A Decentralized Learning Architecture for Medical Prescription Anomaly Detection via Hybrid Federated-Swarm Learning

Ravelly C. Zanatta, Vinícius J. B. Vanzin, Saulo N. Matos, Rodrigo D. Garcia, Dilvan A. Moreira, and Jó Ueyama

***Abstract*—The increasing use of electronic medical records (EMRs) has improved efficiency, accuracy, and accessibility of patient data. However, conventional centralized architectures suffer from single points of failure and data privacy issues. To address these challenges, this study proposes a decentralized machine learning architecture that combines concepts from Federated Learning (FL) and Swarm Learning (SL) for anomaly detection in medical prescriptions. The proposed architecture leverages blockchain and the InterPlanetary File System (IPFS) to enable secure model sharing and decentralized storage, thereby reducing communication complexity and establishing a transparent, decentralized parameter repository. Experimental evaluations were conducted using logistic regression (LR), a multi-layer perceptron (MLP), and a decision tree (DT) model. Compared with the FL baseline, the proposed system achieved superior efficiency, lower resource consumption, and improved latency, along with smaller block sizes. It, however, exhibited slightly lower transaction throughput and longer training rounds, reflecting the added complexity of decentralization. In predictive performance on the anomaly classification task, DT achieved the highest precision and recall under the evaluated dataset (F_1-score = 0.9912), followed by MLP (0.5504) and LR (0.2442). The decentralized training approach led to negligible performance loss relative to centralized models, less than 4% for LR and below 1% for both MLP and DT. Overall, the proposed system demonstrates a robust and efficient alternative for decentralized learning in healthcare applications, maintaining strong predictive performance while enhancing architectural transparency.**

Link to graphical and video abstracts, and to code:
https://latamt.ieeer9.org/index.php/transactions/article/view/10586

***Index Terms*—Blockchains, Decentralized applications, Federated learning, InterPlanetary File System (IPFS), Medical information systems, Smart contracts**

I. Introduction

THE increasing use of electronic medical records has brought significant benefits, such as greater efficiency, accuracy, and rapid access to patient information. However, despite the adoption of these technologies, challenges related to data integrity, security, and privacy still persist. These risks associated with sensitive data are among the key drivers of legislation such as the General Data Protection Law (LGPD) [1], [2] in Brazil and the General Data Protection Regulation (GDPR) [3] in the European Union.

The practice of fraud in medical data has both clinical and financial impacts on the healthcare system [4]–[6]. Fraudulent data in medication prescriptions poses significant risks to patient health, including adverse drug reactions, unnecessary hospitalizations, and, in severe cases, even fatalities [4], [5]. Moreover, approximately 10% of global healthcare expenditures are lost due to fraud in the healthcare sector [6].

Strategies to combat healthcare fraud can be classified into three main categories: prevention, detection, and the use of artificial intelligence and data mining [7]. Recent studies have further demonstrated the robustness of machine learning models in critical healthcare scenarios, such as patient shock prediction [8] and cardiac pattern classification [9].

In this context, machine learning represents a promising alternative, enabling the analysis of large volumes of data and the identification of suspicious patterns and prescriptions [6], [10]. Similar anomaly-based detection approaches have proven effective in monitoring complex, distributed infrastructures [11]. However, current prescription systems employ centralized architectures. This reliance creates single points of failure and increases vulnerability to attacks. It also raises concerns regarding patient data privacy [12]–[14].

Given the sensitivity of healthcare data, approaches that prioritize privacy are essential. Federated Learning (FL) addresses this by allowing nodes to train models locally while retaining data at the edge [15]. In this framework, nodes transmit only the trained model parameters to a central server, which aggregates them into a global model and redistributes the update [16], [17]. However, this reliance on a central server introduces vulnerabilities regarding security and fault tolerance. The central aggregator becomes a single point of failure, susceptible to software outages and malicious attacks, such as the injection of adversarial data [18].

To mitigate these centralization risks, Swarm Learning (SL) offers a decentralized alternative. Unlike FL, SL eliminates the central server. Nodes instead form a "swarm" to exchange parameters directly. The network collaboratively refines the global model in an interative process [18]–[20]. To maintain coordination without a central authority, training is typically organized into rounds where a random node is selected as the temporary leader for aggregation [19], [21]. Despite its advantages, SL faces significant challenges, including high

The associate editor coordinating the review of this manuscript and approving it for publication was Mónica K. Huerta (*Corresponding author: Ravelly Carvalho Zanatta*).

Ravelly Carvalho Zanatta, V. J. B. Vanzin, S. N. Matos, R. D. Garcia, D. A. Moreira, and J. Ueyama are with the Institute of Mathematical and Computer Sciences at the University of São Paulo, São Carlos, Brazil (e-mails: ravellyzanatta@usp.br, vinicius.vanzin@usp.br, saulo.matos@usp.br, rgarcia@usp.br, dilvan@icmc.usp.br, joueyama@icmc.usp.br).

communication complexity and node synchronization issues which can overload the network, as well as difficulties regarding the security and traceability of malicious behavior [21].

Our work builds on the FL and SL methodologies by introducing a decentralized machine learning framework for anomaly detection in medication prescriptions. The proposed approach integrates a permissioned blockchain infrastructure with the InterPlanetary File System (IPFS), a P2P communication protocol designed to store and share data in a distributed and decentralized manner.

To preserve decentralization and coordinate the training process, we implement a smart contract that orchestrates key tasks. This contract designates the leader node for aggregation (analogous to the mechanism in Swarm Learning), assigns the testing node, regulates access to parameters and models, defines stopping criteria, and manages the initiation and completion of each training step, among other decisions. In addition, to increase traceability and integrity, model parameters are stored and shared via the IPFS, with each version tracked through a Content Identifier (CID) address recorded in the smart contract.

In summary, this study introduces a novel coordination mechanism and hybrid topology for decentralized machine learning for anomaly detection in medical prescriptions. Our main contributions are as follows:

- We propose a topology that bridges the gap between Federated Learning (FL) and Swarm Learning (SL). By utilizing the IPFS as a decentralized parameter repository, the architecture successfully eliminates the single point of failure of FL. It also mitigates the P2P communication overhead associated with SL.
- We introduce an on-chain orchestration model that governs the training lifecycle. This includes a training synchronization strategy, a randomized leader/aggregator selection mechanism, and strict access control over the off-chain IPFS parameters.
- We present an adaptable, decentralized framework capable of supporting different machine learning paradigms. The system accommodates multiple aggregation strategies, such as parameter-averaging and prediction-level majority-voting. It maintains predictive performance on par with traditional centralized baselines for the task of medical prescription anomaly detection.

To evaluate the proposed approach, we utilized the dataset created by dos Santos et al. [28], which contains labeled medical prescriptions collected in a hospital production environment. The task of detecting anomalies in these prescriptions is framed as a binary classification problem. Each sample in the dataset is classified based on features such as medication type, dosage, and frequency.

II. Related Work

Several studies investigate FL as an alternative to mitigate privacy concerns in healthcare-related machine learning applications. For example, the study conducted by Kumar et al. [22] proposes an architecture for COVID-19 detection using deep learning models and the integration of FL supported by blockchain to ensure patient data privacy. The approach involves data normalization, segmentation using Capsule Networks, and classification to identify patients with COVID-19. Similarly, the work of Settipalli et al. [23] employs a federated blockchain implemented with Quorum and the PBFT consensus algorithm to integrate health insurance claim data from different organizations, providing enhanced security and privacy.

Similarly, Baucas et al. [24] developed an FL platform integrated with blockchain for wearable devices in predictive healthcare applications. In their approach, models are trained locally on each device to preserve data privacy and are subsequently aggregated on a server for further analysis. Finally, the work conducted by Zutião [25] proposes an architecture that integrates blockchain with FL via the Flower framework, aiming to enhance security, privacy, and reliability in the machine learning process for medical prescription anomaly detection.

Other studies highlight Swarm Learning (SL) as an alternative approach for ensuring patient-data privacy and security. Warnat-Herresthal et al. [19] present the SL architecture as a decentralized and privacy-preserving approach. This methodology shows promise for the collaborative analysis of clinical data, such as diagnostics, prognostics, and treatment, under stringent reliability requirements. Meanwhile, Han et al. [18] provide an overview of the SL architecture, describing its properties, challenges, and opportunities, as well as discussing its differences from Federated Learning (FL). They also examine its advantages compared to other machine learning methods.

Saldanha et al. [26] propose a study on the direct prediction of genetic aberrations in gastric cancer using Swarm Learning with pathology images. The study demonstrates strong performance in predicting molecular biomarkers, indicating that Swarm Learning achieves performance comparable to that of other deep learning models.

Still within the healthcare domain, Gana and Jamil [27] examine the integration of Swarm Learning, Federated Learning, blockchain, and Directed Acyclic Graphs in IoT-based healthcare systems through a systematic literature review. It highlights the potential of DAGs to enhance scalability and performance compared to traditional blockchain architectures, while also identifying key challenges such as interoperability, technical complexity, and regulatory compliance.

Unlike Federated Learning, which exhibits limitations in data security and fault tolerance due to its reliance on a centralized aggregation server [18], Swarm Learning operates as a decentralized mechanism. However, because of its architecture, SL faces challenges related to communication complexity and node synchronization, which may overload the network [19], [21].

Approaches integrating FL with blockchain (e.g., [22], [23]) successfully enhance trust and auditability. However, they generally suffer from two critical trade-offs: they either retain a central aggregation server (single point of failure), or they attempt to store heavy model parameters directly on-chain, leading to block size bloat. Conversely, SL designs (e.g., [19], [26]) eliminate the central server, but their reliance on direct peer-to-peer parameter exchange introduces communication

TABLE I
COMPARISON OF STATE-OF-THE-ART APPROACHES IN HEALTHCARE APPLICATIONS CONSIDERING BLOCKCHAIN INTEGRATION, DATA PRIVACY, AND DECENTRALIZED MACHINE LEARNING

Reference	Coordinator	On-chain Storage	Off-chain Storage	Decentralized aggregation	Communication
22	Central server	Model weights and metadata	CT images	No	P2P
23	Central server	Insurance data	Raw data and smart contract	–	Client-server
24	Central server	Local model results and history	Raw and private patient data	No	Client–server
25	Central server	Model results and metadata	Sensitive training data	No	Client-server
19	Swarm Learning Library (SLL)	Onboarding metadata, permissions and smart contracts	Raw training data	Yes	P2P
18	Swarm coordinator node	Model state metadata and identities (SVIDs)	Raw training data	Yes	P2P
26	Ethereum-based blockchain	Metadata and global model state information	Raw training data	Yes	P2P
27	Protocol consensus and smart contract	Metadata, identities, and logs	Raw training data	Yes	P2P
Ours	Smart contract	CID addresses and metadata	Model parameters and sensitive training data	Yes	IPFS and Smart contract

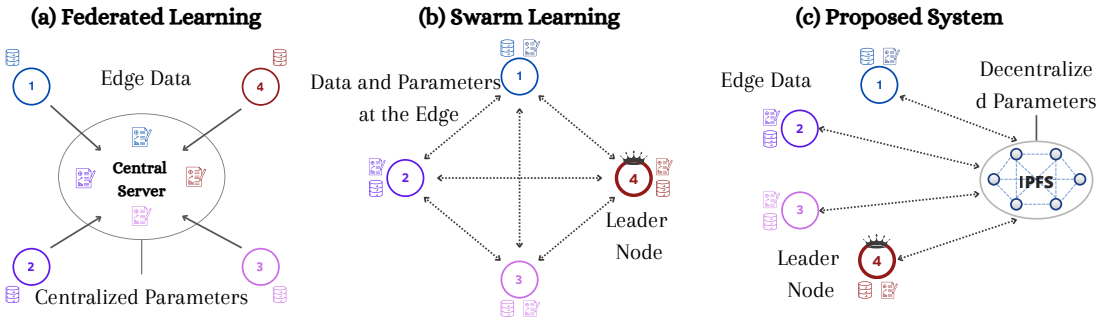


Fig. 1. Comparison between decentralized learning mechanisms: (a) Federated Learning, (b) Swarm Learning, and (c) the proposed system.

overhead and synchronization challenges. Therefore, a critical gap remains for an architecture that can achieve the decentralized aggregation of SL while maintaining the communication efficiency and off-chain storage flexibility of FL. Table I summarizes the key features of each study in comparison with our proposed approach. In this context, the aim of this study is to propose a fault-tolerant system that leverages the decentralized aggregation characteristics of SL. Furthermore, the use of IPFS as a data-sharing mechanism reinforces decentralization while reducing communication complexity, similarly to FL.

III. METHODOLOGY

We propose a decentralized machine learning framework for anomaly detection in medical prescriptions. In addition, a smart contract coordinates the system by managing node participation in training, controlling rounds, regulating access to parameters stored in IPFS, and selecting both the leader node (for aggregation) and the testing node.

Fig. 1 illustrates the similarities and differences among FL, SL, and our proposed system. Our approach resembles FL in terms of training data storage and sharing, as nodes also send training outputs to a single location. However, instead of a central server controlling all model parameters, IPFS is used. This ensures that parameters remain decentralized, since IPFS fragments uploaded files into blocks, which are distributed and stored across its network nodes, tracking them through Content Identifiers (CIDs) that uniquely identify the content [29].

Compared to SL, the proposed framework adopts a similar training coordination structure. Specifically, training is organized into rounds, and in each round, a leader node is randomly selected for parameter aggregation, while a testing node is also designated (which does not participate in training during that round). The key difference is that, whereas in SL, nodes share model parameters directly with one another through P2P connections, in the proposed system, the parameters are stored in IPFS, and their corresponding CIDs are recorded in the coordinating smart contract. As a result, parameter sharing is mediated through IPFS, with the smart contract controlling access to the stored information.

The proposed architecture eliminates the need for multiple P2P connections among training nodes, since communication is mediated by the blockchain and IPFS. By routing parameter sharing through IPFS and recording CIDs on the blockchain, the system creates a fully transparent and traceable ledger of the training lifecycle without relying on direct node-to-node parameter exchange.

The decentralized training process is governed by the smart contract through a synchronous algorithmic cycle:

1) The global model and its respective hyperparameters are initialized and stored off-chain in IPFS.
2) Nodes register their participation via the smart contract. The contract enforces a synchronous barrier to handle communication delays and prevent inconsistencies. Training operations cannot begin until the required number of participating nodes is available.
3) At the beginning of each round, the contract uses a pseudo-random mechanism computed via the current block hash to assign one node as the aggregator (leader) and one as the held-out testing node.
4) The designated trainer nodes query the contract for the current global model's CID, fetch the parameters from IPFS, and execute local training entirely on their isolated edge data partitions.
5) Upon completion, each trainer node uploads its locally updated parameters to IPFS and submits the resulting CID to the smart contract via a blockchain transaction.
6) The smart contract acts as a synchronization lock, waiting until all trainer CIDs for the current round are received. The leader node fetches all local parameters from IPFS, applies the model-specific aggregation formula, stores the new global model on IPFS, and registers the final aggregated CID on-chain.
7) The testing node, which was excluded from the training phase, fetches the newly aggregated global model, evaluates it against its held-out test set, and records the performance metrics to IPFS.

Fig. 2 illustrates the training workflow, showing all decisions and actions carried out by the nodes and the smart contract.

Training proceeds in multiple rounds until the stopping criterion is met (e.g., the maximum number of rounds is reached). In each round, the smart contract randomly selects a leader node responsible for aggregating the parameters of the locally trained models, as well as a node tasked with testing the global model produced in that round. These assignments occur at the beginning of every round, ensuring decentralization by changing both leader and testing roles. The testing node does not participate in training; it only evaluates the global model at the end of the round.

The way nodes train the model resembles a cross-validation scheme. Each node in the network holds a distinct subset of the data and independently performs local training. Afterward, the parameters of the trained models are aggregated to compose the global model. This process allows different subsets to be evaluated, preventing the training from overfitting to a specific portion of the data, in a manner analogous to k-fold cross-validation. Furthermore, the testing node uses its own data partition (an unseen held-out subset) exclusively for evaluation and does not participate in training.

A. Smart Contract

The coordinating smart contract, previously described, is responsible for managing the stages of the decentralized machine learning process. Its primary role is to orchestrate interactions with the network nodes, ensuring synchronization of the training, aggregation, and validation phases, as well as the secure recording of operations on the blockchain. To preserve network security and integrity, the contract defines two modifiers (*onlyOwner* and *onlyRegistered*), ensuring that only authorized nodes can participate in the training process.

The smart contract is structured around four core functions. First, nodes are registered for participation in the training process. This function enforces two conditions: (i) a node must not already be registered, and (ii) training can only begin once the number of registered nodes reaches the minimum specified by the user. Once these conditions are satisfied, the training procedure is initiated. At the beginning of each round, a leader node and a testing node are randomly selected using a pseudo-random mechanism native to the smart contract, computed via the current block hash.

The second function (training) requires that the node be registered (*onlyRegistered*) and not designated as the testing node. Upon completion, the CIDs of the local training outputs are recorded, and an event is emitted for confirmation. The third function (aggregation) may only be invoked by the leader node of the respective round, which is responsible for storing the CID of the global model. This step occurs after all training nodes have submitted their local CIDs.

In the final function (testing), only the designated testing node is permitted to perform the evaluation and record the results on the contract. The testing node does not participate in the training phase, ensuring that evaluation is conducted on data unseen during training. Finally, the contract verifies whether the maximum number of rounds has been reached; if not, the next round is initiated.

B. Experimental Settings

To study the network behavior of the proposed framework, we simulated a permissioned blockchain network in a local test environment employing the Hyperledger Besu platform [30]. The experiments were conducted on hardware equipped with an Intel Core i7-7700 CPU @ 3.60GHz, 16GB of RAM, and the Ubuntu 22.04 operating system. Network participants were deployed as containerized Docker nodes. Each node's routine was automated using a Python orchestration script that interfaces with the blockchain via the Web3.py library. To ensure reproducibility, fixed random seeds were set across data splitting procedures and model parameter initializations.

The proposed approach enabled decentralized machine learning, in which nodes train models locally on their own data, share parameters and results via IPFS, and coordinate task execution through a smart contract on the blockchain. Our code is available publicly at: https://github.com/ravelly-zanatta/decentralized-machine-learning.

In this test environment, the network was configured with the IBFT2 consensus protocol and WebSocket communication to ensure synchronization of events between nodes and the smart contract. The smart contract coordinates the training, aggregation, and testing phases of the system, manages the order of training rounds, records the CIDs of the trained

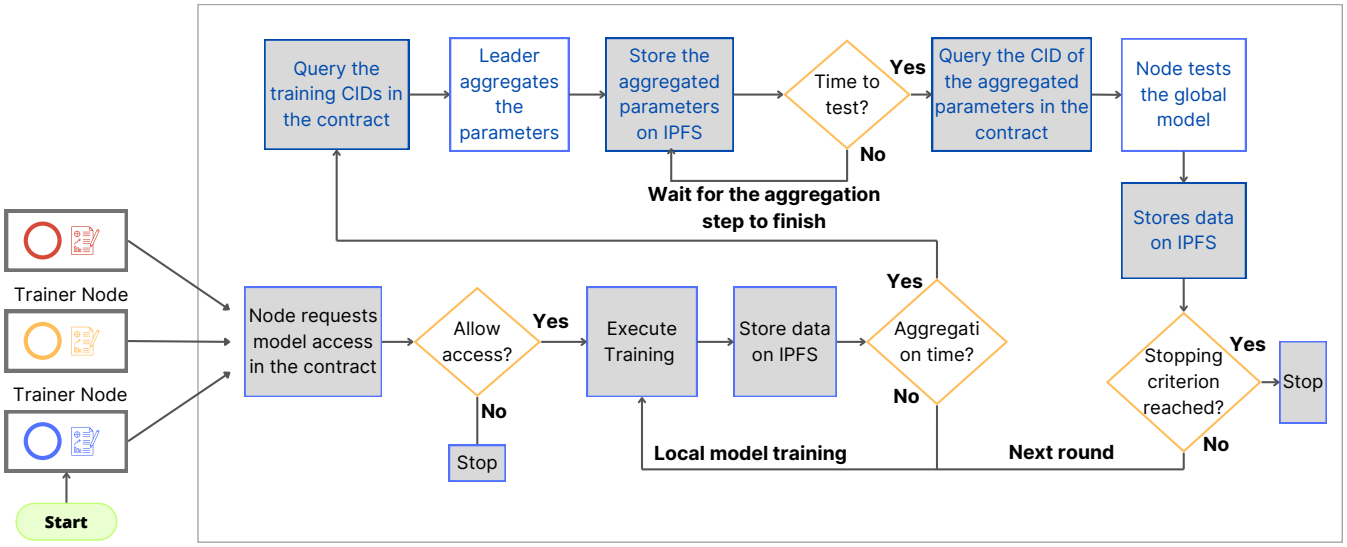


Fig. 2. Training steps flowchart for the proposed decentralized learning system (adapted from [19]).

models, and randomly assigns both a leader node and a testing node for each round.

The contract was developed in Solidity (compiler version 0.8.0). Deployment to the network was performed via the MetaMask wallet, which served as the intermediary for signing and submitting the deployment transaction. For data storage, a local instance of IPFS was used during the experiments in the testing environment.

For the task of anomaly detection in medical prescriptions, we evaluate multiple machine learning techniques, including LR, MLP, and DT, using medication type, dosage, and frequency as classification features. The proposed architecture is designed to be agnostic to the choice of learning algorithm, supporting different parameter aggregation strategies. For Logistic Regression, aggregation is performed by averaging the parameters of the local models to form the global model weights W_{global}, defined as:

$$W_{global} = \frac{1}{N} \sum_{i=1}^{N} W_i \tag{1}$$

Where N is the number of local trainer nodes and W_i is the coefficient vector of the i-th local model.

For the Decision Tree and Multi-layer Perceptron models, aggregation is performed at the prediction level. The global model is composed of the combined predictions of the local models following a majority-voting ensemble scheme:

$$\hat{y}_{global} = \text{mode}(\hat{y}_1, \hat{y}_2, ..., \hat{y}_N) \tag{2}$$

Where $\hat{y}_i$ is the class prediction of the i-th local model. Thus, the architecture can be adapted to various machine learning methods.

1) Dataset: For anomaly detection evaluation, we used a medical prescription dataset[1] from Nossa Senhora da Conceição Hospital in Brazil. The original dataset contained 563,171 prescription orders collected from January to September 2017. In the original implementation, anomaly detection was performed through unsupervised outlier detection, reaching an F_1-score of 68% in the best experimental setting [28]. In our experiments, we used the publicly available subset of the dataset, comprising 150,113 samples. We instead reframed the anomaly detection problem as a supervised binary classification task, dividing the data into a training set (75%) and a held-out test set (25%) in a per-class stratified manner. The training set was further partitioned evenly across the training nodes in the network, simulating isolated edge data. The held-out test set was allocated exclusively to the assigned testing node. Table II describes each feature in the dataset.

TABLE II
DESCRIPTION OF DATASET FEATURES FOR MEDICAL PRESCRIPTION ANOMALY DETECTION

Feature	Datatype	Description
Medication	String	Medication name
Frequency	Numeric	Daily frequency
Dosage	Numeric	Prescribed dosage (in milligrams)
Target	Boolean	Prescription label. 0 for valid prescriptions and 1 for anomalous prescriptions

Being an anomaly detection task, the dataset is highly imbalanced. In the publicly available version used for our experiments, 147,811 samples correspond to valid prescriptions (class 0), whereas 2,302 samples are labeled as anomalous (class 1). The majority class represents over 98% of the total data. The categorical feature of "Medication" (21 unique values) was encoded through one-hot encoding. The dataset was intentionally kept in its original unbalanced form to preserve the real-world distribution of anomalies. Artificially balancing the classes (e.g., via undersampling or synthetic oversampling techniques) could distort the inherent rarity of irregular or fraudulent medical prescriptions.

2) Machine Learning Models: To evaluate the performance of the proposed approach on the medical prescription anomaly

[1]https://github.com/nlp-pucrs/prescription-outliers

detection task, we compared three machine learning methods: logistic regression (LR), decision tree (DT), and multi-layer perceptron (MLP). For the MLP, hyperparameter optimization was performed using the Optuna framework [31], conducting 30 trials and optimizing for the F_1-score on the training set. The exact configurations and hyperparameter values for all three models are summarized in Table III.

TABLE III
MACHINE LEARNING MODEL HYPERPARAMETERS

Model	Hyperparameters	Values
LR	Solver / Penalty	L-BFGS/L2
	Max iterations	1000
	Warm-start / Intercept	True/True
DT	Split criterion	Gini impurity
	Max depth	Unrestricted
	Min samples split / Leaf	2 / 1
MLP	Hidden layers (neurons)	4 layers (55, 77, 28, 40)
	Learning rate / Optimizer	0.0721 (constant) / SGD
	L2 Regularization / Activation	1.084×10^{-5} / ReLU
	Max epochs	200

3) Evaluation Framework: We evaluated the proposed approach according to three main criteria: network performance, scalability, and anomaly classification performance.

To analyze network performance, we measured the transaction rate (transactions per second - TPS), minimum, maximum, and average latency, average duration of training rounds, RAM and CPU utilization, and average block size (in bytes). Each metric was averaged over 10 independent runs, with each run consisting of 5 training rounds. Hardware resource consumption was continuously monitored and captured using a Prometheus and Grafana stack. Transaction latency was measured as the elapsed time from transaction submission until confirmation by the smart contract.

To maintain consistency across evaluations, the experimental environment utilized a total of 8 nodes. These were divided into two distinct groups: 4 participating machine learning nodes and 4 blockchain validator nodes. Among the 4 participating nodes, in any given round, 3 acted as trainers (with one randomly selected as the leader/aggregator) and 1 acted exclusively as the testing node. The 4 blockchain validator nodes were strictly responsible for maintaining the ledger via the IBFT2 consensus protocol and did not participate in the machine learning tasks. For scalability, we analyzed the average transaction latency and average round duration while varying the number of participating nodes from 2 to 6. As in the network performance evaluation, we report the average values across 10 runs for each experimental configuration.

For the anomaly classification task, we report standard binary classification metrics on the held-out test set, including loss, accuracy, precision, recall, and F_1-score. The evaluated machine learning models (LR, DT, and MLP) were tested under identical experimental conditions, with 4 participating nodes and 4 blockchain validation nodes. To assess the impact of our decentralized training, we compared the classification performance of our proposed approach against a traditional centralized training setup using the same dataset and model configurations.

IV. RESULTS AND DISCUSSION

We evaluated the proposed decentralized training framework for medical prescription anomaly detection using three machine learning algorithms: logistic regression, decision tree, and multi-layer perceptron. Performance was assessed across three categories: network (under a simulated environment), scalability, and classification performance on the medical prescription dataset.

A. Network Performance

For network performance, we compared our approach against a blockchain-based federated learning baseline [25]. This specific baseline was selected because it tackles the same medical prescription anomaly dataset using a decentralized learning paradigm. This choice provides a highly controlled environment for direct comparison. It represents a fully on-chain trust model, where all artifacts are stored immutably on the ledger. This serves as a counterpoint to evaluate the efficiency of our hybrid architecture, which separates on-chain coordination from off-chain storage.

We evaluate the baseline system using the Decision Tree (DT) model because it demonstrated the strongest predictive performance in the anomaly classification task. Consequently, it is the most representative and viable candidate for a realistic comparative assessment.

All experiments were conducted under the same conditions, using the 4 participating machine learning nodes and the 4 blockchain validator nodes. Each experiment consisted of five training rounds on the same dataset, with identical data partitions assigned to each trainer node. The results indicate that, while the proposed architecture improves system latency and resource usage, implementing a more complex smart contract increases execution time and reduces the transaction throughput compared to the baseline.

1) Transactions per Second (TPS): The results for the TPS metric reflect the network's ability to process transactions within a given time interval. A higher TPS value indicates a greater capacity to handle a larger volume of concurrent requests without significantly increasing latency. Conversely, lower TPS values may reveal limitations related to the consensus mechanism, computational complexity of operations, or overhead introduced by additional system layers.

The proposed system achieved an average TPS of 159.66 with logistic regression, 154.93 with a decision tree, and 157.20 with MLP. In comparison, the baseline system reported an average TPS of 172.62.

For all tested models, the proposed system achieved a lower TPS than the baseline. This decrease is likely caused by the overhead of additional operations by the decentralized framework. The system must manage communication through IPFS for model storage and retrieval, and the selection of leader and testing nodes for every round. These introduce extra coordination and processing steps within the network.

2) Latency: Transaction latency is a key metric in blockchain-based systems, as it directly affects response time and coordination between the nodes and the smart contract.

The evaluated latency corresponds to the elapsed time between the submission and confirmation of network transactions.

As shown in Table IV, the baseline system exhibited higher latency compared to the proposed system for every machine learning technique. The proposed approach presented relatively consistent latency results across all models.

TABLE IV
COMPARISON OF TRANSACTION LATENCY (IN SECONDS) ACROSS EXPERIMENTS

Model	Max Latency (s)	Min Latency (s)	Avg Latency (s)
LR	**0.9898**	**0.1086**	0.6468 ± 0.0221
MLP	1.0154	0.1123	0.6029 ± 0.0321
DT	1.0956	0.1094	**0.5974** ± 0.0180
Baseline	1.1162	0.3326	0.8975 ± 0.0392

Overall, the proposed system achieved lower average latency for all tested machine learning models compared to the baseline. The combined transaction submission and validation time in our architecture is comparatively shorter. This improvement is attributed to the design choice of registering only the CID addresses of the models on the blockchain. In contrast, the baseline records both the models and their corresponding metrics on-chain. As a result, the proposed system reduces the amount of information processed by the consensus mechanism, leading to faster transaction confirmation times.

3) Average Round Duration: Round duration accounts for all operations performed during the training process, both on-chain and off-chain. Table V presents the results for the average time per round in each scenario.

TABLE V
COMPARISON OF AVERAGE ROUND DURATION (IN SECONDS) ACROSS EXPERIMENTS

Rounds	LR	MLP	DT	Baseline
1	25.60	22.01	12.74	9.44
2	16.32	23.95	11.98	10.05
3	16.58	19.99	14.01	9.09
4	15.26	18.13	12.97	9.48
5	14.26	19.29	13.96	10.13
Avg	17.60 ± 4.08	20.67 ± 2.06	13.13 ± 0.76	9.63 ± 0.39

The baseline achieved shorter training times across all five rounds compared to the proposed system. This difference can be attributed to the overhead introduced by communication between the blockchain and IPFS, data serialization and upload times, aggregation complexity (influenced by the random leader selection process), and testing procedure (performed by the designated testing node).

4) RAM usage: RAM usage is a relevant factor in distributed systems, particularly in scenarios with hardware constraints. We evaluated the memory consumption of each network node during the training process.

Memory usage data were collected and averaged across all nodes for both the proposed system and the baseline, as shown in Table VI. With the same number of nodes in the network, the baseline consumed more memory than the proposed system.

TABLE VI
COMPARISON OF RAM USAGE (IN MB) FOR PARTICIPATING AND VALIDATOR NODES

Node	LR	MLP	DT	Baseline
Participating 1	269.85	259.15	274.12	377.03
Participating 2	240.84	259.46	255.46	342.78
Participating 3	247.71	272.27	252.61	377.33
Participating 4	272.45	291.26	278.01	389.70
Validator 1	276.38	294.29	277.04	347.85
Validator 2	264.28	285.42	266.69	370.14
Validator 3	274.80	290.39	269.71	347.49
Validator 4	287.56	296.54	268.79	358.66
Avg	*266.73 ±14.45*	*281.10 ±14.35*	*268.79 ±8.79*	*363.87 ±16.03*

This difference may be attributed to the aggregation structure of the federated learning baseline, which requires higher memory allocation on participating nodes to store and process local model parameters. Additionally, the management of communication and synchronization with the central aggregation server can introduce extra overhead, further increasing memory usage.

5) CPU usage: In addition to memory usage, we also evaluated the raw CPU utilization during training for each approach. Measurements were collected during training for both the proposed decentralized system (using the three machine learning techniques) and the federated learning baseline system. CPU consumption data were sampled at constant time intervals (a total of 11 timesteps).

Fig. 3 illustrates the CPU usage of each network node across five training rounds. Overall, the evaluated approaches demonstrate notable variation across network nodes. In the proposed system using logistic regression, all nodes maintained CPU usage below 4%, with validators showing slightly higher averages. The MLP model exhibited a similar pattern, with all nodes remaining below 4.4%. For the decision tree model, participating nodes showed even lower consumption, while validator nodes presented slightly higher values, reaching up to 4.01% in one case. Conversely, the baseline system showed consistently higher CPU usage across all nodes, with validators averaging above 5.78%, indicating higher computational overhead.

These findings reinforce the computational efficiency of the proposed decentralized architecture, demonstrating both reduced resource consumption and a more balanced workload distribution among participating nodes.

6) Block Size: The block size is a metric directly related to the volume of data being processed and validated during training rounds. Smaller blocks tend to reduce propagation time between nodes and overhead on the consensus mechanism, which can contribute to lower latency. Larger blocks may indicate higher computational costs and longer validation times. Average block size reflects how the system's architecture impacts overall network performance. As shown in Table VII, the block sizes in the federated learning baseline system were larger compared to those of the proposed system using any of the machine learning techniques.

This difference can be attributed to architectural distinc-

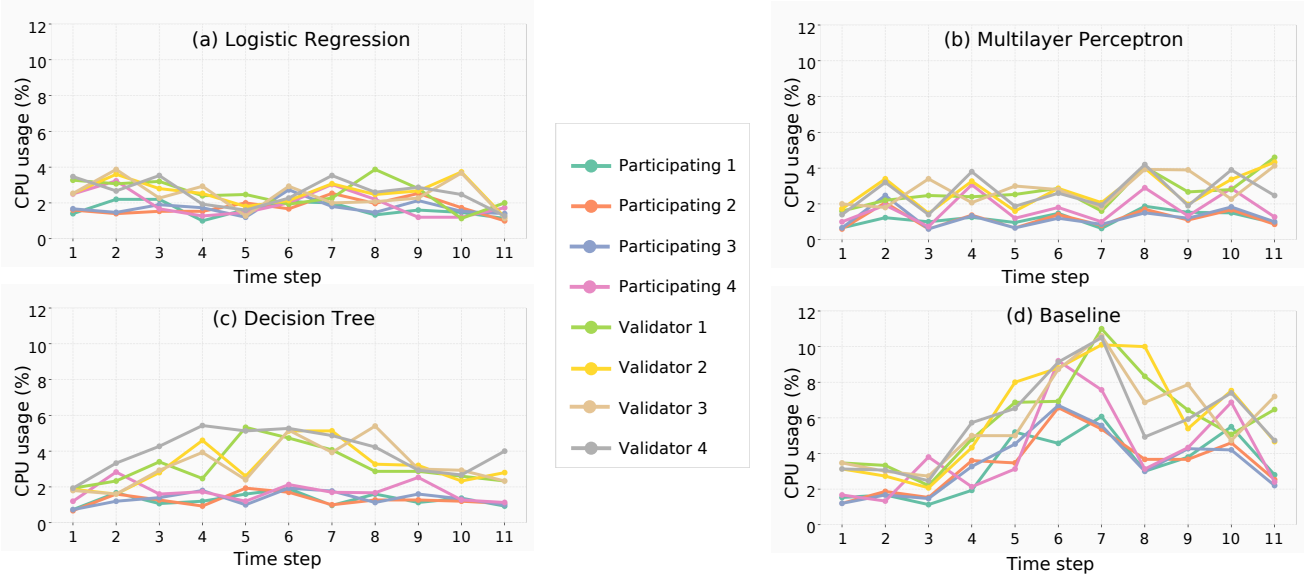


Fig. 3. CPU usage (%) during training for different learning models: (a) LR, (b) MLP, (c) DT, (d) Baseline.

TABLE VII
COMPARISON OF AVERAGE BLOCK SIZE (IN BYTES) ACROSS EXPERIMENTS

Model	Round 1	Round 2	Round 3	Round 4	Round 5
LR	1084	1132	1084	1132	1075
MLP	1084	1083	1084	1083	1084
DT	1085	1085	1085	1085	1085
Baseline	42873	41753	40473	42873	42841

tions: in the proposed framework, only CID addresses referring to the models are stored on the blockchain, whereas the baseline stores both the models and their associated metrics directly on-chain. This design choice significantly reduces the volume of data transmitted and stored within the network. Moreover, the proposed system also exhibited lower average latencies relative to the baseline, supporting these findings, since block size is directly correlated with data volume, which in turn impacts latency.

B. Scalability

We evaluated the behavior of the proposed decentralized machine learning system under different configurations. Experiments were conducted by varying the number of participating nodes, ranging from 2 to 6 participating nodes.

We employed the LR, DT, and MLP models across all configurations to assess how each model behaves as the number of nodes in the network increases. The metrics analysis included transaction latency on the blockchain and the average training time per round, with a standard of five training rounds. Even as the number of trainer nodes changed, the dataset was uniformly partitioned across them, ensuring that all nodes trained under equivalent conditions.

As shown in Fig. 4, the proposed system using the decision tree model achieved lower average round times compared to the other models. Based on average round times, both LR and DT demonstrate scalable behavior, as the increase in training time follows a near-linear trend. In contrast, the MLP model does not scale as well, as its average round time grows non-linearly with the number of training nodes, likely due to the higher computational overhead of off-chain operations. The MLP requires longer training for convergence and greater memory usage, which increases read and write latency.

Regarding average latency (also shown in Fig. 4), all models maintained relatively stable values even as the number of nodes increased. Across all experimental configurations, latency remained within the range of 0.5506 and 0.6103 seconds per round, showing a low rate of variation when varying the number of participating nodes.

While the scalability evaluation was constrained by the computational limits of a local simulated environment, the architectural design provides indicators of how the system is projected to behave in a larger setting. Unlike traditional SL, which can suffer from exponentially increasing communication complexity due to direct peer-to-peer parameter exchange, the proposed architecture isolates communication. Nodes interact exclusively with the smart contract and the IPFS network.

However, as the number of participants increases, the synchronization overhead will shift to the aggregation phase. The leader node must execute a growing number of read operations from IPFS to retrieve all local models, making the aggregation bandwidth and processing time linearly dependent on the network size. In environments with a larger number of nodes, these factors could introduce latency bottlenecks, which may require scaling optimizations such as hierarchical aggregation.

C. Anomaly Detection Performance

In scenarios with a high degree of class imbalance, such as the dataset used in this study, accuracy alone is not an adequate measure of model performance, since a classifier that predicts

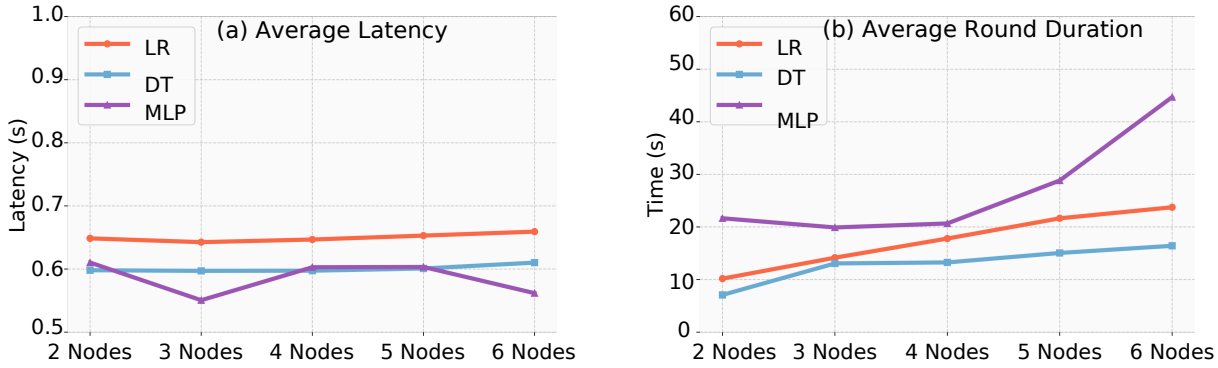


Fig. 4. (a) Average latency (in seconds) and (b) average round duration (in seconds) across experiments.

only the majority class may achieve high accuracy without discriminative capability. Therefore, the metrics analyzed included loss, accuracy, precision, recall, and the F_1-score, allowing for a comprehensive evaluation of the predictive performance of each approach.

For all three tested models (LR, MLP, and DT), experiments were carried out with four nodes: three trainers (with one selected as aggregator) and one separate testing node per round. The training set was uniformly partitioned among the trainer nodes. We report performance metrics on the held-out test set at the end of five training rounds. We compare against the same models trained in a conventional centralized manner, in order to assess differences in performance between the two training structures.

Table VIII presents the performance metrics for the evaluated machine learning models. The LR model was not particularly suitable for this scenario, given the highly imbalanced nature of the dataset. While LR is efficient for tabular data, its reliance on linear decision boundaries makes it ill-equipped to handle the relationships that exist between medication types and continuous variables like dosage and frequency. Because LR aggregation is performed by averaging the parameters of the local models, the rarity of the positive (anomaly) class means that local trainer nodes may compute highly different coefficients depending on their specific data partitions. However, because LR bases its decisions on probabilistic thresholds, it tends to favor the majority class in datasets with severe class imbalance [32], so averaging these coefficients during aggregation degrades the global model's discriminative power. In the dataset under consideration, where over 98% of the samples belong to the negative class, the aggregated LR model overwhelmingly defaults to the majority class, resulting in the significant volume of false negatives and low F_1-score observed.

As illustrated in Fig. 6, the LR model produces a large number of false positives and false negatives, along with a relatively few true positives, resulting in significantly low recall and F_1-score values. The MLP model obtained a higher number of true positives, resulting in improved precision, recall, and F_1-score metrics. The DT model achieved the best performance under this highly imbalanced scenario, demonstrating greater precision and recall. Its hierarchical structure results in increased sensitivity to the minority class, with few false positives and false negatives.

In summary, the decision tree approach proved to be the most suitable technique for medical prescription anomaly detection in this dataset. As shown in Table VIII, the decentralized training process did not result in a significant loss of performance. Based on the F_1-score metric, the LR model trained under the proposed decentralized system exhibits a difference of less than 5% compared to its centralized counterpart. Similarly, the MLP and the DT models showed a performance difference below 1% relative to the traditional centralized version.

To evaluate the models' discriminative capacity, we have included Receiver Operating Characteristic (ROC) curves for the decentralized models (Fig. 5). These corroborate our findings: the DT exhibits near-perfect discriminative ability across thresholds, while the LR and MLP models don't separate the classes as effectively. It is important to note that while the feature space is of low dimensionality, these attributes represent the complete set of variables utilized in the original hospital production environment [28]. Using these features ensures that our evaluation reflects a real-world clinical use case.

While the DT model achieved strong performance under this highly imbalanced dataset, electronic medical records often contain complex, high-dimensional data, where the explicit hierarchical boundaries that advantage the DT might not generalize as well. Our methodology incorporated an isolated held-out test set and a decentralized training process that

TABLE VIII
COMPARISON OF PERFORMANCE METRICS ACROSS EXPERIMENTS

Training	Model	Accuracy	Precision	Recall	F_1-Score
Decentralized (ours)	**LR**	0.9859	0.7083	0.1475	*0.2442 ± 0.0793*
	MLP	0.9900	0.8945	0.3975	*0.5504 ± 0.0043*
	DT	0.9997	0.9964	0.9861	*0.9912 ± 0.0015*
Centralized	**LR**	0.9857	0.6056	0.1892	0.2884
	MLP	0.9901	0.8949	0.3993	0.5522
	DT	0.9998	1.0000	0.9861	0.9930

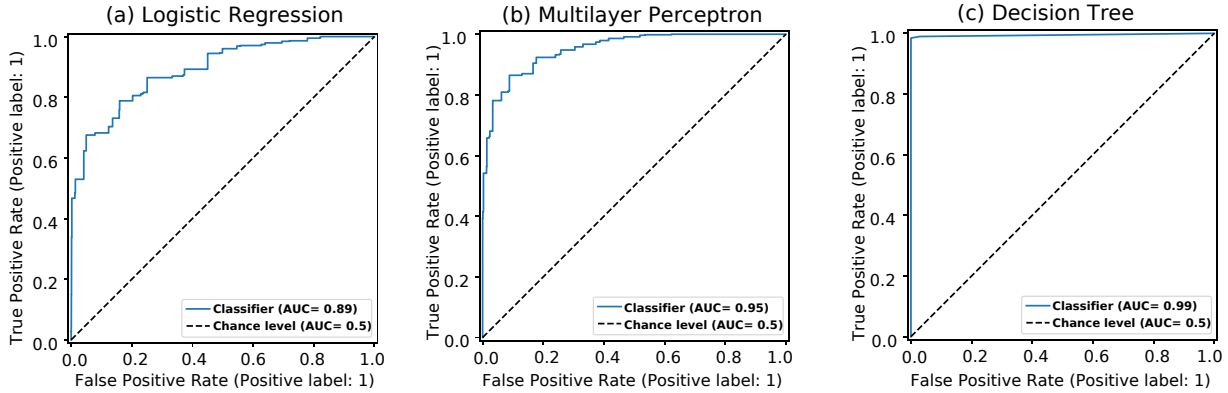


Fig. 5. ROC curves on the test set for (a) Logistic Regression, (b) Multi-layer Perceptron, (c) Decision Tree.

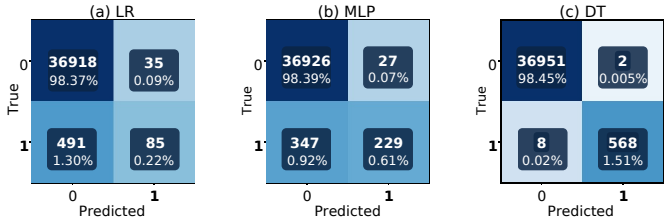


Fig. 6. Confusion matrices on the test set for (a) LR, (b) MLP, (c) DT.

functions as a form of distributed cross-validation to mitigate overfitting.

The generalizability lies primarily in the architecture itself, rather than the specific performance of the DT. The proposed decentralized training and aggregation framework is model-agnostic. As demonstrated by our multi-model evaluations, the system supports different aggregation strategies, such as parameter-averaging and majority-voting ensemble schemes. As data complexity increases in real-world clinical scenarios, the architecture can integrate models with a higher capacity for complex feature representation. While network performance may exhibit some variation depending on the specific algorithm deployed, the overall feasibility and applicability of the decentralized approach remain evident.

V. Conclusion

This study proposed a decentralized training architecture designed to mitigate single points of failure inherent to traditional FL and eliminate the reliance on centralized parameter aggregation through the integration of IPFS for decentralized storage and data sharing. This design simplifies the synchronization mechanisms compared to SL, improving scalability, fault tolerance, and system robustness.

Experimental results on the medical prescription anomaly detection task demonstrated that the proposed approach outperformed the baseline federated system across most performance metrics, such as latency and resource usage, despite a slight increase in training round duration and a decrease in transactions per second due to architectural complexity. Scalability tests confirmed stable latency behavior across all models, while the decision tree (DT) model showed the best overall efficiency.

In classification performance, logistic regression (LR) and multi-layer perceptron (MLP) were less suitable for the highly imbalanced dataset (0.2442 and 0.5504 F_1-score, respectively), whereas the decision tree (DT) model achieved superior results (F_1 = 0.9912). Notably, the decentralized setup introduced minimal performance degradation compared to centralized training: less than 4% for LR and below 1% for both MLP and DT.

Despite strong performance, the system's reliance on IPFS introduces network latency variability depending on node availability and connectivity. To address this and further advance the architecture, future research will focus on mitigating IPFS-induced variability through node replication and cache optimization strategies. We intend to expand scalability evaluations by testing the framework under higher transaction volumes with a significantly larger network of participating nodes to identify upper-bound performance limits.

Future work will evaluate the architecture's resilience against adversarial threats, such as data poisoning and Byzantine node behaviors. We also intend to explore reputation-based mechanisms. These will strengthen trust, auditability, and traceability within the decentralized network. Additionally, the proposed architecture will be validated in real-world clinical environments. This evaluation will require the development of robust integration interfaces to connect the decentralized framework with legacy, centralized medical record systems, ensuring interoperability while strictly adhering to data protection regulations.

Acknowledgments

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001. The Article Processing Charge for the publication of this research was also covered by CAPES (id ROR: 00x0ma614).

REFERENCES

[1] E. T. V. de Castro, G. R. Silva, and E. D. Canedo, "Ensuring privacy in the application of the brazilian general data protection law (lgpd)," in *Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing*, 2022, pp. 1228–1235. doi: 10.1145/3477314.3507023 .

[2] M. V. P. Prestes, D. M. S. Bonini, F. C. d. Melo, M. Bastos, J. S. Bonini, and W. C. F. N. d. Silva, "General data protection law nº 13.709/2018: notes on its context as a legal framework in brazil," *Research, Society and Development*, vol. 10, no. 12, p. e568101220906, 2021. doi: 10.33448/rsd-v10i12.20906 .

[3] C. G. Granmar, "Global applicability of the gdpr in context," *International Data Privacy Law*, vol. 11, no. 3, pp. 225–244, 06 2021. doi: 10.1093/idpl/ipab012 .

[4] E. Ates Bulut and A. T. ISIK, "Abuse/misuse of prescription medications in older adults," *Clinics in Geriatric Medicine*, vol. 38, no. 1, pp. 85–97, 2022. doi: 10.1016/j.cger.2021.07.004 .

[5] S. Chatterjee and A. Biswas, "An assessment of legibility and rationality of prescriptions through prescription audit," *QAI Journal for Healthcare Quality and Patient Safety*, vol. 5, no. 2, pp. 30–35, 2024. doi: 10.4103/QAIJ.QAIJ_14_24 .

[6] K. B. Johnson, W.-Q. Wei, D. Weeraratne, M. E. Frisse, K. Misulis, K. Rhee, J. Zhao, and J. L. Snowdon, "Precision medicine, ai, and the future of personalized health care," *Clinical and translational science*, vol. 14, no. 1, pp. 86–93, 2021. doi: 10.1111/cts.12884 .

[7] A. Rashidian, H. Joudaki, and T. Vian, "No evidence of the effect of the interventions to combat health care fraud and abuse: a systematic review of literature," *PLOS ONE*, vol. 7, no. 8, pp. 1–8, 08 2012. doi: 10.1371/journal.pone.0041988 .

[8] J. D. E. Caro and C. Fajardo, "Predicting shock in pediatric patients through thermal gradients and machine learning: A multi-model approach," *IEEE Latin America Transactions*, vol. 24, no. 2, pp. 144–152, 2026. doi: 10.1109/TLA.2026.11369471 .

[9] A. Vidales-Esquivel, F. Ornelas-Tellez, and J. Ortiz-Bejar, "Harmonic analysis and pattern classification of electrocardiograms for heart disease diagnosis," *IEEE Latin America Transactions*, vol. 24, no. 2, pp. 164–173, 2026. doi: 10.1109/TLA.2026.11369468 .

[10] J. M. Johnson and T. M. Khoshgoftaar, "Medicare fraud detection using neural networks," *Journal of Big Data*, vol. 6, no. 1, pp. 1–35, 2019. doi: 10.1186/s40537-019-0225-0 .

[11] J. G. Almaraz-Rivera, "An anomaly-based detection system for monitoring kubernetes infrastructures," *IEEE Latin America Transactions*, vol. 21, no. 3, pp. 457–465, 2023. doi: 10.1109/TLA.2023.10068850 .

[12] C. Zhang, X. Xiao, and C. Wu, "Medical fraud and abuse detection system based on machine learning," *International journal of environmental research and public health*, vol. 17, no. 19, p. 7265, 2020. doi: 10.3390/ijerph17197265 .

[13] S. Tanwar, K. Parekh, and R. Evans, "Blockchain-based electronic healthcare record system for healthcare 4.0 applications," *Journal of Information Security and Applications*, vol. 50, p. 102407, 2020. doi: 10.1016/j.jisa.2019.102407 .

[14] R. Y. Gupta, S. S. Mudigonda, and P. K. Baruah, "A comparative study of using various machine learning and deep learning-based fraud detection models for universal health coverage schemes," *International Journal of Engineering Trends and Technology*, vol. 69, no. 3, pp. 96–102, 2021. doi: 10.14445/22315381/IJETT-V69I3P216 .

[15] E. Moore, A. Imteaj, S. Rezapour, and M. H. Amini, "A survey on secure and private federated learning using blockchain: Theory and application in resource-constrained computing," *IEEE Internet of Things Journal*, vol. 10, no. 24, pp. 21 942–21 958, 2023. doi: 10.1109/JIOT.2023.3313055 .

[16] Y. Li, C. Chen, N. Liu, H. Huang, Z. Zheng, and Q. Yan, "A blockchain-based decentralized federated learning framework with committee consensus," *IEEE Network*, vol. 35, no. 1, pp. 234–241, 2020. doi: 10.1109/MNET.011.2000263 .

[17] Y. Qu, M. P. Uddin, C. Gan, Y. Xiang, L. Gao, and J. Yearwood, "Blockchain-enabled federated learning: A survey," *ACM Computing Surveys*, vol. 55, no. 4, pp. 1–35, 2022. doi: 10.1145/3524104 .

[18] J. Han, Y. Han, Y. Zhang, X. Jing, and Y. Ma, "Demystifying swarm learning: An emerging decentralized federated learning system," in *2024 IEEE 24th International Symposium on Cluster, Cloud and Internet Computing (CCGrid)*. IEEE, 2024, pp. 367–373. doi: 10.1109/CCGrid59990.2024.00045 .

[19] S. Warnat-Herresthal, H. Schultze, K. L. Shastry, S. Manamohan, S. Mukherjee, V. Garg, R. Sarveswara, K. Händler, P. Pickkers, N. A. Aziz *et al.*, "Swarm learning for decentralized and confidential clinical machine learning," *Nature*, vol. 594, no. 7862, pp. 265–270, 2021. doi: 10.1038/s41586-021-03583-3 .

[20] H. A. Madni, R. M. Umer, and G. L. Foresti, "Blockchain-based swarm learning for the mitigation of gradient leakage in federated learning," *IEEE Access*, vol. 11, pp. 16 549–16 556, 2023. doi: 10.1109/ACCESS.2023.3246126 .

[21] E. Hallaji, R. Razavi-Far, M. Saif, B. Wang, and Q. Yang, "Decentralized federated learning: A survey on security and privacy," *IEEE Transactions on Big Data*, vol. 10, no. 2, pp. 194–213, 2024. doi: 10.1109/TBDATA.2024.3362191 .

[22] R. Kumar, A. A. Khan, J. Kumar, N. A. Golilarz, S. Zhang, Y. Ting, C. Zheng, W. Wang *et al.*, "Blockchain-federated-learning and deep learning models for covid-19 detection using ct imaging," *IEEE Sensors Journal*, vol. 21, no. 14, pp. 16 301–16 314, 2021. doi: 10.1109/JSEN.2021.3076767 .

[23] L. Settipalli and G. Gangadharan, "Qfbn: Quorum based federated blockchain network for healthcare system to avoid multiple benefits and data breaches," *IEEE Consumer Electronics Magazine*, vol. 13, no. 2, pp. 24–35, 2022. doi: 10.1109/MCE.2022.3188880 .

[24] M. J. Baucas, P. Spachos, and K. N. Plataniotis, "Federated learning and blockchain-enabled fog-iot platform for wearables in predictive healthcare," *IEEE Transactions on Computational Social Systems*, vol. 10, no. 4, pp. 1732–1741, 2023. doi: 10.1109/TCSS.2023.3235950 .

[25] G. A. Zutião, "Detecção de anomalias em prescrições médicas com aprendizagem federada e gerenciamento de armazenamento em blockchain," Master's Thesis in Computer Science and Computational Mathematics, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2023. doi: 10.11606/D.55.2023.tde-19122023-122357 .

[26] O. L. Saldanha, H. S. Muti, H. I. Grabsch, R. Langer, B. Dislich, M. Kohlruss, G. Keller, M. van Treeck, K. J. Hewitt, F. R. Kolbinger *et al.*, "Direct prediction of genetic aberrations from pathology images in gastric cancer with swarm learning," *Gastric cancer*, vol. 26, no. 2, pp. 264–274, 2023. doi: 10.1007/s10120-022-01347-0 .

[27] D. Gana and F. Jamil, "Dag-based swarm learning approach in healthcare: A survey," *IEEE Access*, vol. 13, pp. 13 796–13 815, 2025. doi: 10.1109/ACCESS.2025.3531216 .

[28] H. D. Dos Santos, A. H. D. Ulbrich, V. Woloszyn, and R. Vieira, "Ddc-outlier: preventing medication errors using unsupervised learning," *IEEE journal of biomedical and health informatics*, vol. 23, no. 2, pp. 874–881, 2018. doi: 10.1109/JBHI.2018.2828028 .

[29] J. Jayabalan and N. Jeyanthi, "Scalable blockchain model using off-chain ipfs storage for healthcare data security and privacy," *Journal of Parallel and distributed computing*, vol. 164, pp. 152–167, 2022. doi: 10.1016/j.jpdc.2022.03.009 .

[30] C. Fan, C. Lin, H. Khazaei, and P. Musilek, "Performance analysis of hyperledger besu in private blockchain," in *2022 IEEE International Conference on Decentralized Applications and Infrastructures (DAPPS)*, 2022, pp. 64–73. doi: 10.1109/DAPPS55202.2022.00016 .

[31] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 2623–2631. doi: 10.1145/3292500.3330701 .

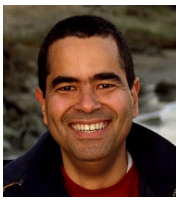
[32] H. Luo, X. Pan, Q. Wang, S. Ye, and Y. Qian, "Logistic regression and random forest for effective imbalanced classification," in *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*, vol. 1. IEEE, 2019, pp. 916–917. doi: 10.1109/COMPSAC.2019.00139 .

Ravelly C. Zanatta is a M.Sc. student at the Institute of Mathematical and Computer Sciences of the University of São Paulo (ICMC-USP). He received his B.S. degree in Applied and Computational Mathematics from the State University of Central-West (UNICENTRO), in Guarapuava, PR, Brazil, in 2019. His research interests include computer networks, blockchain, machine learning, and data science.

Vinícius J. B. Vanzin received the B.S. degree in computer engineering from UNIFTEC, Caxias do Sul, RS, Brazil in 2018. He is currently pursuing the Ph.D. degree in computer science at University of São Paulo, São Carlos, SP, Brazil. His research interests include machine learning applied to the biomedical sciences, entity normalization and structuring, interoperability of healthcare information, natural language processing, and large language models.

Saulo N. Matos is a Ph.D. student at the Institute of Mathematical and Computer Sciences of the University of São Paulo (ICMC-USP), with research focused on Machine Learning, Instrumentation, and Internet of Things. He holds a master's degree in Instrumentation, Control, and Automation of Mining Processes from the University of Ouro Preto and the Vale Technological Institute (2022). He graduated in Control and Automation Engineering from the Federal University of Ouro Preto (2020). He has published papers and patents on instrumentation, embedded systems, machine learning, and control theory.

Rodrigo D. Garcia is a Ph.D. candidate at the Institute of Mathematical and Computer Sciences of the University of São Paulo (ICMC-USP), where he also received his M.Sc. He is currently working as a research visitor at the University of Southern California (USC). His research interests include privacy-preserving protocols and blockchain applications.



Dilvan A. Moreira received the Ph.D. degree in electronics engineering from the University of Kent at Canterbury, UK., in 1995. He conducted post-doctoral research in biomedical informatics with Stanford University, in 2008. He is currently an Associate Professor at the University of São Paulo (USP), Brazil. His research interests include biomedical informatics, microelectronics, and data science applications in healthcare.

Jó Ueyama is a Full Professor at the Institute of Mathematical and Computer Sciences (ICMC) of the University of São Paulo (USP). He has been a Brazilian Research Council (CNPq) fellow since 2014. He received his Ph.D. in computer science from the University of Lancaster in 2006 and was a research fellow at the University of Kent at Canterbury before joining USP. Jó has a publication record with 72 journal articles and over 100 conference papers. His research interests are focused on computer networks, security, and blockchain.