RESEARCH ARTICLE OPEN ACCESS

Proof-of-Learning: A Federated Approach for Decentralized AI Model Training

Aditya Kandari[1], Sumanth Bhargav Reddy[2], Sai Charan Kamarthi[3], Chetan Meluseema[4], Anita Patil[5]

[1,2,3,4] UG Students, Dept. of Computer Science-Artificial Intelligence, Ballari Institute of Technology and Management, Ballari, Karnataka, India

Email : adityakandari789@gmail.com

Corresponding Author Email : nsumanthbhargavreddy@gmail.com

[5] Professor, Dept. of Computer Science-Artificial Intelligence, Ballari Institute of Technology and Management, Ballari, Karnataka, India

----------**********----------

Abstract:

Blockchain systems like Bitcoin rely on Proof-of-Work (PoW), where large amounts of computational power are used to solve complex mathematical puzzles. While this approach is effective for maintaining security, it consumes significant energy and produces no meaningful output. This paper explores an alternative approach called Proof-of-Learning (PoL), where computational effort is redirected toward training machine learning models. In this framework, multiple participants collaboratively train a shared model using federated learning, allowing data to remain private on individual devices. Instead of simply demonstrating computational effort, each participant contributes by improving the model. A validation mechanism is introduced to ensure that these contributions are genuinely useful before being incorporated into the global model. The goal is to transform traditionally wasted computational resources into productive work while preserving the security and fairness of decentralized systems. By combining blockchain with artificial intelligence, this approach aims to make distributed computing more efficient, practical, and meaningful.

Keywords — **Proof-of-Learning, Blockchain, Federated Learning, Decentralized Systems, Machine Learning, Proof-of-Work, Distributed Computing, Model Validation.**

----------**********----------

I. INTRODUCTION

Blockchain technology enables secure and decentralized systems without relying on a central authority. Most existing platforms, including Bitcoin, use the Proof-of-Work (PoW) mechanism, where nodes solve complex cryptographic puzzles to validate transactions and maintain network integrity [1].

Although PoW is effective, it requires a large amount of computational power and energy, while producing no useful output beyond maintaining the system itself. This has raised concerns about its efficiency and long-term sustainability [2].

To address this issue, recent research has focused on replacing wasteful computations with tasks that provide real-world value. One such approach is Proof-of-Learning (PoL), where computational effort is used to train machine learning models instead of solving arbitrary problems [7]. This allows the system to generate meaningful outcomes while preserving decentralization.

Federated learning further strengthens this idea by allowing multiple distributed clients to collaboratively train a shared model without exchanging raw data, thereby preserving privacy [4]. Techniques such as secure aggregation also ensure that individual model updates remain confidential during the training process [5].

However, a major challenge in these systems is verifying whether the contributions made by participants are both genuine and useful. Unlike

PoW, where computational effort is easy to verify, validating learning-based contributions is far more complex.

To address this, this paper proposes a Proof-of-Learning-based framework that combines federated learning with a validation mechanism to ensure meaningful participation and efficient decentralized AI model training.

II. LITERATURE SURVEY

A. Satoshi Nakamoto (2008) – Bitcoin: A Peer-to-Peer Electronic Cash System

Nakamoto Satoshi developed Bitcoin (2008), a peer-to-peer electronic cash system that allows conducting transactions through a network of distributed systems without the presence of intermediary financial services [1]. The idea was primarily based on eliminating intermediaries in transaction processing because their involvement usually leads to higher cost, additional time required, and possible security threats.

One of the core features of Bitcoin is that it does not require trusted third parties since it utilizes cryptography. Moreover, users may transfer money through the creation of a chain of digital signatures. The problem with such approach is related to ensuring that a transaction cannot be repeated (double-spending). Nakamoto solved this issue by creating a distributed timestamp service in which all transactions are broadcasted across the network in the correct chronological sequence, thus allowing reaching consensus.

Another important element of the presented technology is blockchain – a sequence of blocks linked via cryptographic hashes. Thus, in blockchain, each subsequent block contains a hash function of the preceding one. In turn, this implies that it will be impossible to change any information stored on the blockchain without recalculation of all subsequent data. As shown in Fig. 1, this hash-linked structure provides both integrity and transparency in the network .

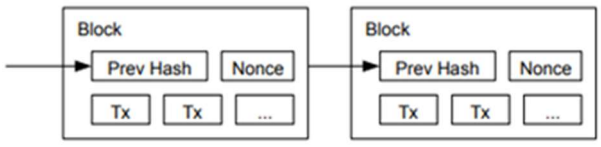

Fig. 1: Blockchain structure showing hash-linked blocks (adapted from [1])

Consensus among distributed nodes is achieved via Proof-of-Work (PoW) mechanism. This algorithm requires solving complex computational problems, and the participant who managed to find the required value (nonce) gets a reward in the form of cryptocurrencies and transaction commissions. Solving this task requires lots of resources but verifying the received result is rather simple. Thus, PoW allows maintaining safety of a system in an efficient manner but at the same time it wastes many computational resources without creating any useful results.

Additionally, Merkle trees are used in Bitcoin, which allows checking transactions efficiently and saving storage space.

However, there are still some limitations of Bitcoin. For example, the proof of Work principle requires lots of resources (electricity, time, etc.) for its implementation. In addition, the system also has scalability-related limitations.

Overall, Satoshi Nakamoto made a great contribution to development of decentralized systems and technologies. However, it has some limitations that were eliminated in further works of researchers.

B. Dwork and Naor (1992) – Pricing via Processing or Combatting Junk Mail.

One interesting way of regulating access to shared computational resources was provided by Dwork and Naor (1992). The method aimed at discouraging spamming in electronic communication systems [2]. Unlike traditional approaches, such as penalties or fines, here computational effort was imposed as a means of restricting spam without affecting regular users.

As stated above, the proposed scheme is based on introducing a pricing function. The essence of it consists in the need to perform certain computational operations before using a resource. Thus, it becomes less effective and more costly to send messages in bulk, as every one of them will have its own computational requirement.

Thus, in the case of an email server, the sender needs to first compute the value, which is produced based on the message's content, time stamp, and destination, by applying the price function. Then, the

receiver verifies this value, and if the message does not meet the criteria, the recipient rejects it.

An important achievement made by the authors is the introduction of definitions for pricing functions. For a good pricing function, the following three conditions should be met: computational hardness, anti-amortization, and fast-verifiability. Thus, such function creates an asymmetry when verifying a message is much easier than computing it beforehand.

In addition, the concept of a shortcut is introduced. It represents a way to calculate a message price for an authorized party, which is also known as a cryptographic trapdoor.

There were multiple proposals concerning how a pricing function can look in practice. Among the discussed algorithms are the computation of square root modulo a prime p where a price function looks like $f\_p(x) = \sqrt{x} \bmod p$ and is verified by a relation $y^2 \equiv x \bmod p$. Another example involves using the Fiat-Shamir construction allowing adjusting the difficulty level.

Hash functions are also discussed as a tool to reduce the computation complexity of pricing functions. Thus, instead of applying the pricing algorithm to every bit of data, one computes it using a hash value. There are multiple hashing schemes mentioned in the paper, for instance DES-based, MD4, subset sum, and Snefru.

However, this paper had some disadvantages that make the algorithm inefficient in practice. Firstly, there is no practical outcome from the computation apart from restricting access to a certain resource. Secondly, the difficulty levels should be chosen appropriately for the algorithm to resist brute-force attacks.

Overall, despite having numerous flaws, this paper has greatly contributed to the development of the field. Most importantly, this work became the basis for further research concerning distributed systems and their security features. One of the main achievements was introducing the notion of requiring proof of work to access resources, which is now widely used in blockchains [1]. This idea led to such concepts as Proof-of-Work and Proof-of-Learning.

C. McMahan et al. (2017) – Communication-Efficient Learning of Deep Networks from Decentralized Data

Federated Learning was proposed by McMahan et al. (2017) to provide a decentralization of machine learning. Specifically, it allows multiple clients to cooperate and develop a global model using the local data of the clients [4]. The main motivation of this work is to mitigate problems related to centralized machine learning, such as privacy and communication issues when moving user data to a centralized place. Therefore, the key idea here is to perform learning locally at the clients' side.

This framework implies that local clients store their data locally. Instead, the model parameters are trained locally and sent to the central server that aggregates all local parameters to build a better global model. Thus, this approach allows for protecting private user data, mitigating privacy issues.

The learning algorithm here is formulated in terms of a distributed optimization problem where the global objective is considered to be a linear combination of the local objectives, reflecting the amount of data stored by each client.

One of the most important problems mentioned by the authors is that local datasets can have different distributions (non-IID and imbalance). As a result, the distributed learning algorithm cannot work properly since it assumes a homogeneous distribution of datasets.

To solve this problem, the authors suggest an algorithm called FederatedAveraging (FedAvg). The main idea of this algorithm is to combine local stochastic gradient descent with aggregation steps. Specifically, selected clients perform local training in several rounds and then upload their models to the central server to update the global model. Such an approach is efficient in terms of performance and communication cost.

First, the aggregation algorithm prioritizes contributions from clients having larger local datasets. Moreover, an illustration of the whole process is provided in terms of algorithms and diagrams of client-side and server-side processing as shows in Fig. 2.

Algorithm 1 `FederatedAveraging`. The K clients are indexed by k; B is the local minibatch size, E is the number of local epochs, and η is the learning rate.

Server executes:
initialize w_0
for each round $t = 1, 2, \dots$ **do**
$m \leftarrow \max(C \cdot K, 1)$
$S_t \leftarrow$ (random set of m clients)
for each client $k \in S_t$ **in parallel do**
$w_{t+1}^k \leftarrow \text{ClientUpdate}(k, w_t)$
$w_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^k$

ClientUpdate(k, w): *// Run on client k*
$\mathcal{B} \leftarrow$ (split $\mathcal{P}_k$ into batches of size B)
for each local epoch i from 1 to E **do**
for batch $b \in \mathcal{B}$ **do**
$w \leftarrow w - \eta \nabla \ell(w; b)$
return w to server

Fig. 2: FederatedAveraging algorithm (adapted from [4])

Additionally, this framework uses gradient updates, where the contribution of local clients comes in the form of gradients used to improve the global model.

It should be noted that the framework faces several problems related to the limited bandwidth, available hardware, and participation rate. Namely, communication is the key bottleneck, but the authors solve this problem by performing several local iterations in advance, thus, minimizing the number of communication steps.

Finally, experiments show that the increase in the number of local computations helps to decrease the communication costs. Specifically, the FedAvg algorithm reduces the number of communication steps to achieve a similar level of accuracy by 10x to 100x compared to other distributed algorithms.

As for the privacy issues, federated learning can help with them since data is not uploaded to a centralized place. Still, some additional approaches, such as differential privacy and secure aggregation, should be applied for more robust solutions.

Although federated learning has many advantages, this framework also has certain limitations. First, it requires a centralized server. Second, model updates can cause privacy problems since some information can leak. Moreover, federated learning can underperform because of non-IID data and low participation rates.

All in all, federated learning is a great invention that allows for solving problems associated with centralized machine learning. Thus, this concept is essential for the development of decentralized systems and new approaches, such as Proof-of-Learning.

D. Bonawitz et al. (2017) – Practical Secure Aggregation for Privacy-Preserving Machine Learning.

Bonawitz et al. proposed a method for secure aggregation [5], which allows for calculating the aggregate of the user data without access to individual entries in it. Federated learning does not require sharing raw user data between parties. However, model updates obtained during the training process can contain sensitive information. In contrast to traditional machine learning, the federated learning framework relies on the client-side to train the models locally and upload the corresponding model updates.

The problem statement is associated with secure computations involving high-dimensional vectors. Every user computes his or her local contribution to the aggregate based on their private data. Only the final value is needed to be transmitted to the central server.

The solution is based on the principles of secure multiparty computation. Each user encrypts their vector using pairwise masks, which are secret values shared between pairs of participants. Upon aggregation, these masks are canceled out, allowing the server to receive only the desired aggregate. The core idea is to use pairwise masking, where two users share a random number. While one party adds this number, another subtracts it from the data, preserving correctness and hiding individual contributions.

To reduce communication costs, the proposed method avoids sharing random vectors directly. Instead, it implements the Diffie–Hellman key exchange and uses pseudorandom generators to expand seeds into vectors. Additionally, the approach considers the case where users might drop out of the network. To mitigate this problem, the researchers implement Shamir's Secret Sharing.

An advanced version of the scheme involves double masking. Using this modification, each user randomly generates a second mask, which is used to encode the vector. Such a technique helps to prevent any attacks related to inference.

The overall algorithm includes several steps, namely key exchange, masking, verification, and

unmasking. The protocol can resist both honest-but-curious and active adversaries. Moreover, its security level is guaranteed.

In terms of performance, the protocol is computationally efficient. Communication costs are 1.7× to 2× larger compared to unencrypted transmissions.

One disadvantage of the approach is that it involves complex cryptographic procedures and multiple communication rounds. Furthermore, it requires coordination between participants and reliable data transfer.

In summary, this research significantly enhances privacy in federated learning systems and extends federated learning schemes [4]. The protocol is vital for ensuring secure and privacy-preserving distributed learning, especially in the Proof-of-Learning paradigm.

E. Bravo-Marquez et al. (2019) - Proof-of-Learning: A Blockchain Consensus Mechanism Based on Machine Learning Competitions.

Bravo-Marquez et al. (2019) introduced the notion of Proof-of-Learning (PoL), which serves as an alternative to traditional Proof-of-Work (PoW) [7]. PoL consists in training machine learning models rather than performing unnecessary calculations.

A major disadvantage of PoW consensus mechanism consists in its inefficiency due to the wasteful nature of the computations performed by nodes in the network. PoL aims to eliminate this problem by transforming the computing power into something valuable for real-world applications.

The core idea of PoL is to consider consensus as the competition between machine learning algorithms. Nodes are supposed to train their machine learning models to generate blocks based on the results achieved in the process of training.

Nodes need to prove that they can generate good-quality models, and the best one gets the opportunity to add a new block. In other words, the network turns mining activities into productive actions.

It means that a participant who is able to generate a model that outperforms other models will get rewards. To win, nodes need to provide models as well as metrics of their performance, and then the network evaluates these metrics.

As it can be seen, verification of models' performance is another crucial issue for the network. The evaluation procedure involves using test data sets to avoid cheating and achieve fairness and consistency among participants.

The second significant contribution of the work includes the idea of using a repository of models and experiments. As a result, the evolution of different models will be traced, creating valuable information that can be used in further research.

As for the benefits of the suggested algorithm, it implies that computational efforts are spent efficiently. Indeed, instead of solving meaningless mathematical puzzles, participants train useful machine learning models.

The only problem in the implementation of PoL is selecting appropriate data sets for evaluation and making sure that the metric selected is appropriate. Besides, verifying machine learning models is a time-consuming and complex task.

As can be noted, PoL continues the research on blockchain consensus mechanisms that started with PoW [1]. It is a natural development of the idea according to which mining can be applied to solve practical issues.

The suggested algorithm can form the basis of a powerful decentralized architecture combining the capabilities of blockchain and machine learning technologies. Integrating PoL into federated learning systems [4] and implementing secure aggregation [5] will help to develop intelligent distributed systems.

TABLE I
COMPARISON OF LITERATURE SURVEY PAPERS

S.No.	Paper	Technique Used	Objective	Key Contribution	Advantages	Limitations
1	Nakamoto (2008) [1]	Blockchain with Proof-of-Work	Enable decentralized digital transactions without central authority	Introduced blockchain and PoW consensus mechanism	Secure, decentralized, tamper-resistant	High energy consumption, computational waste, scalability issues

2	Dwork & Naor (1992) [2]	Pricing via Processing (PoW concept)	Prevent spam and misuse of network resources	Introduced computational effort as access control	Simple and effective for limiting abuse	No useful output from computation, inefficient
3	McMahan et al. (2017) [4]	Federated Learning	Train models without sharing raw data	Introduced decentralized ML training with local updates	Privacy preservation, reduced data transfer	Communication overhead, non-IID data challenges
4	Bonawitz et al. (2017) [5]	Secure Aggregation	Protect privacy of model updates	Developed cryptographic protocol for secure model aggregation	Strong privacy guarantees, scalable	Additional computation and communication overhead
5	Bravo-Marquez et al. (2019) [7]	Proof-of-Learning	Replace PoW with useful ML tasks	Introduced ML-based consensus using model performance	Efficient use of computation, meaningful output	Complex verification, dataset dependency

III. RESEARCH GAP

However, despite the advances made in blockchain technologies and distributed learning, some essential limitations still persist in existing models. First of all, classical blockchain solutions like Bitcoin work with the Proof-of-Work (PoW) mechanism, which requires massive computation resources and energy consumption in order to provide only one function – network security [1]. The same could be said about early solutions where computation effort was used as a means of limiting access to resources while being wasted because nothing productive was done with it [2].

The latest improvements made in the field of federated learning have solved some privacy problems by making the process of distributed learning independent of data transfer [4]. Still, there are issues such as inefficient communication, distribution of non-IID unbalanced data among clients, as well as lack of tools for verifying the relevance and credibility of model updates. Though secure aggregation helps to protect updates from disclosure, it does not evaluate whether the contribution to the global model is positive.

On the other hand, recent approaches like Proof-of-Learning try to solve the issue of efficiency by making computations useful since they involve machine learning tasks. Although this solution is more productive in terms of the usage of computational capabilities, it raises another problem as to the fair evaluation of models and verifying their contributions to the global pool.

In conclusion, after examining the existing body of literature, it is possible to state that existing solutions solve these issues separately, and there is no single approach that would guarantee all three criteria: effective use of computing resources, privacy, and reliability of model contributions.

Thus, it can be stated that the research problem is to find a way to integrate these concepts in a single framework.

IV. PROPOSED FRAMEWORK

To overcome the described research gap, this paper presents a theoretical framework of integrating

blockchain-based consensus algorithm with federated learning and a validation mechanism.

According to the proposed scheme, different parties collaborate in training a single machine learning model in which each participant trains its local model based on their private datasets, and then all model updates are aggregated through the use of federated learning while the blockchain serves as a tool for decentralizing, securing, and making the process more transparent.

In addition, a validation mechanism is implemented to make sure that only updates which lead to improvement of the model's performance are incorporated in the model. Hence, only relevant updates are included in the system, which prevents from wasting any effort for meaningless computations.

Moreover, secure aggregation technique should be used to ensure confidentiality of the contribution in case it is shared between the nodes.

Thus, combining federated learning with blockchain and introducing a validation mechanism, the described approach combines the strengths of decentralization, privacy protection, and relevant contributions.

V. APPLICATIONS

There are many areas where the Proof-of-Learning framework could find practical application. For example, in healthcare, it will be possible to train a joint model in collaboration with multiple institutions without the need to share patient data; in the financial sector – to use machine learning models to detect fraudulent activities while preserving confidentiality; in smart cities – to train a model using data collected by distributed sensors; and finally, in IoT systems – to train a global model using edge devices' contributions while still protecting their data.

VI. FUTURE WORK

Some promising areas for future research are developing solutions for scaling Proof-of-Learning systems, enhancing validation mechanisms, implementing additional privacy protection measures such as differential privacy and homomorphic encryption, minimizing the communication overhead in federated learning systems, and training non-IID data.

VII. CONCLUSIONS

The current paper has provided a thorough overview of existing solutions applied in the context of blockchain technology and decentralized machine learning. Specifically, traditional blockchain protocols like Bitcoin use Proof-of-Work consensus algorithm for conducting decentralized transactions securely; however, they suffer from high computational power costs and inefficiency of usage [1]. In turn, approaches based on pricing via processing have demonstrated the idea of using computational effort as an effective control mechanism. Nevertheless, they do not provide any meaningful results of their operation [2].

In light of rapid advances in machine learning technologies, Federated Learning protocol ensures the decentralized nature of the training process while maintaining users' privacy by storing sensitive data on local devices [4]. Furthermore, secure aggregation approach has become an efficient technique of increasing security by ensuring confidentiality of model updates [5]. Although these innovations have helped to develop advanced systems, there is still no reliable way to verify the contribution and usefulness of nodes that participate in the process.

Finally, Proof-of-Learning consensus protocol that uses computational tasks of machine learning for ensuring decentralization shows a great potential as an alternative to traditional consensus algorithms in terms of computational power. However, there are also some difficulties connected with the model verification, fair evaluation of participants, and additional complexities introduced by the approach.

According to the analysis of current research findings, there is currently no approach that is able to ensure efficient computation and verification, as well as privacy of communication simultaneously. Therefore, the task of integrating the mentioned

concepts should be considered the direction of future development of decentralized intelligence.

To sum up, current technologies allow creating efficient decentralized machine learning models; however, their integration requires further efforts.

REFERENCES

[1] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008.
[2] C. Dwork and M. Naor, "Pricing via Processing or Combatting Junk Mail," in Advances in Cryptology — CRYPTO '92, 1992, pp. 139–147.
[3] S. Haber and W. S. Stornetta, "How to Time-Stamp a Digital Document," Journal of Cryptology, vol. 3, no. 2, pp. 99–111, 1991.
[4] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in Proc. 20th Int. Conf. Artificial Intelligence and Statistics (AISTATS), 2017, pp. 1273–1282.
[5] K. Bonawitz et al., "Practical Secure Aggregation for Privacy-Preserving Machine Learning," in Proc. ACM SIGSAC Conf. Computer and Communications Security (CCS), 2017, pp. 1175–1191.
[6] A. Back, "Hashcash – A Denial of Service Counter-Measure," 2002.
[7] F. Bravo-Marquez, S. Reeves, and M. Ugarte, "Proof-of-Learning: A Blockchain Consensus Mechanism Based on Machine Learning Competitions," in Proc. IEEE Int. Conf. Decentralized Applications and Infrastructures (DAPPCON), 2019.

