International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

An Innovative Reliable Client-Centric Deep Learning Inference Methodology

[1]Ganesh D. Govindwar, [2]Dr. Sheetal S. Dhande

Submitted: 23/12/2023 **Revised**: 29/01/2024 **Accepted**: 07/02/2024

Abstract: Mobile phones and tablets have access to a very huge amount data that may be utilized to train learning models, potentially improving the user experience significantly. Nevertheless, the data available is often both extensive and sensitive, making it challenging to collect at centralize server and train within a centralized server using conventional methods. In this study, we investigate the utilization of blockchain technology with decentralized digital ledger to create a decentralized client-centric distributed learning system with the flexibility to support various machine learning models. This system enables the training of machine learning models directly on local machines, thereby addressing the constraints imposed by centralized servers. We demonstrate our system design, which includes two decentralized blockchain models built using Python Tensor Flow to ensure the system's reliability and efficiency. Ultimately, Block-CCL serves as an experimental environment for evaluating and distinguishing the impact of decentralized client centric i.e. federated learning from synchronization of model methods on the performance of the entire system. This highlights the validity and effectiveness of a federated learning system as a viable alternative to more centralized machine learning models.

Keywords: *Client-centric learning, Privacy, Homomorphic Encryption, Blockchain, Distributed Ledger.*

1. Introduction

In today's data market, individuals submit or generate data in various formats, encompassing activities on social media, online shopping preferences, and the maintenance of healthcare records. Companies then collect this data for either sale or in-house data analytics and machine learning purposes. Consequently, individuals essentially donate their personal or official resources to these companies. Furthermore, these companies have unrestricted access to our data, which can pose a significant invasion of privacy depending on the nature of the collected data. To address this issue of ownership and privacy, client-centric learning [1],[2] emerges as a proposed technique. In this approach, when the objective is private machine learning, data owners provide a training model to users who train on local data and transmit only the updated model weights. This process ensures that the user's data is never exposed to the server, allowing them to maintain control over their data. This approach encourages users with sensitive information, like healthcare data, to engage in the training procedure, thereby enabling the data owner to accumulate more data for training purposes. The issue of sharing our data freely, a valuable resource for organizational model training get resolved with this new approach. We suggest harnessing blockchain for user data uploading and tracking, while also offering compensation to users for the data utilized in analyses. Furthermore, the adoption of blockchain's distributed ledger technology enhances the security of updates, rendering them immutable and therefore more secure. This address concerns related to data security, confidentiality, and the safety of data uploads, attracting a wider audience and enabling organizers to gather a more extensive dataset from a larger user base.

Integrating blockchain with deep learning at client will always be a critical combination for increased security and privacy of user data, and hence much research is now being conducted in this space. To improve security, sophisticated encryption algorithms such as homomorphic encryption can be used to make the network more secure.

We introduced a novel framework called client-centric learning (Block-CCL) for constructing learning systems using blockchain technology features in this article. We also used customized homomorphic encryption (HE) to make it safer and to keep the information more private. This system demonstrates confidentiality, efficiency, and security in deep learning with blockchain. Block-CCL is based on “proof of concept” tool for spawning mini-systems that illustrate various features of our technology on a tiny scale. We used PyTorch tool to implement the Block-CCL architecture to the MNIST dataset for testing reasons. The results were fairly encouraging, and they will be applicable to more sensitive and confidential data, such as in the medical and military fields.

2. Related Work

For more than a decade, blockchain has been an established and significant technology for decentralized

[1]*Sipna COET Amravati, SGB University Amravati, Maharashtra, India ganeshgovindwar@gmail.com*
[2]*Sipna COET Amravati, SGB University Amravati, Maharashtra, India sheetaldhandedandge@gmail.com*

data management. When both of these technologies like blockchain and federated learning are readily available in the market, it will be intriguing to explore the possibilities that emerge from the synergy between these two technologies, as it can significantly bolster the security and privacy of decentralization. Numerous efforts have been dedicated to the advancement of client centric learning, incorporating a highly secure decentralized digital ledger that stores all data in a safe manner. while others leverage alternative decentralized protocols model. The primary motive is to remove the necessity for a centralized server for data collection and training the model. This is motivated by a dual purpose: first, it reinforces privacy, and second, it reduces the minimum processing power required by distributing computing across the network. An increasing body of research is focusing on the implementation of such learning using secure decentralized methodology, underscoring the evident harmony between these two technologies. Numerous articles have proposed the use of blockchain for preserving the global model among the community and forming a consensus [3], [4], [5]. While these explorations have focused on blockchain primarily as a secure and organized repository for the global model, they have not fully harnessed the capabilities of smart contract technology in effectively coordinating the learning process and calculating the evaluation functions regarding each agent's contributions to the global model.

In recent research [6], contributions to enhancing the global model have been delineated for both horizontal and vertical categories of the federated learning (FL). Horizontal FL scenario works on the 'deletion technique', involving multiple training rounds where data points from a specific client are successively omitted, with changes in testing accuracy serving as the basis for measuring each client's contributions. On the other hand, shapely values are introduced for vertical federated learning to quantify the relevance of each feature. It's worth noting that the implementation of shapely values can yield varying outcomes [7].

Furthermore, Monik Raj Behera and colleagues explored the use of smart contracts in a consortium blockchain network to create a just, clear, safeguarded, and unchanging incentive system for client-centric distributed learning. Their novel approach calculates federated contributions, providing a unique scalar measurement of each participant's role in client-centric learning. This pooled input is compatible with machine learning techniques employing gradient descents for determining weight parameters.

T. Hai et al. proposed a framework for creating a personalized recommendation system by merging blockchain with federated deep learning. This research involves two components: blockchain-based storage for electronic medical records, which employs Hyperledger Fabric to continuously monitor and record modifications in these records on the cloud server. The study then utilizes ‘Light GBM’ and ‘N-Gram’ models in group learning to prescribe customized therapies based on the patient's cloud-based database. The results demonstrate the effectiveness of this approach, as evidenced by various metrics, like F1 scores, recall and accuracy[8].

To uphold fairness within FL schemes, Y. Zhang and colleagues introduced a Pseudorandom Number Generation (BPNG) model using blockchain technology was founded which involves Verifiable Random Functions (VRFs). Furthermore, they deployed a Gradient-Random-Noise-Addition (GRNA) model, which relies on the zero knowledge proof and differential privacy to safeguard data privacy within federated learning schemes. These protocols were executed on the Hyperledger Fabric platform, and their practical feasibility was confirmed through performance evaluations conducted under experimental conditions [9].

Omar El Rifai and colleagues introduced a collaborative learning system through a coordinating website, aiming to share knowledge and make informed predictions while preserving data openness and user permissions. The approach was exemplified using a diabetic dataset and a predictive decision support tool, with in-depth discussions on its applicability in medical settings and an initial implementation to validate the methodology [10].

Z. Wang, B. Yan, and A. Dong has suggested distinguished federated learning method for data sharing in federated learning using an innovative blockchain architecture. Additionally, they presented a mechanism of incentive centered around reputation points and Shapley values to enhance the long-term viability of the federated learning system, encouraging active participation and equitable rewards. Experimental results and analyses indicated that federated learning exhibited smoother loss curves compared to centralized machine learning [11].

M. Shayan and colleagues proposed, a full decentralized peer-to-peer (P2P) solution for multiple client-server collaborative learning. Biscotti harnesses Blockchain and cryptographic principles to orchestrate a secure machine learning process among interconnected clients. The research showcased Biscotti's scalability, fault tolerance, and resilience against known attacks, even when faced with adversarial entities [12].

James E Short et al. presented an experimental distributed database and intelligent contract network architecture for tracking analytic workloads in a high-performance computing (HPC) environment. They integrated the FL/FA model into a bitcoin based architecture and tested

the interaction of platforms with servers located globally and the blockchain network. The model aimed to generate an encrypted audit trail of computer analytic activities and federate such operations across multiple supercomputer installations [13].

Yuxia Chang et al. developed a new federated learning model including blockchain technology for intelligent healthcare. This setup includes, edge nodes manage the blockchain to prevent single points of failure, and MIoT devices employ federated learning to leverage dispersed clinical data. To protect data privacy, the authors devised an adaptive differential privacy method and a gradient verification-based consensus mechanism to identify poisoning attacks. Experimental findings indicate that the proposed approach achieves high model accuracy within an acceptable timeframe while minimizing privacy budget usage and resisting poisoning attempts [14].

3. Methodologies

This section outlines our strategy for developing a system that bears a resemblance to a decentralized variant of client-centric learning. Building upon existing research, we opted to introduce modifications aimed at reinforcing privacy, enhancing access control, and bolstering security. To elevate security to an even higher level, we employed Homomorphic Encryption (HE) when working with our MNIST data, all while contemplating the principles of client-centric learning. Although the primary focus lies in data security, the system also takes into account communication efficiency. The discussion in this section will revolve around three key aspects of a distributed learning system: parallelism, the extent of centralization, and synchronization.

3.1 Parallelism

In most scenarios, the most time-intensive element of the machine learning process is training a Machine Learning (ML) model [15]. So, how can distributed learning alleviate this challenge? Distributed learning typically presents two different paradigms to parallelize the process: Data-Parallelism and Model-Parallelism [16]. For our system, we opted for Data-Parallelism. The key limitation of Model-Parallelism is its specificity to modelling and algorithms, which makes it challenging for use in a general-purpose system. Consequently, we harnessed data parallelism to train the ML model.

Data parallelism allows, every node trains an entire model only on a segment of the training dataset. All the model updates created are subsequently incorporated in the global model. This approach offers the benefit of faster parallel computation, particularly on stochastic processes like stochastic gradient descent (SGD), commonly used in distributed model learning. Additionally, it is highly adaptable to various parallelizable processes, such as gradient computation and approximation [17]. Therefore, we chose this parallelism model because it significantly simplifies the underlying model architecture, resulting in a flat array of weights from which the complete model can be reconstructed.

3.2 Centralization

A fully centralized system would essentially consist of a single device responsible for managing the entire training process from start to finish. In contrast, traditional learning involves a single server or a cluster of servers overseeing the training carried out by other nodes, representing a more distributed and less centralized approach. Given the focus of this research on distributed learning, we exclusively examine the distributed centralized version and its entirely decentralized counterpart. Consequently, we do not delve into traditional gradient descent methods, which are challenging to implement in such a distributed system. Instead, we concentrate on a stochastic variation to estimate the true gradient while considering the speed of convergence of the approximation. Our system, like other client-centric learning systems, utilizes parallel stochastic gradient descent (P-SGD) through Federated Averaging. The fundamental idea is straightforward: individual nodes train on distinct subsets of MNIST datasets, and their updates are aggregated or averaged into a single model, progressively converging towards an accurate global model.

3.3 Synchronization

The idea of a synchronization threshold, similar to other parallel and highly parallel paradigms, applies to the context of a distributed learning system. In this particular situation, each barrier represents a cycle for sharing updates, during which updates are consolidated into an updated model. It is noteworthy that more frequent synchronization barriers lead to faster model convergence and reduced performance degradation.

Bulk Synchronous Parallel (BSP) is often considered the simplest method for maintaining uniformity through a sequence of computation rounds interspersed with communication intervals. The primary advantage of this model lies in its ability to deliver the utmost level of consistency, consequently leading to the fastest convergence. In our adaptation of the BSP model, each training round is constrained by a predetermined duration, concluding within a tight time constraint beyond which no further trainer modifications are considered for that round.

4. System Architecture

In our adaptation of BSP, each training session adheres to a predefined timeframe, culminating in a strict deadline beyond which no further modifications by trainers qualify

for that specific round. Block-CCL, a distributed system that harnesses blockchain and the client-centric learning mechanism, comprises several integral components as illustrated in Figure 1. The system comprises the following core services, each with well-defined functionality and precise interactions with other layers:

Blockchain service: This serves as the principal communication and data storage network for all validator nodes. It employs a sophisticated architecture to facilitate decentralized services, ensuring consistent data across all validators.

Client-centric distributed deep dearning service: Positioned atop the blockchain layer, this service mirrors the client-centric deep learning functionality. It manages machine learning tasks like gradient enabling sharing through interaction with the local memory structure to oversee transaction execution, encompassing aspects such as model construction and aggregation. The PyTorch client-centric distributed learning paradigm was implemented for this purpose.

Training validation service: This layer combines elements from bitcoin and the instruction layer. It is responsible for assessing incoming changes and making determinations regarding the acceptance or denial of specific transactions.

Model training service: On the trainer side, this service conducts model training, including intermediate operations like model flattening and reconstruction. It handles the transmission and reception of flattened images and gradients.

Local Data Exchange service: Local data is accessible on the local machine, gathering decentralized data from each individual block. This data is locally processed using the client-centric distributed learning-trained model, and it is subsequently transmitted in an encrypted format with the blockchain model. In our proposed research, the Homomorphic Encryption (HE) technique is employed.

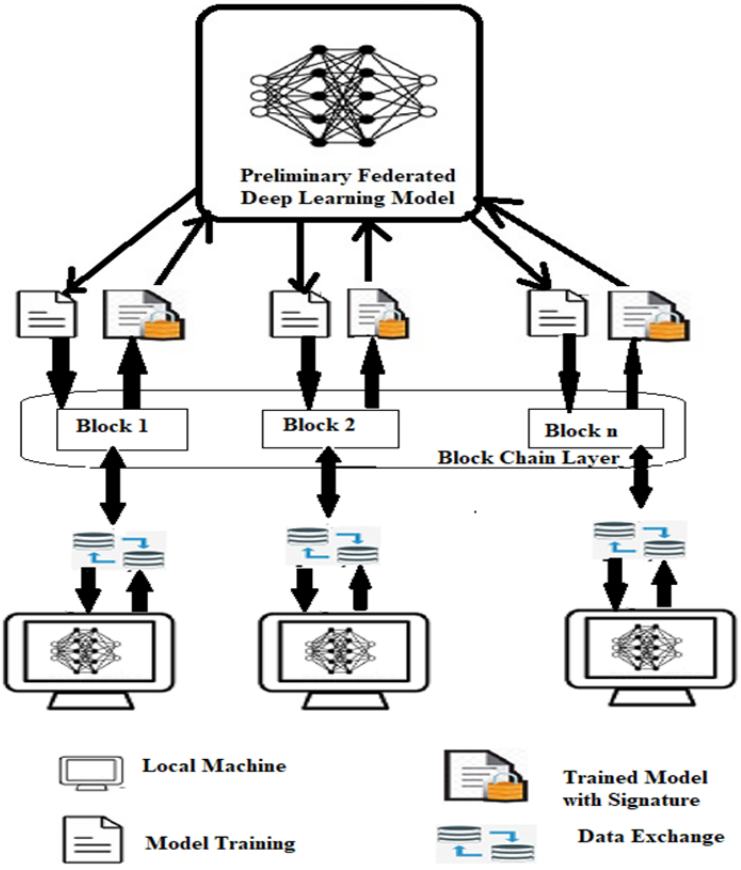


Fig 1: System Architecture

Here, we conducted experiments within the framework of Block-CCL. All of the experiments described herein revolved around the training and evaluation of a CNN i.e., convolutional neural network using the standard MNIST dataset, which consists of images sized at 28x28 pixels and is used for recognizing handwritten MNIST digits. Each system trainer employed a random 30% subsample from the training dataset of MNIST.

To gauge the effectiveness of the training, we utilized the loss metric in these experiments. The Python Deep Learning with PyTorch was the tool of choice for training our client-centric distributed learning model through

blockchain, taking full advantage of the platform's blockchain layer. Additionally, as mentioned earlier, we incorporated Homomorphic Encryption (HE) at the client-centric distributed level to enhance the encryption used during training.

In the context of the MNIST image classification dataset, we employed the Federated Average technique (FedAvg), which naturally divides the data among clients created during the client-centric distributed learning process.

Algorithm 1: Federated Averaging. The parameter k indexes the K client's nodes, B indicate the local batch size, E denotes local epochs, and n signifies the rate of learning.

Execution on the Server Side:

Initialize $\omega 0$.

for each round, denoted as t=1, 2, ...,

do the following:

Set m to be the maximum of (C * K,1)

Choose a randomly generated group of 'm' clients, denoted as 'st'.

for each client, represented as 'k', within 'st', execute the following steps in parallel.

$$\omega_{t+1}^{k} \leftarrow ClientUpdate(k, w_t)$$

end for

$$w_{t+1} \leftarrow \sum_{k \in s_t} \frac{n_k}{m_t} \omega_{t+1}^{k}$$

end for

Update on Client Side (k,w):

$$m_t \leftarrow \sum_{k \in s_t} nk$$

β is updated by splitting P_k in batches of size B.

for every round i ranging from 1 to E do

for each batch, denoted as b, from the set B:

Update ω using the formula

$\omega \leftarrow \omega - n\nabla l(w; b)$.

end for

end for

Send the updated ω back to the server.

5. Experimental Results

Here in our study, we assess the performance of the decentralized model with client-centric distributed learning and compare it to the decentralized model with client-centric distributed learning using the blockchain method. We've adopted the most effective configuration, employing two trainers as client-centric distributed learning clients. The benchmark for our system's performance is the decentralized model with client-centric distributed learning. The objective of this experiment is to determine if the added advantages of decentralization through blockchain significantly impact the quality of the model.

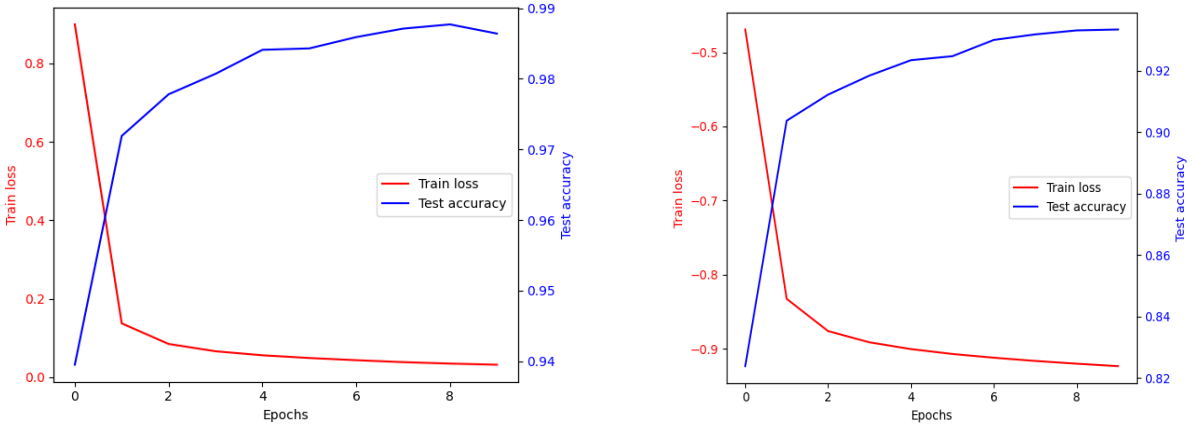
In this experiment, the decentralized model with client-centric distributed learning involves two clients, and the entire MNIST training dataset is distributed across these two clients. Both the decentralized and decentralized blockchain models undergo 30 training epochs. The architecture of the convolutional neural network (CNN) includes two 4x4 convolutional layers including first with 16 channels and the second with 32, followed by 3x3 max pooling, a fully connected layer with 256 units utilizing ReLu activation function and the softmax output layer, resulting in a total of 1,54,280 parameters.

To delve into client-centric distributed optimization, it's crucial to outline how data is partitioned among clients. In our investigation, we partition the MNIST data, which is initially shuffled and then split into two clients, each receiving 3600 samples.

6. Results and Discussion

According to Figure 2 shown below, the decentralized model with client-centric distributed learning and blockchain exhibited nearly identical accuracy and loss values compared to decentralized client-centric distributed learning without the blockchain layer. The model was trained for ten epochs, and the decentralized model training is done by two trainers slightly outperformed the decentralized model without blockchain, with a loss difference of less than 0.8. Accuracy of the Client-centric distributed Learning model without Blockchain stood at approximately 98.5%, while the accuracy with the Blockchain layer was around 94.0%. This experiment was conducted on a system equipped with a Core i5 (4.6 GHz) processor and 16 GB of RAM.

These findings indicate that the additional benefits of decentralization and secure training do not exert a significant impact on model performance. "It's essential to emphasize that this does not mean that the decentralized version outperforms its centralized counterpart in terms of performance. Instead, it underscores their equivalency in this aspect.



(a) **(b)**

Fig 2: Loss and Accuracy Curve with Federated Learning without Blockchain(a) with Blockchain(b)

The performance metrics utilized in the evaluation and analysis of our proposed Block-CCL framework include a temporal dimension. In our experimental setup, we implemented client-centric distributed learning with two clients, one with blockchain and one without. Given that the data is distributed across multiple blocks, it's reasonable to expect that client-centric distributed learning with blockchain would consume more time compared to client-centric distributed training without blockchain. When we conducted the experiment to assess this temporal complexity, we obtained the following outcomes. The data from this analysis was used to generate the graph presented in Figure 3.

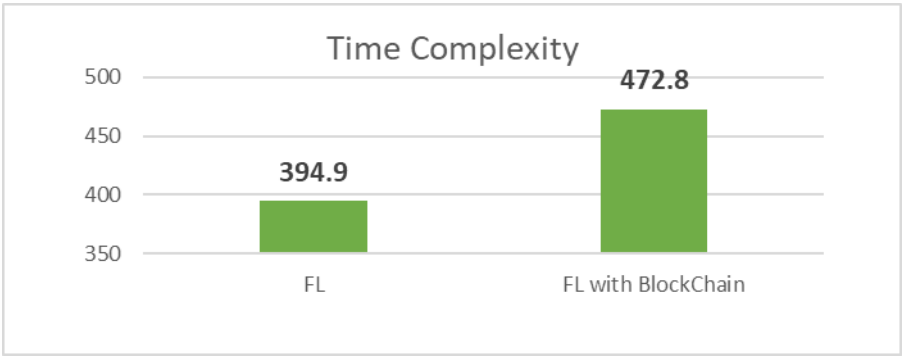


Fig 3: Time Complexity Comparison with and without Blockchain

7. Conclusion

Block-CCL is a proof-of-concept designed to demonstrate that its components may be combined to build a cohesive system with the necessary variability in the scope, decentralisation, efficiency, and privacy. Our Blockchain-based decentralised system has demonstrated its suitability as a compatible choice to pure decentralised training, marginally underperforming where the loss in increased by 0.8 and accuracy decreased by around 4%, when compared to decentralised training with only Client-centric distributed Learning. We proved that even a basic reward & penalty mechanism may have a significant better influence on model quality. However, our goal of implementing Blockchain technology with Client-centric distributed Learning as the platform with enhanced privacy and data decentralisation yielded positive results, and we will definitely want to try this model in the future for Military, Banking and Finance, and other industries to reap the full benefits of blockchain integrated with Client-centric distributed Learning.

References

[1] S. Zhou, H. Huang, W. Chen, P. Zhou, Z. Zheng, & S. Guo, "PiRATE: A Blockchain-Based Secure Framework of Distributed Machine Learning in 5G Networks," IEEE Netw., vol. 34, no. 6, pp. 84–91, Nov. 2020, doi: 10.1109/MNET.001.1900658.

[2] Y. J. Kim and C. S. Hong, "Blockchain-based Node-aware Dynamic Weighting Methods for Improving Federated Learning Performance," 2019 20th Asia-Pacific Netw. Oper. Manag. Symp. Manag. a Cyber-Physical World, APNOMS 2019, Sep. 2019, doi: 10.23919/APNOMS.2019.8893114.

[3] U. Majeed and C. S. Hong, "FLchain: Federated Learning via MEC-enabled Blockchain Network," 2019 20th Asia-Pacific Netw. Oper. Manag. Symp. Manag. a Cyber-Physical World, APNOMS 2019, Sep. 2019, doi: 10.23919/APNOMS.2019.8892848.

[4] X. Bao, C. Su, Y. Xiong, W. Huang, and Y. Hu, "FLChain: A Blockchain for Auditable Federated Learning with Trust and Incentive," Proc. - 5th Int.

Conf. Big Data Comput. Commun. BIGCOM 2019, pp. 151–159, Aug. 2019, doi: 10.1109/BIGCOM.2019.00030.

[5] Y. Li, C. Chen, N. Liu, H. Huang, Z. Zheng, and Q. Yan, "A Blockchain-Based Decentralized Federated Learning Framework with Committee Consensus," IEEE Netw., vol. 35, no. 1, pp. 234–241, Mar. 2021, doi: 10.1109/MNET.011.2000263.

[6] G. Wang, C. X. Dang, and Z. Zhou, "Measure Contribution of Participants in Federated Learning," Proc. - 2019 IEEE Int. Conf. Big Data, Big Data 2019, pp. 2597–2604, Dec. 2019, doi: 10.1109/BIGDATA47090.2019.9006179.

[7] M. Sundararajan and A. Najmi, "The many Shapley values for model explanation," 37th Int. Conf. Mach. Learn. ICML 2020, vol. PartF168147-12, pp. 9210–9220, Aug. 2019.

[8] T. Hai, J. Zhou, S. R. Srividhya, S. K. Jain, P. Young, and S. Agrawal, "BVFLEMR: an integrated federated learning and blockchain technology for cloud-based medical records recommendation system," J. Cloud Comput., vol. 11, no. 1, pp. 0–15, 2022, doi: 10.1186/s13677-022-00294-6.

[9] Y. Zhang et al., "Blockchain-Based Practical and Privacy-Preserving Federated Learning with Verifiable Fairness," Math. 2023, Vol. 11, Page 1091, vol. 11, no. 5, p. 1091, Feb. 2023, doi: 10.3390/MATH11051091.

[10] O. El Rifai, M. Biotteau, X. de Boissezon, I. Megdiche, F. Ravat, and O. Teste, "Blockchain-Based Federated Learning in Medicine," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 12299 LNAI, pp. 214–224, 2020, doi: 10.1007/978-3-030-59137-3_20/COVER.

[11] Z. Wang, B. Yan, and A. Dong, "Blockchain Empowered Federated Learning for Data Sharing Incentive Mechanism," Procedia Comput. Sci., vol. 202, pp. 348–353, Jan. 2022, doi: 10.1016/J.PROCS.2022.04.047.

[12] M. Shayan, C. Fung, C. J. M. Yoon, and I. Beschastnikh, "Biscotti: A Blockchain System for Private and Secure Federated Learning," IEEE Trans. Parallel Distrib. Syst., vol. 32, no. 7, pp. 1513–1525, Jul. 2021, doi: 10.1109/TPDS.2020.3044223.

[13] J. E. Short, K. Miyachi, C. Toouli, and S. Todd, "A field test of a federated learning/federated analytic blockchain network implementation in an HPC environment," Front. Blockchain, vol. 5, p. 893747, Aug. 2022, doi: 10.3389/FBLOC.2022.893747.

[14] Y. Chang, C. Fang, and W. Sun, "A blockchain-based federated learning method for smart healthcare," Comput. Intell. Neurosci., vol. 2021, 2021, doi: 10.1155/2021/4376418.

[15] M. Andrychowicz et al., "Learning to learn by gradient descent by gradient descent," Adv. Neural Inf. Process. Syst., pp. 3988–3996, Jun. 2016.

[16] J. Verbraeken, M. Wolting, J. Katzy, J. Kloppenburg, T. Verbelen, and J. S. Rellermeyer, "A Survey on Distributed Machine Learning," ACM Comput. Surv., vol. 53, no. 2, Dec. 2019, doi: 10.1145/3377454.

[17] L. Bottou, "Large-scale machine learning with stochastic gradient descent," Proc. COMPSTAT 2010 - 19th Int. Conf. Comput. Stat. Keynote, Invit. Contrib. Pap., pp. 177–186, 2010, doi: 10.1007/978-3-7908-2604-3_16/COVER.

[18] P. Kairouz et al., "Advances and Open Problems in Federated Learning," Found. Trends Mach. Learn., vol. 14, no. 1–2, pp. 1–210, Jun. 2021, doi: 10.1561/2200000083.

[19] Y. Liu, Y. Kang, C. Xing, T. Chen, and Q. Yang, "A Secure Federated Transfer Learning Framework," 2020, doi: 10.1109/MIS.2020.2988525.