Federated Learning Meets Blockchain: a Power Consumption Case Study

Published Version:
Romandini, N., Mazzocca, C., Montanari, R. (2023). Federated Learning Meets Blockchain: a Power Consumption Case Study [10.1109/pdp59025.2023.00040].

Availability:
This version is available at: https://hdl.handle.net/11585/962281 since: 2024-02-26

Published:
DOI: http://doi.org/10.1109/pdp59025.2023.00040

Federated Learning Meets Blockchain: a Power Consumption Case Study

1st Nicolò Romandini
Department of Computer Science and Engineering
University of Bologna
Bologna, Italy
nicolo.romandini@unibo.it

2nd Carlo Mazzocca
Department of Computer Science and Engineering
University of Bologna
Bologna, Italy
carlo.mazzocca@unibo.it

3rd Rebecca Montanari
Department of Computer Science and Engineering
University of Bologna
Bologna, Italy
rebecca.montanari@unibo.it

***Abstract*—Federated learning (FL) is emerging as the most promising approach to collaboratively train a machine learning (ML) model on a common task without centralizing data. During each FL round, participants locally train a partial model with its on-premises data. Such models are subsequently aggregated to derive a global one. How these partial models are combined is a primary concern. Traditional approaches usually rely on a parameter server that introduces many weaknesses such as single point of failure, lack of trustworthiness among unknown participants, and incapacity to handle the traffic generated from millions of devices.**

Thus, to overcome such concerns, blockchain has recently been proposed as a valuable solution to improve the robustness of FL approaches. The full-blown benefits of using blockchain enable tackling the limits of centralized servers. However, energy consumption is still one of the significant factors inhibiting its widespread due to the current discussions on climate change and sustainability. Recently, a growing number of research works have been focusing on integrating FL and blockchain, nevertheless, adequate analysis and estimate of their energy and power consumption are often lacking.

This paper presents an estimate of the power consumption of FlowChain, an architecture that integrates FL with blockchain to simplify the use of FL. Experimental results demonstrate that the overall power consumption significantly depends on the ML model adopted.

***Index Terms*—Blockchain, Federated Learning, Power Consumption, Energy Consumption, Sustainability**

I. Introduction

Nowadays, an unprecedented amount of data is generated from multiple locations leading the way to novel machine learning (ML) solutions. Despite the increasing data availability, how to effectively use them is a challenging task that is further hindered by modern privacy laws and regulations. In this direction, federated learning (FL) [1] is envisioned as the most promising solution [2] to leverage distributed data while guarantying privacy. These capabilities enable overcoming the limits of traditional ML approaches that demand data centralization. FL enables involved parties to collaboratively train a global model on a common task without outsourcing on-premises data. An FL process consists of multiple communications rounds: each participant, typically referred to as a client, trains a local model through its private data and produces an update. All clients' contributions are then aggregated into a global model that serves as the starting point for the next round. How the aggregation is performed is a critical task that is, in its original version, entrusted to a parameter server.

However, using a central server for model aggregation poses some challenges that must be carefully tackled. Clients may not fully trust the server which also introduces single-point-of-failure. A unique server may have some biases to prefer a partial model over others. In addition, it cannot manage the traffic generated from millions of distributed devices. To address such concerns, blockchain has been proposed as an attractive solution to develop a more robust and decentralized FL approach [3]. For example, through the consensus protocol, the blockchain can guarantee that the final global model will be created without any kind of bias. This trend is also corroborated by many recent works [4]–[7] that have enriched FL with blockchain for different purposes such as accountability, data provenance, or improving trustworthiness among unknown participants. Despite the full-blown benefits of using blockchain, its widespread adoption is often inhibited by energy consumption which is one of the significant concerns also due to the current discussions on climate change and sustainability [8]. In blockchain environments, determining the exact amount of energy consumption is a hard task that depends on several factors such as the number of participants and the consensus mechanism adopted. For example, proof of work (PoW) consensus requires a large extent of electrical energy, as the entities involved in the validation process (i.e., miners) demand a huge amount of computational resources to validate blocks [9]. Miners compete to confirm a block and who wins the race advances the blockchain status. All the other block candidates are discarded, resulting in a huge waste of electricity put into their calculation. The difficulty in accurately estimating the general energy consumption of blockchain technology is further confirmed by the limited results available in the literature [10]. Some studies [11], [12] have provided general estimates for the lower and upper bounds of the energy consumed, but a more comprehensive analysis is needed.

Energy is one of the major concerns also for FL environments [13] because participants may have heterogeneous capabilities. In FL, clients do not offload heavy computations to cloud-based resources, thus, their computing resources are directly involved in the training phase. For this reason, Multi-Access Edge Computing (MEC) servers are often leveraged to allow users to offload a portion of their dataset for model training [14]. Despite the increasing interest in integrating blockchain and FL while keeping energy consumption under control, energy and power analysis are often neglected and hardly ever presented in novel proposals. At the state-of-the-art, there are no works that offer an estimate of the energy and power consumption of a blockchain employed for FL purposes.

To fill this gap, this paper aims to estimate the power consumption of integrating FL and blockchain. In particular, we conduct a power consumption analysis of FlowChain [15], a framework that integrates FL with blockchain to simplify the use of FL. Our evaluation mainly focuses on the power consumption of the blockchain for those operations needed to enable FL (e.g., aggregating partial models). Experimental results demonstrate that the overall power consumption heavily depends on the considered ML model.

The remainder of the paper is structured as follows. Section II provides the background to understand the framework analyzed. Then, Section III presents the power consumption model adopted. In Section IV, we discuss the experiments conducted. Finally, Section V draws our conclusions and introduces future work.

II. Background

This section provides the needed background to understand the case study on which we conduct our analysis.

A. Federated learning

FL is a collaborative ML technique that enables training a global ML without outsourcing local data, contrary to traditional ML which needs to centrally information. This feature is one of the key strengths since it avoids all the concerns related to data sharing and privacy regulations. Each participant that joins FL locally trains a partial model using its on-premises data. These models are then aggregated through a predefined aggregation strategy. FL strategies comprise the algorithm to aggregate partial models and the type of synchronization adopted (i.e., synchronous or asynchronous).

The primary algorithms used to aggregate partial models are federated SGD (FedSGD) and federated averaging (FedAVG). The former applies stochastic gradient descent to optimize federated problems. During each round, on-premises data are used to take one step of gradient descent on the current model, then the server performs a weighted average of the resulting models. The latter is a slightly different version, proposed by the same authors. Each client sends weights instead of gradients. Partial models are typically aggregated synchronously, only after all participants have provided their contributions. However, asynchronous approaches are also becoming more prevalent due to the highly distributed nature of FL.

B. Blockchain

A blockchain is a chain of immutable blocks linked through hashes. Blocks comprise tamper-proof information that in FL is represented by the partial models submitted by clients. The integrity of blocks is achieved by leveraging cryptographic techniques (i.e., hash and digital signature). Any modification to data produces a completely different hash, making it impossible to tamper stored information. Blocks are validated thanks to a consensus protocol that has a significant impact in terms of energy and power consumption. For example, PoW requires that entities responsible for validating blocks perform heavy-burden computations to solve a complex mathematical puzzle. Another consensus protocol, which is an energy-efficient alternative, is proof of stake (PoS). In this method, the chance of an entity to validate a block depends on its amount of wealth in that network. Hence, such an approach saves energy and is more sustainable than PoW [16]. Such mechanisms protect the network against malicious adversaries. Another relevant property of blockchain refers to the access models, indeed, a blockchain can be permissionless and permissioned. The former does not foresee any restrictions, while the latter participants may not be allowed to validate or both access and validate.

Recently, an increasing number of blockchains have started integrating smart contracts, which are applications directly executed on the blockchain. Therefore, consensus protocols, as for information embedded in the blocks, assure the authenticity of the smart contract execution. This is remarkably relevant in FL scenarios since it provides trustworthiness on how the global model was produced.

C. FlowChain

FlowChain is a decentralized and distributed framework for executing FL training. Its architecture is depicted in Fig. 1. FlowChain exploits blockchain to store partial models and aggregate them in a fully automated way by leveraging smart contracts. Due to the immutability property of the blockchain, partial models can be retrieved not only from the smart contract state but also from the blockchain by analyzing the sequence of all executed transactions. Using a smart contract increases the level of trust among unknown participants of the FL training because it avoids potential biases to prefer a model over others.

It is designed to be compatible with Flower [17], a platform devised to favor the use of FL. For this reason, each client comprises a Flower Client that trains partial models and a connector to interact with the blockchain. Upon completion

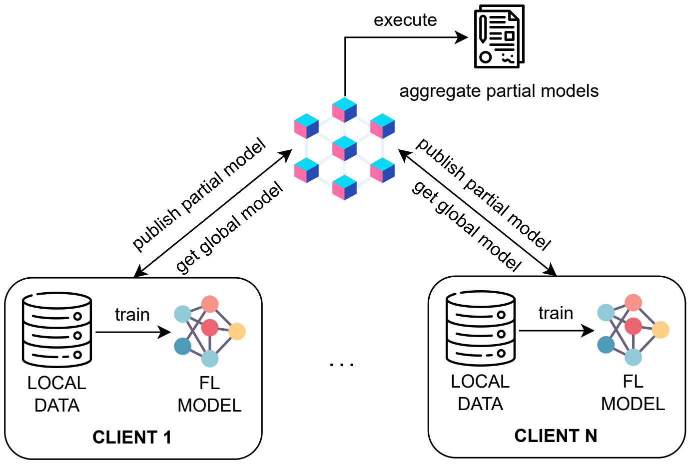

Fig. 1. FlowChain architecture.

of the partial model training, the connector pushes the model by executing a transaction to a smart contract deployed on the blockchain. Hence, the blockchain replaces all the functionalities that are usually performed by a central server (i.e., storing partial models and aggregating them). Concerning the FlowChain blockchain, it is implemented through Hyperledger Fabric[1], an open-source framework for deploying permissioned blockchains. Fabric employs the Raft consensus algorithm, which is implemented through an orderer node. The orderer node is responsible for receiving transactions, ordering, and creating blocks, ensuring that every validated transaction is final and correct. This mechanism enables fast and reliable consensus while reducing energy and power consumption compared to traditional consensus methods like PoW and PoS.

The FlowChain smart contract can implement different algorithms and aggregation strategies, but at present, it only provides the FedAVG algorithm and a synchronous aggregation strategy, where the smart contract waits for all clients before aggregating the models. In addition, FlowChain also supports identity management through decentralized identifiers (DIDs). Thus, each participant must be registered on the blockchain making it possible to control who or what can participate in the training process and track who has published a particular partial model.

III. Power Consumption Model

To effectively enable the use of blockchain in FL environments, the power consumption of a blockchain node while performing the operations connected to an FL process needs an adequate investigation. It is clear that all the parties involved in FL consume energy and saving energy is fundamental to reducing costs and avoiding environmental damages associated with carbon emission [18].

We model the power consumption of the overall FL process P_f as a function of the different modules involved in the training phase. Let us consider the power consumption of an FL training during the i-th round. The client that sends partial models consumes P_c^i. The power consumption of a blockchain node that collects partial model P_b^i is a combination of power needed to run the node P_n^i and that P_s^i to execute the smart contract to aggregate partial models and generate the $i+1$-th global model. We denote with C the number of FL clients and with B the number of blockchain nodes. Therefore, the power consumption of the i-th FL round can be defined as follows:

$$P_f^i = C * P_c^i + B * P_b^i \tag{1}$$

where P_b^i is defined as:

$$P_b^i = P_n^i + P_s^i \tag{2}$$

Therefore, given N rounds of FL, the overall power consumption of an FL process is given by:

$$P_f = \sum_{i=1}^{N} P_f^i = C * \sum_{i=1}^{N} P_c^i + B * \sum_{i=1}^{N} P_b^i \tag{3}$$

The formulas above clearly show that the total power consumption of an FL training depends on C and B. However, determining the optimal number of FL clients and blockchain nodes that constitute the network goes beyond the scope of this paper.

IV. Evaluation

In this section, we evaluate the feasibility of employing blockchain in an FL environment. We deploy the blockchain component of FlowChain on 2 nodes each equipped with an Intel(R) Core(TM) i5-3470 CPU running at 3.20GHz and 12 GB of RAM.

The main focus of our analysis is to evaluate the power consumption of the blockchain while performing FL operations. Therefore, we did not consider the amount of power consumed by clients while training partial models. We also measure the consumption of the connector that allows Flower Clients to interact with the blockchain. To do this, we use the well-known PowerTop[2] tool available for Linux. Concerning the blockchain, since in FlowChain each component is run inside a Docker container, we obtain power metrics through docker-activity[3], a tool developed to monitor the statics of container and their power consumption.

A. Experiments

To estimate the power consumption comprehensively, we conducted two experiments varying the complexity of the employed ML models and datasets. To obtain a more accurate estimate, each experiment was repeated 10 times and the results are aggregated. For the first case, we used the Fashion-MNIST dataset, a collection of Zalando item images consisting of a training set of 60.000 examples and a test set of 10.000 examples. Each example is a 28x28 grayscale image associated with a label of 10 classes. The used neural network has only

[1]https://www.hyperledger.org/use/fabric

[2]https://github.com/fenrus75/powertop

[3]https://github.com/jdrouet/docker-activity

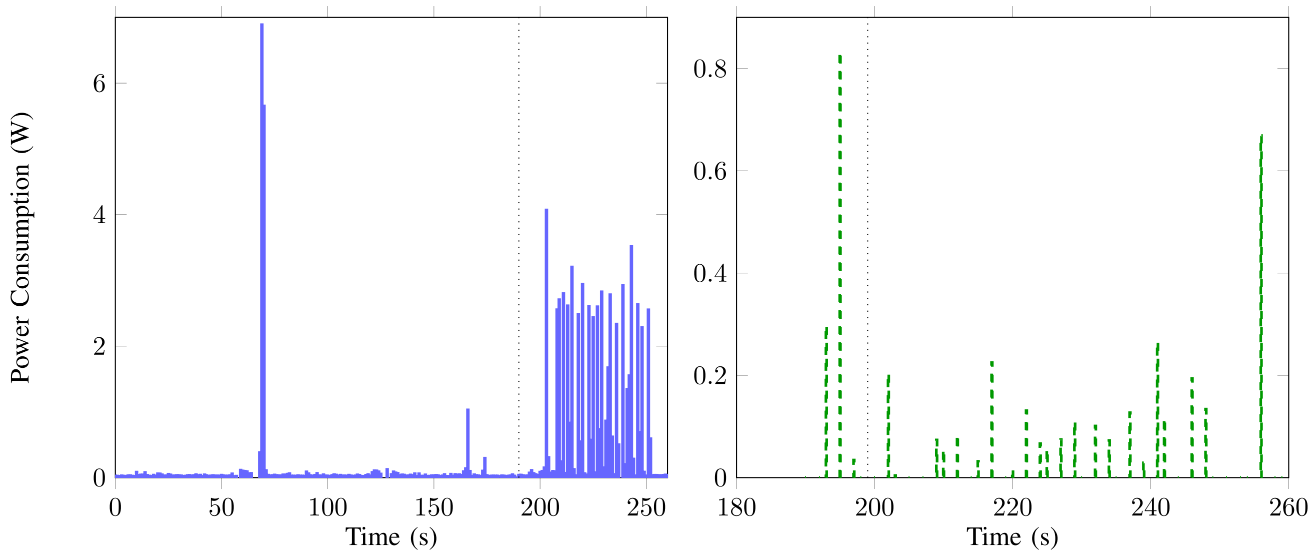

Fig. 2. **First experiment**. Power consumption of a blockchain node while performing FL operations (blue) and client connector while interacting with the blockchain (green).

three layers: a Flatten input layer, a Dense layer consisting of 128 neurons, and an output layer consisting of 10 neurons, one per class. In total, the network has 101.770 parameters.

Then, for the second experiment, we increased the complexity by employing CIFAR-10 and the neural network MobileNetV2 proposed in [19]. CIFAR-10 is a well-known dataset of 60.000 32x32 color images in 10 classes, with 6.000 images per class. There are 50.000 training images and 10.000 test images. MobileNetV2 has 2.270.794 parameters, about 20 times more than the network used for the first experiment. In both cases, the neural network is trained locally for 5 epochs, 10 FL rounds are performed, and datasets were fairly split among all the FL clients.

B. Results

Figures 2 and 3 give an idea of the overall power consumption of FL processes, while Table I reports the memory and CPU usage. In particular, for each experiment, we plot the results obtained during one of the 10 executions. The graphs sharply highlight the difference in terms of seconds needed to perform the FL processes. Indeed, the simpler case was deemed about 850 seconds less than the experiments whit a more complex setting. In the second experiment, the blockchain's power consumption is about an order of magnitude higher than in the first. In contrast, Figures 4 and 5 report the aggregate results of the 10 executions for both experiments, calculating the maximum and mean value of power consumption in different time slots. Table I proves that, in both experiments, the blockchain node has a greater consumption of memory and CPU compared to the client connector. Thus, it deems more powerful resources to be executed. In the following, we analyze in detail the power consumption of the blockchain node and the client while performing FL-related operations.

TABLE I
MEMORY (MB) AND CPU (%) USAGE.

		Memory (MB)		CPU (%)	
		Mean	Max	Mean	Max
First Experiment	Blockchain Node	161	315	1.53	32.09
	Client Connector	106	124	1.57	9.77
Second Experiment	Blockchain Node	1357	2688	4.16	49.78
	Client Connector	329	385	4.17	18.81

1) Blockchain: Blockchain power consumption can be divided into two main phases: the instantiation of the smart contract and the execution of the transactions to get partial models and combine them. In the blockchain part of Figures 2 and 3, the vertical dotted line divides the instantiation of the smart contract and the execution of the FL process. Concerning the power consumption during the instantiation, there are no remarkable differences since instantiating a smart contract does not depend on the complexity of the ML model. The second peak, which can be appreciated in Figure 2, represents the automatic invocation of the smart contract initialization function. This is clearly shown in Figure 4 where the mean power consumption during the smart contract instantiation phase is almost equal between the two experiments. The FL process involves three main transactions: publishing one model, publishing a second model with subsequent aggregation to create the global model, and reading the latter. Figures 2 and 3 highlight that, in both experiments, the aggregation is more power-intensive. Furthermore, there is a sharp difference in terms of power consumption between the two experiments: the second one is deemed 5 times the consumption of the former. Figure4 shows where the peak of

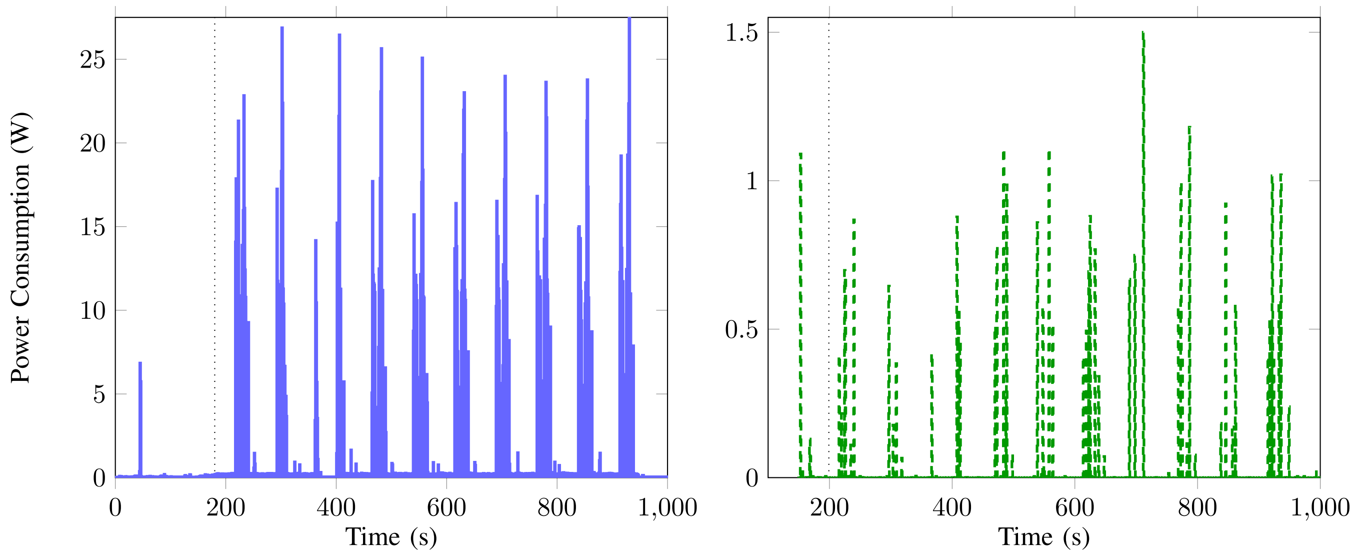

Fig. 3. **Second experiment**. Power consumption of a blockchain node while performing FL operations (blue) and client connector while interacting with the blockchain (green).

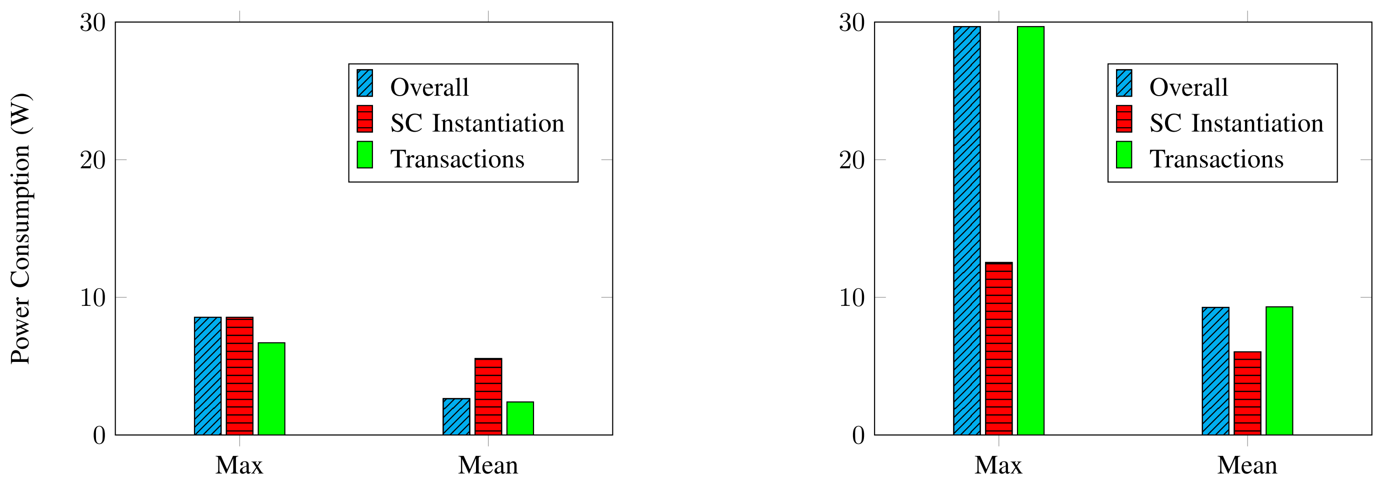

Fig. 4. Power consumption statistics of a blockchain node, respectively, for the first and second experiments.

maximum power consumption occurs throughout the execution of the experiments. In the first one, it is interesting to note that the consumption is higher during the smart contract instantiation than during the transaction execution part. In the second experiment, however, consumption is significantly higher during transaction execution. In general, Figure 4 shows how a more complex ML model markedly affects power consumption during the FL process.

2) Client: On the client side, also the execution can be divided into two phases: the creation of the connector and the sending of the various transactions to the blockchain. In the client connector part of Figures 2 and 3, the vertical dotted line represents the division between the creation of the connector and the execution of the FL process. By observing the figures, we can state that the creation of the connector, barring small variations, has comparable power consumption between the first and second experiments. In both cases, as Figure 5 shows, the peak power consumption occurs precisely during the creation of the connector. The consumption during sending and receiving of the various FL models is markedly different between the first and second experiments. As Figure 5 shows, the mean power consumption during the transaction phase increases in the second experiment by having to send a much larger amount of data. In general, as is the case with the first experiment, Figure 5 shows how a more complex ML model noticeably affects power consumption during the FL process. However, since the overall power consumption is lower than 3W, it can be deployed in real-world scenarios. For example, a client connector can be deployed on a Jetson Nano[4] since it provides satisfying compute performance with 5-10W of power consumption.

[4]https://developer.nvidia.com/embedded/jetson-nano-developer-kit

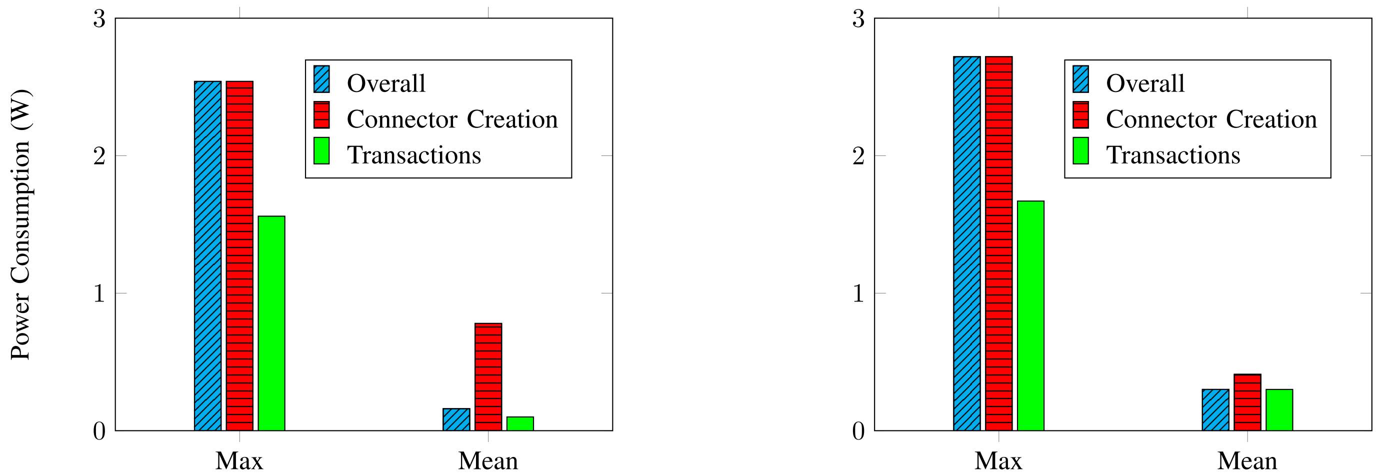

Fig. 5. Power consumption statistics of a client connector, respectively, for the first and second experiments.

V. Conclusions and Future Work

The interest in integrating FL and blockchain has increased remarkably over the last few years due to the blockchain's ability to address most of the weaknesses of using of a central parameter server and enhance trust among unknown participants, which are major barriers that hinder the widespread use of FL. However, although the blockchain brings astonishing advantages, there are many concerns about its power and energy consumption, which heavily depends on several factors such as the adopted consensus protocol, blockchain platform, and ML models

This paper presents an estimation of the power consumption of FlowChain, a blockchain-based platform that simplifies the use of FL. In our experiments, we demonstrated that the overall power consumption heavily depends on the complexity ML model employed. Furthermore, on the basis of the results collected, we can state that FlowChain can be deployed in real-world scenarios. In order to provide a more comprehensive discussion, in our future research, we plan to conduct further analysis by varying the deployment under study in order to determine which configuration can best satisfy FL requirements.

References

[1] P. Bellavista, L. Foschini, and A. Mora, "Decentralised Learning in Federated Deployment Environments: A System-Level Survey," *ACM Comput. Surv.*, vol. 54, no. 1, feb 2021.

[2] M. Ali, H. Karimipour, and M. Tariq, "Integration of blockchain and federated learning for Internet of Things: Recent advances and future challenges," *Computers Security*, vol. 108, p. 102355, 2021.

[3] D. C. Nguyen, M. Ding, Q.-V. Pham, P. N. Pathirana, L. B. Le, A. Seneviratne, J. Li, D. Niyato, and H. V. Poor, "Federated Learning Meets Blockchain in Edge Computing: Opportunities and Challenges," *IEEE Internet of Things Journal*, vol. 8, no. 16, pp. 12 806–12 825, 2021.

[4] C. Mazzocca, N. Romandini, M. Colajanni, and R. Montanari, "FRAMH: A Federated Learning Risk-Based Authorization Middleware for Healthcare," *IEEE Transactions on Computational Social Systems*, pp. 1–12, 2022.

[5] M. H. u. Rehman, A. M. Dirir, K. Salah, E. Damiani, and D. Svetinovic, "TrustFed: A Framework for Fair and Trustworthy Cross-Device Federated Learning in IIoT," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 12, pp. 8485–8494, 2021.

[6] S. K. Lo, Y. Liu, Q. Lu, C. Wang, X. Xu, H.-Y. Paik, and L. Zhu, "Towards Trustworthy AI: Blockchain-based Architecture Design for Accountability and Fairness of Federated Learning Systems," *IEEE Internet of Things Journal*, pp. 1–1, 2022.

[7] Z. Yang, Y. Shi, Y. Zhou, Z. Wang, and K. Yang, "Trustworthy Federated Learning via Blockchain," *IEEE Internet of Things Journal*, pp. 1–1, 2022.

[8] R. Beck, "Beyond Bitcoin: The Rise of Blockchain World," *Computer*, vol. 51, no. 2, pp. 54–58, 2018.

[9] W. Wang, D. T. Hoang, P. Hu, Z. Xiong, D. Niyato, P. Wang, Y. Wen, and D. I. Kim, "A Survey on Consensus Mechanisms and Mining Strategy Management in Blockchain Networks," *IEEE Access*, vol. 7, pp. 22 328–22 370, 2019.

[10] J. Sedlmeir, H. U. Buhl, G. Fridgen, and R. Keller, "The Energy Consumption of Blockchain Technology: Beyond Myth," *Business & Information Systems Engineering*, vol. 62, no. 6, pp. 599–608, Dec 2020.

[11] H. Vranken, "Sustainability of bitcoin and blockchains," *Current Opinion in Environmental Sustainability*, vol. 28, pp. 1–9, 2017, sustainability governance.

[12] M. J. Krause and T. Tolaymat, "Quantification of energy and carbon costs for mining cryptocurrencies," *Nature Sustainability*, vol. 1, no. 11, pp. 711–718, Nov 2018.

[13] Z. Yang, M. Chen, W. Saad, C. S. Hong, and M. Shikh-Bahaei, "Energy Efficient Federated Learning Over Wireless Communication Networks," *IEEE Transactions on Wireless Communications*, vol. 20, no. 3, pp. 1935–1949, 2021.

[14] C. W. Zaw, S. R. Pandey, K. Kim, and C. S. Hong, "Energy-Aware Resource Management for Federated Learning in Multi-Access Edge Computing Systems," *IEEE Access*, vol. 9, pp. 34 938–34 950, 2021.

[15] P. Bellavista, L. Foschini, R. Montanari, and N. Romandini, "FlowChain: The Playground for Federated Learning in Industrial Internet of Things Environments," *IEEE Internet of Things Magazine*, vol. 5, no. 2, pp. 78–83, 2022.

[16] A. A. Monrat, O. Schelén, and K. Andersson, "A Survey of Blockchain From the Perspectives of Applications, Challenges, and Opportunities," *IEEE Access*, vol. 7, pp. 117 134–117 151, 2019.

[17] D. J. Beutel, T. Topal, A. Mathur, X. Qiu, T. Parcollet, and N. D. Lane, "Flower: A friendly federated learning research framework," *arXiv preprint arXiv:2007.14390*, 2020.

[18] M. Uddin, Y. Darabidarabkhani, A. Shah, and J. Memon, "Evaluating power efficient algorithms for efficiency and carbon emissions in cloud data centers: A review," *Renewable and Sustainable Energy Reviews*, vol. 51, pp. 1553–1563, 2015.

[19] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.