FedRL: Federated Learning with Non-IID Data via Review Learning

Jinbo Wang
1694300437@qq.com
University of Electronic Science and Technology of China
Chengdu, China

Ruijin Wang*
ruijinwang@uestc.edu.cn
University of Electronic Science and Technology of China
Chengdu, China

Xikai Pei
peixikai@cdatc.com
University of Electronic Science and Technology of China
Chengdu, China

ABSTRACT

Federated Learning epitomizes a sophisticated distributed machine learning methodology, enabling collaborative neural network model training across multiple entities without necessitating the transfer of local data, thereby fortifying data privacy protection. A significant challenge in federated learning lies in the statistical heterogeneity, characterized by non-independent and identically distributed (Non-IID) local data across diverse parties. This heterogeneity can engender inconsistent optimization within individual local models. Although previous research has endeavored to tackle issues stemming from heterogeneous data, our findings indicate that these attempts have not yielded high-performance neural network models. To confront this fundamental challenge, we introduce the FedRL framework in this paper, which facilitates efficient federated learning through review learning. The core principle of FedRL involves leveraging the knowledge representation generated by the global and local model layers to conduct periodic layer-by-layer comparative learning in a reciprocal manner. This strategy rectifies local model training, leading to enhanced outcomes. Our experimental results and subsequent analysis substantiate that FedRL effectively augments model accuracy in image classification tasks, while demonstrating resilience to statistical heterogeneity across all participating entities.

CCS CONCEPTS

• **Computing methodologies** → **Distributed computing methodologies**;

KEYWORDS

Machine learning, Federated learning, Heterogeneity data, Image classification

ACM Reference Format:
Jinbo Wang, Ruijin Wang, and Xikai Pei. 2024. FedRL: Federated Learning with Non-IID Data via Review Learning. In *2024 16th International Conference on Machine Learning and Computing (ICMLC 2024), February 02–05, 2024, Shenzhen, China.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3651671.3651704

ICMLC 2024, February 02–05, 2024, Shenzhen, China

ACM ISBN 979-8-4007-0923-4/24/02
https://doi.org/10.1145/3651671.3651704

1 INTRODUCTION

Distributed collaborative machine learning (DCML) has been widely adopted due to its ability to protect data privacy. Unlike traditional centralized machine learning (ML) approaches, DCML does not require the transfer of data from local sources to untrusted third-party processing centers. Instead, it involves transferring the ML model to a processing center for analysis and processing. Crucially, the processing center does not have access to the raw data used to train the model.

Currently, the most popular approach to DCML is federated learning (FL). FL involves training neural network models on distributed terminals using local data, and subsequently aggregating the local models uploaded by the terminals on a central server to obtain a global model. FL's main advantage lies in its ability to enable parallel training of models on multiple terminal nodes, while ensuring knowledge aggregation without the need to share local data. This approach effectively safeguards data privacy [6, 8, 10, 16], making it a highly efficient ML paradigm. The Federated Averaging Algorithm (FedAvg) [15] is the most widely used aggregation algorithm for FL. The algorithm involves using stochastic gradient descent (SGD) [3] to minimize the local model loss at each round of communication on the terminal. The central server subsequently obtains a weighted average of the local models from all parties, based on the volume of local data on each terminal, to derive the global model. The global model is then sent back to the terminals to update the local model.

However, some studies [7, 9, 12, 13, 17, 18, 22] show that FedAvg cannot effectively converge in highly statistically heterogeneous scenarios. In FL, each terminal participating samples training data independently based on locally generated data, leading to significant variation in data distribution across all parties. Non-IID data can result in inconsistent optimization directions for all parties'models, and the parameters of local models can be scattered. Consequently, the global model produced by aggregating the local models on the central server may deviate significantly from the local model [1, 5, 14]. To improve FL's robustness in the face of statistical heterogeneity scenarios, some studies have proposed the use of regularization items to limit local model updates and prevent model parameters from "drift". FedProx [13] proposes adding a proximal term to the objective function to limit the difference between the local model and the global model. SCAFFOLD [9] adjusts model parameters by controlling variables to reduce "drift". MOON [12] utilizes contrastive learning and the SimCLR [2] clustering algorithm to minimize the similarity between the representation of the global model and the local model, ultimately improving accuracy. While the aforementioned methods aim to minimize the gap

between local and global models to improve robustness and minimize local loss, they have not yielded satisfactory results in image classification tasks. In certain data distributions, their performance can even be as poor as that of FedAvg.

Building on the concept of reducing parameter "drift", we can consider implementing a constraint method during local model optimization and updates. This approach would enable the local model to fully learn the knowledge contained in the global model while minimizing local classification loss. As knowledge from the global model is transferred to the local model on the terminal, their similarity will increase, leading to convergence between the global and local optima. Inspired by the way people learn new subjects in daily life, which typically involves starting with a shallow understanding and gradually delving deeper, we can apply a similar iterative approach to learning the global model via the local model in FL. In FedRL, training is periodic, and the cycle length is determined by the depth of the model. During each cycle, the local model first learns the simplest knowledge representation of the global model (i.e., the shallowest representation). As communication rounds progress, the local model gradually begins to learn deeper knowledge representations of the global model until it reaches the final layer. At this point, the local model starts the next cycle of knowledge review and learning, starting again from the shallowest representation and progressing to deeper representations.

This paper has the following main contributions.

1. Our proposed FL framework addresses statistical heterogeneity in an efficient manner. The framework leverages both the shallow and deep knowledge representations of the model to correct local model training.
2. We incorporate a periodic knowledge review loss term into local model optimization, which significantly enhances FedRL's performance.
3. We have implemented FedRL and conducted extensive experiments on several datasets, including MNIST [4], FashionMNIST [21], CIFAR10 [11], and CIFAR100 [11]. Our results demonstrate that FedRL achieves state-of-the-art performance in terms of both model accuracy and convergence speed.

2 BACKGROUND AND RELATED WORK

FedAvg [15] is the inaugural algorithm proposed for FL. The main framework diagram of FedAvg is depicted in Fig. 1, which primarily comprises four steps. The first step involves terminals receiving an initialization model from the central server, followed by local model training using SGD on local data. The second step involves uploading the trained local model to the central server, and in the third step, the central server performs a weighted average of all local models to derive a global model. Finally, the central server sends the global model to all terminals for updating their local models in preparation for the next round of model training. However, numerous studies [1, 5, 7, 9, 12–14, 17, 18, 22] have demonstrated that FedAvg exhibits local model "drift" on Non-IID data, which leads to relatively poor convergence of the global model.

To enhance the expressiveness of FedAvg on Non-IID data, a number of studies have implemented regularization terms to constrain local model updates. FedProx [13] proposes adding a proximal term

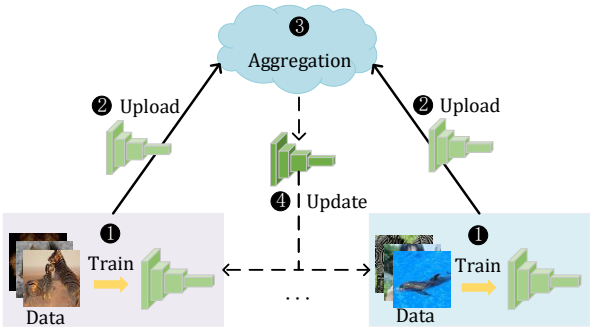

Figure 1: FedAvg overall architecture diagram.

to the objective function during local model optimization, which is computed using the two-norm between the local model and the global model. While FedProx can reduce the difference between the local and global models, it also restricts the global model's approach to the optimal point. SCAFFOLD [9] suggests incorporating control variables at each party and the aggregation center to reduce local model "drift". However, this approach can only minimize the "drift" rather than entirely eliminate it. MOON [12] proposes using the SimCLR [2] self-supervision approach to bring the local and global models progressively closer and to ensure that local models are distinct from each other across communication rounds. FedProc [17] also employs SimCLR, but differs from MOON in that it uses the feature map generated by the middle layer of the model as the input for SimCLR, whereas MOON uses the feature representation generated by an additional projector as input. In addition, FedDyn [1] and FedDC [5] both optimize model convergence by adding regular terms.

Other studies have addressed the Non-IID problem from different perspectives. For instance, FedBN [14] leverages local batch normalization to mitigate feature shift during local model optimization. Zhao et al. [22] used the earth mover's distance (EMD) to gauge the distance between local data distribution and the global distribution, elucidated the connection between EMD and weight divergence, and suggested that a smaller shared dataset could improve global model updates. Due to privacy concerns associated with shared dataset sampling in the aforementioned schemes, Tang et al. [18] proposed using virtual shared datasets in lieu of real shared datasets. Several studies [7, 19, 20] have demonstrated that the aggregation method of the central server can be improved. FedAvgM [7] proposes incorporating server momentum during aggregation to alleviate Non-IID issues. FedNova [20] suggests using the normalized average method to resolve model objective inconsistency while ensuring rapid convergence. FedMA [19] proposes hierarchical matching and averaging of aggregated weights using Bayesian non-parametric methods.

3 PERIODIC KNOWLEDGE REVIEW FEDERATED LEARNING

3.1 Problem Statement

Assuming N terminals participate in FL, each marked as Ter_1, Ter_2, ..., Ter_N, the local samples of terminal Ter_i are denoted as $\mathcal{D}_i$, and the local samples of different terminals belong Non-IID distributed datasets. The data from all terminals are aggregated together as $\mathcal{D} = \cup_{i\in[N]}\mathcal{D}_i$. The objective is to collaboratively train a model $F(w)$ such that it exhibits the best possible performance on $\mathcal{D}$, without the terminal nodes exchanging data with one another. The objective function of model training can be expressed as

$$\arg\min_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w}) = \sum_{i=1}^{N} \frac{|\mathcal{D}_i|}{|\mathcal{D}|}\mathcal{L}_i(\boldsymbol{w}). \tag{1}$$

where $\mathcal{L}_i(\boldsymbol{w})$ is local objective function of Ter_i, $\mathcal{L}_i(\boldsymbol{w}) = \frac{1}{|\mathcal{D}_i|}$ $\sum \ell_i(F_i(\boldsymbol{w}; x), y)$, $(x, y) \in \mathcal{D}_i$.

3.2 Method

From the above comparison, we deduce that mitigating local model "drift" is the key to resolving the Non-IID problem. To this end, we propose a more effective FL algorithm, FedRL, based on FedAvg, which seeks to slow down the local model "drift" by minimizing the local loss function while moving closer to the global model. The primary contribution of FedRL lies in the local training process of the terminal, where the best model classification performance is achieved through iterative knowledge transfer and learning from the global model, moving from shallow to deep. The overall framework is presented in Algorithm 1, where $F(\boldsymbol{w}^t)$ denotes the global model in round t, and $F_i(\boldsymbol{w}_i^t)$ represents the Ter_i local model in round t. For the sake of clarity, we use $F(\boldsymbol{w}[0, m])$ to denote the knowledge representation of the first m layers of the model. In the following section, we will elaborate on FedRL in detail, focusing on two aspects: periodic knowledge transfer and local objective function design.

Periodic Knowledge Review We propose a periodic knowledge review method to facilitate the local model's learning from the global model and mitigate local model "drift". Fig. 2 illustrates the framework of this approach. Assuming the model has M layers, the knowledge review cycle is M. At the m-th time node of a cycle, the first m layers of the global model and the local model $F(\boldsymbol{w}[0, m])$ map the input data x to the corresponding knowledge representation $\boldsymbol{kr}_m^t$ and $\boldsymbol{kr}_{i,m}^t$ ($\in \mathbb{R}^Q$), which is used to compute the periodic loss term $\boldsymbol{\ell}_{pkr}$. The local model proceeds with forward propagation using $\boldsymbol{kr}_{i,m}^t$, and the predicted value logits $\boldsymbol{output} \in \mathbb{R}^K$ is obtained in the final layer, which is used to calculate $\boldsymbol{\ell}_{ce}$.

Local Objective In FedRL, the loss function for terminal local model training comprises two components: the classic cross-entropy loss function $\boldsymbol{\ell}_{ce}$ for image classification and the periodic knowledge review loss $\boldsymbol{\ell}_{rl}$ proposed in this paper. By learning the knowledge of the global model from shallow to deep iteratively, the local model moves closer to the global model and alleviates local model "drift". Simultaneously, a constant term μ is utilized to determine the impact of the knowledge review loss on the overall loss function. The local loss can be defined as follows

$$\boldsymbol{\mathcal{L}} = \boldsymbol{\ell}_{ce} + \frac{\mu}{2}\boldsymbol{\ell}_{rl} \tag{2}$$

This loss function can bring the local model closer to the global model while minimizing the local classification loss, thereby slowing down local model "drift" caused by Non-IID local data. The detailed loss function is defined as follows

$$\boldsymbol{\mathcal{L}}_i(\boldsymbol{w}_i) = \frac{1}{\mathcal{D}_i}\sum_{b=0}^{|\mathcal{D}_i|/B}\sum_{j=0}^{B}\boldsymbol{\ell}_i(F_i(\boldsymbol{w}_i; x_j), y_j) + \frac{\mu}{2}||\boldsymbol{kr}_m^t - \boldsymbol{kr}_{i,m}^t||_2 \tag{3}$$

where B denotes the batch size. $\boldsymbol{kr}_m^t$ represents the knowledge representation generated by the m-th layer network of the global model during the t-th round of communication, and $\boldsymbol{kr}_{i,m}^t$ represents the knowledge representation generated by the m-th layer network of the local model on the i-th terminal. All terminals use the SGD algorithm to update the local model according to the local data, and the update objective is defined as in the above (3).

Algorithm 1 The FedRL framework.

Input: The number of terminals N, local dataset of Ter_i $\mathcal{D}_i$, layers of model M, communication rounds T, local train epoch E, batch size B, learning rate η;
Onput: The final global model $\boldsymbol{w}^T$;
SERVERAGGREGATION()
  Initialize $\boldsymbol{w}^0$;
  For t in $[0, T]$ **do**
    For i in $[0, N]$ parallel **do**
      Send $\boldsymbol{w}^t$ to Ter_i;
      $\boldsymbol{w}_i^{t+1} \leftarrow$ TERMINALLOCALTRAIN(i, t, $\boldsymbol{w}^t$);
    EndFor
  EndFor
  $\boldsymbol{w}^{t+1} \leftarrow \sum_{[N]}^{i=0} \frac{|\mathcal{D}_i|}{|\mathcal{D}|}\boldsymbol{w}_i^{t+1}$;
  return $\boldsymbol{w}^T$;
TERMINALLOCALTRAIN(i, t, $\boldsymbol{w}^t$)
  Initialize $m \leftarrow 0$;
  $\boldsymbol{w}_i^t \leftarrow \boldsymbol{w}^t$
  For e in $[0, E]$ **do**
    For batch $\{x_j, y_j \mid j < B\}$ of $\mathcal{D}_i$ **do**
      # get knowledge representation (kr) of $\boldsymbol{w}_i^t$, $\boldsymbol{w}^t$;
      $\boldsymbol{kr}_m^t \leftarrow F(\boldsymbol{w}^t[0, m])$;
      $\boldsymbol{kr}_{i,m}^t \leftarrow F_i(\boldsymbol{w}_i^t[0, m])$;
      $\boldsymbol{\ell}_{pkr} \leftarrow ||\boldsymbol{kr}_m^t, \boldsymbol{kr}_{i,m}^t||_2$;
      $output \leftarrow F_i(\boldsymbol{w}_i^t; x_j)$;
      $\boldsymbol{\ell}_{ce} \leftarrow CrossEntropy(output, y_j)$;
      $\boldsymbol{\mathcal{L}} \leftarrow \boldsymbol{\ell}_{ce} + \frac{\mu}{2} \cdot \boldsymbol{\ell}_{pkr}$;
      m+ = 1;
      $m\%$ = M;
      $\boldsymbol{w}_i^{t+1} \leftarrow \boldsymbol{w}_i^t - \eta \cdot \nabla\boldsymbol{\mathcal{L}}$;
    EndFor
  EndFor
  Send $\boldsymbol{w}_i^{t+1}$ to server;

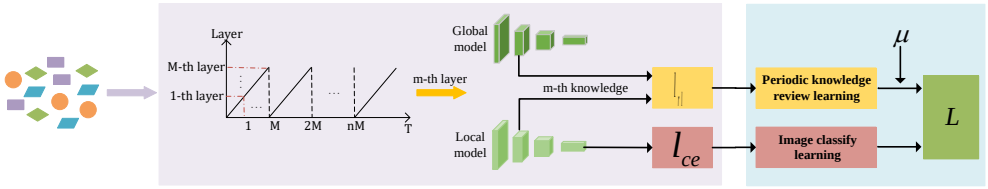

Figure 2: Periodic Knowledge Review Learning framework.

4 EXPERIMENT

4.1 Experimental Setup

1) *Dataset*: The efficacy of the FedRL framework is validated using four standard datasets: MNIST [4], FashionMNIST [21], CIFAR10 [11], and CIFAR100 [11]. We sample $p_k \sim DirN(\beta)$ and allocate a $p_{k,j}$ proportion of the instances of class k to party j, where $Dir(\beta)$ is the Dirichlet distribution with a concentration parameter β (0.5 by default). We set number of terminal is 10 by default. For some classifications, each party may contain little or no data, as shown in Fig 3. In order to simulate real-life scenarios more realistically, terminal does not *shuffle* when using *dataloader* to load training data.

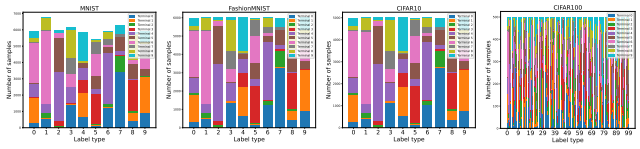

Figure 3: Data distribution of different data sets on different terminals.

2) *Model*: For MNIST and FashionMNIST, we use a convolutional neural network comprising two convolutional layers, two pooling layers, and two fully connected layers. For CIFAR10, we also use a simple convolutional neural network comprising two convolutional layers, two pooling layers, and three fully connected layers [12]. For CIFAR100, we use ResNet50.

3) *Baseline*: We compare FedRL with several state-of-the-art FL algorithms, including FedAvg [15], FedProx [13], and SCAFFOLD [9], where FedProx and FedRL are compatible for FedAvg (When μ=0, FedProx and FedRL are equal to FedAvg. We also include the SOLO baseline method [12], which refers to the end-point results obtained by training a local model using local data.

4) *Training details*: We use pytorch to implement FedRL and other baselines, all codes are running on 3090-GPU. For all FL algorithms, we use SGD when the terminal locally trains the model, and the initial learning rate lr is set to 0.01. lr is divided by 10 in the 40-th and 80-th rounds of communication. During model training, the SGD weight decay is set to 0.00001 and the SGD momentum is set to 0.0001, and the batch size is set to 32. The entire training process comprises 120 rounds of communication, with 3 rounds of local training performed in each round of communication. The number of local training rounds for SOLO is set to 300 rounds.

4.2 Accuracy Comparison

We experimented with multiple values of μ within the range [0.0001, 0.01] for FedRL and identified the optimal values for MNIST, FashionMNIST, CIFAR10, and CIFAR100 datasets as {0.005, 0.004, 0.001, 0.0001} respectively. We determined the optimal μ values for the four datasets in FedProx, while maintaining consistency in the other hyperparameters used across all FL algorithms throughout the experimental process.

Using the aforementioned configurations, we evaluated the highest Non-IID test accuracy of various FL algorithms on different datasets. Table 1 presents the recorded top-1 test accuracies. The results indicate that the SOLO scheme consistently yields the lowest accuracy across different datasets, highlighting the advantages of FL. In addition, the SCAFFOLD algorithm did not achieve the expected level of accuracy and performed worse than FedProx and FedRL, particularly on CIFAR100. The FedProx algorithm achieved an accuracy very similar to that of FedAvg, suggesting that the proximal term has a negligible impact on model optimization. However, this comes at the expense of increased computational overhead at the terminal. The proposed FedRL algorithm in this paper achieved the highest accuracy on the four datasets, surpassing FedAvg by 0.53%, 3.61%, 0.48%, and 0.51% respectively. The improvement is particularly noteworthy on the FashionMNIST dataset, where it outperformed FedProx by almost 4 percentage points.

Table 1: Top-1 test accuracy of different FL algorithms on different datasets.

Method	MNIST	FashionMNIST	CIFAR10	CIFAR100
FedAvg	97.19%	77.66%	62.97%	65.12%
FedProx	97.28%	77.85%	63.17%	64.86%
SCAFFOLD	97.55%	73.49%	59.86%	51.67%
SOLO	81.76%	53.78%	37.96%	25.35%
FedRL	**97.72%**	**81.27%**	**63.45%**	**65.63%**

4.3 Communication Efficiency

In addition to accuracy, communication efficiency is a crucial metric for evaluating the effectiveness of FL. Our comparison of FedRL with various baseline methods revealed that FedRL can achieve the same accuracy with the fewest number of communication rounds, thereby demonstrating superior communication efficiency. To establish a benchmark, we recorded the number of communication rounds required for FedAvg to achieve a specific model accuracy

Table 2: Communication efficiency of different FL algorithms on different datasets.

Method	MNIST		FashionMNIST		CIFAR10		CIFAR100	
	#rounds	speedup	#rounds	speedup	#rounds	speedup	#rounds	speedup
FedAvg	120	1.00x	120	1.00x	120	1.00x	120	1.00x
FedProx	86	1.40x	91	1.32x	97	1.24x	/	/
SCAFFOLD	104	1.15x	/	/	/	/	/	/
FedRL	**51**	**2.35x**	**50**	**2.40x**	**80**	**1.50x**	**59**	**2.03x**

at 120 rounds of communication. We then noted the number of communication rounds and the multiples of improvement in communication efficiency for other FL algorithms when they first surpassed the benchmark accuracy. Table 2 presents the recorded data, which indicates that FedRL has improved communication efficiency on different datasets. Specifically, it achieved a 2.35x and 2.40x improvement on MNIST and FashionMNIST, respectively. Notably, on FashionMNIST, FedRL matched the accuracy of FedAvg at 120 rounds of communication in just 50 rounds of communication. However, FedProx only achieved a 1.40x and 1.32x improvement in communication efficiency, and SCAFFOLD performed even worse. On CIFAR10 and CIFAR100, FedProx and SCAFFOLD performed even worse, with communication efficiency not even as good as FedAvg. In contrast, FedRL still achieved a speed increase of 1.5x and 2.03x on CIFAR10 and CIFAR100, respectively. The experimental results demonstrate that FedRL can accelerate the convergence speed and improve communication efficiency while also enhancing model training accuracy.

4.4 Runtimes

While achieving high-performance collaborative training with various FL algorithms, it is also important to ensure that these algorithms do not introduce excessive computational burden at the terminal. To evaluate the computational burden at the terminal, we measured the time taken for local training to complete at each round of communication for all terminals. We then calculated the average time taken per round of communication over 120 rounds and compared the running time of different FL algorithms, as shown in Table 3. The results indicate that the training time of FedAvg is consistently the lowest, as it does not introduce any additional computational overhead. FedProx incurs a relatively large computational cost due to the calculation of the norm of the entire model parameters, resulting in the longest training time among all algorithms. In contrast, FedRL only utilizes a specific layer of the model parameters, resulting in a significantly lower computational cost compared to FedProx. Therefore, the training time for FedRL is also lower. On the four datasets, FedRL achieved higher accuracy and communication efficiency while also reducing training time by 5.33s, 3.37s, 5.68s, and 53.48s compared to FedProx, respectively. Although SCAFFOLD has a training time similar to FedRL, it significantly underperformed in terms of accuracy and communication efficiency compared to FedRL.

5 CONCLUSION AND FUTURE WORK

This paper outlines the fundamental challenges associated with Non-IID data in FL and provides an overview of existing solutions.

Table 3: Runtimes of different FL algorithms on different datasets.

Method	MNIST	FashionMNIST	CIFAR10	CIFAR100
FedAvg	40.64s	40.96s	73.91s	217.46s
FedProx	53.17s	51.35s	87.99s	351.37s
SCAFFOLD	49.25s	48.19s	**79.30s**	319.52s
FedRL	**47.84s**	**47.98s**	82.31s	**297.89s**

Nevertheless, it is worth noting that these solutions are considerably intricate and may pose difficulties in achieving optimal performance when training models in Non-IID scenarios. To this end, this paper presents FedRL, a straightforward and effective approach to Federated Learning that involves regular knowledge review. Specifically, FedRL periodically employs various neural network layers, ranging from shallow to deep, for comparative learning, with the aim of mitigating local model “drift". Our extensive experimentation demonstrates that FedRL can significantly accelerate model convergence and enhance the resilience of Federated Learning in Non-IID settings.

We currently focuses on the application of FedRL to the task of image classification. However, the potential applicability of FedRL to other fields, such as target detection, reinforcement learning, natural language processing, and more, is an area of ongoing research and development. Future work will explore the effectiveness of FedRL in these different domains and assess its potential for improving the performance of FL in a broader range of applications.

ACKNOWLEDGMENTS

This work is supported by National Natural Science Foundation of China(62271128), national key R&D projects(2022YFB3304303) and key R&D projects of Sichuan Science and Technology Plan (2022ZDZX0004, 2023YFG0029, 2023YFG0150, 2023ZHCG004, 2022YFG0212, 2021YFS0391, 2021YFG0027).

REFERENCES

[1] Durmus Alp Emre Acar, Yue Zhao, Ramon Matas Navarro, Matthew Mattina, Paul N Whatmough, and Venkatesh Saligrama. 2021. Federated learning based on dynamic regularization. *arXiv preprint arXiv:2111.04263* (2021).
[2] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*. PMLR, 1597–1607.
[3] J Michael Cherry, Caroline Adler, Catherine Ball, Stephen A Chervitz, Selina S Dwight, Erich T Hester, Yankai Jia, Gail Juvik, TaiYun Roe, Mark Schroeder, et al. 1998. SGD: Saccharomyces genome database. *Nucleic acids research* 26, 1 (1998), 73–79.
[4] Li Deng. 2012. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE signal processing magazine* 29, 6 (2012), 141–142.

[5] Liang Gao, Huazhu Fu, Li Li, Yingwen Chen, Ming Xu, and Cheng-Zhong Xu. 2022. Feddc: Federated learning with non-iid data via local drift decoupling and correction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 10112–10121.
[6] Robin C Geyer, Tassilo Klein, and Moin Nabi. 2017. Differentially private federated learning: A client level perspective. *arXiv preprint arXiv:1712.07557* (2017).
[7] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. 2019. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335* (2019).
[8] Hai Huang, Jiaming Mu, Neil Zhenqiang Gong, Qi Li, Bin Liu, and Mingwei Xu. 2021. Data poisoning attacks to deep learning based recommender systems. *arXiv preprint arXiv:2101.02644* (2021).
[9] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. 2020. Scaffold: Stochastic controlled averaging for federated learning. In *International conference on machine learning*. PMLR, 5132–5143.
[10] Pang Wei Koh, Jacob Steinhardt, and Percy Liang. 2022. Stronger data poisoning attacks break data sanitization defenses. *Machine Learning* (2022), 1–47.
[11] Alex Krizhevsky, Geoffrey Hinton, et al. 2009. Learning multiple layers of features from tiny images. (2009).
[12] Qinbin Li, Bingsheng He, and Dawn Song. 2021. Model-contrastive federated learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 10713–10722.
[13] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. 2020. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems* 2 (2020), 429–450.
[14] Xiaoxiao Li, Meirui Jiang, Xiaofei Zhang, Michael Kamp, and Qi Dou. 2021. Fedbn: Federated learning on non-iid features via local batch normalization. *arXiv preprint arXiv:2102.07623* (2021).
[15] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*. PMLR, 1273–1282.
[16] Nima Mohammadi, Jianan Bai, Qiang Fan, Yifei Song, Yang Yi, and Lingjia Liu. 2021. Differential privacy meets federated learning under communication constraints. *IEEE Internet of Things Journal* 9, 22 (2021), 22204–22219.
[17] Xutong Mu, Yulong Shen, Ke Cheng, Xueli Geng, Jiaxuan Fu, Tao Zhang, and Zhiwei Zhang. 2023. Fedproc: Prototypical contrastive federated learning on non-iid data. *Future Generation Computer Systems* 143 (2023), 93–104.
[18] Zhenheng Tang, Yonggang Zhang, Shaohuai Shi, Xin He, Bo Han, and Xiaowen Chu. 2022. Virtual homogeneity learning: Defending against data heterogeneity in federated learning. In *International Conference on Machine Learning*. PMLR, 21111–21132.
[19] Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris Papailiopoulos, and Yasaman Khazaeni. 2020. Federated learning with matched averaging. *arXiv preprint arXiv:2002.06440* (2020).
[20] Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H Vincent Poor. 2020. Tackling the objective inconsistency problem in heterogeneous federated optimization. *Advances in neural information processing systems* 33 (2020), 7611–7623.
[21] Han Xiao, Kashif Rasul, and Roland Vollgraf. 2017. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747* (2017).
[22] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. 2018. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582* (2018).