A Blockchain-Empowered Federated Learning System and the Promising Use in Drug Discovery

Xueping Liang, Eranga Bandara, Juan Zhao, and Sachin Shetty

Abstract Federated learning is a collaborative and distributed machine learning model that addresses the privacy issues in centralized machine learning models. It emerges as a promising technique that addresses the data sharing concerns for data-private multi-institutional collaborations. However, most existing federated learning systems deal with centralized coordinators and are vulnerable to attacks and privacy breaches. We propose a blockchain-empowered coordinator-less decentralized, federated learning platform "Rahasak-ML" to solve issues in centralized coordinator-based federated learning systems by providing better transparency and trust. It uses an incremental learning approach to train the model by multiple peers in the blockchain network. Rahasak-ML is integrated into the Rahasak blockchain as its data analytics and machine learning platform. Each peer in the blockchain can establish supervised or unsupervised machine learning models with the existing data on its own off-chain storage. Once a peer generates a model, it can be incrementally/continuously trained and aggregated by other peers through the blockchain using the federated learning approach without requiring a centralized coordinator. The model parameters sharing, local model generation, incremental model training, and model sharing functions are implemented in the Rahasak-ML platform. We discussed the promise of Rahasak-ML machine learning in medicine.

Keywords Federated learning · Blockchain · Medicine · Drug discovery · Big data

X. Liang (✉)
Department of Information Systems and Supply Chain Management, University of North Carolina at Greensboro, 488 Bryan Building, Greensboro, NC 27402, USA
e-mail: x_liang@uncg.edu

E. Bandara · S. Shetty
Virginia Modeling, Analysis, and Simulation Center, Old Dominion University, Norfolk, VA, USA
e-mail: cmedawer@odu.edu

S. Shetty
e-mail: sshetty@odu.edu

J. Zhao
Department of Biomedical Informatics, Vanderbilt University Medical Center, Nashville, Tennessee, USA
e-mail: juan.zhao@vumc.org

W. Charles (ed.), *Blockchain in Life Sciences*, Blockchain Technologies,
https://doi.org/10.1007/978-981-19-2976-2_6

1 Introduction

Federated learning is a new technique for training machine learning models across decentralized participants without accessing any party's private data [1, 2, 3]. It emerges as a promising paradigm for data-private multi-institutional collaborations by distributing the model training to the data owners and aggregating their results, solving the concerns of sharing data [4]. In a federated learning system, the central server (centralized coordinator) coordinates the learning process and aggregates the parameters from local machine learning models trained on each participant's data [5]. Although such a design minimizes the risk of privacy leakage, the centralized coordinator is vulnerable to attacks and privacy breaches, becoming the single point of failure and trust issues.

While blockchain is a technology that offers assurances of reliability and usage transparency in decentralized settings, researchers started to investigate the combinations of the two promising technologies [6, 7]. In this study, we took advantage of blockchain and federated learning and proposed a platform called Rahasak-ML [8]. Rather than using centralized coordinators to aggregate and learn the global model, the Rahasak-ML used an incremental learning technique [9, 10] to continuously train the models by multiple peers in the blockchain network. Each peer in the blockchain manages its local storage and establishes local models [11]. Once a peer generates a model, it can be incrementally trained and aggregated by other peers through the blockchain by using the federated learning approach. Rahasak-ML stores information (e.g., participating clients who generate and aggregate local models, generation times, etc.) into the blockchain ledger that all participating parties can view. It provides a way to audit the system. All actions performed on the model are entirely traceable by each user giving a clear history of all operations and incremental versions that existed. This system adds more transparency to the federated learning system by providing a traceable record of the model development, potentially alerting to adversarial machine learning attempts or fraudulent actions. Rahasak-ML makes the following contributions:

- Integrates federated learning with blockchain to enable model sharing and aggregations without having centralized authority, increasing the transparency, trust, and provenance of the model generation;
- Adds the ability to audit the federated learning system by storing task details (e.g., who generates local models and aggregates them, model generation times, etc.) in the blockchain;
- Offers different functions in the platform that are implemented as independent services (microservices) that are easy to scale and deploy; and
- Introduces a way to integrate the models in smart contracts to predict the output of real-time data.

The chapter is organized as follows. In Sect. 2, we briefly introduce federated learning, blockchain, and the role of these two technologies in drug discovery. In

Sect. 3, we introduce the architecture of federated learning in the Rahasak-ML platform. In Sect. 4, we further explain the training process and the implementation in a medical use case and offer insights into related work. Finally, in Sect. 5, we discuss the future directions and open questions.

2 Overview of Federated Learning and Blockchain

2.1 Federated Learning

Machine learning represents a set of methods that can automatically uncover patterns in data and then use detected patterns to predict future data. Machine learning models show promise in aiding decision-making in healthcare [12, 13] and finance [14]. However, a large, diverse labeled dataset is the key to making a supervised machine learning model broadly effective. Collaborative learning is an efficient way to increase the data size and diversity, via multi-institutional data sharing for the training of a single model [4]. The current approach to achieving collaborative learning requires sharing the data with a third party to train a global model, such as using data repositories for different purposes (*Fiscal Service Data Registry*, [15]). However, this centralized approach presents many issues, such as high costs for data transmission and storage, security and privacy at high risk, lack of auditing, data ownership, and restrictions of data sharing, e.g., the Health Insurance Portability and Accountability Act (HIPAA) regulations in healthcare [16].

To address these security and privacy issues, a decentralized machine learning approach, i.e., federated learning [17, 18], has been proposed to build a shared machine learning model without storing or having access to any party's private data. In federated learning, the central server coordinates the learning process and aggregates the information from multiple participants (i.e., referred to here as "parties") in a decentralized manner while keeping each participant's raw data private. Each party downloads the global model parameters from the central server at each iteration, locally trains it with their private/local dataset, and sends each of their local model parameters to the central server for aggregation. Then, the central server gathers all the local model parameters, aggregates them, and updates the global model parameters for the next iteration. This learning process continues until pre-defined termination criteria are met. For example, if the maximum number of iterations is reached, or if the model accuracy is greater than a threshold, the learning process is finished and will exit automatically.

2.2 Barriers and Challenges in Drug Discovery

Drug discovery involves identifying potential new medicines, which involves and requires the knowledge of a wide range of scientific disciplines, such as biology, chemistry, and pharmacology. Developing a new drug is a complex, lengthy, and costly process, entrenched with a high risk of uncertainty that a drug will succeed. The drug development pipeline included multiple stages, from identifying targeted therapeutic agents to clinical trial designs, including Phases I, II, and III. Each stage is critical but faces challenges, such as insufficient knowledge about the underlying mechanisms of disease, the heterogeneity of patients who have diverse clinical phenotyping and endotyping, a lack of targets and biomarkers, small or biased samples in clinical trials, and regulatory challenges [19]. These hurdles create barriers to the development of the drugs, leading to increased costs and time, thus increasing the risk of failure. To minimize these challenges, researchers moved toward computational approaches to accelerate pipeline, such as using high-throughput virtual screening and molecular docking to reduce the number of compounds that need to be screened experimentally [20]. However, these approaches have inaccuracy and inefficiency problems. Therefore, new methods and computing technologies to automate analytical model building for pharmaceuticals are needed and could transform drug discovery.

Today, the advances in high-throughput approaches to biology and disease present opportunities to pharmaceutical research and industry [21]. For example, multi-omics ranging from genome, proteome, transcriptome, metabolome, and epigenome are generated at unprecedented speed, improving the capabilities of systematically measuring and mining biological information. In addition, widely adopted electronic health records (EHR) and smart technologies capture detailed phenotypic patterns, allowing researchers to monitor patient outcomes and study medication treatments. The booming of such "big data," including omics, images, clinical characteristics, social/environmental information, and literature, has driven much of researchers' interest in harnessing machine learning to analyze and uncover novel findings and hidden patterns from the massive data [22, 23, 24].

Machine learning and deep learning are fundamental branches of artificial intelligence (AI), which refer to computer systems' ability to learn from input or past data. AI has achieved successful applications in many domains, such as imaging detection and natural language processing. Recently, AI algorithms have been increasingly being applied in all stages of drug discovery, including screening chemical compounds, identifying novel targets [25], examining target–disease associations [26], improving the small-molecule compound design and optimization, studying disease mechanisms [27], evaluating drug toxicity and physicochemical properties [28], predicting and monitoring the drug response [29], and identifying new indications for an existing drug, known as drug repositioning. Moreover, researchers utilized machine learning models to optimize the clinical trials, such as estimating the risks of clinical trials more accurately [30] and improving the patient pre-screening process, as well as approaches to feasibility, site selection, and trial selection [31].

From a machine learning viewpoint, it is desirable to have large and diverse data to inform model training, but access to data remains a challenge in drug

discovery. Several public databases contain millions of biological assay results, such as ChEMBL [32] and PubChem [33], which can provide input for machine learning models to retrieve training models and then predict biological activities or physical properties for drug-like molecules. However, the data only represents a small fraction of what has been measured, which might bias the machine learning models and affect the model reliability and reproducibility [34]. Furthermore, many larger datasets are proprietary to pharmaceutical companies or publishers and are not publicly and freely available. To overcome the barriers, researchers seek federated learning to solve data acquisition and data bias problems faced by AI drug discovery by keeping confidentiality and customizing models for users [35].

Federated learning is a new machine learning paradigm where multiple sites collaboratively learn a shared machine learning model while keeping all the training data on a single site [2]. Developing federated health AI technologies are essential and highly demanding in medicine [13]. Examples include the European Union Innovative Medicines Initiative's (https://www.imi.europa.eu/) projects for privacy-preserving federated machine learning. Chang et al. explored data-private collaborative learning methods for medical models for image classification [36]. Xiong et al. [37] proposed using a federated learning work in predicting drug-related properties. The architecture of federated learning is that each participating pharma company (peer) will locally train the model without sharing the training data. Each peer only encrypts and uploads the model updates, and a coordinator server aggregates all the updates from the local client and broadcasts the latest shared global model to them. Thus, individual pharma companies will be able to fine-tune the machine learning model and effectively tailor it to their specific field of inquiry, with the individual research data remaining confidential.

2.3 Challenges in Federated Learning

While the federated learning process has significant improvements to minimize the risk of privacy leakage by avoiding storing raw datasets to a third party, it still presents some major vulnerability issues in the model architecture and the training process.

- First, the central server for coordinating a shared and trained global model presents the single point of failure and trust issues. A malicious behavior or malfunction from the central server could bring inaccurate global model parameters updates, which would misrepresent the local model parameters update sent by the parties. Therefore, decentralization of the entire federated learning process was necessary.
- Second, during the learning process, malicious parties could send manipulated local model parameters to the central server, affecting the global model parameters. If such malicious local parameters are not detected or removed before aggregation, they will compromise the global model and lower the overall model accuracy [1, 38]. Some studies [39] have proposed approaches to verify model parameters, but they mainly rely on the data sample size and the computation time, which could be easily altered by malicious participants to avoid detection.

- In addition, these studies do not address the quality of the data sample that would affect the accuracy and the convergence analysis of the federated learning process. A more difficult malicious behavior, colluding attack, has shown the vulnerabilities of existing defenses based on Sybil [40]. Thus, it is essential to note that verifiable local model parameters update is important for the accuracy of the global model parameters.

2.4 Blockchain Benefit for Federated Learning

Blockchain provides a shared digital ledger that records data in a public or private peer-to-peer network. It guarantees a decentralized trust system without involving trusted third parties. Multiple partners (nodes) can exist in the blockchain network, and each partner (node) has a copy of the data being maintained [41]. The data on the blockchain are organized into blocks. A block contains a set of records (transactions). Each block is linked to its previous block by containing the previous block's hash in its header. If someone was to tamper with the contents of one block, then all blocks in the blockchain following that block would be invalidated.

Depending on the type of access and from where the nodes that support the blockchain are selected, there are two primary types of blockchains: permissionless and permissioned. Permissionless blockchains deal with entirely untrusted/byzantine parties; examples are Bitcoin, Ethereum, and Rapidchain. Permissioned blockchains deal with trusted/known parties; examples are BigchainDB, Hyperledger, and HbasechainDB. Many blockchains, such as Bitcoin, are used for cryptocurrencies. For example, Ethereum and Hyperledger support different transaction storage models related to other business or e-commerce activities. Recently, blockchain has quickly been applied to other areas, including the healthcare and drug industry [42, 43]. For example, studies have integrated blockchain with EHRs, to allow the different stakeholders to manage EHR transparently while guaranteeing fairness and usage (records access) consent [44].

To address the challenges of federated learning, we propose integrating blockchain with federate learning to replace the centralized coordinator. The blockchain network can be deployed among different peers, and the peers can train machine learning modes with the data on their own local storages (e.g., off-chain storage). Then the local models generated by different peers can be aggregated/averaged into a global model using the federated learning approach without using a centralized coordinator. In blockchain-enabled federated learning systems, the model parameter sharing, local model generation, incremental model training, and model sharing functions can be implemented with smart contracts. All federated learning tasks happening in the system (e.g., generate local models and aggregate them) and stored in the blockchain ledger are viewed by all participating parties. It provides a means to audit the system and adds more transparency to the federated learning process. Once local models are generated, these models can be integrated into blockchain smart contracts (e.g., a program that directs client requests to the blockchain) to predict real-time data.

This system adds more transparency to the federated learning system by providing a traceable history of the model development, potentially alerting to adversarial machine learning attempts or fraudulent actions.

2.5 The Benefits of Blockchain-Empowered Federated Learning for Drug Discovery

The blockchain-enabled federated learning enhanced such infrastructure by decentralizing the architecture further and making the training process and model sharing more transparent and traceable. As a result, hospitals, institutions, and drug companies can achieve an accurate and generalizable model; more sites contribute their local insights while remaining in full control and possession of their data. This approach allows complete traceability of data access, limiting the risk of misuse by third parties. There is a consortium of pharmaceutical, technology, and academic partners, the Machine Learning Ledger Orchestration for Drug Discovery (MELLODDY, https://www.melloddy.eu/), that uses deep learning methods on the chemical libraries of ten pharma companies to create a modeling platform that can more quickly and accurately predict promising compounds for development, all without sacrificing the data privacy of the participating companies. Specifically, the benefits of a blockchain-empowered federated learning system are as follows:

- Entails training algorithms across decentralized sites or devices holding data samples without exchanging those samples.
- Small pharmaceutical companies and research institutions would achieve accurate, less biased models by gaining insights from other sites containing diverse data.
- Provides a platform with more transparency, trust, and provenance for model training and sharing.
- Provides the ability to audit the system and make sure local data and models are traceable. For example, the task information related to who generates models, aggregate parameters, and model generation time would be recorded in the blockchain.
- Offers flexibility with connecting more participating sites and devices.
- Provides the ability to process real-time data.

3 The Rahasak-ML Platform

3.1 Overview

The Rahasak-ML platform integrates federated learning with blockchain to enable model sharing and model training without having a centralized coordinator, which keeps the data private [45, 46]. The proposed platform has been implemented on

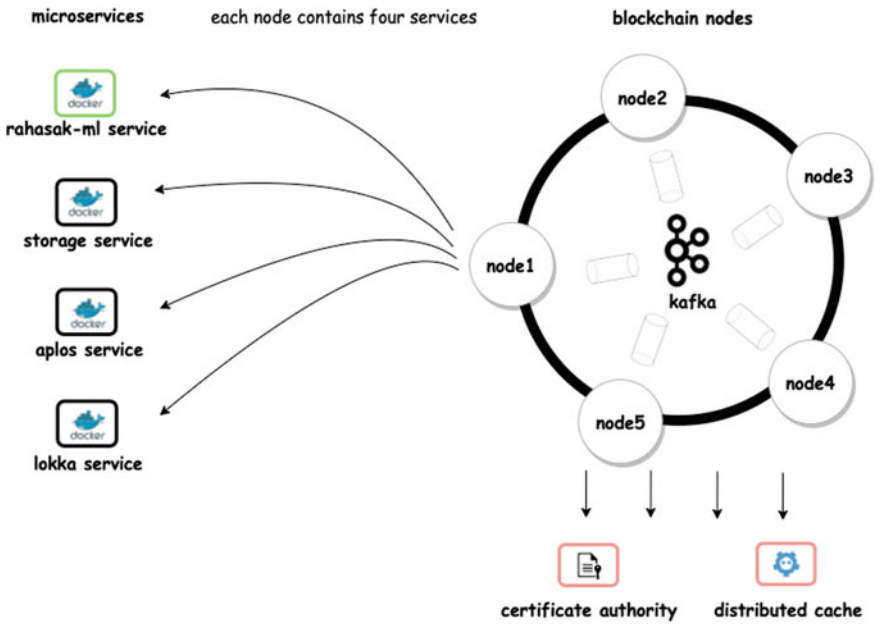

Fig. 1 Rahasak-ML platform's microservices-based architecture. Each blockchain node contains four services: Rahasak-ML service, Storage service, Aplos service, and Lokka service

top of the Rahasak blockchain [5], a highly scalable blockchain system for big data. The architecture of the Rahasak-ML federated learning environment is discussed in Fig. 1.

Its proposed platform has been designed with microservice-based distributed system architecture [47]. In Rahasak-ML, all the functionalities are implemented as independent microservices. These services are Dockerized [48] and available for deployment using Kubernetes [49]. The following are the main services/components of the Rahasak-ML platform:

- Storage service: Apache Cassandra-based block, transaction, and asset storage service [50].
- Aplos service: smart contract service implemented using Scala functional programming language and Akka actors [51].
- Lokka service: block creating service implemented using Scala and Akka streams [52].
- Distributed message broker: Apache Kafka-based distributed publisher/subscriber service used as consensus and message broker platform in the blockchain, Rahasak-ML service federated machine learning service.
- Distributed cache: Etcd-based distributed key-value pair storage (open-source distributed key-value storage system).
- Certificate authority: certificate authority that issues certificates for peers and clients.

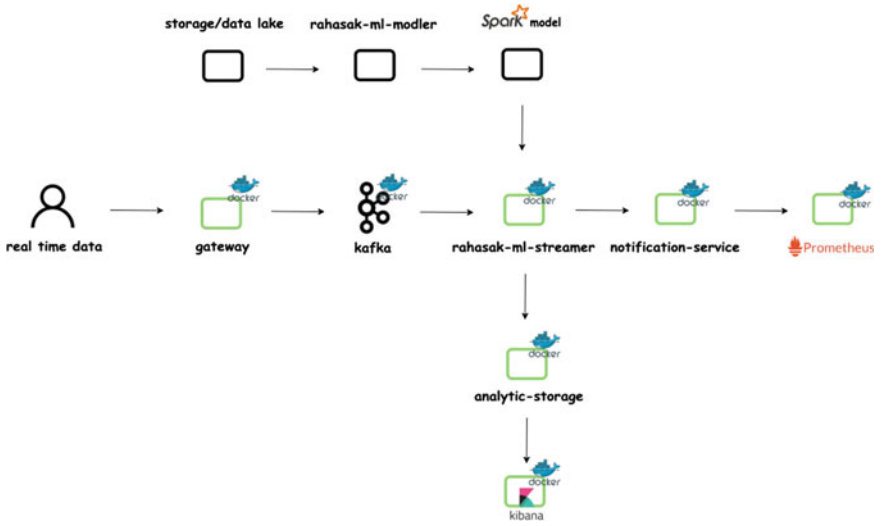

Fig. 2 Rahasak-ML service architecture. Each blockchain peer has its own Rahasak-ML service. Machine learning models will be generated with the data on each peer's off-chain storage

Each peer in the network has its own off-chain storage for storing the raw data. The hash of these data is published to a blockchain ledger and shared with other peers. The blockchain storage on the Rahasak-ML platform keeps all its transactions, blocks, and asset information (hash of the data in off-chain storage) on Cassandra-based Elassandra Storage (https://github.com/strapdata/elassandra). It exposes Elasticsearch application programming interfaces [53] for transactions, blocks, and assets on the blockchain. Each peer in the blockchain can establish supervised or unsupervised machine learning models with the existing data on its own off-chain storage. Once a peer generates a model, it can be incrementally trained and aggregated by other peers through the blockchain by using the federated learning approach. The model parameter sharing, local model generation, incremental model training, and model sharing functions are implemented in the Rahasak-ML platform. Once machine learning models are generated, these models can be integrated into blockchain smart contracts to predict real-time data. Figure 2 shows the architecture of the Rahasak-ML services in a single blockchain peer.

Each peer in the network runs its own Rahasak-ML service. The Rahasak-ML service contains the following components. All these components are Dockerized and deployed via Kubernetes.

- Storage Service.
- Rahasak-ML Modeler Service.
- Rahasak-ML Streamer Service.
- Gateway Service.
- Apache Kafka.

3.2 Key Components

3.2.1 Storage Service

Each peer in the Rahasak-ML platform has two storage mechanisms: off-chain and on-chain storage. Both are built with Apache Cassandra-based Elassandra storage. Off-chain storage stores the data generated by the peers. The hash of these data is published to on-chain storage and shared with other peers. Blockchain keeps all its transactions, blocks, and asset information on this on-chain storage. The on-chain storage in each peer is connected in a ring cluster architecture. The data saved in one node will be replicated with other nodes via this ring cluster. After executing transactions with smart contracts, the state updates in a peer are saved in Cassandra storage and distributed with other peers, Fig. 3.

Blockchain can keep any data structure as blockchain assets since it uses Cassandra as the underlying asset storage. As a use case of Rahasak-ML, the authors built a blockchain-based secure NetFlow network packet storage and network anomaly detection (e.g., network attack) service. It stored actual NetFlow packet data in the blockchain peers' off-chain storage. The hash of the data was stored in the on-chain storage as a blockchain asset. The smart contracts in the blockchain parsed the NetFlow packets coming through the router and stored them in the blockchain storage. Rahasak-ML can build machine learning models with the data saved in the peers' off-chain storage. In federated learning scenarios, the local models are stored in the off-chain storage. The hash of the model and storage Uniform Resource Identifier (URI) of the model are stored in on-chain storage and distributed with other peers.

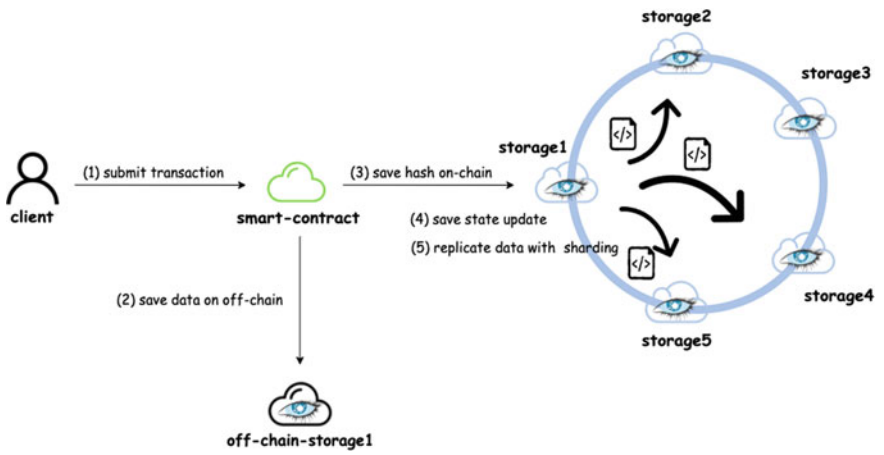

Fig. 3 Rahasak-ML storage service architecture. Each peer comes with two types of storage: on-chain storage and off-chain storage. Off-chain storage stores the actual data generated by the peers. The hash of these data is published to on-chain storage and shared with other peers

3.2.2 Rahasak-ML Modeler Service

Rahasak-ML modeler service is responsible for building the machine learning model by analyzing the peers' off-chain storage data. It supports building both supervised (e.g., Decision Tree, Random Forest, and Logistic Regression) and unsupervised (e.g., K-Means and Isolation Forest) machine learning models. To build a new machine learning model, the first step is training, which uses a dataset as an input and adjusts the model weights for the model accuracy. The second step is testing, which takes in an independent dataset for testing the accuracy.

Figure 4 shows the overall flow of these steps, which is performed by the Rahasak-ML Modeler service. Once the prediction model is built and trained by the Rahasak-ML Modeler service, it can be used to perform tasks on new data. In a federated learning environment, each peer in the network will continuously train the generated model with the data on their off-chain storage using an incremental training approach. The continuous model training can be done with Spark Streams [54], such as real-time training libraries. More information about the continuous model training is discussed in Sect. 4.

Following the model generation, the training models can be used in smart contracts to predict/cluster real-time data. For example, Rahasak-ML Modeler can be used to build the Isolation Forest and K-Means-based models to detect outliers of network traffic data. This model will split network data into two clusters: normal network traffic and suspicious (attacks) network traffic. Once local models are built and aggregated, the models can be integrated into blockchain smart contracts to predict

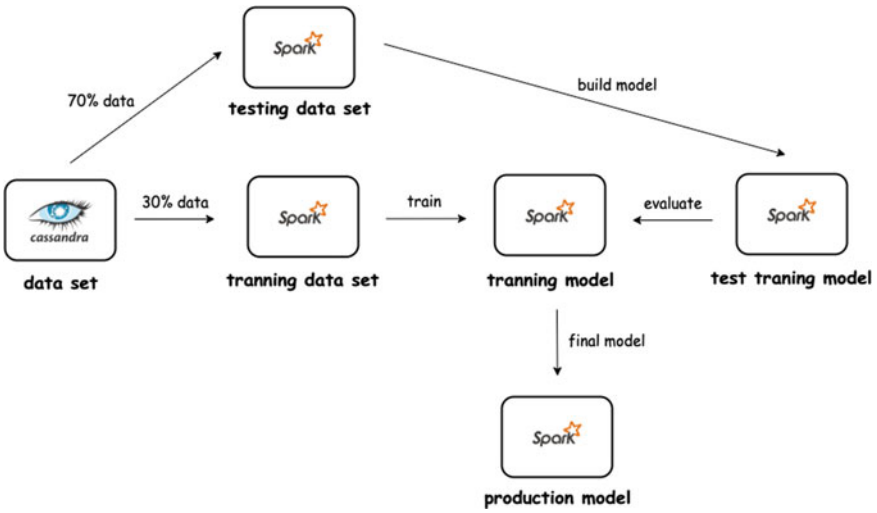

Fig. 4 Rahasak-ML modeler service architecture. Seventy percent of the data is used to train the model, and 30% will be used for testing

the real-time network data. When new network packets come to the blockchain, smart contracts can use the model and predict the category (normal or suspicious) of real-time network traffic.

3.2.3 Rahasak-ML Streamer Service

Rahasak-ML streamer service clusters the real-time data with the machine learning models built by the Rahasak-ML Modeler service. It uses blockchain smart contracts [55, 56] to run the machine learning model with the newly generated data. Smart contract functions are written to use the model and predict the cluster output. This service consumes real-time data via Kafka (e.g., Kafka Streams and Spark Streams). For example, in the previously mentioned network traffic analysis scenario, the Rahasak-ML streamer will consume real-time network packets via Apache Kafka and run through the model built by the Rahasak-ML Modeler service. It will decide the clustering output (normal and suspicious) of the new packets, and if a suspicious packet is found, it will publish the entry to a notification service. Alerts will be generated, notifying experts via notification dashboards (e.g., Prometheus and Grafana), as shown in Fig. 5.

3.2.4 Gateway Service

When analyzing real-time data, the Gateway service is used as the entry point to the Rahasak-ML platform. It fetches (or pushes from other services) real-time data from various data sources, such as log fields, NetFlow, TCP, UDP, and database. For example, the gateway service can receive real-time network traffic data via NetFlow. Once data arrive, they are prepared (by removing noise, parsing the data, etc.) and published to the Rahasak-ML streamer service via Kafka as JSON encoded objects.

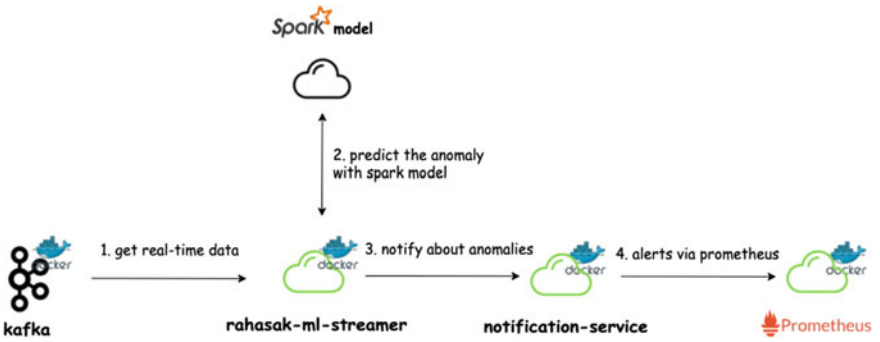

Fig. 5 Rahasak-ML streamer service architecture. Streamer service clusters the real-time data with the machine learning models built by the Rahasak-ML Modeler service

Fig. 6 Gateway service architecture. Gateway service is used as the entry point to the Rahasak-ML platform. It fetches (or pushes from other services) real-time data from various data sources such as log fields, NetFlow, TCP, UDP, and database

When the platform receives NetFlow packets, it extracts relevant fields, aggregates them, constructs a JSON object, and forwards it to the Rahasak-ML streamer service via Kafka, as shown in Fig. 6.

3.2.5 Kafka Message Broker

Apache Kafka is the consensus and message broker service in the Rahasak-ML blockchain environment. The authors use a Reactive Programming and Reactive Streaming model [57] where the services published events/messages with Kafka. The events will be subscribed by relevant services and take corresponding actions. The real-time data that come through the gateway service are published into Kafka first. Then Rahasak-ML streamer service consumes them and runs with the model, which is built by the Rahasak-ML Modeler service, as shown in Fig. 7.

4 Rahasak-ML Federated Learning Process

4.1 Overview

Rahasak-ML proposed a blockchain-based federated learning approach to build and share the models. With this approach, model generation, incremental model training, model aggregation, and sharing can be done without having centralized authority. Federated learning approaches increase privacy but still rely on centralized control

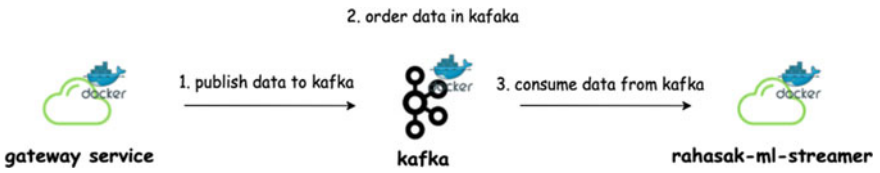

Fig. 7 Rahasak-ML message broker architecture. Apache Kafka is the message broker of the Rahasak-ML platform. Each microservice communicates with other services via Kafka

to manage the process. Centralized control can be compromised, causing a potential weak link in the system and a lack of trust in the authority that owns the centralized server [2]. A blockchain-based decentralized system provides a logical ruleset that all participants are aware of and agree on, allowing participants to audit operations to ensure that all parties follow the rules. It improves the ability to audit and adds more transparency to the federated learning process. Each peer in the blockchain network incrementally trains the machine learning models with the data on its own local off-chain storage. Once all peers (or a majority of peers) are trained, the finalized model details will be integrated into a block and published to the other peers in the network by the block-generating service of Rahasak-ML (Lokka service).

4.2 Incremental Training Flow

Assume a scenario where blockchain nodes are deployed in three companies, Companies A, B, and C. The blockchain is configured to store the data related to network traffic. Each company has its own off-chain storage, which stores the actual network traffic data. The hash of the network traffic data is published into the blockchain ledger. First, the Lokka service (that generates blocks) creates a genesis block with the incremental learning flow and the model parameters, as shown in Algorithm 1. Each peer in the network has its own Lokka service. The Block Creator is determined in a round-robin distributed scheduler. Consider the scenario in Fig. 8, which has three Lokka services, and assume that the first block is created by Lokka A, the second block will be created by Lokka B, and Lokka C creates the third block. This process is repeatedly performed to generate future blocks.

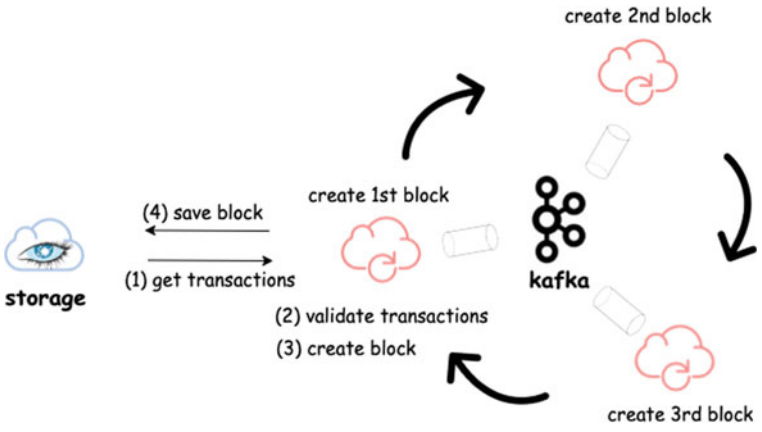

Fig. 8 The block creator is determined in a round-robin distributed scheduler. The block approval process is performed via the federated consensus implemented between Lokka services

Algorithm 1 Training pipeline initialization

1 INITIALIZE TRAINING PIPELINE:
2 Choose Lokka node l_i by the round-robin scheduler to initialize the training pipeline
3 Find available blockchain peers p(1, ..., n) from the distributed cache
4 Define incremental learning flow based on each peer join time (ttl) to the network
5 Define ML model training parameters and algorithm information
6 GENERATE GENESIS BLOCK:
7 Create genesis block b_i with model parameters and incremental training flow
8 Save b_i in ledger and broadcast it to other peers in the network

Incremental learning flow defines the order of the model training process. When defining a learning flow, the Lokka service finds the existing nodes in the network via distributed cache service in the Rahasak-ML. Rahasak-ML uses Etcd distributed key/value pair storage as the distributed cache and service registry. Etcd stores the health information of the blockchain nodes in the network. When a blockchain node is added to the network, it registers a node name (with meta-information) in the Etcd with the time to live (TTL) key. The node will periodically update this TTL key (before TTL reach) to prove it is alive. If a node is dead/exits, the TTL key will automatically be removed from Etcd. By using the TTL keys in Etcd, other nodes can know the available nodes in the network. The order of the incremental learning flow is decided by the TTL key created timestamp in the Etcd. This timestamp defines the blockchain nodes' added time to the network. Assume the Lokka service has the incremental learning flow as A→B→C based on the TTL keys in the Etcd registry. This flow represents that peer A will produce a model, and then this model will be incrementally trained by peer B and then peer C. Once a miner node publishes the genesis block with model parameters and incremental flow to the blockchain ledger, other peers take the block and process it according to the defined flow, as shown in Fig. 9.

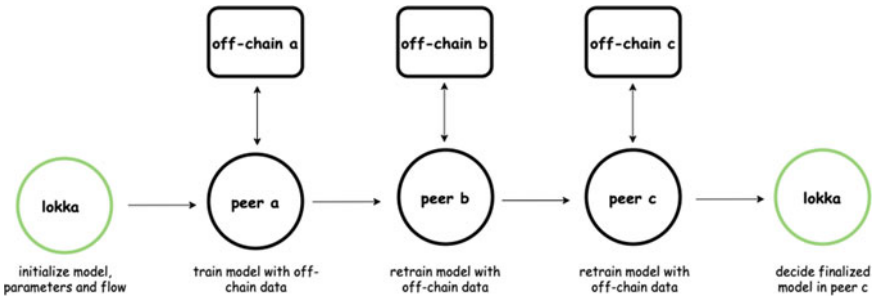

Fig. 9 Rahasak-ML training pipeline. Once a miner node publishes the genesis block with machine learning model parameters and incremental flow to the blockchain ledger, other peers take the block and process it according to the defined flow

According to the incremental learning flow, first, peer A generates the anomaly detection model with the data on the off-chain storage based on the model parameters in the genesis block. Then it saves the model built on its off-chain storage. The actual model is not published onto the blockchain ledger or any central storage. The hash and URI of the built model saved in the off-chain storage are published to the blockchain ledger as a transaction. Then peer B starts to incrementally train the model built by Peer A. To achieve this, peer B fetches the model built by peer A from peer A's off-chain storage using the given URI. Then it trains that model with the data on peer B's off-chain storage. This training model will be saved on peer B's off-chain storage, and peer B will publish the model hash and off-chain storage URI of the model to the blockchain ledger as a transaction. Next, peer C will incrementally train the model trained by peer B and publish the details to the blockchain ledger as a transaction, as shown in Algorithm 2.

Algorithm 2 Incremental training flow

```
1 Wait till publishing genesis block b_i
2 for each peer p= 1, ..., n do
3    INCREMENTAL MODEL TRAINING:
4    if p == 1 then
5       Fetch genesis block b_i from the ledger and get model training parameters
6       Build initial model with the data in the off-chain storage
7    else
8       (assume p=x)
9       Fetch ML model from the peer p=x−1 off-chain storage
10      Incrementally train that model with the data on the peer
          p=xoff-chain storage
11   end
12   Save built ML model in off-chain storage
13   PUBLISH MODEL UPDATES:
14   Create transaction t_i with ML model hash and off-chain storage URI of the model
15   Publish t_i to the ledger
16 end
```

The flow of the incremental learning process is described in Fig. 10.

4.3 Finalizing Model

Assume all three companies (or a majority of the companies) incrementally train the prediction model and publish the model hash and URI to the blockchain ledger as a transaction. Then Lokka service takes these transactions and creates a block with finalized model details with the final model stored in the peer C's off-chain storage. Currently, the model trained by the last peer (peer C in this scenario) is identified as the finalized model. In future work, there are plans to determine the

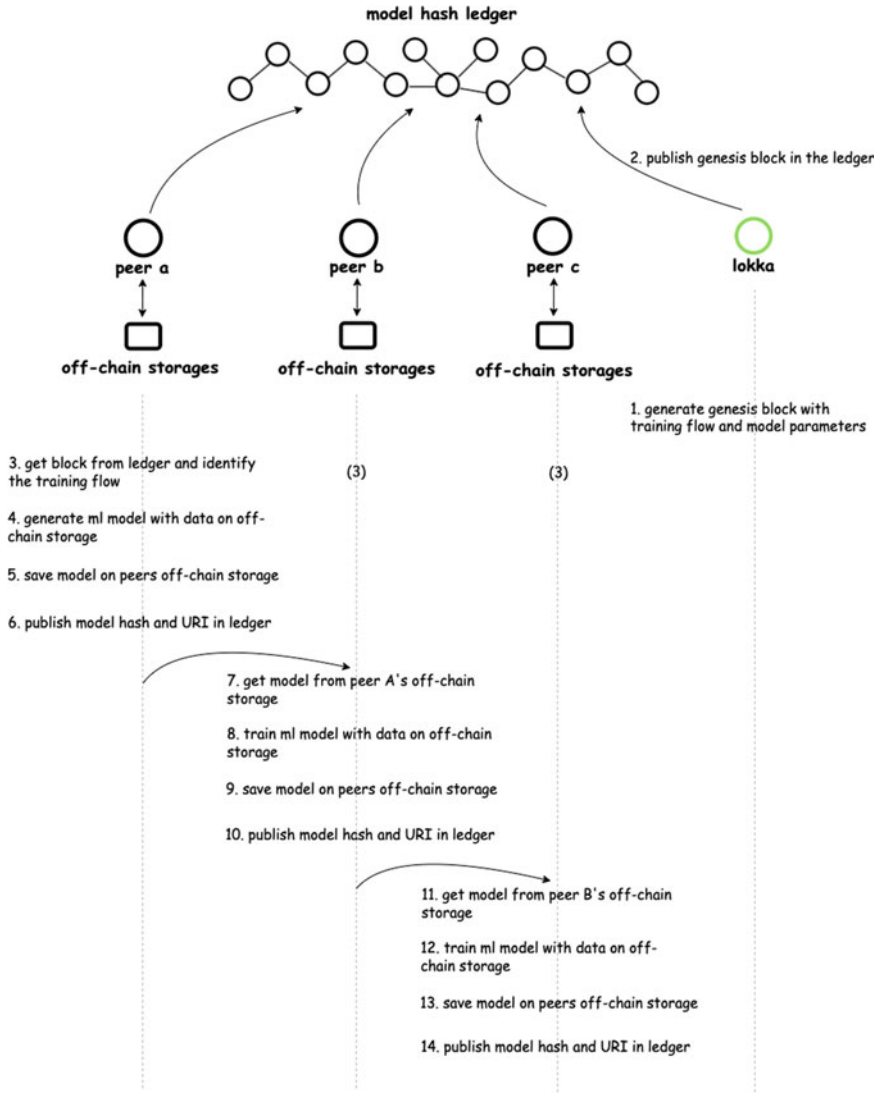

Fig. 10 Rahasak-ML incremental training flow. Each peer trains the model with the data on the off-chain storage. The state update in each training step will be published to the blockchain ledger

finalized model by evaluating the accuracy of each model trained by its peers. Lokka service includes the URI of peer C's off-chain storage (which stores the final model) and model training transaction details into the block. Then Lokka service saves the generated block in the ledger and distributes it to other peers. Once the peers receive the new block, they validate the learning process with the transactions in the block. If the process is valid, peers fetch the final model stored in peer C's off-chain storage

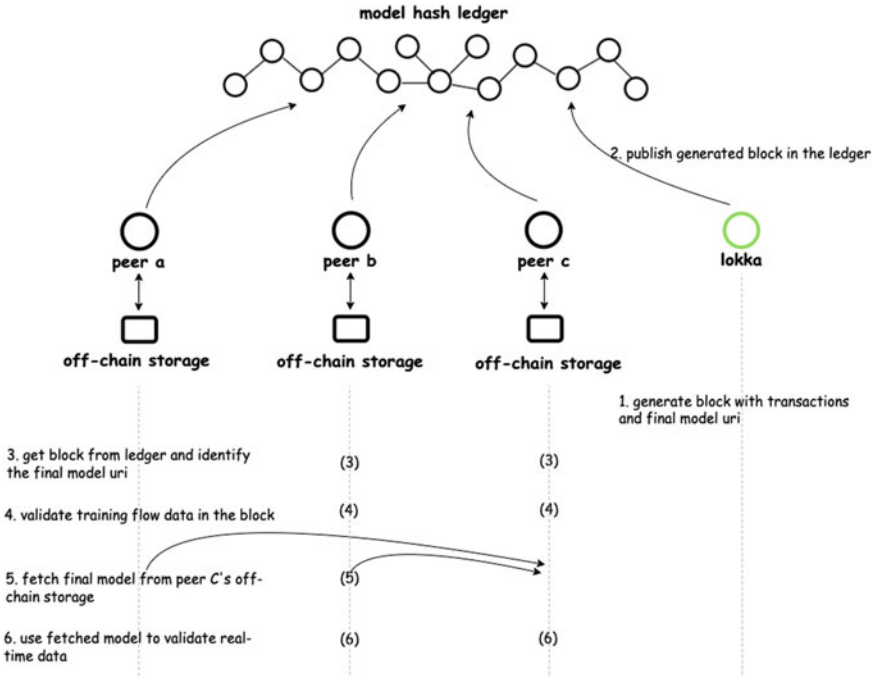

Fig. 11 Rahasak-ML finalizes the machine learning model. The final model will be decided by the Lokka service when generating the final block

via the given URI in the block. The incrementally trained model sharing process is described in Fig. 11. Once the finalized model is fetched, it can be used in smart contracts for prediction.

For the Lokka service to decide the final model, the majority of the nodes in the network need to complete the incremental learning process. If there are five nodes in the federated learning flow, three of these nodes need to finish the incremental learning flow to decide on the finalized model. Once the Lokka service has generated the block with the finalized model details, other Lokka services in the network need to approve that block. When approving, they first validate the transactions in the block. If all transactions in the block are valid, it gives a vote for the block (mark block as valid or invalid), as shown in Algorithm 3. To handle the voting process, the Lokka service digitally signs the block hash and adds the signature to the block header. When the majority of Lokka services submit the vote for a block, that block is considered as a valid/approved block.

Algorithm 3 Choose final model

1 Wait till the majority of the peers complete the incremental training process in the training pipeline
2 DEFINE FINAL MODEL:
3 Get transaction list t(1, ..., n) from ledger
4 Find the transaction t_n which submitted by the last peer p_n (model trained by the last peer identified as the finalized model)
5 Create block $b_{i+1}$with final model URI, model hash and transactions
6 Save blockbi+1in the ledger and broadcast it to other peers
7 UPDATE FINAL MODEL:
8 **for** each peer p= 1, ..., n **do**
9 Fetch blockbi+1from ledger
10 Verify transactions in the block
11 If the block is valid, fetch final ML model from peer p_n
12 **end**

4.4 The Use Case of Blockchain-Empowered Federated Learning in the Medical Field

Blockchain-empowered federated learning provides a secure, transparent, and privacy-preserving computing solution for building accurate and robust predictive models using biomedical data from multiple parties (e.g., institutions, hospitals, and drug companies). It does not need a centralized server to collect data from various parties, which is often difficult to share due to HIPAA. As a proof of concept, the authors built blockchain-empowered federated learning for diagnosing acute inflammation of the bladder. We used inflammation of the bladder health dataset [58] and chose logistic regression as the prediction model. In this use case, a blockchain network is deployed at five peers (five hospitals). Each peer has its own dataset and trains and validates a local logistic regression model. Finally, these local models are averaged. The loss and accuracy of the models were computed, and block generation time was measured in the blockchain-enabled federated learning system. The preliminary study can be extended to more scenarios in medicine and drug discovery use cases.

4.4.1 Federated Model Accuracy and Training Loss

In the federated learning scenario, the model was trained with 1000 iterations. A copy of the shared model is sent to all peers participating in the iteration. Each peer trains its own model with its own dataset locally. Each local model is improved in its own direction. Then total loss and accuracy were computed as shown in Fig. 12. Figure 13 shows how the total training loss varies at different peers in each iteration.

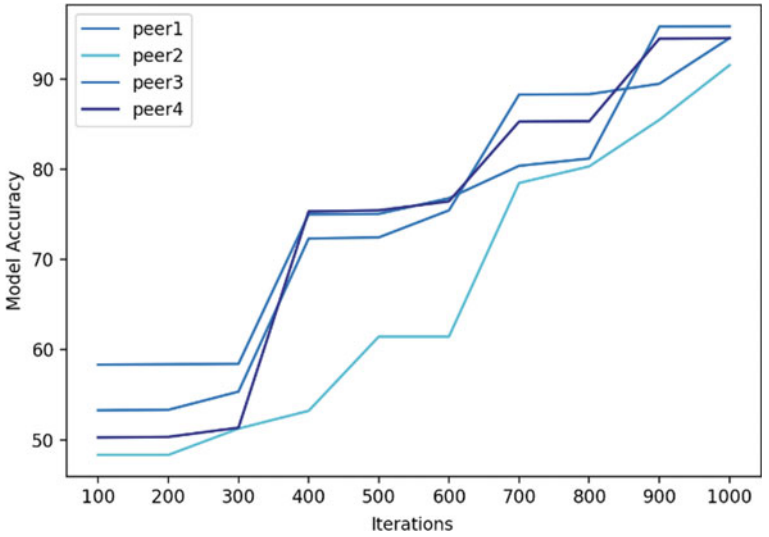

Fig. 12 Federated model accuracy in different peers

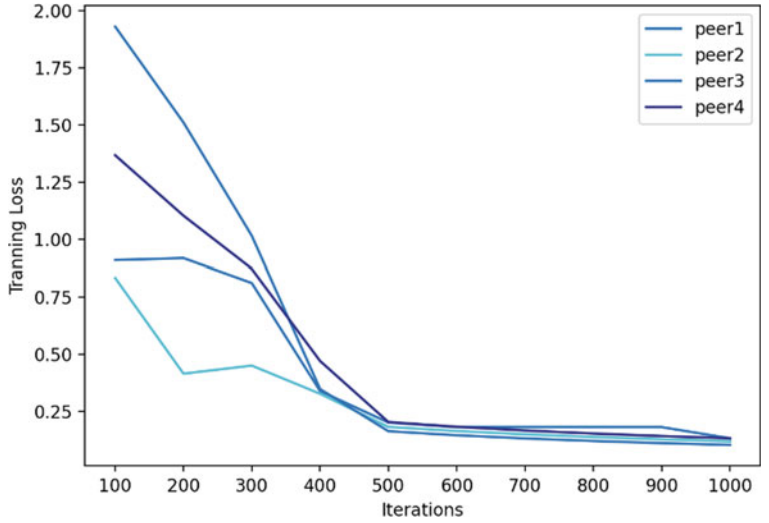

Fig. 13 Federated model training loss in different peers

4.4.2 Block Generation Time

Block generation time was measured in the Bassa-ML federated learning system with a different number of blockchain peers (up to 7). Figure 14 shows the average block generation time when having a different number of blockchain peers in the network.

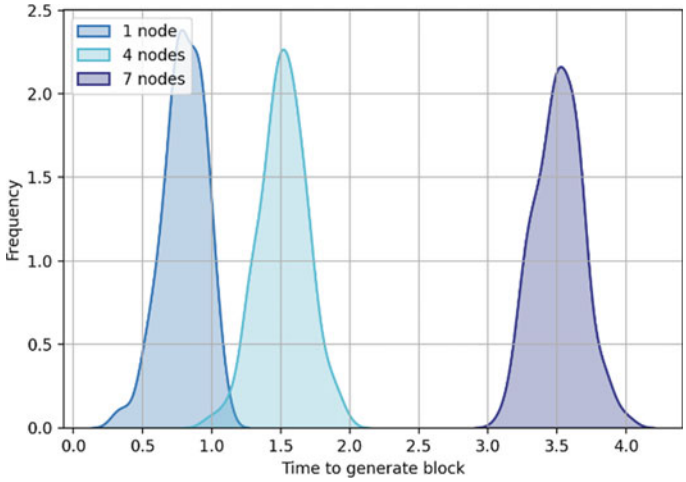

Fig. 14 Average block generation time

Each experiment was repeated 100 times in this evaluation—each with different peer sets—and average values were plotted. When adding peers to a cluster, each peer needs to validate transactions in the block and recalculate the block header. Accordingly, block generation time increases as peers are added.

5 Future Directions

The proposed platform took full advantage of blockchain and AI technologies to provide a more efficient and secure solution with a promise to accelerate the research in medicine. The following is a summary of future work and several open directions.

The proposed system overcomes several key concerns faced in centralized systems. While individual nodes (peers) develop local models based on their local data, the resulting models and parameters are shared through the blockchain platform. The model parameter sharing, local model generation, model averaging, and model sharing functions are implemented with smart contracts implemented on the platform. Most recently, the Rahasak-ML federated learning system was integrated into Rahasak blockchain version 3.0. The following are features of the Rahasak-ML platform that are planned to release in the future:

- Decide the finalized model by evaluating the accuracy of each model trained by the peers.
- Support more supervised/unsupervised machine learning algorithms with Rahasak-ML.
- Automate the deployment of the Spark cluster in Rahasak-ML with Kubernetes.
- Integrate TensorFlow-Federated libraries into Rahasak-ML.

5.1 Data Heterogeneity

Medical data are particularly diverse—in terms of the variety of modalities, dimensionality, and characteristics—even for a specific protocol, there are acquisition differences, a brand of the drugs, or local demographics [59]. Although federated learning can address the data bias issue by collecting more data sources, inhomogeneous data distribution is still challenging, as many assume independently and identically distributed data across their peers. Another challenge is the different data standards and data heterogeneity among peers. For example, hospitals may adopt EHR systems from different vendors, and different countries use different diagnostic and procedure coding systems. For example, health systems in the United Kingdom use the International Classification of Diseases ICD-10 code, but the United States adopted ICD-10-CM. This heterogeneity may lead to a situation where the optimal global solution may not work well for an individual local participant.

5.2 Efficiency and Effectiveness

From the technical view, efficiency and effectiveness are the major concerns of federated learning. Federated learning needs peers to share and update the models, and thus, the communication cost between different peers is an issue. Especially when integrated with blockchain, how to minimize the communication time and improve the efficiency of the training process is important. Studies have focused on improving the framework to jointly improve the federated learning convergence time and the training loss [60], but the tradeoff between accuracy and communication expenditure should also be considered.

5.3 Model Interpretation

Integrating machine learning models is important, particularly for healthcare and medicine. The core question of interpretability is whether humans understand why the model makes such predictions on unseen instances. Many machine learning models, such as deep learning, are a “black-box” to humans, and thus, many studies have explored tools to interpret the models [61, 62, 63, 64]. In a federated learning context, as the model was kept updated through multiple parties, the interpretation would be a challenge.

To summarize, federated learning for life sciences will benefit the process of data sharing among multiple organizations without a central authority. The data sharing process will monitor and track the data operations efficiently to ensure data integrity and provenance. Still, the data ownership problem is the key to adopting Rahasak-ML in FDA- or EMA-regulated research.

6 Conclusions

Federated learning emerges as a new technique that uses collaboration and distribution to train machine learning models without sharing the local raw data. It promises to benefit the medical field and drug industry that require strict data protection. However, most of the existing federated learning systems deal with centralized coordinators that are vulnerable to attacks and privacy breaches. We proposed a blockchain-empowered coordinator-less decentralized federated learning platform, named Rahasak-ML, to solve issues in centralized coordinator-based federated learning systems by providing better transparency and trust. We introduced the architecture and learning process of Rahasak-ML. We introduced a use case of using Rahasak-ML to train a machine learning model for diagnosis, which could be applied to other biomedical data to facilitate decision-making. Still, data standardization, communication efficiency, and model interpretation need to be resolved.

References

1. Kairouz P, McMahan HB, Avent B, Bellet A, Bennis M, Bhagoji AN, Zhao S (2021) Advances and open problems in federated learning. Found Trends Mach Learn 14(1–2):1–210. https://doi.org/10.1561/2200000083
2. Konečný J, McMahan HB, Yu FX, Richtarik P, Suresh AT, Bacon D (2016) Federated learning: strategies for improving communication efficiency. NIPS Workshop on Private Multi-Party Machine Learning. Retrieved from https://arxiv.org/abs/1610.05492
3. McMahan B, Moore E, Ramage D, Hampson S, Arcas BA (2017) Communication-efficient learning of deep networks from decentralized data. In: Proceedings of the 20th international conference on artificial intelligence and statistics; Proc Mach Learn Res 54:1273–1282. https://proceedings.mlr.press/v54/mcmahan17a.html
4. Sheller MJ, Edwards B, Reina GA, Martin J, Pati S, Kotrotsou A, Bakas S (2020) Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data. Sci Rep 10(1):12598. https://doi.org/10.1038/s41598-020-69250-1
5. Yang T, Andrew G, Eichner H, Sun H, Li W, Kong N, Beaufays F (2018) Applied federated learning: improving google keyboard query suggestions. https://arxiv.org/abs/1812.02903
6. Lu Y, Huang X, Zhang K, Maharjan S, Zhang Y (2020) Blockchain empowered asynchronous federated learning for secure data sharing in the internet of vehicles. IEEE Trans Veh Technol 69(4):2020 4298–4311. https://doi.org/10.1109/TVT.2020.2973651
7. Qu Y, Gao L, Luan TH, Xiang Y, Yu S, Li B, Zheng G (2020) Decentralized privacy using blockchain-enabled federated learning in fog computing. IEEE Internet Things J 7(6):5171–5183. https://doi.org/10.1109/JIOT.2020.2977383
8. Bandara E, Liang X, Foytik P, Shetty S, Ranasinghe N, De Zoysa K (2021) Rahasak-scalable blockchain architecture for enterprise applications. J Syst Archit 102061. https://doi.org/10.1016/j.sysarc.2021.102061
9. Nallaperuma D, Nawaratne R, Bandaragoda T, Adikari A, Nguyen S, Kempitiya T, Pothuhera D (2019) Online incremental machine learning platform for big data-driven smart traffic management. IEEE Trans Intell Transp Syst 20(12):4679–4690. https://doi.org/10.1109/TITS.2019.2924883

10. Shan N, Ziarko W (1994) An incremental learning algorithm for constructing decision rules. In: Ziarko WP (ed) Rough sets, fuzzy sets and knowledge discovery. Springer, pp 326–334. https://doi.org/10.1007/978-1-4471-3238-7_38
11. Sathya R, Abraham A (2013) Comparison of supervised and unsupervised learning algorithms for pattern classification. Int J Adv Res Artif Intell 2(2):34–38. https://doi.org/10.14569/IJARAI.2013.020206
12. Johnson KB, Wei W-Q, Weeraratne D, Frisse ME, Misulis K, Rhee K Snowdon JL (2021) Precision medicine, AI, and the future of personalized health care. Clin Transl Sci 14(1):86–93. https://doi.org/10.1111/cts.12884
13. Wang F, Preininger A (2019) AI in health: state of the art, challenges, and future directions. Yearb Med Inform 28(1):16–26. https://doi.org/10.1055/s-0039-1677908
14. Emerson S, Kennedy R, O'Shea L, O'Brien J (2019) Trends and applications of machine learning in quantitative finance (SSRN Scholarly Paper No. ID 3397005). Rochester, NY, Social Science Research Network. Retrieved from Social Science Research Network, https://papers.ssrn.com/abstract=3397005
15. List of registries (2015) Retrieved August 30, 2021, from National Institutes of Health (NIH). https://www.nih.gov/health-information/nih-clinical-research-trials-you/list-registries
16. Annas GJ (2003) HIPAA regulations—a new era of medical-record privacy? N Engl J Med 348(15):1486–1490. https://doi.org/10.1056/NEJMlim035027
17. Federated learning: Collaborative machine learning without centralized training data (2017) Retrieved August 30, 2021, from Google AI Blog, http://ai.googleblog.com/2017/04/federated-learning-collaborative.html
18. Liao W, Luo C, Salinas S, Li P (2019) Efficient secure outsourcing of large-scale convex separable programming for big data. IEEE Trans Big Data 5(3):368–378. https://doi.org/10.1109/TBDATA.2017.2787198
19. Forum on Neuroscience and Nervous System Disorders, Board on Health Sciences Policy, Institute of Medicine (2014) Drug development challenges. In: Improving and Accelerating Therapeutic Development for Nervous System Disorders: Workshop Summary. National Academies Press (US). Retrieved from https://www.ncbi.nlm.nih.gov/books/NBK195047/
20. Liao C, Peach ML, Yao R, Nicklaus MC (2013) Molecular docking and structure-based virtual screening. In: Future Science Book Series. In Silico Drug Discovery and Design. Future Science Ltd, pp 6–20. https://doi.org/10.4155/ebo.13.181
21. Subramanian I, Verma S, Kumar S, Jere A, Anamika K (2020) Multi-omics data integration, interpretation, and its application. Bioinform Biol Insights 14. https://doi.org/10.1177/1177932219899051
22. Pendergrass SA, Crawford DC (2019) Using electronic health records to generate phenotypes for research. Curr Protoc Hum Genet 100(1):e80. https://doi.org/10.1002/cphg.80
23. Zhao J, Feng Q, Wu P, Lupu RA, Wilke RA, Wells QS, Wei W-Q (2019) Learning from longitudinal data in electronic health record and genetic data to improve cardiovascular event prediction. Sci Rep 9(1):717. https://doi.org/10.1038/s41598-018-36745-x
24. Zhao J, Grabowska ME, Kerchberger VE, Smith JC, Eken HN, Feng Q, Wei W-Q (2021) ConceptWAS: a high-throughput method for early identification of COVID-19 presenting symptoms and characteristics from clinical notes. J Biomed Inform 117:103748. https://doi.org/10.1016/j.jbi.2021.103748
25. Jeon J, Nim S, Teyra J, Datti A, Wrana JL, Sidhu SS, Kim PM (2014) A systematic approach to identify novel cancer drug targets using machine learning, inhibitor design and high-throughput screening. Genome Med 6(7):57. https://doi.org/10.1186/s13073-014-0057-7
26. Ferrero E, Dunham I, Sanseau P (2017) In silico prediction of novel therapeutic targets using gene-disease association data. J Transl Med 15(1):182. https://doi.org/10.1186/s12967-017-1285-6
27. Godinez WJ, Hossain I, Lazic SE, Davies JW, Zhang X (2017) A multi-scale convolutional neural network for phenotyping high-content cellular images. Bioinformatics 33(13):2010–2019. https://doi.org/10.1093/bioinformatics/btx069

28. Brown N, Ertl P, Lewis R, Luksch T, Reker D, Schneider N (2020) Artificial intelligence in chemistry and drug design. J Comput Aided Mol Des 34(7):709–715. https://doi.org/10.1007/s10822-020-00317-x
29. Bibault J-E, Giraud P, Housset M, Durdux C, Taieb J, Berger A, Burgun A (2018).Deep learning and radiomics predict complete response after neo-adjuvant chemoradiation for locally advanced rectal cancer. Sci Rep 8(1):12611. https://doi.org/10.1038/s41598-018-30657-6
30. Harrer S, Shah P, Antony B, Hu J (2019) Artificial intelligence for clinical trial design. Trends Pharmacol Sci 40(8):577–591. https://doi.org/10.1016/j.tips.2019.05.005
31. Calaprice D, Galil K, Salloum W, Zariv A, Jimenez B (2020) Improving clinical trial participant prescreening with artificial intelligence (AI): a comparison of the results of AI-assisted vs standard methods in 3 oncology trials. Ther Innov Regul Sci 54(1):69–74. https://doi.org/10.1007/s43441-019-00030-4
32. Gaulton A, Hersey A, Nowotka M, Bento AP, Chambers J, Mendez D, Leach AR (2017) The ChEMBL database in 2017. Nucleic Acids Res 45(D1):D945–D954. https://doi.org/10.1093/nar/gkw1074
33. Kim S, Chen J, Cheng T, Gindulyte A, He J, He S, Bolton EE (2019) PubChem 2019 update: improved access to chemical data. Nucleic Acids Res 47(D1):D1102–D1109. https://doi.org/10.1093/nar/gky1033
34. Parikh RB, Teeple S, Navathe AS (2019) Addressing bias in artificial intelligence in health care. JAMA 322(24):2377–2378. https://doi.org/10.1001/jama.2019.18058
35. Schneider P, Walters WP, Plowright AT, Sieroka N, Listgarten J, Goodnow RA, Schneider G (2020). Rethinking drug design in the artificial intelligence era. Nat Rev Drug Discov 19(5):353–364. https://doi.org/10.1038/s41573-019-0050-3
36. Chang K, Balachandar N, Lam C, Yi D, Brown J, Beers A, Kalpathy-Cramer J (2018) Distributed deep learning networks among institutions for medical imaging. J Am Med Inform Assoc 25(8):945–954. https://doi.org/10.1093/jamia/ocy017
37. Xiong Z, Cheng Z, Lin X, Xu C, Liu X, Wang D, Zheng M (2021) Facing small and biased data dilemma in drug discovery with enhanced federated learning approaches. Sci China Life Sci https://doi.org/10.1007/s11427-021-1946-0
38. Li L, Xu W, Chen T, Giannakis GB, Ling Q (2019) Rsa: Byzantine-robust stochastic aggregation methods for distributed learning from heterogeneous datasets. [Cs, Math]. Retrieved from http://arxiv.org/abs/1811.03761
39. Kim H, Park J, Bennis M, Kim S-L (2020) Blockchained on-device federated learning. IEEE Commun Lett 24(6):1279–1283. https://doi.org/10.1109/LCOMM.2019.2921755
40. Xu G, Li H, Liu S, Yang K, Lin X (2020) Verifynet: secure and verifiable federated learning. IEEE Trans Inf Forensics Secur 15:911–926. https://doi.org/10.1109/TIFS.2019.2929409
41. Androulaki E, Barger A, Bortnikov V, Cachin C, Christidis K, De Caro A, et al (2018) Hyperledger fabric: A distributed operating system for permissioned blockchains. In: Proceedings of the thirteenth EuroSys conference, vol 30. ACM. https://doi.org/10.1145/3190508.3190538
42. Azaria A, Ekblaw A, Vieira T, Lippman A (2016) Medrec: using blockchain for medical data access and permission management. In: 2016 2nd International conference on open and big data (OBD). pp 25–30. https://doi.org/10.1109/OBD.2016.11
43. Liang X, Shetty S, Zhao J, Bowden D, Li D, Liu J (2018) Towards decentralized accountability and self-sovereignty in healthcare systems. In: Qing S, Mitchell C, Chen L, Liu D (eds) Information and communications security. ICICS 2017. Lecture notes in computer science, vol 10631. pp 387–398. https://doi.org/10.1007/978-3-319-89500-0_34
44. Yang G, Li C (2018) A design of blockchain-based architecture for the security of electronic health record (EHR) systems. IEEE international conference on cloud computing technology and science (CloudCom) 2018:261–265. https://doi.org/10.1109/CloudCom2018.2018.00058
45. Abadi M, Chu A, Goodfellow I, McMahan HB, Mironov I, Talwar K, Zhang L (2016) Deep learning with differential privacy. In: Proceedings of the 2016 ACM SIGSAC conference on computer and communications security. pp 308–318. https://doi.org/10.1145/2976749.2978318

46. Bonawitz K, Ivanov V, Kreuter B, Marcedone A, McMahan HB, Patel S, Seth K (2017) Practical secure aggregation for privacy-preserving machine learning. In: Proceedings of the 2017 ACM SIGSAC conference on computer and communications security. pp 1175–1191. https://doi.org/10.1145/3133956.3133982
47. Thönes J (2015) Microservices. IEEE Softw 32(1):116–116. https://doi.org/10.1109/MS.2015.11
48. Merkel D (2014) Docker: Lightweight Linux containers for consistent develop-ment and deployment. Linux J 2014(239):2. https://doi.org/10.5555/2600239.2600241
49. Burns B, Grant B, Oppenheimer D, Brewer E, Wilkes J (2016) Borg, omega, and kubernetes. Queue 14(1):70–93. https://doi.org/10.1145/2898442.2898444
50. Lakshman A, Malik P (2010) Cassandra: a decentralized structured storage system. Oper Syst Rev (ACM) 44(2):35–40. https://doi.org/10.1145/1773912.1773922
51. Gupta M (2012) Akka essentials. Packt Publishing Ltd. Retrieved from https://www.packtpub.com/product/akka-essentials/9781849518284
52. Davis AL (2019) Akka streams. In: Reactive streams in java. Apress, pp 57–70 https://doi.org/10.1007/978-1-4842-4176-9_6
53. Gormley C, Tong Z (2015) Elasticsearch: the definitive guide: a distributed real-time search and analytics engine. O'Reilly Media, Inc. Retrieved from https://www.oreilly.com/library/view/elasticsearch-the-definitive/9781449358532/
54. Beneventi F, Bartolini A, Cavazzoni C, Benini L (2017) Continuous learning of hpc infrastructure models using big data analytics and in-memory processing tools. In: Design, automation and test in Europe conference and exhibition (DATE). IEEE, 1038–1043. https://doi.org/10.23919/DATE.2017.7927143.
55. Bandara E, Liang X, Foytik P, Shetty S, Ranasinghe N, Zoysa KD, Ng WK (2021) Saas—microservices-based scalable smart contract architecture. In: Thampi SM, Wang G, Rawat DB, Ko R, Fan C-I (eds), Security in computing and communications. Singapore, Springer Singapore, pp. 228–243. https://doi.org/10.1007/978-981-16-0422-5_16
56. Bandara E, Ng W, Ranasinghe N, Zoysa K (2019) Aplos: Smart contracts made smart. BlockSys. https://doi.org/10.1007/978-981-15-2777-7_35
57. Wan Z, Hudak P (2000) Functional reactive programming from first principles. In: PLDI '00: Proceedings of the ACM SIGPLAN 2000 conference on programming language design and implementation, vol 35. ACM, pp 242–252. https://doi.org/10.1145/349299.349331
58. Upstill R, Eccles D, Fliege J, Collins A (2013) Machine learning approaches for the discovery of gene–gene interactions in disease data. Brief Bioinform 14(2):251–260. https://doi.org/10.1093/bib/bbs024
59. Rieke N, Hancox J, Li W, Milletarì F, Roth HR, Albarqouni S, Cardoso MJ (2020) The future of digital health with federated learning. Npj Digit Med 3:119. https://doi.org/10.1038/s41746-020-00323-1
60. Chen M, Shlezinger N, Poor HV, Eldar YC, Cui S (2021) Communication-efficient federated learning. PNAS 118(17). https://doi.org/10.1073/pnas.2024789118
61. Lundberg SM, Erion G, Chen H, DeGrave A, Prutkin JM, Nair B, Lee S-I (2020) From local explanations to global understanding with explainable AI for trees. Nat Mach Intell 2(1):56–67. https://doi.org/10.1038/s42256-019-0138-9
62. Lundberg SM, Nair B, Vavilala MS, Horibe M, Eisses MJ, Adams T, Lee S-I (2018) Explainable machine-learning predictions for the prevention of hypoxaemia during surgery. Nature Biomed Eng 2(10):749–760. https://doi.org/10.1038/s41551-018-0304-0
63. Montavon G, Samek W, Müller K-R (2018) Methods for interpreting and understanding deep neural networks. Digit Signal Process 73:1–15. https://doi.org/10.1016/j.dsp.2017.10.011
64. Ribeiro MT, Singh S, Guestrin C (2016) Why should i I trust you?: explaining the predictions of any classifier. [Cs, Stat]. Retrieved from http://arxiv.org/abs/1602.04938

Xueping Liang is an Assistant Professor in the Department of Information Systems and Supply Chain Management at the University of North Carolina at Greensboro. Prior to that, she was an Assistant Professor of Computer Science at Virginia State University. Her research is centered around data provenance mechanisms, cybersecurity, blockchain, privacy protection, and the Internet of Things (IoT). Specifically, she is interested in distributed consensus models in blockchain technology, cyber-resiliency in IoT, and various practical issues in cloud computing security. She received her Ph.D. in Cybersecurity from the University of Chinese Academy of Sciences. She has published more than 30 conference and journal papers and book chapters at reputed venues.

Eranga Bandara worked as a Senior Research Scientist at the Virginia Modeling Analysis and Simulation Center (VMASC) Virginia, USA. His research interests include Distributed Systems, Blockchain, Big Data, Actor-based Systems, and Functional programming. He worked as a Lead Engineer at Pagero AB Sweden. With Pagero AB, he was involved with research and developments in Distributed Systems, Functional Programming, Big Data, Actor-based systems, and DevOps.

Juan Zhao is a Research Assistant Professor in the Department of Biomedical Informatics at Vanderbilt University Medical Center. She achieved a Ph.D. degree in Computer Science at the University of Chinese Academy in Beijing, China. Her current research interests focus on deep learning, machine learning, natural language processing, and blockchain, especially on leveraging such technologies to further the understanding of complex diseases and improve clinical outcomes and treatment. This research combines her educational background in computer science, and machine learning, her programming experience as a senior software engineer, and her training as a postdoctoral fellow in cybersecurity at Tennessee State University and in Biomedical Informatics at Vanderbilt. Dr. Zhao's study was funded by the American Heart Association (AHA) and the National Institutes of Health (NIH). She has published ~20 research papers and achieved four software copyrights. She is also serving as a program committee member in the Association for the Advancement of Artificial Intelligence Conference (AAAI) and IEEE Big Data, the Associate Editor of the Journal of Network Modeling Analysis in Health Informatics and Bioinformatics, and the Special Topic Editor for Frontiers in Big data.

Sachin Shetty is an Associate Director in the Virginia Modeling, Analysis, and Simulation Center at Old Dominion University and an Associate Professor in the Department of Computational Modeling and Simulation Engineering. Sachin Shetty received his Ph.D. in Modeling and Simulation from Old Dominion University in 2007. His research interests lie at the intersection of computer networking, network security, and machine learning. Recently, he has been involved with developing cyber risk/resilience metrics for critical infrastructure and blockchain technologies for distributed system security. His laboratory has been supported by the National Science Foundation, Air Office of Scientific Research, Air Force Research Lab, Office of Naval Research, Department of Homeland Security, and Boeing. He has published over 150 research articles in journals and conference proceedings and four books. He is the recipient of Commonwealth Cyber Initiative Research Fellow, Fulbright Specialist award, EPRI Cybersecurity Research Challenge award, and DHS Scientific Leadership Award and has been inducted into Tennessee State University's million-dollar club.