HHS Public Access

Author manuscript

Inform Med Unlocked. Author manuscript; available in PMC 2024 October 31.

Published in final edited form as:

Inform Med Unlocked. 2024 ; 50: . doi:10.1016/j.imu.2024.101590.

WebQuorumChain: A web framework for quorum-based health care model learning

Xiyan Shao[a,1], **Anh Pham**[b,1], **Tsung-Ting Kuo**[b,c,d,*]

[a]UCSD Department of Computer Science and Engineering, University of California San Diego, La Jolla, CA, USA

[b]UCSD Health Department of Biomedical Informatics, University of California San Diego, La Jolla, CA, USA

[c]Department of Biomedical Informatics and Data Science, School of Medicine, Yale University, New Haven, CT, USA

[d]Department of Surgery, School of Medicine, Yale University, New Haven, CT, USA

Abstract

Background: Institutions interested in collaborative machine learning to enhance healthcare may be deterred by privacy concerns. Decentralized federated learning is a privacy-preserving and security-robust tool to promote cross-institutional learning, however, such frameworks require complex setups and advanced technical expertise. Here, we aim to improve their utilization by offering an intuitive, user-friendly, and secure system that integrates both front-end and back-end functionalities.

Method: We develop WebQuorumChain, an integrated system built upon the QuorumChain schema. We test the system on a 2-site network using two publicly available health datasets and measure the average vertical and horizontal-ensemble AUCs per dataset across 30 trials, as well as the average execution time of the system.

Results: Our system achieved consistently high AUCs for each dataset (0.94–0.96), with reasonable total execution times ranging from 5 to 20 min, inclusive of modeling and all other system overheads. The front-end displays event logs generated from back-end layers in real time, in sync with the progress of the underlying algorithm.

*Corresponding author. UCSD Health Department of Biomedical Informatics, University of California San Diego, La Jolla, CA, USA. tsung-ting.kuo@yale.edu (T.-T. Kuo).

[1]These authors contributed equally to this work.

Declaration of competing interest

The authors declare no competing interests.

CRediT authorship contribution statement

Xiyan Shao: Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis. **Anh Pham:** Writing – original draft, Investigation, Conceptualization. **Tsung-Ting Kuo:** Writing – review & editing, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization.

Ethical statement

The datasets used in this study are publicly available. There was no ethic approval required.

Conclusions: We develop a web-based system that supplies users with visual tools to configure the federated learning network, manage training sessions, and inspect the learning process. WebQuorumChain helps schedule and monitor low-level processes without violating the fundamental security promises of cross-institutional decentralized machine learning. The system also maintains predictive accuracy and runtime efficiency in the presence of additional layers. WebQuorumChain will help promote meaningful collaboration among healthcare institutions, who can retain full control of their data privacy while contributing to data-driven discoveries.

Keywords

Clinical research information systems; Health information infrastructure; Privacy and security; Machine learning; Predictive modeling

1. Background and significance

1.1. Federated learning in healthcare

The recent rise of machine learning (ML) applications in medicine and healthcare research has led to a wide range of successful medical advances, including clinical diagnosis [1–5], radiography analysis [6–9], and drug discovery [10–12]. It is projected that the healthcare sector may be among the industries most affected by ML-backed innovations [13–16], the majority of which rely on frequent access to data. However, the unique nature of health information dictates that data are wanted both in large quantities and with stringent privacy protection [17]. This dual desire to facilitate big data analysis within the constraint of restricted privacy calls for the use of federated learning (FL) techniques [18,19]. In contrast to the conventional model of centralized learning, where pooled data are deposited and analyzed at one central database and thus exposed to privacy and security drawbacks [20], FL does not require local data to be transferred externally [21,22]. With FL, each site participant (agent/learner) retains full control of their collected data and stays in charge of local model construction; the only information being exchanged for global model aggregation would be derived statistics and covariates [23,24]. However, since FL relies on a central server for the process of model aggregation and distribution, it still inherits the Single-Point-of-Failure (SPoF) security vulnerability from conventional ML [25,26]. That is, should the central authority entrusted with learning coordination go offline, whether due to either routine maintenance or a hostile attack, the entire collaboration would cease. Such disruptive events have been observed in recent times [27–29], thus highlighting the need for proactive measures to ensure the integrity and robustness of collaborative learning.

1.2. Decentralized federated learning for immutability, transparency and high-availability

Decentralized federated learning (DFL) in which no central server is needed can incorporate both the privacy-preserving nature of FL and the security robustness against SPoF [30,31]. As a distributed ledger technology, blockchain can facilitate the transmission of model statistics among network learners in the absence of a central authority [50–52]. Moreover, the innate design of blockchain offers the added benefits of immutability, transparency, and high availability [32], making it more resistant to SPoF and data tampering threats [20,25,31]. Several state-of-the-art DFL algorithms including ModelChain

[21], HierarchicalChain [23], and QuorumChain [33] employ blockchain technology as their infrastructural frameworks. ModelChain outlines a technical design that enables dissemination of model updates on a “flattened”, or non-hierarchical, topology, where each individual node of the blockchain represents a single agent [21]. When the network topology is more hierarchically complex, a common scenario in healthcare as a research entity may comprise several subnetworks and subsites. HierarchicalChain [23] is one architectural adaptation for “network of networks” collaborative learning. Next, to handle learning disruption when agents leave the network due to maintenance/attacks, QuorumChain [33] algorithmically allows learning to progress under the agent unavailability scenario. The decision to continue learning rests on the formation of “quorum,” or a subset of learners who remain in the collaboration and whose cumulative amount of data passes a certain threshold [33]. This approach has been shown to significantly improve predictive correctness in site unavailability scenarios [33].

1.3. Usability issues in decentralized federated learning and related works

Although functional, decentralized solutions often demand complex network and client setups, along with extensive cross domain expertise in machine learning, distributed systems, and cryptography [34]. Users may also require significant training to adapt to nonvisual, noninteractive frameworks. For example, a proof of concept architecture [35] explored the usability of blockchain-based learning without implementing an intuitive graphical user interface (GUI) to help the collaborative learning experience; however, studies have advocated for user-friendly interfaces and GUI driven platforms [36–38], such as a web browser access [36] or a layered design where a frontend web GUI separates users from backend tasks [38]. It is seen that the value of enhanced usability in federated learning is greatly anticipated. With regards to blockchain specific federated learning, web enabled access is even more important to health researchers as technical complexity may substantially increase when blockchain is integrated with machine learning. To the best of our knowledge, while the idea of an “application layer” atop the blockchain layer for user level interaction and data aggregation has been proposed [39], a prototype has yet to be developed. The lack of user friendly and intuitive toolkits may discourage users from utilizing DFL algorithms and render the whole process opaque to the general users, despite their demonstrated benefits of immutability and transparency in data provenance. Given that the primary concern of users is the final predictive performance other than the technical steps to expertly deploy and administer a decentralized infrastructure, it is critical to simplify the entry point for DFL systems. In the case of QuorumChain, while the configuration of the “quorum” mechanism is entrusted to a network of immutable, source-verifiable smart contracts, the setup still requires users to start multiple blockchain nodes, deploy the contracts, and interact with a command line interface. These language-specific steps may hinder the observability of the end-to-end system, due to the risks of misconfiguration and a lack of intuitive, GUI based tools. To promote the broader adoption and functionality of DFL schemas such as QuorumChain, it is crucial to enable users to easily establish and monitor the learning process, while providing transparent access to key information and metrics, ensuring a seamless and efficient distributed learning experience.

2. Objective

In this study, we aim to improve the utilization of decentralized federated learning for cross-institutional, privacy-preserving collaborative learning, specifically for the QuorumChain schema. We do this by developing a web based integrated system that allows healthcare researchers to set up the framework at ease, administer training sessions with visual, interactive tools, and automatically inspect learning/model transactions beyond manual inspection.

3. Materials and methods

3.1. Methodology overview

We chose QuorumChain [33] as the underlying DFL algorithm upon which our system would be constructed. It inherits the privacy-preserving nature of previous DFL schemas, as well as the ability to coordinate the transmission of local and global model statistics across a complex hierarchical topology [33]. The algorithm behind QuorumChain is a novel Proof-of-Quorum consensus algorithm, as detailed in the previously published pseudocodes and demonstrated through statistical testing using three health datasets of size 141; 1253; and 157,493 patients, respectively [20,25,33].

WebQuorumChain seeks to enhance the usability of QuorumChain by introducing a web based application layer that abstracts the system's implementation details into intuitive, "clickable" actions. It provides an interactive GUI, allowing users to bypass complex training procedures while still engaging with the learning and networking backend. The system is designed to be delivered as a local executable that can be conveniently loaded on a per-site basis. Three key features, setup tools, learning dashboard, and inspection tools, are offered (Fig. 1). Specifically, the setup tools are designed to aid users in initiating the federated learning rounds, such as starting the blockchain daemon process and connecting to the permissioned network. In addition, these tools allow users to customize learning parameters, such as learning rate, regularization, and number of epochs, and link their local training and testing data (Fig. 1A). Next, the learning dashboard abstracts the implementation details of synchronization among multiple sites, allowing users to focus on the iteration-by-iteration model updates and eventual convergence without having to manage complex inter-site synchronization tasks (Fig. 1B). Meanwhile, the inspection tools enable users to review training logs automatically emitted by the learners and network agents, as well as blockchain transaction logs. This allows users to monitor and review the training progress in real-time, enhancing transparency and control over the system's operations (Fig. 1C).

3.2. User workflow

From a user's perspective, the WebQuorumChain service offers a straightforward and user-friendly approach to administer learning functionalities, all of which are fully equipped with intuitive clickables. Its implementation commences with the user selecting the appropriate command buttons to configure and initiate the blockchain network through the provided GUI; following this initial setup, the user can prepare the learning process by linking the

training and testing datasets and deploying the essential contracts for the DFL protocol (Fig. 2A). After the contracts are deployed, the learning process occurs (Fig. 2B), and the user may actively interact with the graphical inspection tools to track its progress (Fig. 2C). Specifically, logs are generated throughout the learning span to encapsulate critical information pertaining to model-specific details such as gradient contents, and other algorithm state transitions. These logs can be scrutinized in real time with visual indicators for easy comprehension. Once learning completes, the evaluation metrics for all ensemble levels, as well as the final model, are displayed for further analysis and validation.

3.3. Underlying system layers

As a robust encapsulation of the QuorumChain implementation, WebQuorumChain employs an unobtrusive approach that avoids changes to the original architecture. Between the local executable and the decentralized blockchain network, the on chain activities can be effectively relayed, scheduled, and monitored through a front facing web client and its graphical elements, as enumerated in Section 3.2. In return, the client interacts with the intermediate local “server” (local host) within a virtual node that oversees 1) the underlying blockchain transactions (which are coordinated by the blockchain node, i.e., Go-Ethereum (Geth) Process [40]), 2) the message queue in which events and transaction logs are piped, and 3) the algorithmic functions of QuorumChain (the QuorumChain Process) that regulate the fundamental operations of the federated learning. This strategy ensures minimal chance of computational manipulation or state modification, while still offering a convenient view of the execution status. Thus, the enhanced features and functionality of WebQuorumChain do not violate the integrity and reliability of the underlying QuorumChain algorithms. The architecture is depicted in Fig. 3, and more details about each layer are provided in Section 3.4.

3.4. Back-end modules and front-end interface

While the Geth Process converts smart contract definitions into executables on the Ethereum network and allows subsequent interactions with the blockchain, the QuorumChain Process communicates with this on chain component to synchronize the order of model exchange, aggregation, quorum formation, and learning continuation across quorum members. It also dispatches event logs to the Java Message Service (JMS), which are then funneled into a local Message Queue (MQ) for buffering. An event log can contain information such as when a participant enters or leaves the training session, and updates to the model/gradient contents. At the invocation of the inspection tool, the event messages will flow to the web client for visual display. Without this mechanism, such critical information would not be accessible outside of the algorithm. Finally, the web client brings the server’s functionalities to the user by providing a user-friendly and intuitive interface. It listens to the inspection logs and renders them for visualization, while also offering a set of operations that enable easy configuration and control of training states. Some of these interface designs are showcased in Fig. 4.

3.5. Data

We evaluated the system on two public datasets, Cancer Biomarker (CA) [41] and Myocardial Infarction (Edin) [42], both with a binary prediction target of disease presence.

The Myocardial Infarction (Edin) dataset contains 9 features and 1253 patients from the Royal Infirmary Accident and Emergency Department at Edinburg, Scotland [42], including features of pain in left arm, pain in right arm, nausea, sweating, hypoperfusion, new Q waves, ST elevation, ST depression, and T wave inversion. The Cancer Biomarker (CA) dataset has 2 markers as covariates, CA-19 and CA-125, and 141 samples [41]. The details of each dataset are further summarized in Table 1.

3.6. Evaluation setting

To test the WebQuorumChain system, we set up a DFL framework on a private Ethereum network with 2 sites collaborating to train single layer logistic regression models for binary classification tasks over 3 epochs. For the CA dataset, the QuorumChain hyperparameters were set with a waiting-time-period of 4 s, and a quorum percentage threshold of 51 %. For the Edin dataset, the waiting-time-period was 16 s, and the quorum threshold was 60 %. Both datasets used a pooling-time-period of 1 s, and a maximum of 100 iterations per level. All training configurations were initiated via the web client of WebQuorumChain. Next, we ran 30 trials of the collaborative learning per dataset and measured the final AUCs (Area Under the Curve) of the global models, and runtimes either in the training or evaluation phases, or both (i.e., total execution time, which also includes system overheads). These times were measured end-to-end on the client side, considering the overhead of the application layer. Our implementation employed Geth (Go-Ethereum) v1.10.20 [40] for the blockchain node, ActiveMQ 5.17.1 [43] as the Java Message Service (JMS) provider, Spring Boot 2.7.2 [44] as the server framework and React 18.2.0 [45] for the web client framework. These experiments were conducted on Alpine Linux [46] utilizing Java SE 17 [47] within docker containers.

4. Results

4.1. Prediction accuracy and runtime efficiency

The experimental results are presented in Table 2. Both datasets completed their respective 30 trials. The AUC results were consistently high across all trials given the deterministic nature of the logistic regression algorithm. For the CA dataset, the final AUC of the collaborative model was 0.94 whether at the horizontal (i.e., the prediction score of a site is generated using the scores from all models at the same level, as weighted averaged by its respective training data sizes) or vertical (i.e., the weighted average prediction score of a site is generated using the scores from each level related to that site) ensemble [33]. For the Edin dataset, both vertical and horizontal AUC was 0.96. With regards to the execution speed of our system, execution times were measured from the start to the end of the corresponding learning periods on the web client, taking system latencies into account, and thus would exceed the sum of training and evaluation times. Specifically, it took less than 2 min to train the CA dataset, around 2.5 min for evaluation, with a total execution time around 5.3 min inclusive of system overhead. It is noted that the size of the CA dataset is relatively small, as it contained two covariates of cancer markers CA-19 and CA-125 for 141 samples. The standard deviations per phase for this dataset ranges between 3 and 4.5 s. Meanwhile, the larger Edin dataset with 9 covariates and 1253 samples took approximately 9 min

for training, ~7.8 min for evaluation, and ~20 min for total execution time. The standard deviations per phase were less than a second.

4.2. Inspection dashboard

With regards to the inspection function, example strip plots for learning events and blockchain events emitted by WebQuorumChain from both sites and datasets are presented in Fig. 5. Each dot represents the timestamp at which one event log was emitted; for example, at time X, the system alerts that an event of type Y has occurred. These logs were collected in real-time and time-stamped at the web client, showing that the algorithm executed consistently throughout the session. The web client was able to effectively capture information from the ongoing execution process. The learning event categories include INFO, MODEL, and RESULT. INFO events provide general information about various stages along the progression of the algorithm, MODEL events represent emitted gradient and model contents, and RESULT events mark the completion of evaluation stages when AUCs can be captured. Another event category is blockchain (BLOCKCHAIN), which depicts blockchain transaction events.

5. Discussion

5.1. Findings

Our experimental results demonstrate that WebQuorumChain is an efficient extension to the original QuorumChain algorithm for decentralized federated learning. Despite the addition of application layers, the system completed each learning trial within a reasonable timeframe, without errors, and yielded consistently high AUC. The variations in training, evaluation, and/or total execution time may be attributed to nondeterministic network and/or scheduling latencies. Nonetheless, such variations are negligible, as evidenced by their relatively small standard deviations. This suggests that the overhead introduced by our architectural integration is minimal, confirming the technical practicality of WebQuorumChain. The method is also robust as it inherits the advantage of the original QuorumChain algorithm [33]. Even when a site became unavailable during the learning rounds, the algorithm successfully maintained continued learning among the remaining sites, as evidenced by the 100 % completion rate across all 30 trials per dataset.

Another benefit WebQuorumChain offers is the interactive interface, which provides intuitive visual tools for inspection. As shown in Fig. 5, the inspected events align with the stages described in the QuorumChain algorithm. For example, the MODEL logs, which contain model statistics, were emitted during each iteration of the training phase, whereas the RESULT logs reflected the evaluation stages in real-time. Meanwhile, INFO and BLOCKCHAIN events occurred steadily over the course of the learning session, indicating that the workload was evenly distributed between the two sites.

5.2. Enhanced user experience without security compromise

From the perspective of users, WebQuorumChain is an effective upgrade over the command line interfaces provided in previous works. It offers “clickable” setup, learning, and inspection tools with fine grained user control, while seamlessly managing the low-level

jobs and algorithms defined by the QuorumChain algorithm. At the same time, our integrated system remains loyal to the privacy-preserving and security-robust nature of its original counterpart QuorumChain. The newly introduced local host only manages client side logic, while peer-to-peer communication still relies solely on the blockchain network, thus maintaining compliance to its decentralized, transparent, and immutable principle. Overall, this close association between the proposed WebQuorumChain framework and the state-of-the-art QuorumChain protocol will open doors for clinical research users to make use of decentralized federated learning tools, allowing for larger sample sizes, higher statistical power, and more generalizable results without the security risk of exchanging direct data across sites. Such users can trust both the integrity of the immutable/source verifiable algorithm and the ease of use of its web based architecture to support their collaborative efforts.

5.3. Limitations

Our study is limited by the following constraints.

1. Lack of real life user testing. Thorough user testing may establish the efficacy of our usability features such as on-interface, step by step guidance. Further studies may need to focus on the human computer interaction aspects of the system.
2. Limit on datasets and number of sites. The system was validated on a 2-site network using two public health datasets. A larger scale study might be needed to test the scalability of our system as the number of learning agents grows, and/or when the size of the dataset changes.
3. Lack of full authentication/authorization to the web interface. To focus on improving the usability of the system, our proposed design has yet to include full-fledged authentication mechanisms such as 2-factor authentication [48] or role based access control [49]. Further investigation with regards to user identity management is needed to enhance the security and privacy protection of WebQuorumChain.
4. Limit of experimental comparison with existing methods. As comparable approaches for blockchain-based federated learning with graphical user interface are scarce, we have yet carried out experimental comparison against other methods.

6. Conclusion

The key finding we established is that the integrated system WebQuorumChain helps simplify the implementation of complex setup and configuration steps without substantial overheads. Therefore, despite the limitations listed in the previous section, WebQuorumChain can improve the utilization of DFL schemas. From a technical perspective, the system helps schedule and monitor low level jobs and processes without violating the fundamental security promises of QuorumChain. It also maintains both predictive accuracy and runtime efficiency in the presence of additional system layers. In practice, WebQuorumChain can provide intuitive and interactive tools to encourage users to

use the state-of-the-art collaborative QuorumChain algorithm for meaningful collaboration among healthcare institutions. Specifically, since WebQuorumChain is readily available as a web interface with clickable buttons, its intuitive implementation may let users set up, run, and inspect the progression of their machine learning models with ease. Researchers thus can retain full control of their patient data privacy while contributing to the rise of data-driven discoveries. As a result, WebQuorumChain may support the growth of predictive analytics aiming to serve the wider community of physicians and researchers, a beneficial academic contribution to the field of clinical research informatics.

Acknowledgements

The authors would like to thank Cyd Burrows-Schilling, MS, and Randi Sutphin for the technical support of the UCSD Campus AWS cloud network. We also thank Armin Nouri, Maxim Edelson, MS, and Dr. Jennifer Nguyen for providing proofreading help.

Funding

The authors were funded by the U.S. National Institutes of Health (NIH) (R01EB031030). The content is solely the responsibility of the authors and does not necessarily represent the official views of the NIH. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

References

[1]. Agarap AFM. On breast cancer detection: an application of machine learning algorithms on the Wisconsin diagnostic dataset. In: Proceedings of the 2nd international conference on machine learning and soft computing; 2018. p. 5–9. 10.1145/3184066.3184080.

[2]. Battineni G, Sagaro GG, Chinatalapudi N, Amenta F. Applications of machine learning predictive models in the chronic disease diagnosis. J Pers Med 2020;10(2):21. 10.3390/jpm10020021. [PubMed: 32244292]

[3]. Kononenko I Machine learning for medical diagnosis: history, state of the art and perspective. Artif Intell Med 2001;23(1):89–109. 10.1016/S0933-3657(01)00077-X. [PubMed: 11470218]

[4]. Olsen CR, Mentz RJ, Anstrom KJ, Page D, Patel PA. Clinical applications of machine learning in the diagnosis, classification, and prediction of heart failure. Am Heart J 2020;229:1–17. 10.1016/j.ahj.2020.07.009. [PubMed: 32905873]

[5]. Richens JG, Lee CM, Johri S. Improving the accuracy of medical diagnosis with causal machine learning. Nat Commun 2020;11(1):3923. 10.1038/s41467-020-17419-7. [PubMed: 32782264]

[6]. Kassania SH, Kassanib PH, Wesolowskic MJ, Schneidera KA, Detersa R. Automatic detection of coronavirus disease (COVID-19) in X-ray and CT images: a machine learning based approach. Biocybern Biomed Eng 2021;41(3):867–79. 10.1016/j.bbe.2021.05.013. [PubMed: 34108787]

[7]. Wang S, Summers RM. Machine learning and radiology. Med Image Anal 2012;16 (5):933–51. 10.1016/j.media.2012.02.005. [PubMed: 22465077]

[8]. Thrall JH, Li X, Li Q, Cruz C, Do S, Dreyer K, et al. Artificial intelligence and machine learning in radiology: opportunities, challenges, pitfalls, and criteria for success. J Am Coll Radiol 2018;15(3):504–8. 10.1016/j.jacr.2017.12.026. [PubMed: 29402533]

[9]. Choy G, Khalilzadeh O, Michalski M, Do S, Samir AE, Pianykh OS, et al. Current applications and future impact of machine learning in radiology. Radiology 2018; 288(2):318–28. 10.1148/radiol.2018171820. [PubMed: 29944078]

[10]. Lavecchia A Machine-learning approaches in drug discovery: methods and applications. Drug Discov Today 2015;20(3):318–31. 10.1016/j.drudis.2014.10.012. [PubMed: 25448759]

[11]. Patel L, Shukla T, Huang X, Ussery DW, Wang S. Machine learning methods in drug discovery. Molecules 2020;25(22):5277. 10.3390/molecules25225277. [PubMed: 33198233]

[12]. Ekins S, Puhl AC, Zorn KM, Lane TR, Russo DP, Klein JJ, et al. Exploiting machine learning for end-to-end drug discovery and development. Nat Mater 2019;18(5): 435–41. 10.1038/s41563-019-0338-z. [PubMed: 31000803]

[13]. Collier M, Fu R, Yin L. Artificial Intelligence: Healthcare's new nervous system. Accenture. https://www.accenture.com/content/dam/accenture/final/a-com-migration/manual/r3/pdf/pdf-49/Accenture-health-artificial-intelligence-j.pdf [accessed 6 June 2023].

[14]. Future Health Index 2019 Transforming Healthcare Experiences. Phillips. https://images.philips.com/is/content/PhilipsConsumer/Campaigns/CA20162504_Philips_Newscenter/Philips_Future_Health_Index_2019_report_transforming_healthcare_experiences.pdf [accessed 9 October 2023].

[15]. Callahan A, Shah NH. Machine learning in healthcare. In: Sheikh A, Cresswell KM, Wright A, Bates DW, editors. Key advances in clinical informatics. Elsevier; 2017. p. 279–91. 10.1016/B978-0-12-809523-2.00019-4. chap. 19.

[16]. Javaid M, Haleem A, Singh RP, Suman R, Rab S. Significance of machine learning in healthcare: features, pillars and applications. Int J Intell Networks 2022;3: 58–73. 10.1016/j.ijin.2022.05.002.

[17]. Abouelmehdi K, Beni-Hessane A, Khaloufi H. Big healthcare data: preserving security and privacy. J Big Data 2018;5(1):1–18. 10.1186/s40537-017-0110-7.

[18]. Sheller MJ, Edwards B, Reina GA, Martin J, Pati S, Kotrotsou A, et al. Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data. Sci Rep 2020;10(1):12598. 10.1038/s41598-020-69250-1. [PubMed: 32724046]

[19]. Kuo T-T, Pham A Detecting model misconducts in decentralized healthcare federated learning. Int J Med Inform 2022;158:104658. 10.1016/j.ijmedinf.2021.104658.

[20]. Kuo T-T, Gabriel RA, Ohno-Machado L. Fair compute loads enabled by blockchain: sharing models by alternating client and server roles. J Am Med Inform Assoc 2019;26(5):392–403. 10.1093/jamia/ocy180. [PubMed: 30892656]

[21]. Kuo T-T, Ohno-Machado L Modelchain: decentralized privacy-preserving healthcare predictive modeling framework on private blockchain networks. arXiv preprint arXiv:180201746 2018. 10.48550/arXiv.1802.01746.

[22]. Zerka F, Barakat S, Walsh S, Bogowicz M, Leijenaar RT, Jochems A, et al. Systematic review of privacy-preserving distributed machine learning from federated databases in health care. JCO Clin Cancer Inform 2020;4:184–200. 10.1200/CCI.19.00047. [PubMed: 32134684]

[23]. Kuo T-T, Kim J, Gabriel RA. Privacy-preserving model learning on a blockchain network-of-networks. J Am Med Inform Assoc 2020;27(3):343–54. 10.1093/jamia/ocz214. [PubMed: 31943009]

[24]. Kim YJ, Hong CS. Blockchain-based node-aware dynamic weighting methods for improving federated learning performance. IEEE 2019:1–4. 10.23919/APNOMS.2019.8893114.

[25]. Kuo T-T, Gabriel RA, Cidambi KR, Ohno-Machado L. EXpectation Propagation LOgistic REgRession on permissioned block CHAIN (ExplorerChain): decentralized online healthcare/genomics predictive model learning. J Am Med Inform Assoc 2020;27(5):747–56. 10.1093/jamia/ocaa023. [PubMed: 32364235]

[26]. Zhang H, Bosch J, Olsson HH. Federated learning systems: architecture alternatives. IEEE 2020:385–94. 10.1109/APSEC51365.2020.00047.

[27]. Harwell D New York Stock Exchange outage adds to fears on financial markets. https://www.washingtonpost.com/business/economy/nyse-outage-raises-questions-about-whether-regulators-can-keep-up/2015/07/08/e8f7b02a-258d-11e5-b77f-eb13a215f593_story.html. [Accessed 5 April 2024].

[28]. United Airlines Tsidulko J. NYSE outages reveal poor redundancy architecture, insufficient testing. CRN. https://www.crn.com/news/security/300077385/united-airlines-nyse-outages-reveal-poor-redundancy-architecture-insufficient-testing.htm. [Accessed 23 August 2024].

[29]. Strickland E 5 Major Hospital Hacks: Horror Stories from the Cybersecurity Frontlines. IEEE Spectrum. https://spectrum.ieee.org/5-major-hospital-hacks-horror-stories-from-the-cyber-security-frontlines [accessed 23 August 2024].

[30]. Hu C, Jiang J, Wang Z. Decentralized federated learning: a segmented gossip approach. 2019. 10.48550/arXiv.1908.07782. arXiv preprint arXiv: 190807782.

[31]. Kuo T-T. The anatomy of a distributed predictive modeling framework: online learning, blockchain network, and consensus algorithm. JAMIA Open 2020;3(2): 201–8. 10.1093/jamiaopen/ooaa017. [PubMed: 32734160]

[32]. Kuo T-T, Pham A, Edelson ME, Kim J, Chan J, Gupta Y, et al. Blockchain-enabled immutable, distributed, and highly available clinical research activity logging system for federated COVID-19 data analysis from multiple institutions. J Am Med Inform Assoc 2023:ocad049. 10.1093/jamia/ocad049.

[33]. Kuo T-T, Pham A Quorum-based model learning on a blockchain hierarchical clinical research network using smart contracts. Int J Med Inform 2023;169: 104924. 10.1016/j.ijmedinf.2022.104924. [PubMed: 36402113]

[34]. Ludwig H, Baracaldo N, Thomas G, Zhou Y, Anwar A, Rajamoni S, et al. IBM federated learning: an enterprise framework white paper v0. 1. 2020. 10.48550/arXiv.2007.10987. arXiv preprint arXiv:200710987.

[35]. Drungilas V, Vaičiukynas E, Jurgelaitis M, Butkienė R, Čeponienė L. Towards blockchain-based federated machine learning: smart contract for model inference. Appl Sci 2021;11(3):1010. 10.3390/app11031010.

[36]. Jiang W, Li P, Wang S, Wu Y, Xue M, Ohno-Machado L, et al. WebGLORE: a web service for Grid LOgistic REgression. Bioinformatics 2013;29(24):3238–40. 10.1093/bioinformatics/btt559. [PubMed: 24072732]

[37]. Burlachenko K, Horváth S, Richtárik P. Fl_pytorch: optimization research simulator for federated learning. In: Proceedings of the 2nd ACM international workshop on distributed machine learning; 2021. p. 1–7. 10.1145/3488659.3493775.

[38]. Matschinske J, Späth J, Nasirigerdeh R, Torkzadehmahani R, Hartebrodt A, Orbán B, et al. The featurecloud ai store for federated learning in biomedicine and beyond. arXiv preprint arXiv:210505734 2021. 10.48550/arXiv.2105.05734.

[39]. Liu W, Zhang YH, Li YF, Zheng D. A fine-grained medical data sharing scheme based on federated learning. Concurr Comput Pract Exp 2023;35(20):e6847. 10.1002/cpe.6847.

[40]. Official Go implementation of the Ethereum protocol. https://geth.ethereum.org.

[41]. Zou KH, Liu A, Bandos AI, Ohno-Machado L, Rockette HE. In: Statistical evaluation of diagnostic performance: topics in ROC analysis. first ed. CRC Press; 2011. 10.1111/insr.12020_27.

[42]. Kennedy R, Fraser H, McStay L, Harrison R. Early diagnosis of acute myocardial infarction using clinical and electrocardiographic data at presentation: derivation and evaluation of logistic regression models. Eur Heart J 1996;17(8):1181–91. 10.1093/oxfordjournals.eurheartj.a015035. [PubMed: 8869859]

[43]. Christudas B In: Practical microservices architectural patterns: event-based Java microservices with spring Boot and spring cloud. first ed. 2019. p. 861–7. 10.1007/978-1-4842-4501-9.

[44]. Walls C In: Spring Boot in action. first ed. Manning; 2016. https://dl.acm.org/doi/10.5555/3002430.

[45]. Gackenheimer C In: Introduction to React. first ed. Apress; 2015. 10.1007/978-1-4842-1245-5.

[46]. Ely D, Savage S, Wetherall D. Alpine: a {User-Level} infrastructure for network protocol development. USITS; 01); 2001. p. 15. https://dl.acm.org/doi/10.5555/1251440.1251455.

[47]. Java | Oracle. https://www.java.com/en/[accessed 23 August 2024].

[48]. Ometov A, Bezzateev S, Mäkitalo N, Andreev S, Mikkonen T, Koucheryavy Y. Multi-factor authentication: a survey. Cryptography 2018;2(1):1. 10.3390/cryptography2010001.

[49]. Sandhu RS. Role-based access control. Adv Comput 1998:237–86. 10.1016/S0065-2458(08)60206-5. Elsevier.

[50]. Kuo T-T, Kim H-E, Ohno-Machado L Blockchain distributed ledger technologies for biomedical and health care applications. J Am Med Inf Assoc 2017;24(6):1211–20.

[51]. Yu H, Sun H, Wu D, Kuo T-T, editors. Comparison of smart contract blockchains for healthcare applications. AMIA annual symposium 2019: American medical informatics association, bethesda, MD.

[52]. Lacson R, Yu Y, Kuo T-T, Ohno-Machado L. Biomedical blockchain with practical implementations and quantitative evaluations: a systematic review. J Am Med Inf Assoc 2024;31(6):1423–35.

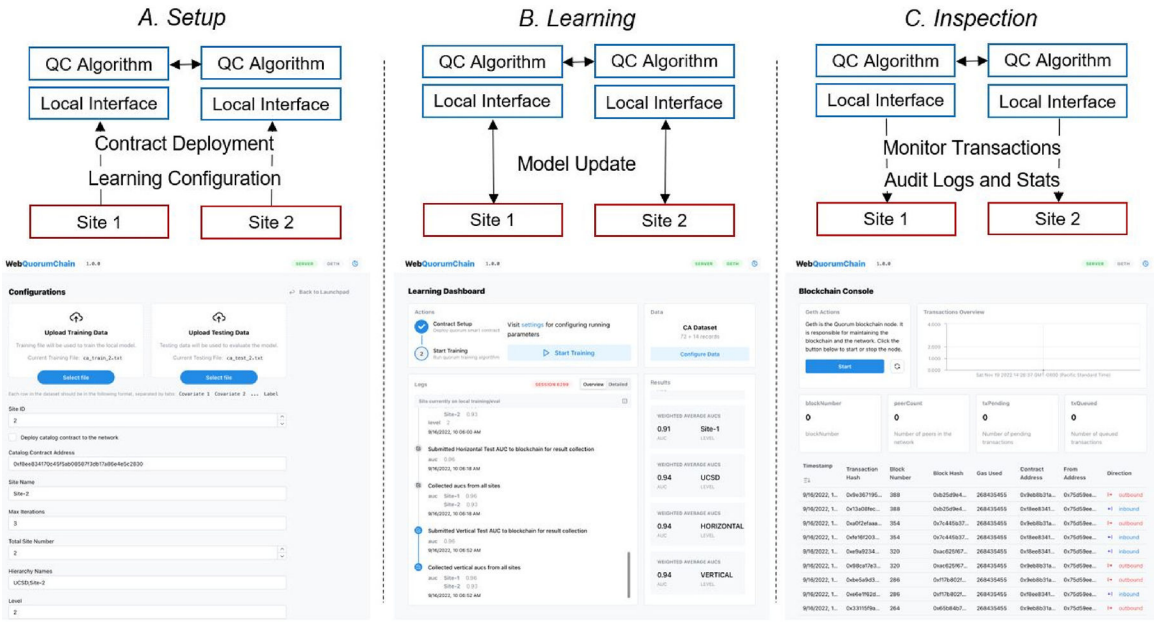

Fig. 1.

High-level overview of WebQuorumChain: **(A)** The setup component helps initiate the learning system by integrating smart contract deployment and hyperparameter configuration through a local interface. **(B)** The learning component helps wrap the core learning algorithm and handles model updates. **(C)** The inspection component helps monitor model transactions and allows the auditing of learning and blockchain logs with viewable statistics (higher resolution versions of the lower images of each panel are provided in Fig. 4).

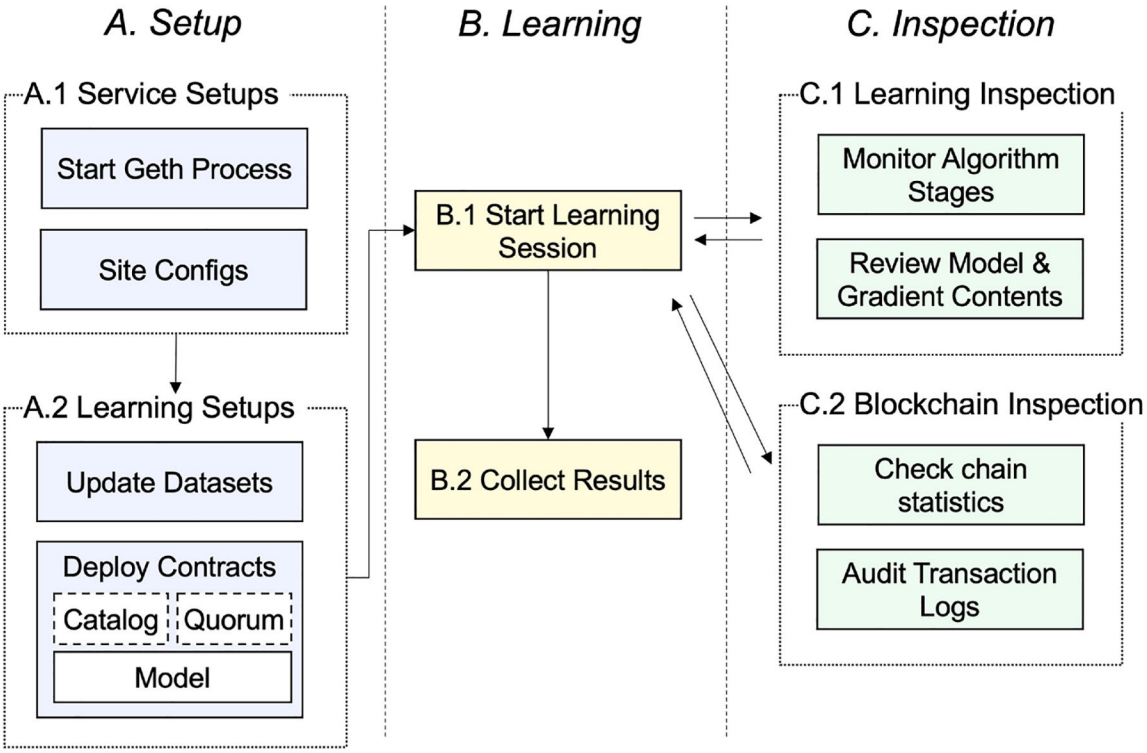

Fig. 2.

User workflow and GUI functionalities: **(A)** To start the service, the user first sets up network configs by starting the Geth blockchain node and editing site configs via the configuration web interfaces. The user then deploys smart contracts Quorum, Model, and optionally Catalog as defined by the QuorumChain algorithm, as well as linking to their datasets. **(B)** The learning process begins and continues until results can be collected. **(C)** Inspection tools are readily available during the life cycle of learning. Event logs including algorithm states and model information such as gradient contents are emitted for review. Upon learning conclusion, evaluation metrics for all ensemble levels as well as the final model are displayed.

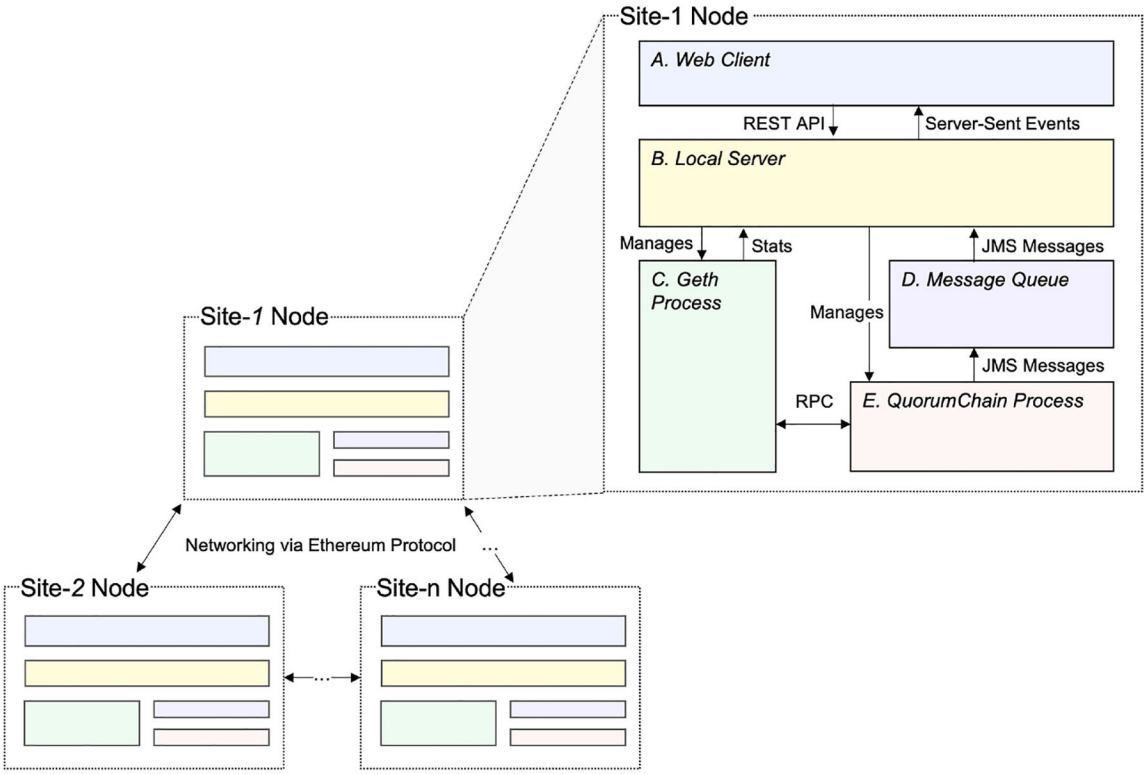

Fig. 3.

Underlying architecture breakdown: *n* sites participate in the FL session and form a peer-to-peer network communicating via blockchain protocols. Per-node architecture is shown in Site-1 Node diagram: **(A)** web client exposes the functionalities of the server by providing a user-friendly interface; **(B)** local server mediates between user interactions and the execution of learning units; **(C)** Geth (Go-Ethereum) Process is the blockchain client that executes smart contract code and the networking agent for communicating among sites; **(D)** message queue of logs and events emitted during the process; and **(E)** the QuorumChain Process carries the fundamental training algorithms.

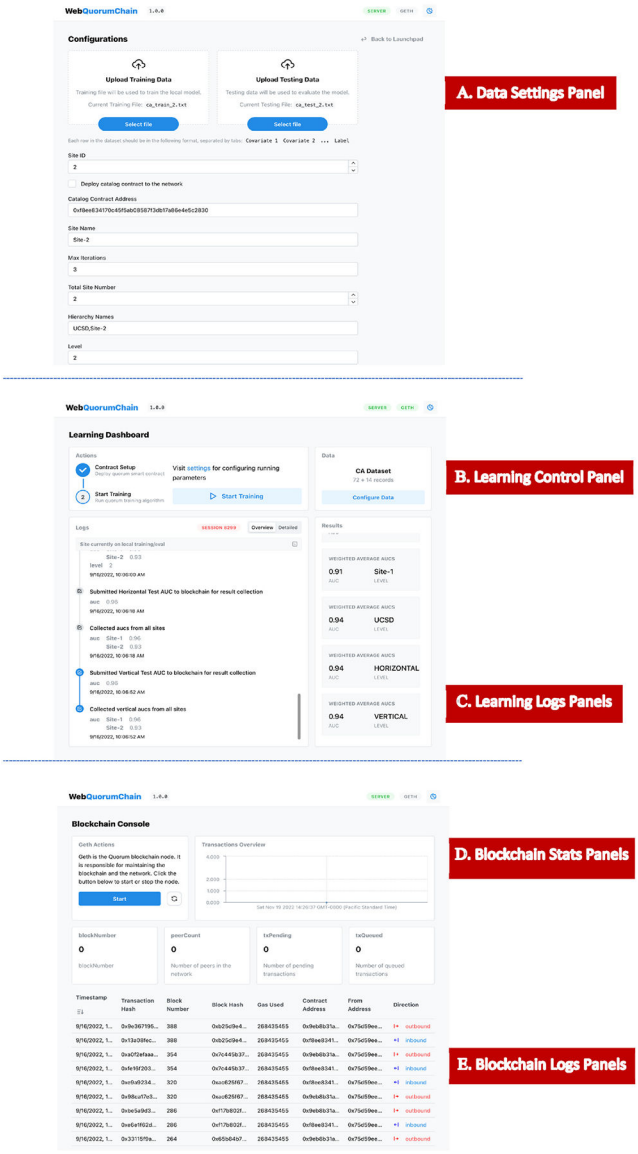

Fig. 4.

Details of GUI design: **(A)** a configuration interface through which users may upload training/testing data via drag-and-drop; **(B)** the learning control panel at a ready state where training session can be started; **(C)** learning logs panel in overview mode such that detailed algorithm logs are filtered out; and exhibited logs presented with previewable model/gradient updates; **(D)** blockchain stats panel including the transaction workload time series plot, current block number, and peer count; and **(E)** blockchain logs that provides insights into the underlying contract transactions; fields like timestamp, block hash, and in/outbound directions are shown.

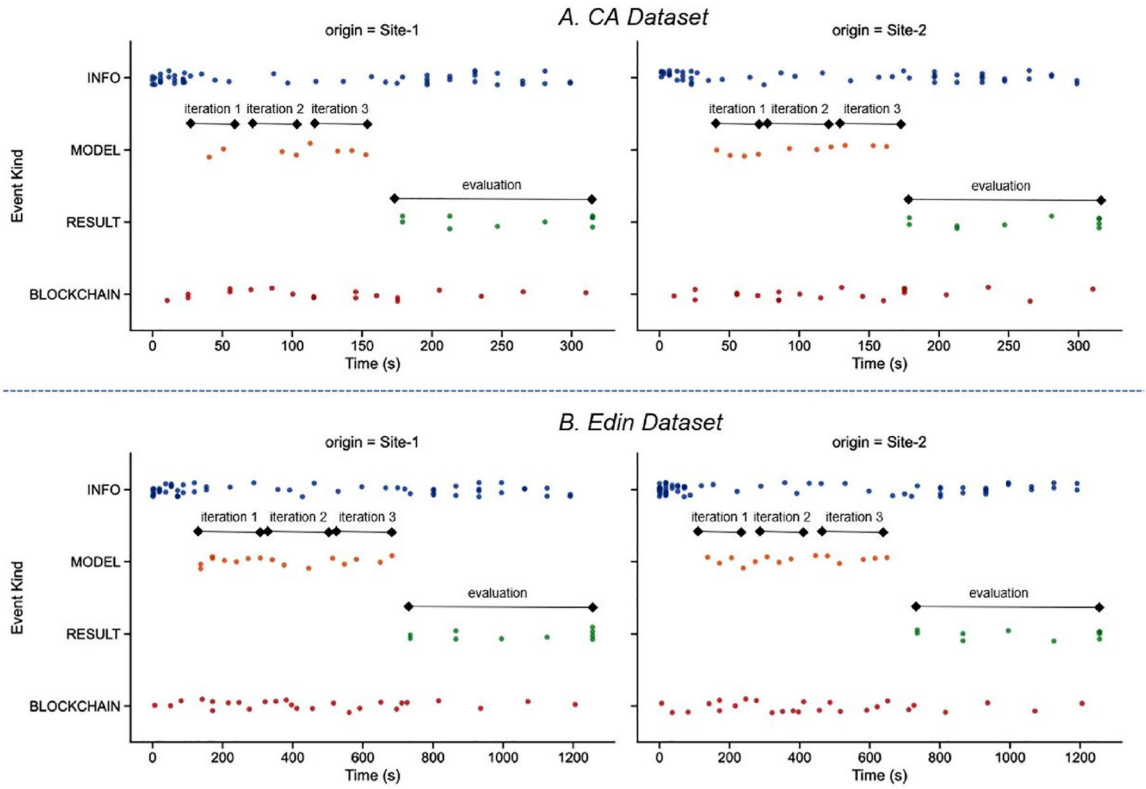

Fig. 5.

Examples of learning events (categorized into INFO, MODEL, and RESULT) and blockchain (BLOCKCHAIN) events for **(A)** CA and **(B)** Edin datasets. We measured all timestamps on the client side of the system. Data points are split into their respective two sites.

Table 1

Description of Cancer Biomarker (CA) and Myocardial Infarction (Edin) datasets including covariates, sample size, class distribution, and outcome.

Dataset	Number of Covariates	Number of Samples	Class Distribution
Cancer Biomarker (CA)	2	141	0.638/0.362
Myocardial Infarction (Edin)	9	1253	0.219/0.781

Table 2

Evaluation results and performance metrics. The metrics include the Area Under the Curve (AUC) for vertical and horizontal ensembles, training time, evaluation time, and total execution time for the entire trial which included network latency (all measured in seconds).

Dataset	Vertical	Horizontal	Training	Evaluation	Total
	AUC	AUC	Time	Time	Time
CA	0.94 ± 0.00	0.94 ± 0.00	107.01 ± 4.37	149.89 ± 3.21	317.03 ± 4.50
Edin	0.96 ± 0.00	0.96 ± 0.00	538.98 ± 0.13	470.00 ± 0.03	1236.00 ± 0.15

Summary Table

What was already known on the topic	• The dual desire to facilitate big data analysis within the constraint of restricted privacy leads to the use of federated learning techniques in healthcare
	• Federated learning may maintain patient privacy, but the centralized design carries innate security risks
	• Decentralized federated learning is a privacy-preserving and security-robust tool to promote cross-institutional learning; however, such frameworks often require complicated setups and user expertise in highly technical fields, rendering the procedure opaque and cumbersome to adopt
What this study added to our knowledge	• Development of an efficient system that integrates the privacy and security promise of the backend QuorumChain algorithm with intuitive, user-friendly front-end GUI, allowing clinical research users to manage and inspect decentralized learning sessions
	• Even with additional system layers, the WebQuorumChain framework still achieves high prediction accuracy and reasonable runtime efficiency, while preserving the strength in privacy/security of QuorumChain