Proceedings of the 2023 International Conference on Machine Learning and Automation
DOI: 10.54254/2755-2721/39/20230592

The experiment of federated learning algorithm

Rui Wang

School of Intelligent Systems Engineering, Sun Yat-sen University, Guangzhou, 510260, China

Wangr267@mail2.sysu.edu.cn

Abstract. As technology advances, concerns regarding data privacy and security have become prominent challenges in machine learning applications. Nevertheless, the introduction of federated learning technology has effectively tackled this concern by concurrently enhancing model performance and preserving data privacy, thereby presenting a more secure and efficient solution within our digital realm. Utilizing discussions about the background of federated learning technology, coupled with pertinent algorithmic procedures and logic, this paper proficiently implements the FedMA algorithm. Additionally, the study performs a comparative analysis of the accuracy and efficiency of the FedMA, FedAvg, FedDyn, and MOON algorithms utilizing the Fashion-MNIST dataset. Moreover, the investigation not only optimizes parameter tuning for the MOON algorithm but also extends experiments to the CIFAR-10 and AGNews datasets, thereby providing additional comparisons of performance and strengths among various federated learning algorithms. In conclusion, the paper provides a comprehensive summary and outlines potential avenues for future research. These insights enhance the comprehension of federated learning and offer valuable guidance to advance and refine its practical applications.

Keywords: Federated Learning, The FedMA Algorithm, The MOON Algorithm, Fashion-MNIST.

1. Introduction

In contemporary society, artificial intelligence plays a pivotal role across diverse domains, driving innovation and progress in a multitude of fields.

The progression of big data technology has led to the emergence of machine learning models as proficient instruments for managing vast datasets. Nonetheless, challenges regarding data privacy and security have arisen as notable obstacles in the implementation of machine learning. In response to this concern, the technology of federated learning has surfaced [1]. Federated learning constitutes a decentralized machine learning methodology that facilitates collaborative model training among numerous data sources, safeguarding the confidentiality of raw data [2].

1.1. Background

The rapid advancement of technologies, including mobile devices, IoT, and cloud computing, has led to the generation, storage, and transmission of vast volumes of data. This data is frequently distributed across various devices and servers, under the ownership of diverse organizations and users [1]. Consequently, challenges arise concerning the efficient utilization of data and the extraction of information through mining processes.

Conventional machine learning and deep learning techniques typically necessitate centralizing data on a central server or within a data center for training purposes. Nevertheless, this practice carries the potential risk of compromising data privacy, particularly in scenarios where sensitive personal information is involved. Furthermore, centralized training can exert substantial strain on computational resources and bandwidth, especially for endpoints with limited resources, such as mobile devices [3].

The emerging distributed learning framework of Federated Learning effectively tackles the challenges of preserving privacy and optimizing resource usage that are inherent in centralized learning [4][5]. Within the scope of Federated Learning, data remains localized on the original devices, avoiding transmission to a central server. At its core, this approach entails training models locally on individual devices, sharing solely model update gradients or parameters to achieve updates on a global model scale. This strategy not only safeguards the privacy of data but also alleviates the load associated with data transmission.

Every device or client (e.g., mobile devices, sensors, edge servers, etc.) is regarded as a participant within the federated system, possessing an individual localized dataset. Via Federated Learning algorithms, these participants can conduct local model training and subsequently integrate their model updates through aggregation algorithms, resulting in the acquisition of global model updates.

Federated Learning demonstrates substantial applicability across a range of real-world scenarios. For example, in the medical domain, Federated Learning facilitates collaborative training of medical models across numerous healthcare institutions, all while upholding patient privacy, ultimately augmenting the accuracy of medical diagnoses [6][7]. In the context of intelligent transportation, Federated Learning has the potential to optimize traffic flow and forecast congestion, all without necessitating access to individual driver travel data [8][9].

1.2. Highlights

Presently, Federated Learning confronts various challenges. These encompass achieving a balance between the accuracy of the global model and the diversity inherent in local models, mitigating the effects of participant heterogeneity, and formulating efficient aggregation algorithms.

The objective of this experiment is to conduct a more in-depth exploration of Federated Learning, with a specific focus on achieving the following objectives:

- Theoretical analysis is conducted on the execution processes and logic of various Federated Learning algorithms.
- A comparison is made among Federated Learning algorithms using diverse datasets.

2. Methodology

2.1. FedAvg Algorithm

The Federated Averaging algorithm (FedAvg) [1] represents a machine learning methodology utilized in the domain of federated learning. This method operates based on gradients and is primarily tailored for the training of extensive machine learning models within a distributed framework.

As shown in Fig. 1, the core principle of this algorithm involves disseminating the training process across multiple devices. Each device utilizes its localized dataset for model training and subsequently uploads the trained parameters to the server. The server computes the mean of the locally trained model parameters uploaded by all devices, yielding the global model parameters. Subsequently, these global parameters are transmitted back to each device, enabling the initiation of iterative training. The FedAvg algorithm presents numerous benefits, encompassing the preservation of privacy, alleviation of communication overhead, accommodation of heterogeneous devices, and enhancement of model performance.

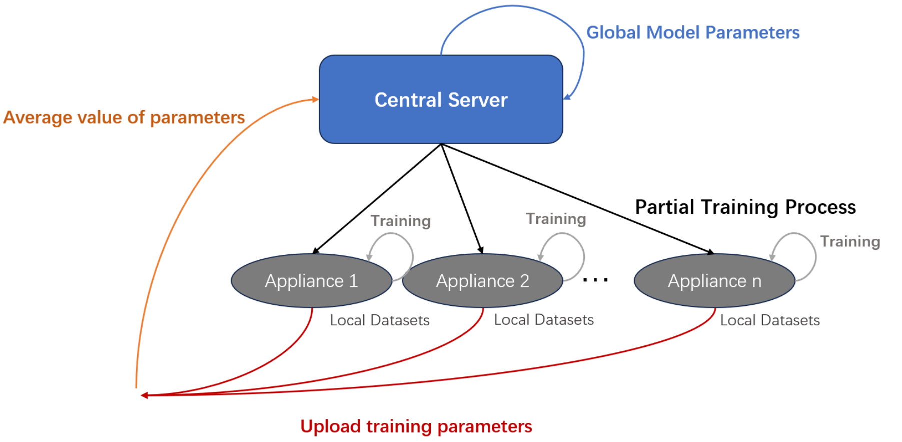

Figure 1. Illustration of the FedAvg Algorithm (Photo/Picture credit: Original).

2.2. *FedMA Algorithm*

FedMA [10] presents an innovative algorithm tailored for Federated Learning. It utilizes a matrix factorization approach to facilitate the training of large-scale machine learning models within a distributed environment.

As shown in Fig. 2, in contrast to the FedAvg algorithm, the FedMA algorithm distributes the training process across multiple devices and preserves a local matrix replica on each device. It individually applies matrix factorization to these local matrices using local data. Afterward, each device sends the outcome of its local matrix's factorization to the server. The server computes the factorization outcome of the global matrix and returns it to each device. This iterative process occurs multiple times to update the local matrices. The FedMA algorithm provides enhanced training efficiency and improved model performance while simultaneously safeguarding user privacy and minimizing communication overhead.

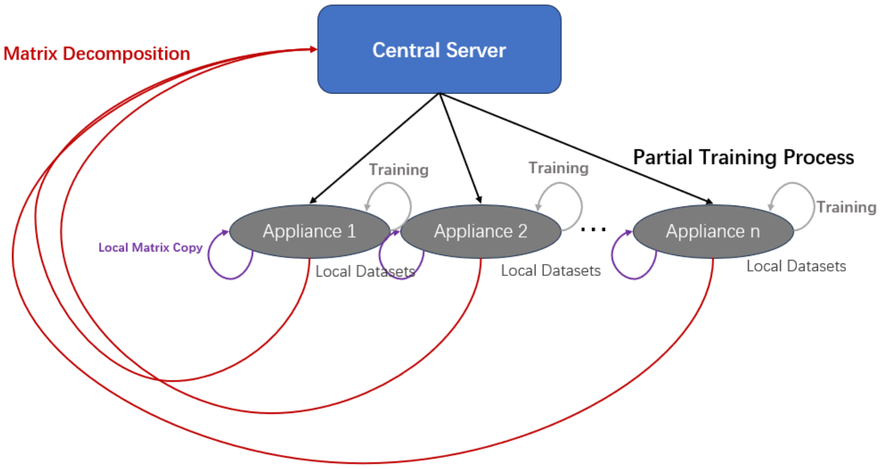

Figure 2. Illustration of the FedMA Algorithm (Photo/Picture credit: Original).

The FedMA algorithm utilizes a technique referred to as 'momentum' to perform parameter updates in each iteration. Momentum encompasses the concept of historical gradients, which serves to expedite parameter updates and amplify training velocity. Specifically, the FedMA algorithm employs momentum on the individual local matrices of each device to effect updates over multiple iterations. Subsequently, the server calculates the global factorized matrix and subsequently transmits it back to each device. This process is recurrently replicated across numerous iterations, facilitating updates to the local matrices until convergence is attained.

2.3. FedDyn Algorithm

The FedDyn (Federated Dynamic Learning) algorithm [11] represents a federated learning methodology grounded in the principles of dynamic federated learning.

Dynamic federated learning constitutes a framework that encompasses both model updates and data updates, which are also recognized as data streams. Diverging from FedAvg, the FedDyn algorithm introduces a dynamic update module depicted in Fig. 1. This module's purpose is to attain a dynamic equilibrium between model updates and data updates. This integration is intended to amplify both model performance and the efficiency of data utilization, as exemplified in Fig. 3. FedDyn possesses the capacity to dynamically adapt the proportion of model updates and data updates, guided by the attributes of the data stream. This dynamic adjustment culminates in enhanced learning outcomes.

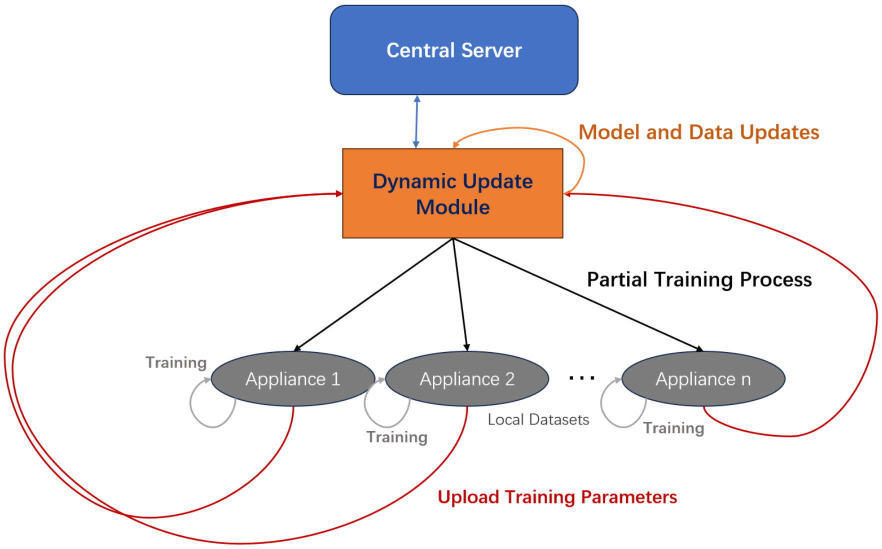

Figure 3. Illustration of the FedDyn Algorithm (Photo/Picture credit: Original).

2.4. MOON Algorithm

The MOON algorithm [12] presents a federated learning solution meticulously crafted to tackle challenges related to scope-constrained scheduling. Illustrated in Figure 4, its foundational principle revolves around arranging devices into a binary tree based on a designated rule. Subsequently, the algorithm transmits the model update information from each device to its corresponding parent node to facilitate aggregation. During the process of aggregation, the algorithm considers various factors, encompassing the weights and quantities of local samples on individual devices, alongside the weights and quantities of parent node devices. This consideration leads to the computation of a novel model parameter vector. Subsequently, the algorithm commences the dissemination of the recently computed model parameter vector, starting from the root node device and propagating down to each successive level. This procedure guarantees that every device obtains the updated model parameter vector for their model updates.

The MOON algorithm exhibits a time complexity of O(n^2), with n signifying the number of devices. Notwithstanding its relatively elevated time complexity, this algorithm showcases exceptional performance in practical applications and is characterized by straightforward implementation.

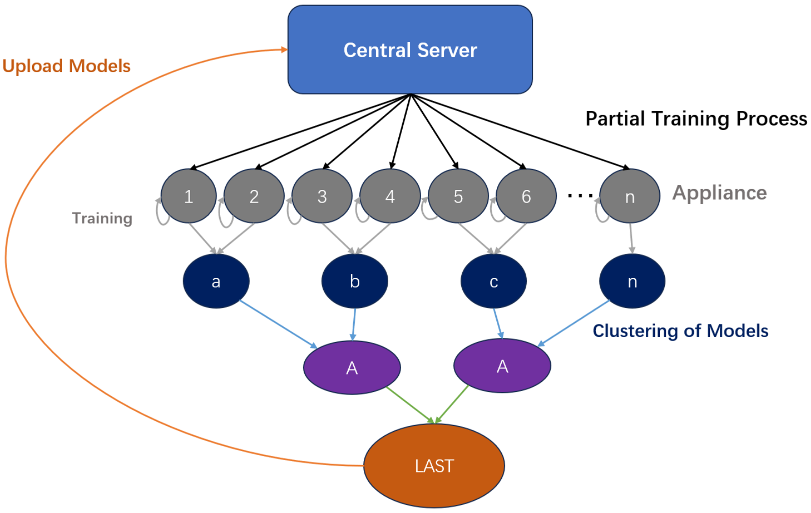

Figure 4. Illustration of the MOON Algorithm (Photo/Picture credit: Original).

This algorithm proves apt for specific federated learning scenarios distinguished by scope-constrained challenges. For example, it finds utility in situations wherein the completion of all device model updates is imperative within a designated timeframe, or when a pre-defined count of device model updates must be achieved within a particular duration.

2.5. PFL-Non-IID

PFL-Non-IID [13] stands as an open-source personalized federated learning platform accessible on GitHub. Presently, the platform provides a repertoire of 29 FL/pFL algorithms, 3 scenarios, and 14 datasets. Moreover, the platform holds the capability to simulate scenarios encompassing more than 2080 clients utilizing a sole GPU card (such as 100Ti with 500GB memory). This holds, for instance, in cases similar to the Cifar10 or Cifar100 datasets.

The framework and code employed in this experiment are constructed upon the underlying framework of this platform.

3. Experiments

3.1. Preparation

3.1.1. Environment and Hardware. This experiment was conducted on a Windows 10 operating system, using an AMD Ryzen 9 5900HX processor and an NVIDIA GeForce RTX 3080 Laptop GPU (32GB). The runtime environment consisted of Python 3.9.16 and Pytorch 2.0.1+cu117.

3.1.2. Dataset. Apart from the expanded experiments, this study utilized the Fashion-MNIST dataset [14].

Fashion-MNIST functions as a dataset dedicated to image classification tasks, representing an enriched and extended version of the renowned MNIST [15] handwritten digit dataset. It encompasses images spanning 10 categories of fashion items: T-shirts/tops, Trousers, Pullovers, Dresses, Coats, Sandals, Shirts, Sneakers, Bags, and Ankle boots.

Every image within the Fashion-MNIST dataset is in grayscale, with dimensions of 28x28 pixels, identical in size to images in the MNIST dataset. Pixel values span the range from 0 to 255, effectively representing grayscale intensity.

The objective underlying the creation of Fashion-MNIST was to subject algorithms to evaluation using more intricate and real-world images. Although the MNIST dataset experienced significant utilization in the past for the validation and comparison of image classification algorithms, it might appear excessively simplistic in light of contemporary deep learning capabilities. Consequently, Fashion-MNIST has emerged as a substitute dataset, extensively employed for the assessment and juxtaposition of diverse deep learning models and algorithms.

Employment of the Fashion-MNIST dataset closely mirrors that of the MNIST dataset, seamlessly integrating into a range of machine learning frameworks and algorithms. It has evolved into a widely acknowledged benchmark dataset within both academic and industrial domains, especially in the realm of computer vision for image classification tasks.

3.1.3. Programming Implementation of FedMA Algorithm. To adapt the FedMA algorithm from the FedDyn code implementation, a simple averaging of the model parameters across all clients is necessary. This contrasts with the approach in FedDyn, where the global model vector is used to modify the loss function.

3.2. Algorithm Comparison

This experiment entails a cross-sectional comparative analysis of the performance demonstrated by the FedMA, FedAvg, MOON, and FedDyn algorithms. The experiment employs the Fashion-MNIST dataset along with a CNN model, spanning 100 epochs.

Table 1 presents a performance comparison of the algorithms, whereas Fig.5 and Fig.6 respectively illustrate the curves depicting the variation in accuracy and loss pertinent to each algorithm.

The experimental outcomes are as follows:

• The FedDyn algorithm demonstrates the swiftest convergence, emerging as the prime selection within 22-40 rounds. Nevertheless, excessive training round numbers might induce the phenomenon of gradient explosion, potentially resulting in instability.

• The FedMA algorithm attains the peak final accuracy and exhibits the most steadfast training process, designating it as the optimal choice beyond 40 rounds.

• The performance of the FedAvg algorithm and the MOON algorithm concerning the Fashion-MNIST dataset displays close similarity. The former achieves a notably faster training pace compared to the latter, and their accuracy reaches its zenith within the initial 22 rounds of training.

Table 1. Accuracy of results and runtime for each algorithm.

Algorithm	Accuracy	Consumption time (s)
FedMA	0.87129	21.43
FedDyn	0.86139	25.71
FedAvg	0.81188	20.37
MOON	0.81188	34.95

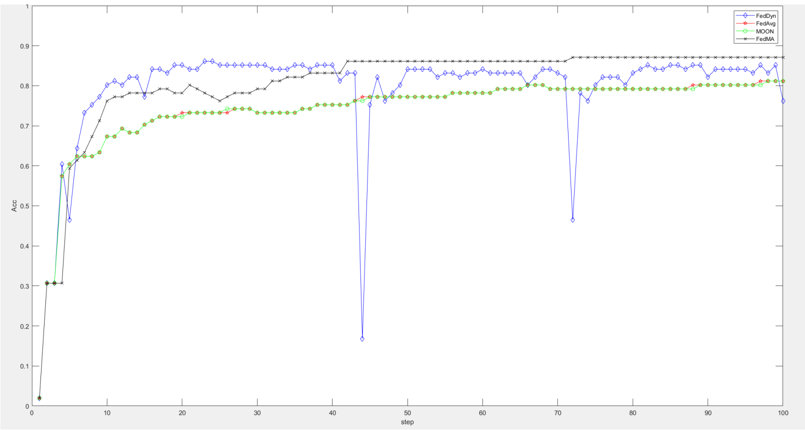

Figure 5. Trends in prediction accuracy (Photo/Picture credit: Original).

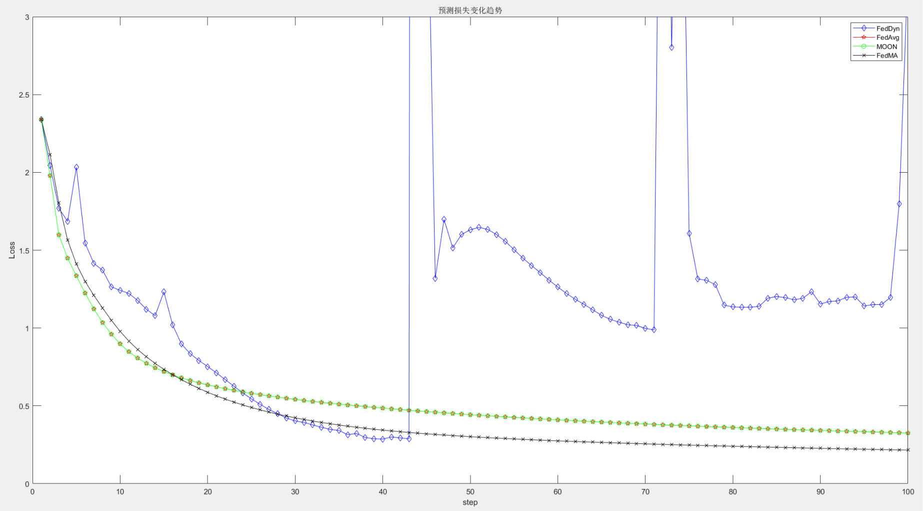

Figure 6. Trends in prediction loss (Photo/Picture credit: Original).

3.3. Parameter Optimization

In the parameter optimization experiments, the MOON algorithm, Fashion-MNIST dataset, and CNN model are employed.

3.3.1. The Number of Clients. Through the manipulation of client numbers, Table 2 presents a performance comparison of the algorithm across varying client quantities. Fig.7 and Fig.8 individually illustrate the curve reflecting the fluctuation in accuracy, and the curve indicating the variation in loss, exhibited by the MOON algorithm across varying client quantities.

The experimental outcomes are as follows:

- A higher client count corresponds to a lengthier runtime duration.
- The algorithm's accuracy does not exhibit a linear correlation with the number of clients. Excessively high or insufficient client numbers can impact the algorithm's accuracy. In this particular experiment, configuring the client count to 5 led to the algorithm attaining peak accuracy and a comparably swifter convergence pace.

Table 2. Algorithmic performance across varying client quantities.

Number of clients	Accuracy	Consumption time (s)
2	0.81188	34.95
5	0.88689	250.12
8	0.78076	901.58

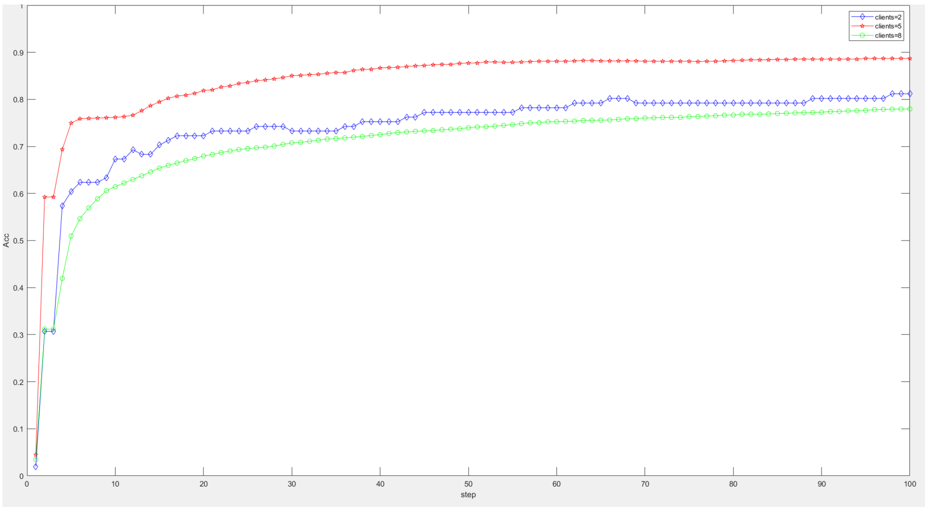

Figure 7. Trends in prediction accuracy (Photo/Picture credit: Original).

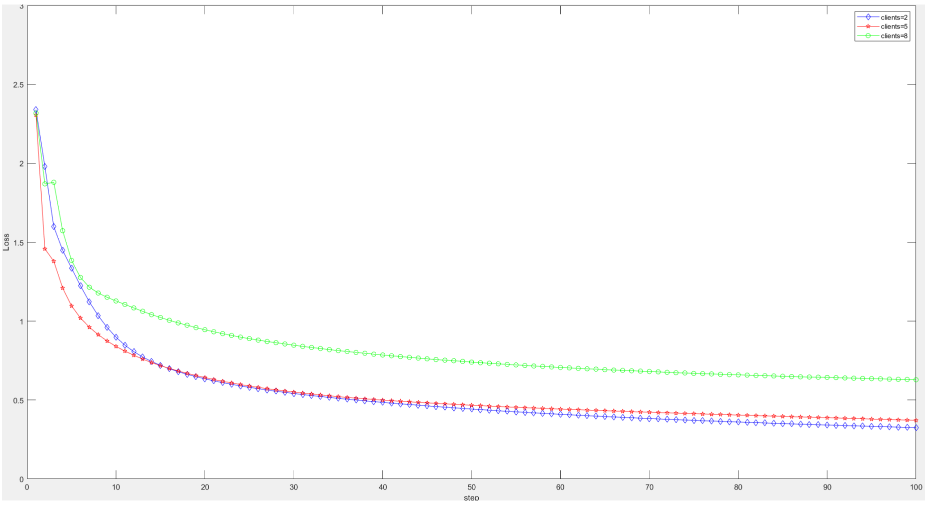

Figure 8. Trends in prediction loss (Photo/Picture credit: Original).

3.3.2. The Number of Global Rounds. Through the manipulation of global rounds, Table 3 presents a performance comparison of the MOON algorithm across diverse global rounds. Fig.9 and Fig.10 individually depict the curves depicting prediction accuracy and loss variation, about the MOON algorithm across distinct aggregation round configurations.

The experimental outcomes are as follows:

- A higher count of global rounds corresponds to a lengthier runtime duration.

Table 3. Algorithmic performance across varying global rounds.

Number of global rounds	Accuracy	Consumption time (s)
50	0.77228	12.78
100	0.81188	34.95
500	0.81188	102.4

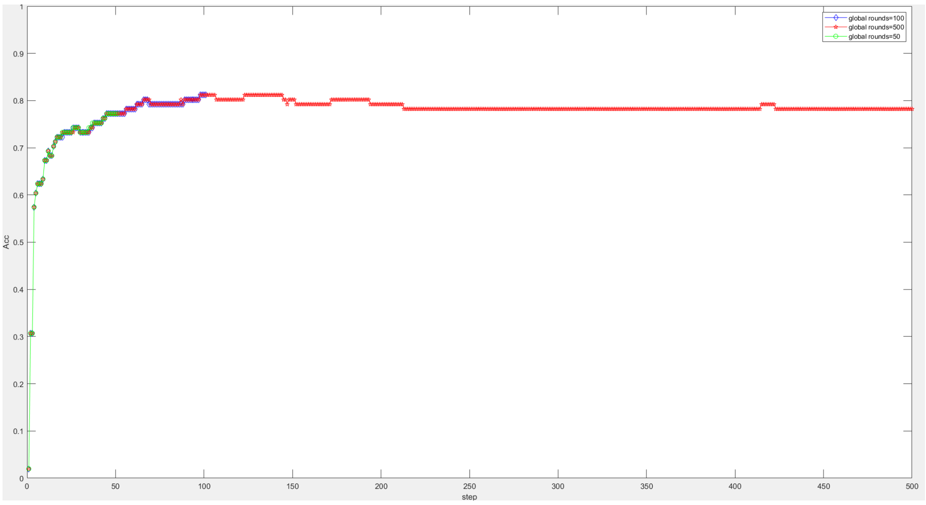

Figure 9. Trends in prediction accuracy (Photo/Picture credit: Original).

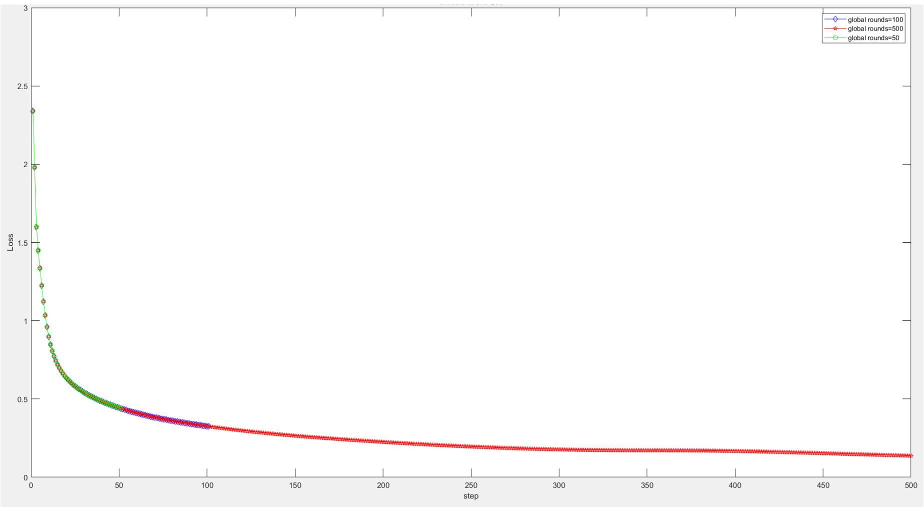

Figure 10. Trends in prediction loss (Photo/Picture credit: Original).

- The algorithm's performance remains unaffected by the number of global rounds. The accuracy and loss curves corresponding to the algorithm retain their shape regardless of the round count. If the round count is excessively low, the algorithm might terminate prior to attaining peak accuracy, leading to diminished accuracy. Once the algorithm stabilizes, augmenting the round count does not yield heightened accuracy.

3.3.3. Learning Rate. Through the manipulation of the learning rate, Table 4 presents a performance comparison of the algorithm across varying learning rate settings. Fig.11 and Fig.12 depict the fluctuations in prediction accuracy and loss associated with the MOON algorithm across distinct learning rate configurations.

Table 4. Algorithmic performance across varying learning rates.

Learning rate	Accuracy	Consumption time (s)
0.05	0.83168	23.19
0.005	0.81188	34.95
0.0005	0.66337	22.53

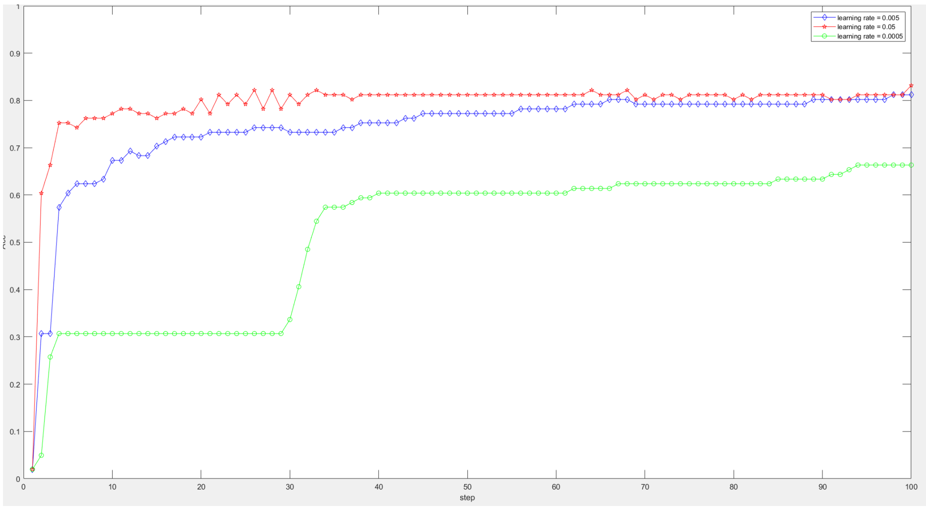

Figure 11. Trends in prediction accuracy (Photo/Picture credit: Original).

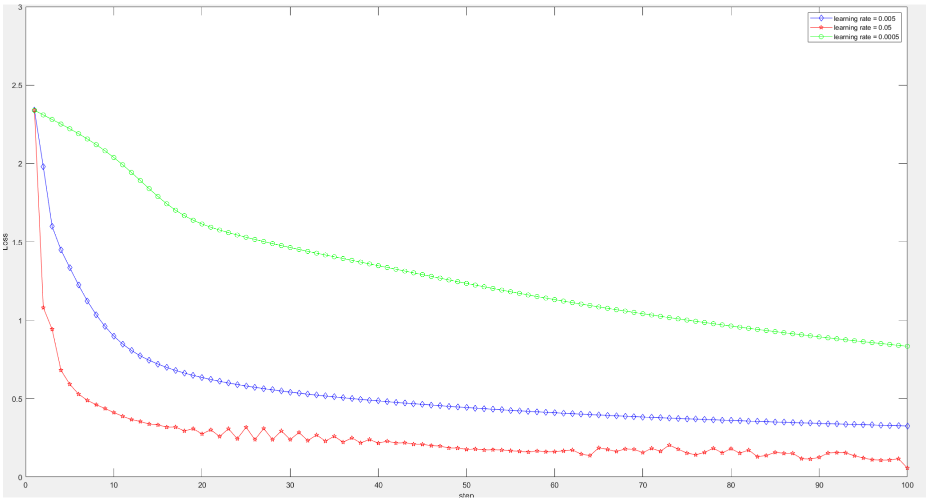

Figure 12. Trends in prediction loss (Photo/Picture credit: Original).

The experimental outcomes are as follows:

- The algorithm's runtime experiences negligible influence from the learning rate.
- The learning rate substantially influences the algorithm's convergence pace. A heightened learning rate prompts swifter convergence. In this specific experiment, a learning rate of 0.05 yielded the algorithm's peak accuracy. Nevertheless, according to theoretical analysis, excessively elevated learning rates could induce overfitting or gradient explosion. Generally, although larger learning rates necessitate a reduced number of training epochs, opting for a larger learning rate is not necessarily optimal.

3.4. Extension Experiments on Additional Datasets

This experiment involves executing the pertinent federated learning algorithms on supplementary datasets to delve deeper into their performance and the merits and drawbacks they present.

3.4.1. Image Dataset (CIFAR-10). CIFAR-10, a widely recognized dataset extensively employed in machine learning and computer vision, encompasses 60,000 32x32 pixel color images, distributed across 10 classes with 6,000 images per class. Its relatively modest image size and class count have established CIFAR-10 as a prominent benchmark for diverse computer vision tasks.

The experiments utilized the ResNet18 [16] model. Table 5 provides a performance comparison of distinct algorithms using the CIFAR-10 dataset, and Fig.13 and Fig.14 respectively depict the fluctuation in predictive accuracy and loss for these algorithms.

Table 5. Algorithmic performance on CIFAR-10.

Algorithms	Accuracy	Consumption time (s)
MOON	0.56163	657.37
FedMA	0.54419	312.9
FedDyn	0.72209	409.05

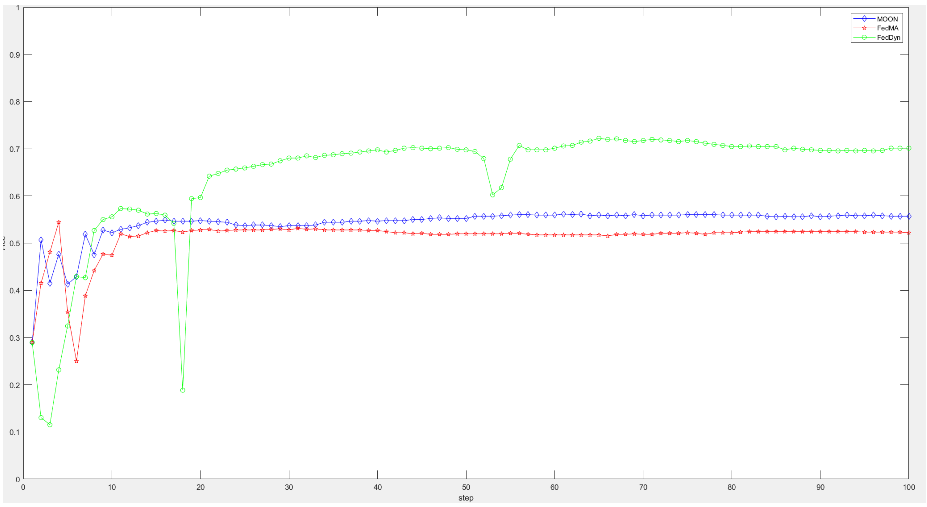

Figure 11. Trends in prediction accuracy (Photo/Picture credit: Original).

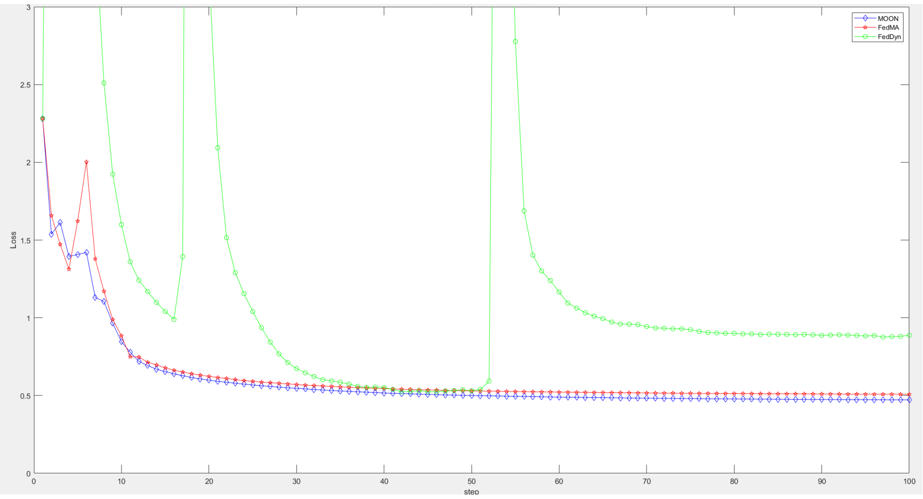

Figure 12. Trends in prediction loss (Photo/Picture credit: Original).

The experimental outcomes are as follows:

- In this experiment, the FedDyn algorithm demonstrates the least stability, yet it yields the most accurate predictions.
- The FedMA and MOON algorithms showcase relatively stable convergence processes. However, their ultimate predictive accuracies tend to be moderate, with FedMA exhibiting marginally lower accuracy.

- Among the algorithms, FedMA boasts the swiftest runtime, whereas MOON exhibits the slowest performance. In comparison, MOON is comparatively less suitable for the present scenario.
- The algorithms' performance on the CIFAR-10 dataset generally lags behind that of the Fashion-MNIST dataset, possibly owing to the elevated complexity of the former dataset.

3.4.2. NLP Dataset (AG News). The AG News dataset serves text classification purposes and holds a primary significance in the realms of natural language processing and machine learning. The dataset gathers internet news articles, classifying them into four distinct categories. Each category includes thousands of news articles, totaling approximately 120,000 articles. Each article is presented in concise text form and has undergone preprocessing, frequently denoted as a Bag of Words representation.

Owing to its moderate scale and diverse array of news subjects, the AG News dataset holds the position of a frequently utilized benchmark dataset in academic and industrial spheres alike. It serves as a mechanism for evaluating the performance and efficacy of various algorithms.

The experiments utilized the LSTM model [17][18]. Table 6 provides a performance comparison of distinct algorithms using the AG News dataset, and Fig.13 and Fig.14 respectively depict the fluctuation in predictive accuracy and loss for these algorithms.

The experimental outcomes are as follows:

- All three algorithms yield equivalent final prediction accuracies; however, FedMA showcases the highest efficiency, whereas MOON demonstrates the lowest efficiency.
- The MOON algorithm achieves the quickest convergence, whereas the loss curve of the FedDyn algorithm displays fluctuations.

Table 6. Algorithmic performance on AG News.

Algorithms	Accuracy	Consumption time (s)
MOON	0.93690	582.37
FedMA	0.93690	274.89
FedDyn	0.93690	294.31

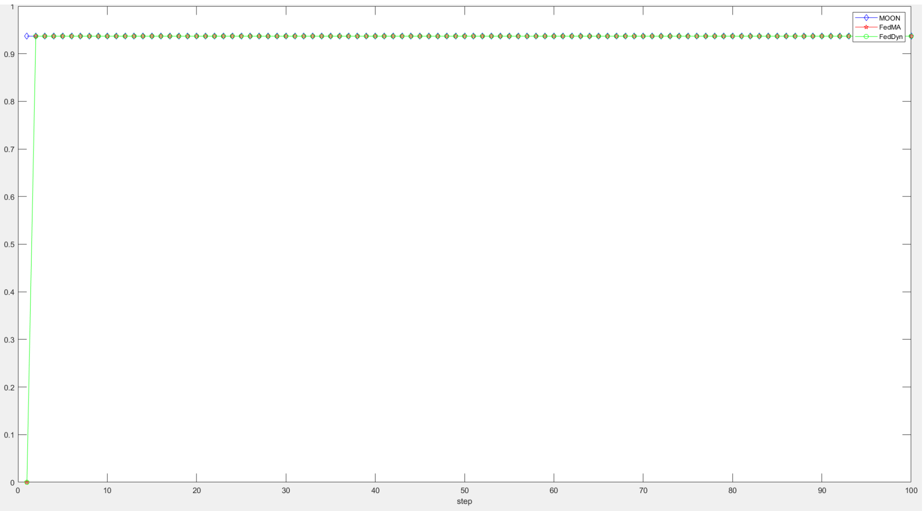

Figure 13. Trends in prediction accuracy (Photo/Picture credit: Original).

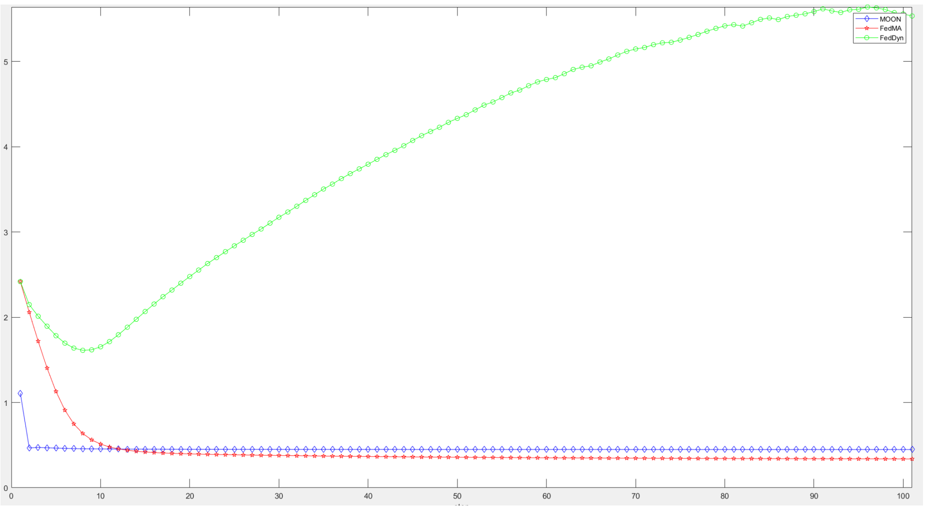

Figure 14. Trends in prediction loss (Photo/Picture credit: Original).

4. Conclusion

4.1. Summary

The experiment commenced by implementing the code for the FedMA algorithm, followed by a comparative analysis of its efficiency and accuracy against the FedAvg, MOON, and FedDyn algorithms. Subsequently, an effort was undertaken to optimize the parameters of the MOON algorithm. The acquired results suggest that the optimal algorithmic accuracy is attained with 5 clients and a learning rate of 0.05. However, it is plausible that these parameters are not optimal, and fine-tuning experiments might yield more refined outcomes. Additionally, in the extended experiment, a comparative analysis of diverse federated learning algorithms was conducted using the CIFAR-10 and AG News datasets.

This experiment has enhanced the comprehension of the principles and framework of federated learning, providing a deeper insight into algorithms like FedMA, MOON, and FedDyn. Among these, through coding and reproducing the FedMA algorithm and conducting a series of comparative experiments, a deeper understanding of its principles, advantages, and limitations has been acquired and experimentally validated. The FedMA algorithm showcases a rapid training pace (attained via multiplication updates, reducing model parameter synchronization frequency), minimal communication overhead (multiplicative updates necessitate transmitting a gradient component, not the entire gradient), and the capacity to sustain superior model performance (multiplicative updates effectively approximate the global gradient, augmenting the model's generalization prowess). Nonetheless, the algorithm also poses specific challenges, including its intricate nature (involving Bayesian nonparametric methods and multiple matrix multiplication operations) and vulnerability to initial parameters (incorrect parameter initialization may induce training instability, affecting model performance).

4.2. Prospect

Given time limitations, ample opportunities remain for further exploration within this experiment. Subsequent research could investigate the following aspects:

4.2.1. Enhanced Parameter Optimization for the MOON Algorithm. The experiment made only limited efforts to optimize parameters for the MOON algorithm, without achieving optimal results. Subsequent research could involve further adjustments to parameters like client count and learning rates, employing gradient descent techniques to ascertain optimal values.

4.2.2. Streamlining the Algorithm. The experiment solely excluded unused algorithms and models from the original code repository, without modifying their specifics. For greater experiment efficiency, elements concerning security and privacy could be excised from the code, enhancing simplicity and fostering a more experimentally oriented environment.

4.2.3. Further Extension Experiments. While this experiment encompassed extension experiments on both image and NLP datasets, numerous suitable datasets for experimentation still exist. Subsequent research could encompass broader datasets, broadening the scope for testing within the experiment.

References

[1] McMahan, B., Moore, E., Ramage, D., Hampson, S., &Arcas, B. A.: Communication-efficient learning of deep networks from decentralized data. In: Artificial Intelligence and Statistics (AISTATS), pp. 1273-1282. PMLR (2017).

[2] Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., ... &Zhao, S.: Advances and open problems in federated learning. Foundations and Trends® in Machine Learning, 14(1–2), 1-210 (2021).

[3] Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., & Zhang, L.: Deep learning with differential privacy. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pp. 308-318 (2016).

[4] Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H. B., Patel, S., & Seth, K.: Practical secure aggregation for privacy-preserving machine learning. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, pp. 1175-1191 (2017).

[5] Yang, Q., Liu, Y., Chen, T., & Tong, Y.: Federated machine learning: Concept and applications. ACM Transactions on Intelligent Systems and Technology (TIST), 10(2), 1-19 (2019).

[6] Cui, J., Zhu, H., Deng, H., Chen, Z., & Liu, D.: FeARH: Federated machine learning with anonymous random hybridization on electronic medical records. Journal of Biomedical Informatics, 117, 103735 (2021).

[7] Chen, S., Duan, B., Zhu, C., Tang, C., Wang, S., Gao, Y., & Liu, Q.: Privacy-preserving integration of multiple institutional data for single-cell type identification with scPrivacy. Science China Life Sciences, 66(5), 1183-1195 (2023).

[8] Yu, F., Xu, Z., Qin, Z., & Chen, X.: Privacy-preserving federated learning for transportation mode prediction based on personal mobility data. High-Confidence Computing, 2(4), 100082 (2022).

[9] Liu, Y., James, J. Q., Kang, J., Niyato, D., & Zhang, S.: Privacy-preserving traffic flow prediction: A federated learning approach. IEEE Internet of Things Journal, 7(8), 7751-7763 (2020).

[10] Wang, H., Yurochkin, M., Sun, Y., Papailiopoulos, D., & Khazaeni, Y.: Federated learning with matched averaging. arXiv preprint arXiv:2002.06440 (2020).

[11] Acar, D. A. E., Zhao, Y., Navarro, R. M., Mattina, M., Whatmough, P. N., & Saligrama, V.: Federated learning based on dynamic regularization. arXiv preprint arXiv:2111.04263 (2021).

[12] Tan, Y., Long, G., Ma, J., Liu, L., Zhou, T., & Jiang, J.: Federated learning from pre-trained models: A contrastive learning approach. Advances in Neural Information Processing Systems, 35, 19332-19344 (2022).

[13] PFL-Non-IID Homepage, https://github.com/TsingZ0/PFL-Non-IID, last accessed 2023/8/15

[14] Xiao, H., Rasul, K., & Vollgraf, R.: "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms." arXiv preprint arXiv:1708.07747 (2017).

[15] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324 (1998).

[16] He, K., Zhang, X., Ren, S., & Sun, J.: Deep residual learning for image recognition. In Proceedings of the IEEE Conference on computer vision and pattern recognition, pp. 770-778. Publisher, Location (2016).

[17] Hochreiter, S., & Schmidhuber, J.: Long short-term memory. Neural Computation, 9(8), 1735-1780 (1997).

[18] Graves, A., & Graves, A.: Long short-term memory. Supervised sequence labeling with recurrent neural networks, 37-45 (2012).