DOI: 10.37943/17BVCN7579

Gulnara Bektemyssova
Candidate of Technical Sciences, Professor of the Department of Computer Engineering
g.bektemisova@iitu.edu.kz, orcid.org/0000-0002-0850-0558
International Information Technology University, Kazakhstan

Gulnaz Bakirova
PhD candidate of the Department of Computer Engineering
g.bakirova@iitu.edu.kz, orcid.org/0009-0006-5643-2349
International Information Technology University, Kazakhstan

COMPARATIVE ANALYSIS OF FEDERATED MACHINE LEARNING ALGORITHMS

Abstract: In this paper, the authors propose a new machine learning paradigm, federated machine learning. This method produces accurate predictions without revealing private data. It requires less network traffic, reduces communication costs and enables private learning from device to device. Federated machine learning helps to build models and further the models are moved to the device. Applications are particularly prevalent in healthcare, finance, retail, etc., as regulations make it difficult to share sensitive information. Note that this method creates an opportunity to build models with huge amounts of data by combining multiple databases and devices. There are many algorithms available in this area of machine learning and new ones are constantly being created. Our paper presents a comparative analysis of algorithms: FedAdam, FedYogi and FedSparse. But we need to keep in mind that FedAvg is at the core of many federated machine learning algorithms. Data testing was conducted using the Flower and Kaggle platforms with the above algorithms.

Federated machine learning technology is usable in smartphones and other devices where it can create accurate predictions without revealing raw personal data. In organizations, it can reduce network load and enable private learning between devices. Federated machine learning can help develop models for the Internet of Things that adapt to changes in the system while protecting user privacy. And it is also used to develop an AI model to meet the risk requirements of leaking client's personal data. The main aspects to consider are privacy and security of the data, the choice of the client to whom the algorithm itself will be directed to process the data, communication costs as well as its quality, and the platform for model aggregation.

Keywords: federated learning; FedAvg; FedAdam; FedYogi; FedSparse; loss; accuracy.

Introduction

The usage of federated Learning helps clients to train a global model without sharing their data. It is a new machine learning paradigm that works with decentralized data from multiple clients to train a global model [1]. These days, "big" data is collected by distributed networks made up of gadgets, cars, and cellphones. Local data storage is getting more and more appealing due to the devices is rising processing capability and worries about the transmission of sensitive information. Clients in federated learning independently gather local data according to their own device [2].

Scientific Journal of Astana IT University
ISSN (P): 2707-9031 ISSN (E): 2707-904X
VOLUME 17, MARCH 2024

The main idea is sending the model to the device and the data stays on the device. This method guarantees that the learning process of the model works right on the end devices. So, you can train the model on datasets located in different places without having to interact with the actual data, to make the universal global model with no necessity to centralize the local data. Individualized data stays localized, minimizing the potential risk of client data exposure [3].

Put differently, the training of a machine learning model through federated learning doesn't necessitate the data to be situated on a central server. Instead, federated learning employs decentralized approaches to train models that are centrally located. The training process in federated learning is iterative, involving multiple rounds of training. This iterative nature ensures ongoing learning and the exchange of knowledge [4].

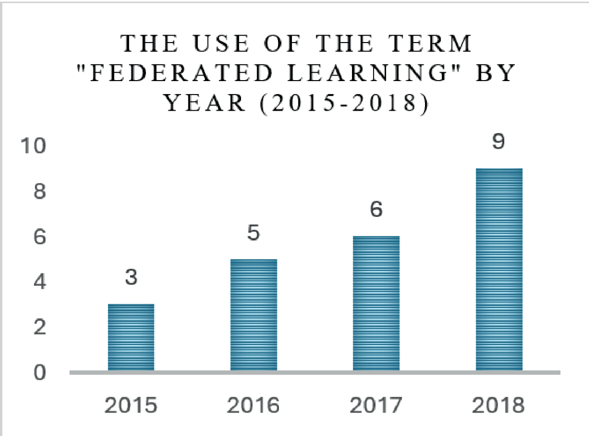
The description of the working process is:

1. to select a model that has either already been trained or is not yet trained at all, and then transfer the global model to local devices or local servers;
2. to train local models on local datasets;
3. to learn a shared prediction model until all the training data stays on the device [5].

The cloud is for transferring the values of the local model, Figure 1. The aggregate data are used by the global models to calculate optimal performance. To incorporate the global model into the local model, the attributes of the global model are then moved to local data centers [6].

Figure 1. The results of the local models are transferred to the cloud

The term "federated" started to appear in academic articles in 2015 year. I looked through platforms such as IEEE, Google Scholar, etc. I present the results of my research. In the past 2015 year [7] and 2016 year [8], the first publications on federated averaging in telecommunication systems have appeared. Another important aspect for active research is to solve the problem of how to reduce the communication load in federated learning. Publications in 2017 year [9] and 2018 year [10] focused on developing strategies for resource allocation, especially for reducing the communication costs between nodes with algorithms and models, and on characterizing robustness to differential privacy attacks. The diagram below shows «The use of the term "federated learning" by year», Figure 2.



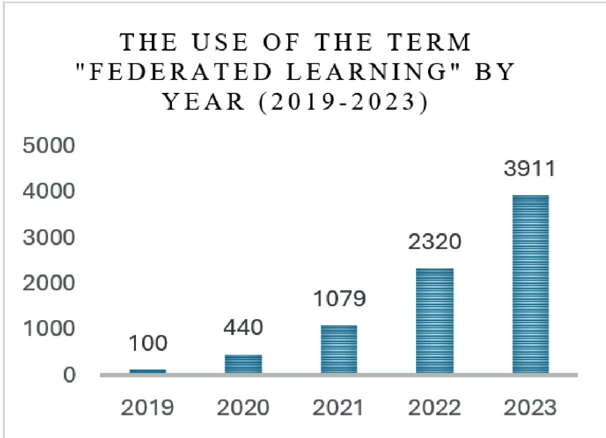


Figure 2. The use of the term "federated learning" by year

Literature review

In work [11] you can see definition of federated learning trains algorithms on decentralized edge devices, guaranteeing congeniality and data sharing among participants. It has applications in privacy-sensitive areas like medicine, banking, and manufacturing. There are three different forms of federated learning: horizontal, vertical and transfer. In federated learning have existed many ready algorithms, some of them in paper [12], including FedAvg (averaging algorithm), FedMA (matched averaging algorithm), FedProx, FedAmp (attentive message passing algorithm), MOON (model-contrastive learning algorithm) and FedLab [13].

In this paper [14] you can find summary of FL methodologies and illustrates how they can be applied in a number of IoT, including smart cities, medicine and industrialization. It also covered research opportunities of FL in IoT. The architecture of FL divided in different categories based on scale of federation, data partitioning and communication architecture. Also, during our research we used framework Flower. The description of it is in work [15]. Flower (FL) is another open-source framework for federated learning. Flower provides an abstraction for federated learning and allows developers to create and manage federated models easily. Flower supports several different algorithms for optimizing model training. In paper [16] are the applications of federated learning: Gboard, medicine, smart retail, finance, autonomous vehicles, smart grids, agriculture, industrialization. Looking through the paper [17], we see various uses of federated learning. The application of federated learning in smart devices and the Internet of Things is now beginning to take off. The researcher [18] concentrated only on examining the AI methodologies used by researchers for device protection and data security aspects, notably for Industrial IoT systems (IIOT) with decentralized design, which create enormous amounts of data and are security crucial.

The authors of [19] discuss the vulnerability of poisoning attacks considering the distributed nature of FL. Byzantine and backdoor attacks are variants of poisoning attacks. Currently, there is interest in federated learning algorithms, that are resistant to poisoning attacks. Defense strategies are anomaly detection, robust aggregation, and perturbation mechanism. In [20], the authors proposed to use fuzzy logic with a strategy to allocate different weights among clients. As a conclusion it was found that Federated Averaging algorithm shows worse results than the above-mentioned method.

Problem statement

The analysis of algorithms of federated learning has shown, that the first main algorithm is FedAvg, and now is the basis of many algorithms, but Per-FedAvg built the concept of indi-

Scientific Journal of Astana IT University
ISSN (P): 2707-9031 ISSN (E): 2707-904X
VOLUME 17, MARCH 2024

vidualized federated learning. Each newly created algorithm corrects weaknesses of previous algorithms by expanding upon robustness [21].

The base of the algorithm FedAvg-DWA is FedAvg. The idea of sampling weights is used in this technique to train the logistic regression model. You can give each sample a weight during the client-side local training phase. This method's main idea is to make the model more sensitive to the loss of minority class samples during training, which will cause the model to concentrate more on correctly predicting positive class samples when the loss function is being optimized [22].

Here, FedVar, which improves data processing on the central server by utilizing the dispersion of weights. By calculating the weight's norm tensors to calculate the mean and variance, and by calculating an advanced weight average including weights within the calculated scale of the standard deviation, a special data distribution is excluded, and a universal model is completed. As a result, the model performs better overall. By contrasting it with the current algorithms FedAvg, the suggested algorithm's performance is confirmed [23]. After comparing FedProx, FedAvg, and Scaffold with alternative local updating approaches, we find that FedProx is consistently slow-going. FedAvg always moves more slowly than FedProx and more quickly than Scaffold [24].

Materials and methods

At first, the datasets needed to be specified before the experiments could be conducted. We have assessed the methods by looking at several federated learning algorithms. We want to identify the dataset; we utilize for algorithm development when designing platforms and algorithms. Our decision is to be utilizing the CIFAR-10 which was assembled by Alex Krizhevsky [25]. Color pictures (60,000 32x32) split into 10 groups, with 6,000 photos in every group. Totally are 10,000 test photos and 50,000 training photos.

10000 photos make up each of the 5 training group and one test package that comprise the dataset. There are precisely 1000 unsystematically chosen photos from every group in the test packages. The last photos are split up randomly across the training package; however, certain training packages could include more photos from one group than another. Every training package has just 5000 photos from all the groups combined.

The next step is about data testing platforms:

1. Flower.
2. Kaggle.

Flower is a federated learning system that works with single-node or multi-node compute clusters as well as actual edge devices to enable training and evaluation in huge cohorts. This makes it possible to explore techniques in a scalable manner in real-world settings, such the constrained computing resources seen in most federated learning workloads. Flower is also fully extensible and can incorporate new algorithms, training strategies and communication protocols. With Flower, you can make experiments that use both algorithmic and system aspects of federated learning in five machine learning processes with up to 15 million clients [26].

At second, Kaggle is a platform service for in machine learning was founded in 2010. Companies and its specialists provide the dataset and themes for competitions, and users propose, for example, analysis models. A system where the companies and institutions providing the challenges purchase outstanding analytical models in exchange for prize money. In addition, the method for analyzing the data sets provided not only by the competition function but also by the kernel function will be published. A kernel is an environment that can run the data input, analysis process, output, etc. on a browser in notebook format, which is the smallest unit of an analysis project on Kaggle. A user can simply run the code in the browser without having to create an analysis environment on their own computer [27].

Results and Discussion

In this section, we have presented the definition of algorithms of federated learning that used in our experiments. But before we move on to the dependence of the algorithms of FedAvg, FedOpt, FedAdam, FedYogi, FedSparse, you can see on the Figure 3.

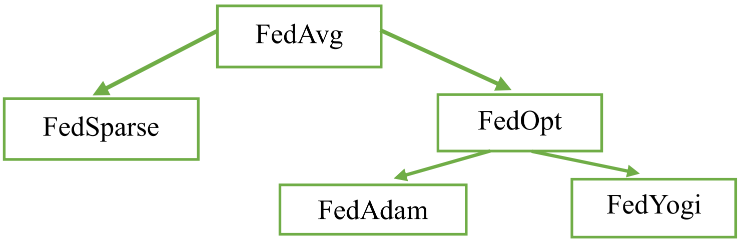


Figure 3. The algorithms of federated learning

Definition:

1. **Algorithm FedAvg**

McMahan was the first, who introduce FedAvg he took notice of minimizing communication costs. The specific workflow of the algorithm is as follows [28]:

1. The K clients are indexed by k; B is the local minibatch size, E is the number of local epochs, and μ is the learning rate
2. *Server executes*:
   $initilize\ \omega_0$
3. $\textbf{for}\ each\ round\ t = 1, 2, \dots\ \textbf{do}$
4. $m \leftarrow \max(C \cdot K, 1)$
5. $S_t \leftarrow (random\ set\ of\ m\ clients)$
6. $\textbf{for}\ each\ client\ k \epsilon S_t\ \textbf{in parallel do}$
7. $\omega_{t+1}^{k} \leftarrow ClientUpdate\ (k, \omega_t)$
8. $m_t = \sum_{k \in S_t} n_k$
9. $\omega_{t+1} = \sum_{k \in S_t} \frac{n_k}{m_t} \omega_{t+1}^{k}$
10. **ClientUpdate** (k, ω):
11. $B \leftarrow (split\ P_k\ into\ batches\ of\ size\ B)$
12. **for** each local epoch i from 1 to E **do**
13. **for** batch b ∈ B **do**
14. $\omega \leftarrow \omega - \mu \nabla l(\omega; b)$
15. return to server

An optimization problem of federated learning in algorithm FedAvg you can see in this formula [29]:

$$\min_{x \in R^d} f(x) = \frac{1}{m} \sum_{i=1}^{m} F_{i(x)},$$

where, $F_{i(x)} = E_{z \sim D_i}[f_i(x.z)]$ – loss function of i-th client, $z \in Z$

D_i – data distribution of i-th client

FedAvg is a method for training algorithms in which the data is distributed across many different servers or devices. It ensures data privacy and security and maintains the locality of the data so that models can be trained without sharing raw data.

Scientific Journal of Astana IT University
ISSN (P): 2707-9031 ISSN (E): 2707-904X
VOLUME 17, MARCH 2024

2. **Algorithm FedOpt**

Algorithm FedOpt is based on FedAvg [30].

1: Input: $x_o, ClientOpt, ServerOpt$
2: **for** $\mathrm{t} = 0, \cdots, \mathrm{T}-1$ **do**
3: Sample a subset S of clients
4: $\mathbf{x_{i,0}^t = x_t}$
5: **for** each client $\mathrm{i} \in \mathrm{S}$ **in parallel do**
6: **for k** $= 0, \cdots, \mathrm{K}-1$ **do**
7: Compute an unbiased estimate $g_{i,k}^t\ of\ \nabla F_i(x_{i,k}^t)$
8: $x_{i,k+1}^t = \mathrm{ClientOpt}(x_{i,k}^t, g_{i,k}^t, \mu_l, \mathrm{t})$
9: $\Delta_{\boldsymbol{i}}^{\boldsymbol{t}} = \boldsymbol{x_{i,K}^t - x_t}$
10: $\Delta_\mathrm{t} = \frac{1}{|\mathrm{S}|}\sum_{i \in S} \Delta_i^t$
11: $x_{t+1} = ServerOpt(x_t - \Delta_{t,\mu,t})$

ServerOpt is one of Adagrad, Yogi, or Adam's adaptive optimization techniques. FedOpt apply adaptive techniques on the server and clients; it is a cross-device algorithm with the equivalent communication expenses as FedAvg.

3. **Algorithm FedAdam**

Before proceeding to the description of the FedAdam algorithm, we should clarify that this algorithm is based on the FedOpt, which algorithm we described above.

1: Initialization: $x_0, \vartheta_{-1} \geq \tau^2$, decay parameters $\beta_1, \beta_2 \in [0,1)$
2: **for** $\mathrm{t} = 0, \cdots, \mathrm{T}-1$ **do**
3: Sample a subset S of clients
4: $\mathbf{x_{i,0}^t = x_t}$
5: **for** each client $\mathrm{i} \in \mathrm{S}$ **in parallel do**
6: **for k** $= 0, \cdots, \mathrm{K}-1$ **do**
7: Compute an unbiased estimate $g_{i,k}^t\ of\ \nabla F_i(x_{i,k}^t)$
8: $x_{i,k+1}^t = \mathrm{x_{i,K}^t - \mu_l g_{i,k}^t}$
9: $\Delta_{\boldsymbol{i}}^{\boldsymbol{t}} = \boldsymbol{x_{i,K}^t - x_t}$
10: $\Delta_\mathrm{t} = \frac{1}{|\mathrm{S}|}\sum_{i \in S} \Delta_i^t$
11: $m_t = \beta_1 m_{t-1} + (1-\beta_1)\Delta_t$
12: $\boldsymbol{\vartheta_t = \beta_2 \vartheta_{t-1} + (1-\vartheta_2)\Delta_t^2}$
13: $\boldsymbol{x_{t+1} = x_t + \mu \frac{m_t}{\sqrt{\vartheta_t} + \tau}}$

4. **Algorithm FedYogi**

1: Initialization: *x0, ϑ−1≥τ2,* decay parameters *β1, β2∈[0,1)*
2: **for** $\mathrm{t} = 0, \cdots, \mathrm{T}-1$ **do**
3: Sample a subset S of clients
4: $\mathbf{x_{i,0}^t = x_t}$
5: **for** each client $\mathrm{i} \in \mathrm{S}$ **in parallel do**
6: **for k** $= 0, \cdots, \mathrm{K}-1$ **do**

Compute an unbiased estimate $g_{i,k}^{t}$ of $\nabla F_i(x_{i,k}^{t})$
$x_{i,k+1}^{t} = \mathrm{x}_{\mathrm{i,K}}^{\mathrm{t}} - \mu_{\mathrm{l}} \mathrm{g}_{\mathrm{i,k}}^{\mathrm{t}}$
$\Delta_{\boldsymbol{i}}^{\boldsymbol{t}} = \boldsymbol{x}_{\boldsymbol{i,K}}^{\boldsymbol{t}} - \boldsymbol{x}_{\boldsymbol{t}}$
$\Delta_{\mathrm{t}} = \frac{1}{|\mathrm{S}|} \sum_{i \in S} \Delta_i^t$
$m_t = \beta_1 m_{t-1} + (1 - \beta_1) \Delta_t$
$\boldsymbol{\vartheta_t = \vartheta_{t-1} - (1 - \beta_2)}\Delta_t^2\ \mathbf{sign}(\boldsymbol{\vartheta_{t-1}} - \Delta_t^2)$
$\boldsymbol{x_{t+1} = x_t + \mu \frac{m_t}{\sqrt{\vartheta_t} + \tau}}$

5. **Algorithm FedSparse** [31]

Initialize v and w
  for $\mathrm{t} = 0, \cdots, \mathrm{T} - 1$ **do**
    $\tau \leftarrow \log(1 + \exp(\mathrm{v}))$
    $\boldsymbol{\theta \leftarrow \delta((|w| - \tau)/\mathrm{T})}$
    $\mathbf{w} \leftarrow \|[\boldsymbol{\theta} > \in]\boldsymbol{w}$ prune global model
    Initialize $\nabla_w^t = 0, \nabla_w^t = 0$
      for s in random subset of the clients **do**
        $\widehat{w}_s^t \leftarrow$ Client (s, w, v)
        $\boldsymbol{z_s} \leftarrow \|[\widehat{w}_s^t \neq 0]$
        $\nabla_w^t += z_s(\widehat{w}_s^t - w)$
        $\nabla_w^t += -(z_s(1 - \theta) - (1 - z_s)\theta$
      end for
      $\boldsymbol{w^{t+1}, v^{t+1}} \leftarrow$ **Adam** $(\boldsymbol{\nabla_w^t})$, **Adamax** $(\boldsymbol{\nabla_v^t})$
  end for

Our goal was to see the results of federated learning algorithms on the Flower and Kaggle platform to practically see the actions and results of federated learning algorithms in practice, to further determine the direction of our research work to create a new federated learning algorithm. For this purpose, we have chosen FedAdam, FedYogi, FedSparse algorithms to compare their results and see the pros and cons of these existing algorithms.

Description of our experiment. To train the models we used algorithms with the ready-made Cifar10 dataset, in the base of the Flower and Kaggle platforms.

After training, we obtain losses (distributed), losses (centralized) and values of accuracy. The values and the average of last 10 evaluations, you can see in the Table 1 and Figures 4-6.

Table 1. The average of last evaluations for the last 10 evaluations

Method	Rounds	Clients	Cifar10		
			Losses (distributed)	Losses (centralized)	Accuracy, %
FedAdam	100	10	3,57	3,86	63,06
FedYogi	100	10	3,77	3,77	59,62
FedSparse	100	10	3,78	3,9	64,04

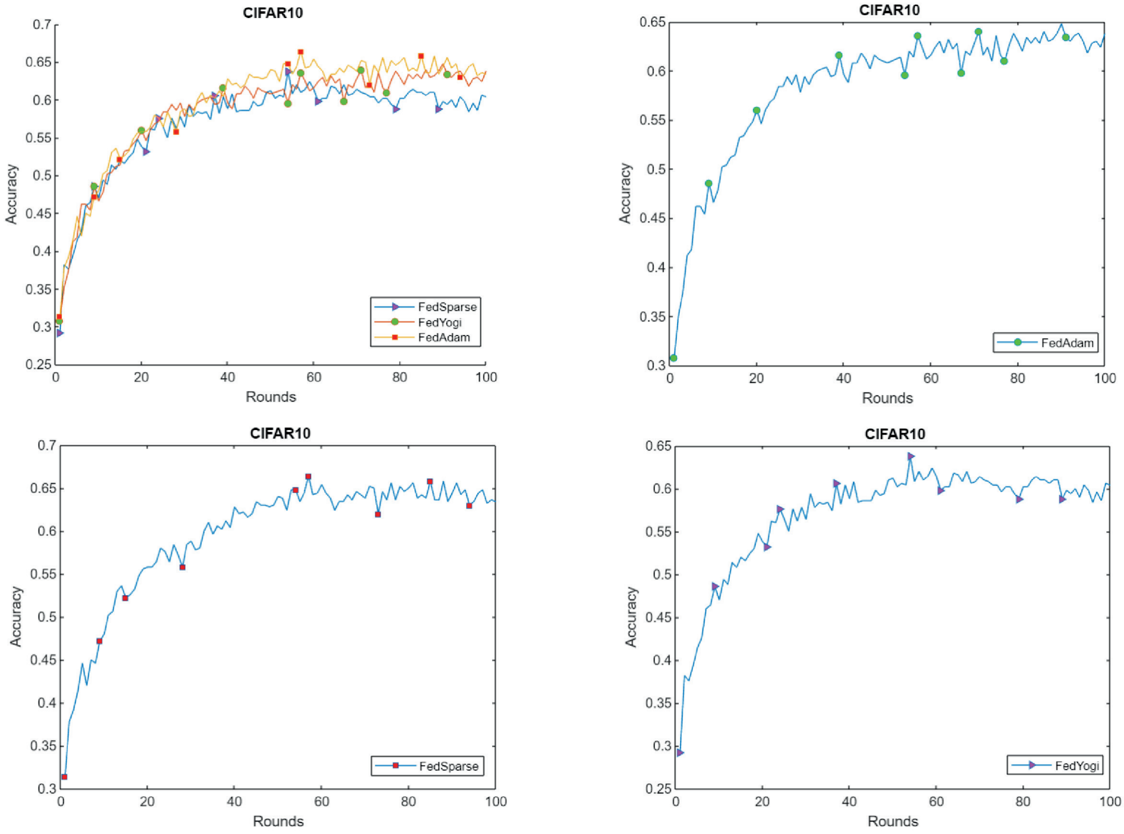


Figure 4. The values of accuracy

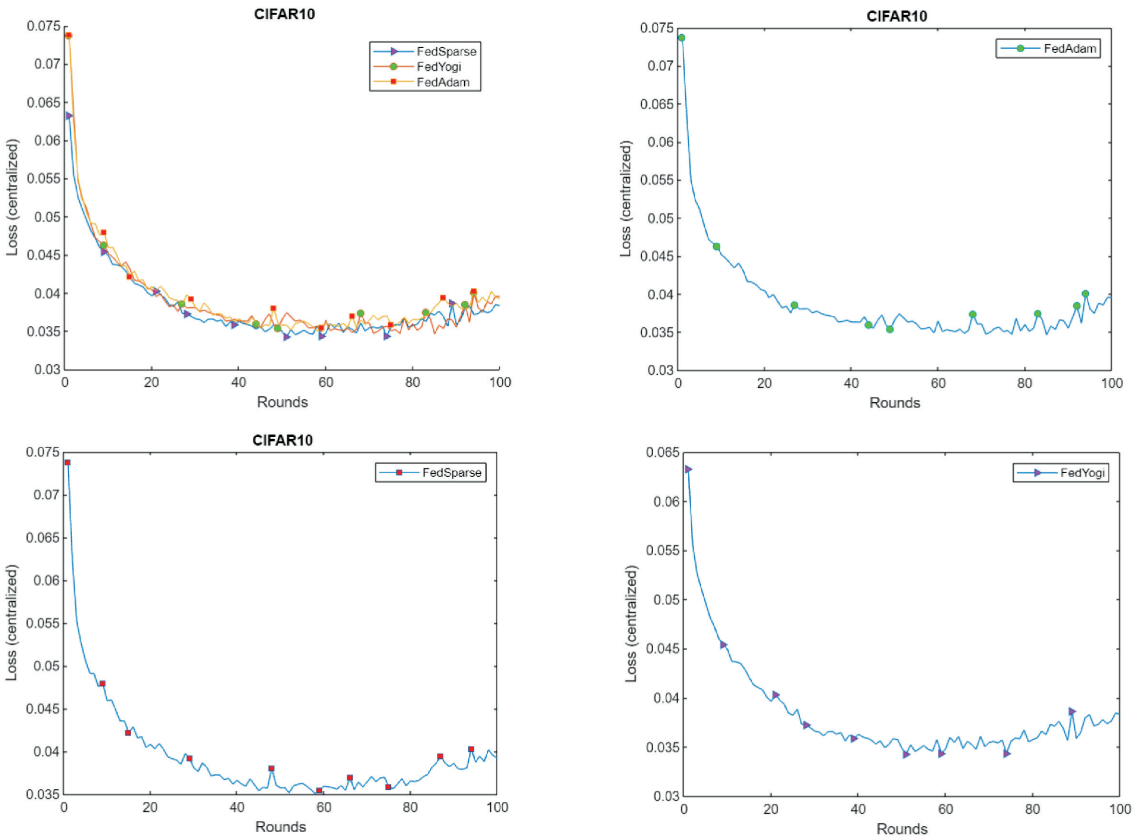


Figure 5. The values of loss (centralized)

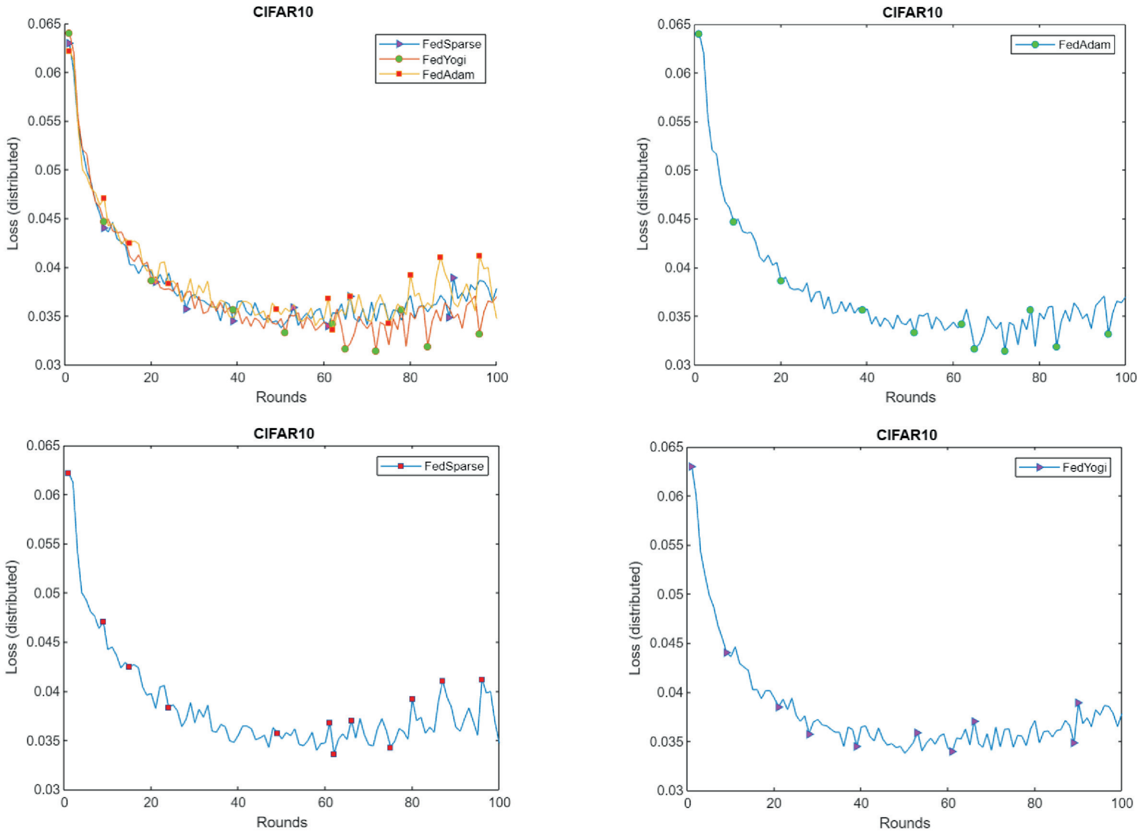


Figure 6. The values of loss (distributed)

We found out experientially, the accuracy of FedAdam - 63,06, FedYogi - 59,62, the accuracy of FedSparse – 64,04, we can conclude that the algorithm FedSparse gives more accurate values of accuracy. Also, regarding losses (distributed and centralized), FedSparse has better results than FedAdam and FedYogi.

It was a good experience to train models in Flower, because on this platform you can connect a desirable number of clients and choose number of rounds. It's a very convenient framework to work with federated learning.

Conclusion

Federated machine learning has advantages over traditional machine learning methods, including data security, data diversity, continuous real-time learning, and hardware efficiency. However, there are key challenges such as communication efficiency, managing multiple systems in one network, handling statistical data heterogeneity, as well as issues related to privacy and confidentiality.

As a result of our research, federated machine learning algorithms, namely FedAdam, FedYogi, and FedSparse, were analyzed and tested. Conducted tests allowed us to identify the advantages and disadvantages of these algorithms using data from various datasets. In our research, the FedAdam algorithm demonstrated the best performance, and we decided to use it in further studies.

Literature review has shown that most researchers in the field of federated learning use the FedAvg algorithm. However, the FedAdam algorithm, based on the FedAvg algorithm, proved to be more effective in our research. Additionally, machine learning methods and neural networks, including fuzzy neural networks, were applied in the study to solve high-dimensional tasks [32].

Scientific Journal of Astana IT University
ISSN (P): 2707-9031 ISSN (E): 2707-904X
VOLUME 17, MARCH 2024

Federated learning technology has broad practical applications. For instance, it is beneficial in smartphones, where it allows for accurate predictions without disclosing original data. In organizations, federated learning can reduce network load and provide private learning between devices. It also aids in developing models for the Internet of Things that adapt to system changes while safeguarding user privacy. Federated learning is also applied in healthcare, where restrictions prevent easy exchange of protected medical information. Furthermore, it is used for developing artificial intelligence models in compliance with regulatory requirements, utilizing vast amounts of data from multiple databases and devices.

References

[1] Gabrielli, E., Pica, G., & Tolomei, G. (2023). A Survey on Decentralized Federated Learning. *ArXiv*, abs/2308.04604.

[2] Li, T., Sahu, A., Talwalkar, A., & Smith, V. (2019). Federated Learning: Challenges, Methods, and Future Directions. *IEEE Signal Processing Magazine*, 37, 50-60.

[3] Aledhari, M., Razzak, R., Parizi, R.M., & Saeed, F. (2020). Federated Learning: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Access*, 8, 140699-140725.

[4] Dinh, T.Q., Tran, T., & Le, T. (2021). Communication cost reduction using sparse ternary compression and encoding for FedAvg. *2021 International Conference on Information and Communication Technology Convergence (ICTC)*, 351-356.

[5] Abhishek, V, A., Binny, S., JohanT, R., Raj, N., & Thomas, V. (2022). Federated Learning: Collaborative Machine Learning without Centralized Training Data. *International journal of engineering technology and management sciences*.

[6] Dinh, T.Q., Tran, T., & Le, T. (2021). Communication cost reduction using sparse ternary compression and encoding for FedAvg. *2021 International Conference on Information and Communication Technology Convergence (ICTC),* 351-356.

[7] Pang, S., Peng, Y., Ban, T., Inoue, D., & Sarrafzadeh, A. (2015). A federated network online network traffics analysis engine for cybersecurity. *2015 International Joint Conference on Neural Networks (IJCNN),* 1-8.

[8] Mosharraf, M., & Taghiyareh, F. (2017). Federated Search Engine for Open Educational Linked Data.

[9] Smith, V., Chiang, C., Sanjabi, M., & Talwalkar, A. (2017). Federated Multi-Task Learning. *Neural Information Processing Systems*.

[10] Zhou, W., Li, Y., Chen, S., & Ding, B. (2018). Real-Time Data Processing Architecture for Multi-Robots Based on Differential Federated Learning. *2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI),* 462-471.

[11] Tyagi, S., Rajput, I.S., & Pandey, R. (2023). Federated learning: Applications, Security hazards and Defense measures. 2023 International Conference on Device Intelligence, Computing and Communication Technologies, (DICCT), 477-482.

[12] Zhao, Y. (2023). Comparison of Federated Learning Algorithms for Image Classification. 2023 2nd International Conference on Data Analytics, Computing and Artificial Intelligence (ICDACAI), 613-615.

[13] Cai, L., Chen, N., Wei, Y., Chen, H., & Li, Y. (2022). Cluster-based Federated Learning Framework for Intrusion Detection. 2022 IEEE 13th International Symposium on Parallel Architectures, Algorithms and Programming (PAAP), 1-6.

[14] Panigrahi, M., Bharti, S., & Sharma, A. (2023). Federated Learning for Beginners: Types, Simulation Environments, and Open Challenges. 2023 International Conference on Computer, Electronics & Electrical Engineering & their Applications (IC2E3), 1-6.

[15] Stano, M., Hluchý, L., Bobák, M., Krammer, P., & Tran, V.D. (2023). Federated Learning Methods for Analytics of Big and Sensitive Distributed Data and Survey. 2023 IEEE 17th International Symposium on Applied Computational Intelligence and Informatics (SACI), 000705-000710.

[16] Reddy, G.P., & Kumar, Y.V. (2023). A Beginner's Guide to Federated Learning. 2023 Intelligent Methods, Systems, and Applications (IMSA), 557-562.
[17] Michálek, J., Skorpil, V., & Oujezský, V. (2022). Federated Learning on Android - Highlights from Recent Developments. 2022 14th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 27-30.
[18] Obarafor, V., Qi, M., & Zhang, L. (2023). A Review of Privacy-Preserving Federated Learning, Deep Learning, and Machine Learning IIoT and IoTs Solutions. 2023 8th International Conference on Signal and Image Processing (ICSIP), 1074-1078.
[19] Gu, Z., Shi, J., Yang, Y., & He, L. (2023). Defending against Poisoning Attacks in Federated Learning from a Spatial-temporal Perspective. 2023 42nd International Symposium on Reliable Distributed Systems (SRDS), 25-34.
[20] Aliyev, S., & Ismayilova, N. (2023). FL2: Fuzzy Logic for Device Selection in Federated Learning. 2023 IEEE 17th International Conference on Application of Information and Communication Technologies (AICT), 1-6.
[21] Reguieg, H., Hanjri, M. E., Kamili, M. E., & Kobbane, A. (2023). A Comparative Evaluation of FedAvg and Per-FedAvg Algorithms for Dirichlet Distributed Heterogeneous Data. 2023 10th International Conference on Wireless Networks and Mobile Communications (WINCOM), 1-6. https://doi.org/10.1109/WINCOM59760.2023.10322899
[22] Bian, K., & Zheng, H. (2023). FedAvg-DWA: A Novel Algorithm for Enhanced Fraud Detection in Federated Learning Environment. 2023 4th International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE), 13-17. https://doi.org/10.1109/ICBAIE59714.2023.10281317.
[23] Shin, W., & Shin, J. (2022). FedVar: Federated Learning Algorithm with Weight Variation in Clients," 2022 37th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC), 1-4. https://doi.org/10.1109/ITC-CSCC55581.2022.9894899.
[24] Karimireddy, S.P., Kale, S., Mohri, M., Reddi, S.J., Stich, S.U., & Suresh, A.T. (2019). SCAFFOLD: Stochastic Controlled Averaging for Federated Learning. International Conference on Machine Learning.
[25] Krizhevsky, A. (2009). Learning Multiple Layers of Features from Tiny Images
[26] Nilsback, M., & Zisserman, A. (2008). Automated Flower Classification over a Large Number of Classes. 2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing, 722-729.
[27] Hayashi, T., Shimizu, T., & Fukami, Y. (2021). Collaborative Problem Solving on a Data Platform Kaggle. ArXiv, abs/2107.11929.
[28] McMahan, H.B., Moore, E., Ramage, D., Hampson, S., & Arcas, B.A. (2016). Communication-Efficient Learning of Deep Networks from Decentralized Data. International Conference on Artificial Intelligence and Statistics.
[29] Nilsson, A., Smith, S., Ulm, G., Gustavsson, E., & Jirstrand, M. (2018). A Performance Evaluation of Federated Learning Algorithms. Proceedings of the Second Workshop on Distributed Infrastructures for Deep Learning.
[30] Reddi, S.J., Charles, Z.B., Zaheer, M., Garrett, Z., Rush, K., Konecný, J., Kumar, S., & McMahan, H.B. (2020). Adaptive Federated Optimization. ArXiv, abs/2003.00295.
[31] Louizos, C., Reisser, M., Soriaga, J.B., & Welling, M. (2021). An Expectation-Maximization Perspective on Federated Learning. ArXiv, abs/2111.10192
[32] Mohammadzadeh, A., Zhang, C., Alattas, K.A., El-Sousy, F.F., & Vu, M.T. (2023). Fourier-based type-2 fuzzy neural network: Simple and effective for high dimensional problems. Neurocomputing, 547, 126316.