subtracts gradient updates to unlearn data (Liu et al., 2021).

This study aims to identify a more efficient method for data removal from trained models. By

^a <https://orcid.org/0009-0006-9073-1254>

combining Orthogonal Gradient Descent (OGD) and Steepest Descent, this paper seeks to improve unlearning efficiency. OGD minimizes the impact on remaining data by projecting the gradient onto an orthogonal vector, while Steepest Descent ensures the most rapid decrease in the gradient. This combination could enhance the unlearning process while maintaining model performance.

2 METHOD

2.1 Dataset Preparation

The dataset utilized in this research is the MovieLens 1M dataset (GroupLens Research, 2024). This dataset comprises 1,000,209 ratings from 6,040 users on 3,883 movies, making it a widely recognized benchmark in the field of recommendation systems. This dataset is chosen for the study to better explore the impact of the experimental unlearning algorithm on user data. Movie preferences often reflect personal tastes or characteristics, making movie preference prediction an ideal test case for assessing the effectiveness of user data removal in the model.

The dataset is divided into three parts: movie data, rating data, and user data. The movie data includes the MovieID, Title, and Genres, with the format: MovieID::Title::Genres. The rating data consists of UserID, MovieID, Rating, and Timestamp, formatted as UserID::MovieID::Rating::Timestamp. The user data includes UserID, Gender, Age, Occupation, and Zip-code, formatted as UserID::Gender::Age::Occupation::Zip-code.

The following preprocessing steps are applied to the dataset:

- For the **Gender** field, the values 'F' and 'M' are converted to 0 and 1, respectively;
- The **Age** field is transformed into ten continuous categories, ranging from 0 to 9;
- The **Genres** field, being categorical, is converted into numerical values. First, a dictionary mapping each genre to a numerical code is created. Then, the **Genres** field for each movie is converted into a list of numbers, as some movies belong to multiple genres;
- The **Title** field is processed similarly to the **Genres** field. A dictionary is created to map the text descriptions to numerical values, and the title descriptions are converted into numerical lists. Additionally, the year of release is removed from the title field;

- Both the **Genres** and **Title** fields are padded to a uniform length to facilitate processing within neural networks, with empty portions filled using the numerical code corresponding to "NA".

While the dataset is primarily concerned with ratings, movies can be categorized by genres, which can be extracted for further analysis about personal tastes or characteristics, which can be taken as part of user privacy.

2.2 Unlearning-based Federated Learning

2.2.1 Designing of the CNN Model

The algorithm in this paper was inspired by the CNN training method for text data proposed by Denny Britz (Denny Britz, 2015), and was adapted to the selection of natural language data sets in the experiment by the experimenter. Although CNNs are typically used for image-related tasks, this method is chosen to better align with the federated learning framework and code employed in this research (Fu et al., 2024 and Xu et al., 2023), facilitating prediction and the analysis of forgetting effectiveness. The research adheres closely to proven frameworks to avoid errors in evaluating the feasibility of the algorithm due to code-related issues. A sample diagram of the CNN model used in this paper is shown Figure 1, which includes the basic structure of the CNN framework.

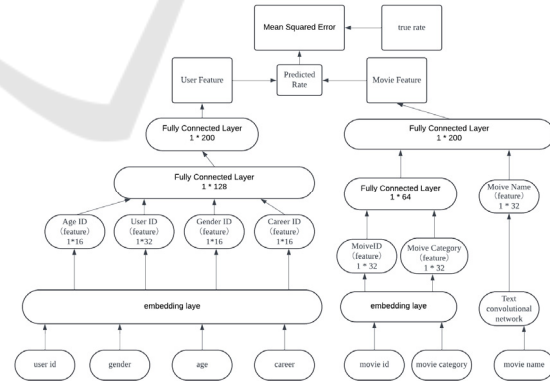


Figure 1: Designing of the CNN model (Photo/Picture credit: Original).

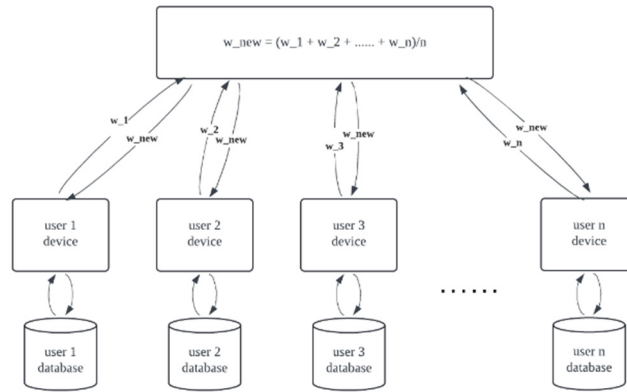


Figure 2: Federal Learning Process (Photo/Picture credit: Original).

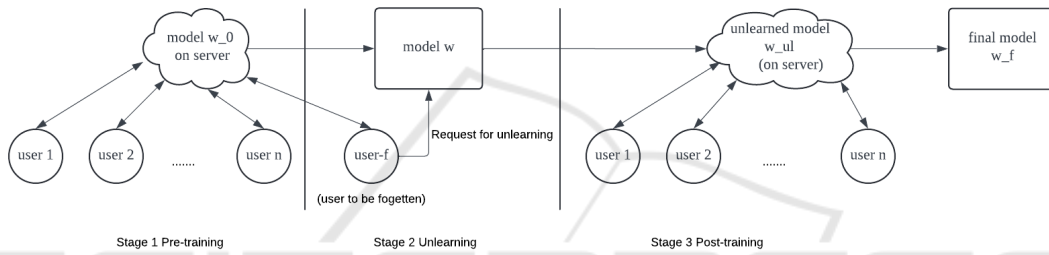


Figure 3: Experiment Design (Photo/Picture credit: Original).

2.2.2 Federated Learning Introduction

classic workflow of federated learning (FedAvg) is depicted in the Figure 2. Initially, users download an initial model neural network w from the server. Taking a linear regression model as an example, w_i represents the local model parameters for the i -th user after training on their individual data. In this federated learning setup, each user's device computes w_i by updating the global model w based on their local dataset. After local training, the updated model w_i is sent back to the server. The server then aggregates and updates the model, which is sent back to users for continued training, as shown in the formula: movies, with the corresponding true ratings being R_i .

$$w_{new} = \sum_{i=0}^n \frac{w_i}{n} \quad (1)$$

where n is the number of users, w_{new} is the updated global model.

The CNN model used in this paper follows a similar process. Due to limited resources, large-scale experiments cannot be conducted. Instead, this experiment simulates the process of different users

training locally by partitioning the dataset and training the model separately for each user. Based on the ratings provided by each user in the MovieLens dataset, this experiment successfully simulates the local training process. Since the dataset is reliably collected from MovieLens website users, partitioning the data by user ID allows to infer that the data contains each user's personal preferences. This partitioned dataset is then used to train the CNN model separately, serving as the foundation for verifying the experimental results.

2.2.3 Unlearn and Recover

The process described in this paper is divided into three stages shown in Figure 3. The first stage is pre-training, where the model is trained as users normally would on a federated learning platform. The second stage is unlearning, in which one user is selected as the opt-out participant, and the unlearning operation is performed using the FedRecovery algorithm (Cao et al., 2023). The final stage is model recovery, where training continues, and the model's subsequent performance is observed, focusing on the potential

improvement in accuracy and whether the forgotten data can be recovered.

This research focuses specifically on addressing the issues that arise after performing unlearning in algorithms such as FedRecovery (Cao et al., 2023), followed by post-retraining. This paper proposes replacing the steepest gradient descent method used in previous algorithms with Orthogonal Gradient Descent (OGD) (Farajtabar et al., 2020) and further applying this approach in post-training to observe whether the OGD algorithm improves the effectiveness of unlearning. The essence of the OGD method used in this experiment involves projecting the gradients from new tasks onto a subspace where the neural network's output on previous tasks remains unchanged, while ensuring that the projected gradient is still useful for learning the new task. Thus, a complete federated learning process for unlearning is constructed from beginning training, forgetting data, and continuing training.

3 RESULTS AND DISCUSSION

3.1 Evaluation Metrics

To demonstrate whether a user's personal data has been removed, this experiment compares the performance of three types of users on the model after unlearning using FedRecovery: User-common (UC), User-forgotten (UF), and User-unknown (UN), who

did not participate in the training. To reflect individual user preferences, score predictions are used to assess whether the model retains user data and remains usable. Theoretically, UC and UF should perform better on the model than UN before unlearning. However, after unlearning, UF and UN should exhibit similar performance on the model. The model predicts the ratings r_i (where i ranges from 0 to 9) for 10 movies, with the corresponding true ratings being R_i .

$$\text{Rate Loss} = \sum_{i=0}^9 (R_i - r_i) \quad (2)$$

3.2 Performance of Users on the Pre-trained Model

The experimental results align with expectations. At this stage, there is no significant difference between UC and UF shown in Figure 4, and both perform better on the model compared to UN.

3.3 Effectiveness and Cost of Unlearning

It can be observed that the model demonstrates a good unlearning effect for UF after the unlearning process, aligning with the experimental results described in the referenced FedRecovery algorithm papers (Cao et al., 2023). With the number of forgetting rounds increases, the effect of UF on the model gradually approaches that of UN as shown in Figure 5.

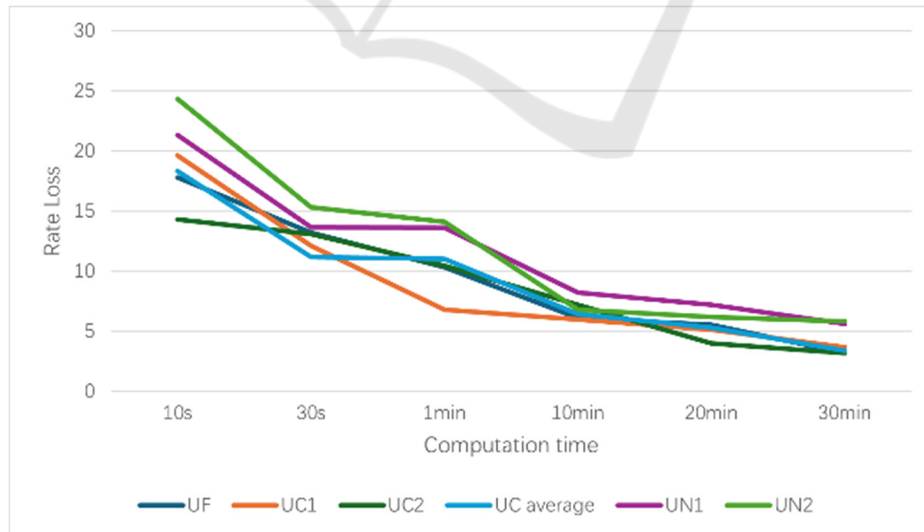


Figure 4: Performance of Users on the Pre-trained Model (Photo/Picture credit: Original).

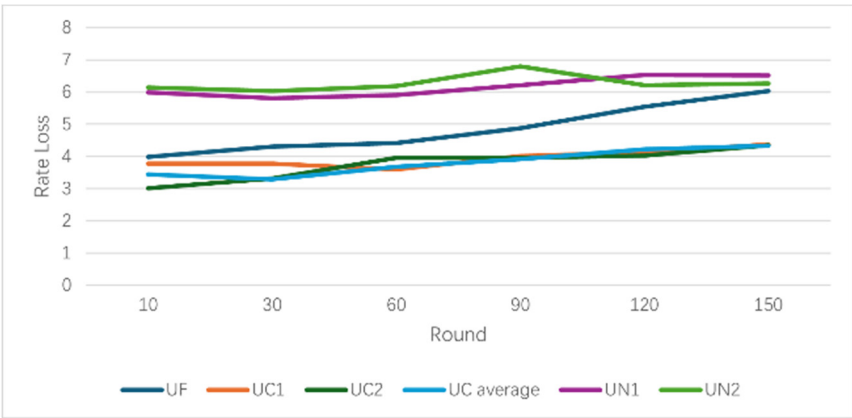


Figure 5: Performance of Users on the Unlearned Model (Photo/Picture credit: Original).

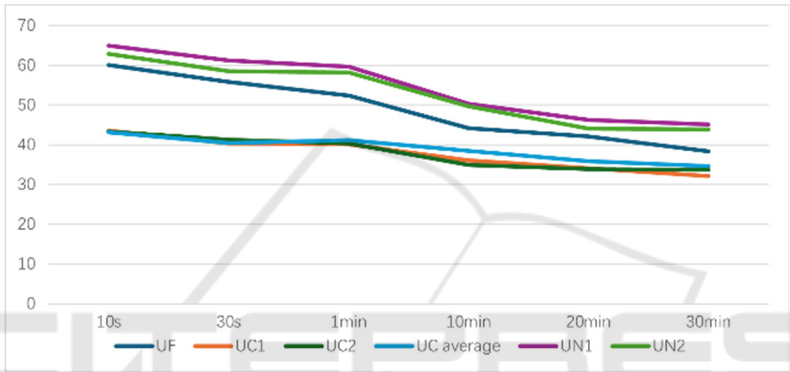


Figure 6: Performance of Users on Post-trained Original Model (Photo/Picture credit: Original).

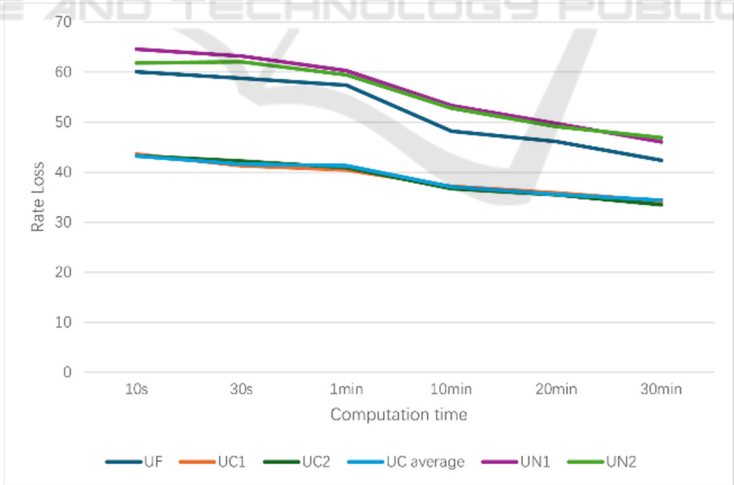


Figure 7: Performance of Users on Post-trained OGD Model (Photo/Picture credit: Original).

3.4 Further Observations on Post-Training After Unlearning

As can be seen in Figure 6, as training progresses, the UF data gradually recovers when the CNN model is

used for post-training. The phenomenon is specifically illustrated in Figure 6, where UF's RateLoss gradually diverges from UN's data and approaches UC's data. To more clearly observe the difference in data, multiply the rate loss by a factor of ten, data are shown in Figure 6 and Figure 7.

It can be seen that the use of the OGD algorithm has mitigated the aforementioned issue, although at the cost of slightly worse RateLoss performance. Additionally, UF's data still does not fully align with UN's data. To further address this issue, the Loss Function of the CNN model was modified, incorporating the accuracy of UF into the Loss Function. The new Loss Function is as follows:

$$\text{New loss} = \text{Loss} - \delta * \text{Sum}(\text{Loss of all UF}) \quad (3)$$

where δ is a hyperparameter, $\delta \geq 0$), aiming to push the model away from UF's data as much as possible.

However, the results were shown in Figure 8 but unsatisfactory, as the performance heavily depended on the setting of the hyperparameter δ , and the model's generalization ability was compromised as training progressed. The approach appeared to

deliberately avoid UF's data rather than bringing UF's performance closer to that of UN, the result is shown in Figure 8.

As a result, after 30 minutes' training, UF becomes even worse than UN when hyperparameters are too large. At the same time, in a larger pre-training (UC is set to 500, UF is set to 20, and the Rate Loss of each UF is included in the Loss Function), this method will reduce the generality of the model. For the same training time, the performance of the model decreases compared to the previous normal training model, the result can be seen in Figure 9.

Figure 9 illustrates a decline in the performance of the User-unknown (UN) category compared to Figure 4, notwithstanding the augmented participation of users in the training process. This observation suggests that merely altering the Loss Function may not effectively enhance the performance of Orthogonal Gradient Descent (OGD).

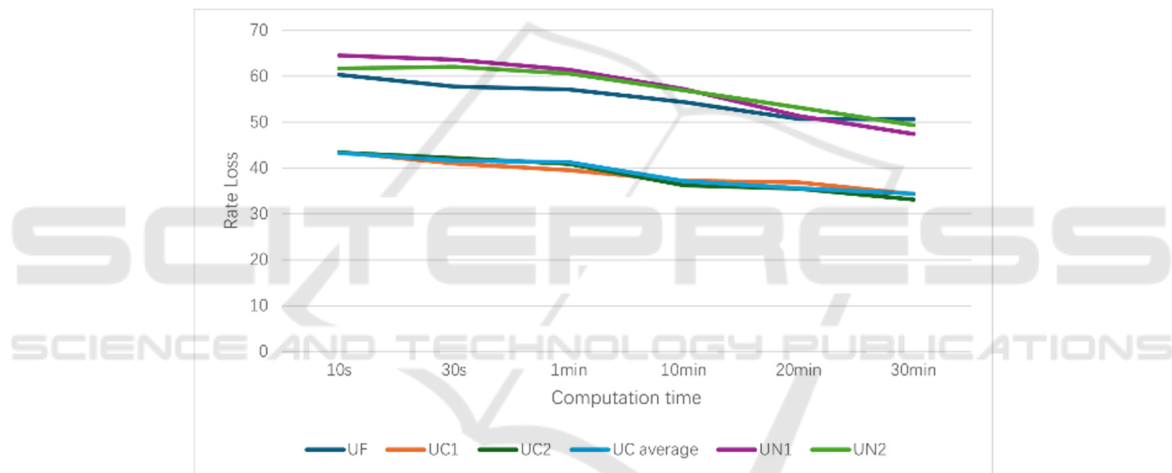


Figure 8: Performance of Users on the Modified OGD Model (Photo/Picture credit: Original).

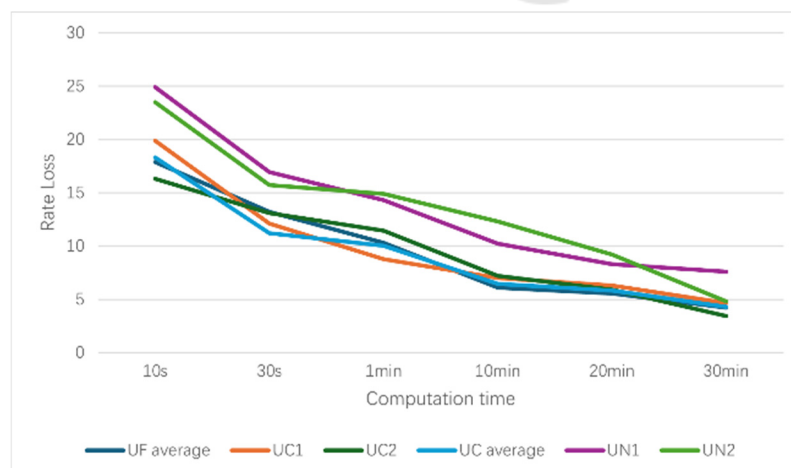


Figure 9: Extended Training (Photo/Picture credit: Original).

Consequently, it is recommended to continue employing the conventional OGD approach without modifications to the Loss Function. This approach aligns with current best practices and ensures the integrity of the model's performance across varying user involvements.

3.5 Discussion

The experiments show that OGD does not outperform SGD in terms of convergence speed. This is an expected issue, as OGD requires more computation to find the appropriate gradients. However, OGD does improve the model's performance during the post-training process.

The reason may lie in the fact that FedRecovery relies on differential privacy (as explained in the underlying principles), which does not entirely eliminate a user's personal data but instead perturbs it with noise (Cao et al., 2023).

The modified Loss Function attempts to push the model away from the user's original data direction, but this might cause issues with the model's generalization. Nevertheless, the experiments confirm OGD's value for unlearning. This is possibly because OGD inherently seeks to store more optimal gradient points on the boundary of gradual changes while providing good approximations for other points' gradients, preventing the storage of the full dataset's gradients (Farajtabar et al., 2020). Consequently, more thorough unlearning appears to occur for UF during the post-training phase.

4 CONCLUSIONS

This paper completes an evaluation of the use of Orthogonal Gradient Descent (OGD) in the context of federated learning and unlearning. The use of OGD significantly increases computational costs, with nearly a 10% increase in time required to achieve the same accuracy on the MovieLens 1M dataset in an hour's training process. However, OGD does have its merits. With adjustments to the hyperparameters, better performance is expected. In situations where time and computational resources are abundant and the need to ensure thorough unlearning of user data is critical, OGD can enhance the effectiveness of FedRecovery in unlearning data. Nevertheless, the application of OGD still faces inherent limitations. For instance, it fails severely when the task changes and becomes dissimilar to the previous task (e.g., rotating images more than 90 degrees for MNIST), and it is sensitive to the learning rate. Additionally,

steepest gradient descent can only be used before the first user requests unlearning; after that, all subsequent training must rely on OGD, which poses an unsolved issue, leading to computational overhead. The researchers hope that better solutions will emerge in the future to combine OGD with federated learning more effectively.

REFERENCES

- Britz, D. 2015. Understanding convolutional neural networks for NLP. Denny's Blog. <https://www.wildml.com/2015/11/understanding-convolutional-neural-networks-for-nlp/>
- Cao, X., Yu, L., Xu, Y., & Ma, X. 2023. FedRecover: Recovering from poisoning attacks in federated learning using historical information. In 2023 IEEE Symposium on Security and Privacy (SP) (pp. 1366–1383). IEEE.
- Chengstone. 2017. Movie recommender [Source code]. GitHub. https://github.com/chengstone/movie_recommender
- Farajtabar, M., Azizan, N., Mott, A., & Li, A. 2020. Orthogonal gradient descent for continual learning. In Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics, Proceedings of Machine Learning Research, 108, 3762–3773.
- Fu, J., Hong, Y., Ling, X., Wang, L., Ran, X., Sun, Z., & Cao, Y. 2024. Differentially private federated learning: A systematic review. arXiv Preprint, arXiv:2405.08299.
- GroupLens Research. 2003. MovieLens 1M dataset. GroupLens. Retrieved May 23, 2024, from <https://grouplens.org/datasets/movielens/1m/>
- Hojjat, K. 2018. MNIST dataset. Kaggle. Retrieved May 29, 2024, from <https://www.kaggle.com/datasets/hojjatk/mnist-dataset>
- Li, T., Sahu, A. K., Talwalkar, A., & Smith, V. 2020. Federated learning: Challenges, methods, and future directions. IEEE Signal Processing Magazine, 37(3), 50–60. <https://doi.org/10.1109/MSP.2020.2975749>
- Liu, G., Ma, X., Yang, Y., Wang, C., & Liu, J. 2020. Federated unlearning. arXiv Preprint, arXiv:2012.13891.
- Liu, G., Ma, X., Yang, Y., Wang, C., & Liu, J. 2021. FedEraser: Enabling efficient client-level data removal from federated learning models. In 2021 IEEE/ACM 29th International Symposium on Quality of Service (IWQOS) (pp. 1–10). IEEE.
- Xue, R., Xue, K., Zhu, B., Luo, X., Zhang, T., Sun, Q., & Lu, J. 2023. Differentially private federated learning with an adaptive noise mechanism. IEEE Transactions on Information Forensics and Security.