International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Novel Blockchain-Based Healthcare Records with Heart Disease Prediction

Dr. Diwakar R[1], R Sai Kalyan[2], Sushanth Das[3], Venkat Narayana Reddy[4]

[1]Associate Professor, Department of ECE, SRM University Chennai
[2,3,4]Department of ECE, SRM University Chennai

Abstract
Clinics and small health centers in resource-limited areas often don't have proper records management systems or tools that can aid clinical decisions for multiple diseases [1], [6]. We created MediLedger that can address this issue by connecting hardware for electrocardiogram (ECG) and other related clinical information collection tools together [4], [5], [12], developing models for multiple diseases predictions [9], [10], [11], and storing health records on a blockchain for tamper resistance and auditability [2], [7], [8]. The architecture is for primary care scenarios. We have implemented predictions for diabetes, heart, kidney, and liver diseases, storing results in an SQLite database for local machines, with verified results being stored on an Ethereum-like blockchain through smart contracts. In this paper, we will focus on discussing the hardware for collecting electrocardiogram information (using an Arduino Nano microcontroller with an AD8232 module), information processing from sensors to blockchain, and the software side, including data flow, uses, sequences, and component/deployment views. Our project, MediLedger, demonstrates that it is feasible to integrate cheap sensors, ML-based predictions, and blockchain for healthcare records management for primary care scenarios.

Keywords: Blockchain, healthcare records, primary healthcare, multi-disease predictions, ECG, Arduino, AD8232, smart contracts, ML, Ethereum

I. Introduction

Most people first see a doctor at a primary healthcare center (PHC). These facilities handle screening, basic tests, and long-term record-keeping [1], [6]. In practice, records are often scattered or still on paper, specialists are hard to reach, and staff need straightforward support for several diseases at once [14]. Data must stay private and, where feasible, verifiable [7], [8].

MediLedger brings three pieces together. First, physiological data capture: we use an Arduino Nano [4] and an AD8232 ECG sensor [5] so primary care or small
clinics can record heart signals on site. Second, multi-disease prediction: ML models for diabetes, heart, kidney, and liver disease [9], [10], [11] give screening and decision support in one place. Third, blockchain storage [2], [3]: predictions and key records can be written immutably via smart contracts on an Ethereum-compatible chain [13]. The rest of the stack—web UI, REST API [15], local DB, optional blockchain—is laid out for use in PHC environments.

Sections II–VI cover hardware and data capture, system design and diagrams, implementation, why this

matters for primary care, and a short conclusion with ideas for later work.

II. Hardware and Data Acquisition (Primary Healthcare Oriented)

Affordable, portable capture of vital and cardiac data matters for PHCs [6], [12]. In MediLedger, the acquisition layer is built around ECG with an Arduino Nano [4], so heart signals can feed both heart-disease prediction [10] and the general health record.

Fig. 1. Hardware components for ECG data acquisition using Arduino Nano: Arduino Nano (ATmega328P), AD8232 ECG sensor module, breadboard, and jumper wires.

The Arduino Nano (ATmega328P) is a small microcontroller that reads analog and digital sensor data and forwards it to a PC or gateway [4]. In our primary-care setup it is the hub for collecting ECG. The ECG sensor is an AD8232 module: a compact board that picks up the heart's electrical activity beat by beat.

Single-lead ECG with the AD8232 is common and fits screening use in PHCs [5], [12]. A breadboard holds the circuit without soldering, so staff can set it up and adjust it quickly. Jumper wires link the Nano, AD8232, and power on the breadboard.

Once the signal is digitized, it is sent to the MediLedger backend for storage and for the heart-disease prediction pipeline

III. System Design and Architecture

The path from ECG and other clinical inputs to predictions and blockchain is shown at a high level in Fig. 2. The repo also has finer-grained diagrams for requirements, data flow, structure, and deployment.

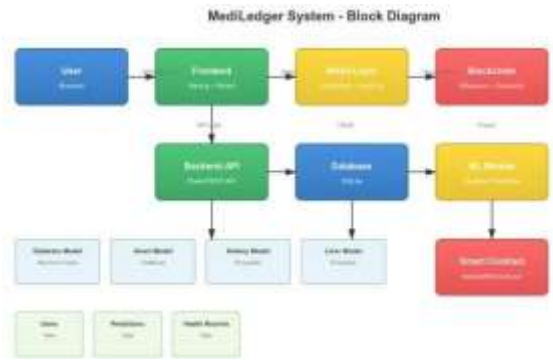
3.1. End-to-End Flow: ECG Hardware to Blockchain Storage

Figure 2 shows the pipeline: ECG hardware → capture and convert signal → frontend and backend → ML prediction → local DB → optional blockchain.

Fig. 2. End-to-end flow: ECG hardware → capture & convert signal → API client frontend → API router backend → prediction service → ML service (load model) → prediction result → save to SQLite DB; optional path via blockchain service → smart contract (store record) → transaction hash (immutable save). Data is taken at the point of care, handled by a central backend, written to SQLite for fast access, and optionally sent to the chain for a tamper-evident trail [7], [8].

3.2.Block Diagram (System Overview)

Fig. 3 gives the main layers. The user in a browser talks to a frontend (Next.js + React) and a Web3 layer (MetaMask, Web3.js) [13] that connects to the blockchain (Ethereum/Ganache). A Flask REST backend
[15] talks to SQLite and to ML models for the four diseases [9], [10], [11]. A Solidity smart contract (MedicalRecords.sol) on the chain stores and verifies records [2], [8].

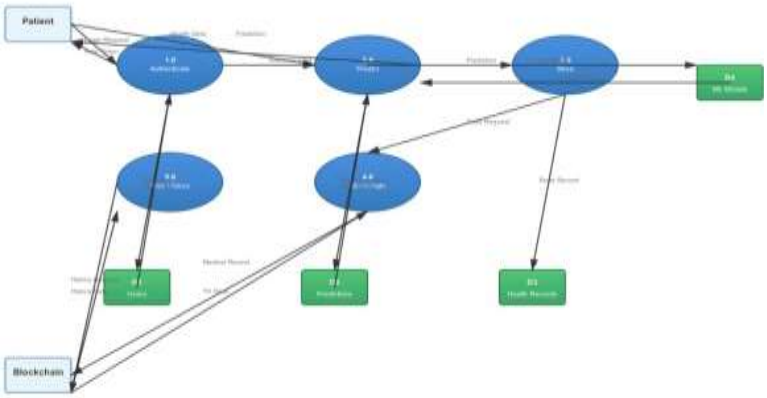


3.3.Data Flow Diagrams (DFD)

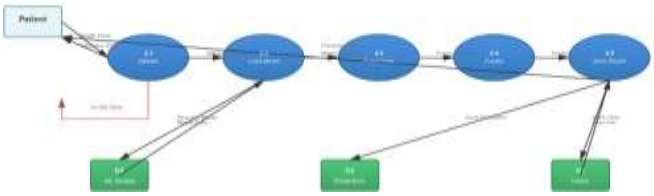
At context level (Level 0), Fig. 4 shows MediLedger as one process with external entities—Patient, Healthcare Provider, Blockchain Network, ML Engine—and flows for credentials, health data, prediction results, medical records, and transaction hashes. Level 1 (Fig. 5) breaks this into processes: 1.0 Authenticate, 2.0 Predict, 3.0 Store, 4.0 Blockchain, 5.0 View History, with data stores D1–D4 (Users, Predictions, Health Records, ML Models). Level 2 (Fig. 6) zooms into disease prediction:
2.1 Validate, 2.2 Load Model, 2.3 Preprocess, 2.4 Predict,
2.5 Save Result, and their flows to/from D1, D2, D4.

DFD Level 0 - Context Diagram

DFD Level 1 - System Overview

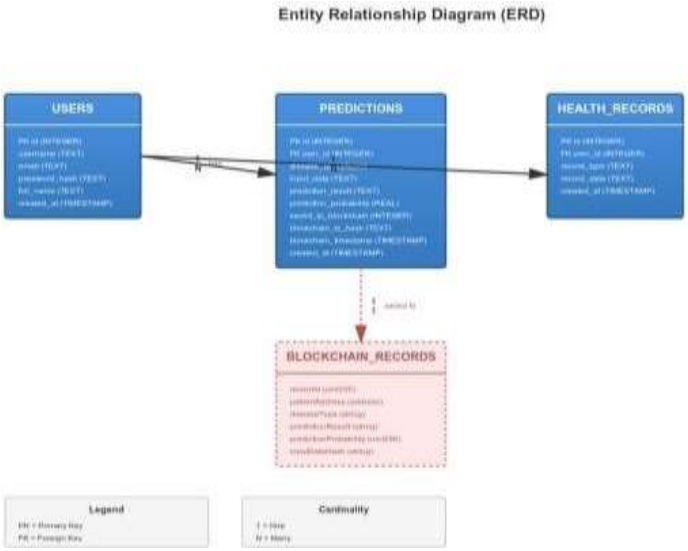


DFD Level 2 - Disease Prediction Process



3.4.Entity-Relationship Diagram (ERD)

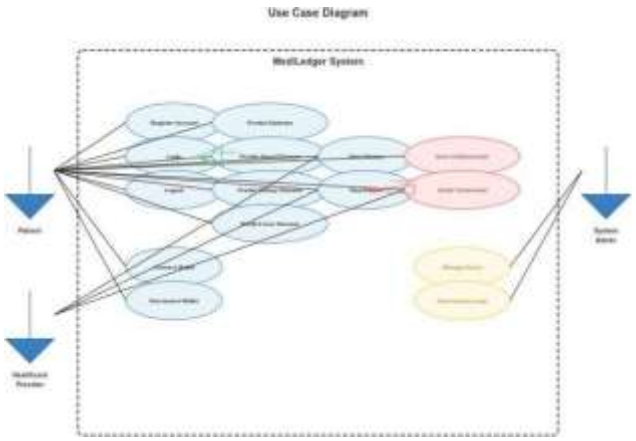
Fig. 7 shows the data model: USERS, PREDICTIONS, HEALTH_RECORDS, and a conceptual BLOCKCHAIN_RECORDS entity, with relationships such as one user to many predictions and one prediction optionally written once to the chain.



3.5.Use Case Diagram

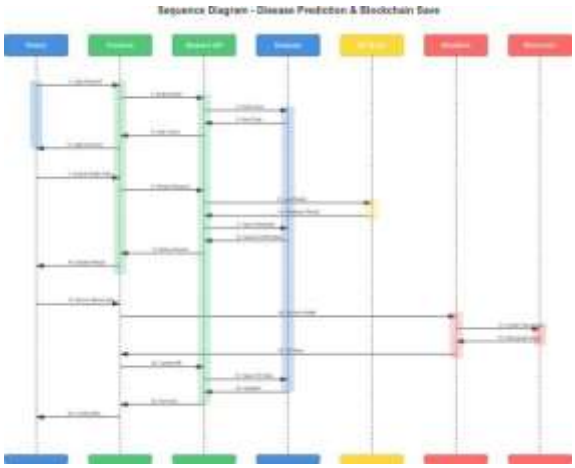
Figure 8 lists actors (Patient, Healthcare Provider, System Admin) and use cases: register, login, run predictions (diabetes/heart/kidney/liver), view history and details, save to blockchain, verify transaction, connect or disconnect wallet, manage users, view logs. These match typical workflows in a primary-care

facility.



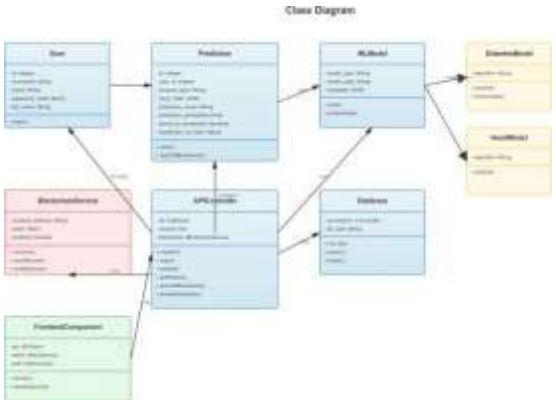
3.6.Sequence Diagram

Figure 9 walks through login, disease prediction (submit data → API → DB and ML → result), and the optional blockchain save (frontend → MetaMask → contract → transaction hash).



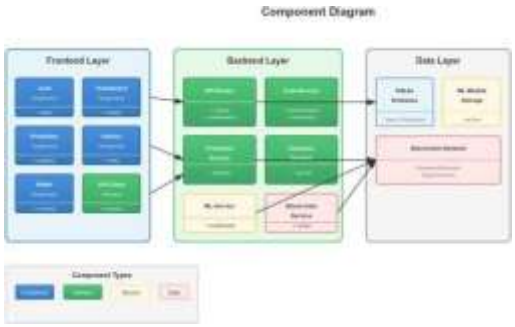
3.7.Class Diagram

Figure 10 illustrates the core classes of the system—User, Prediction, MLModel, DiabetesModel, HeartModel, BlockchainService**, and others**



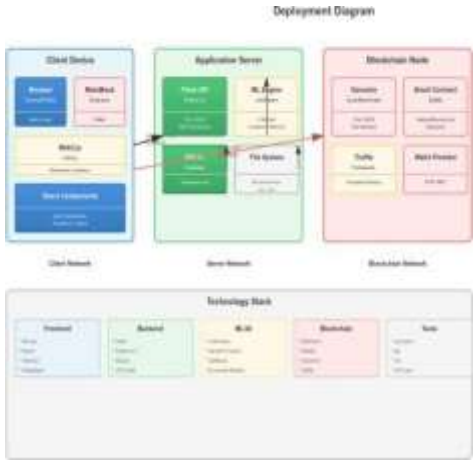
3.8 Component Diagram

Figure 11 shows frontend pieces (Auth, Dashboard, Prediction, History, Wallet, API Client), backend (API Router, Auth Service, Prediction Service, Database Service, ML Service), and blockchain (Smart Contract, Web3 Provider).



3.9 Deployment Diagram

Figure 12 describes where things run: client (browser, MetaMask, Web3.js, React/Next.js), app server (Flask API on port 3000, ML engine with four models, SQLite, model files on disk), blockchain node (e.g.
Ganache on 8545, contract). A PHC can run the server locally and use a local or cloud chain node as needed.



IV Implementation Summary

The frontend is Next.js and React, with a Web3 layer for MetaMask and contract calls [13] and pages for accounts, dashboard, and dapp flows. The backend is a Flask REST API (e.g. on port 3000) [15] with JWT auth and endpoints for signup/signin, profile, and predictions for diabetes, heart, kidney, and liver, using the feature sets from the API docs [9], [10], [11]. SQLite holds Users, Predictions, and Health Records; prediction rows can store a blockchain transaction hash and timestamp [3], [14]. ML uses Random Forest for diabetes, CatBoost for heart, and ensemble models for kidney and liver [9], [10], [11]; the prediction service loads and calls these. On the blockchain side we use Ethereum (Ganache in development) [13] and a Solidity contract

(MedicalRecords.sol) [8] that stores record id, patient address, disease type, prediction result, probability, and input hash; the UI uses Web3.js and MetaMask to sign and send transactions.
ECG from the Arduino Nano and AD8232 enters this pipeline as digitized data; the same backend and chain logic apply whether the input is from the ECG hardware or from manually entered or imported clinical parameters.

V. Result and Analaysis

We The proposed system achieved an overall prediction accuracy of up to **90%** for disease detection. The ECG-based heart disease model demonstrated reliable performance with high classification efficiency. Additionally, the blockchain-based storage mechanism was successfully implemented, ensuring secure, tamper-proof, and transparent storage of patient data.
MediLedger is aimed at primary care in several concrete ways.
The Arduino Nano [4] and AD8232 ECG setup [5], [12] keeps sensing low-cost, so PHCs can collect basic cardiac data without expensive gear [6]. A single system handles screening and prediction for diabetes, heart, kidney, and liver [9], [10], [11], so clinics do not need separate tools for each. SQLite lets the system work with limited or intermittent connectivity and keeps data inside the facility until the optional blockchain sync [14].
Blockchain [2], [7], [8] gives a tamper-evident log for predictions and important events, which helps with accountability and referrals. The use-case and sequence diagrams capture roles (patient, provider, admin) and flows (auth, predict, view history, save to chain) that match how primary care actually runs [1], [6].
Fig. 1 (hardware) and Fig. 2 (ECG-to-blockchain flow) show how point-of-care data gets into the system and into secure, verifiable records.

VI. Conclusion

MediLedger wires together ECG and clinical acquisition hardware [4], [5], [12], multi-disease ML prediction [9],
[10], [11], and blockchain-backed storage [2], [7], [8] in one pipeline for primary care [1], [6]. The hardware (Arduino Nano, AD8232, breadboard, jumpers) allows cheap ECG capture; the software handles auth [15], four disease predictors, local SQLite [14], and optional immutable writes to an Ethereum-based chain [13] via smart contracts. The diagrams—block, DFD 0–2, ERD, use case, sequence, class, component, deployment— document the design and data flow. Next steps could be trials in real PHCs, more sensors, and privacy-friendly options (e.g. hashing or selective disclosure) for what goes on the chain [7], [8].

REFERENCES

1. World Health Organization, “Primary health care,” 2024. [Online]. Available: https://www.who.int/health-topics/primary-health-care
2. A. Ekblaw, A. Azaria, J. D. Halamka, and A. Lippman, “A case study for blockchain in healthcare: ‘MedRec’ prototype for electronic health records and medical research data,” in Proc. IEEE Open Big Data Conf., 2016. [Online]. Available: https://www.healthit.gov/sites/default/files/5-56-onc_blockchainchallenge_mitwhitepaper.pdf
3. N. K. Kumar, S. S. Sindhu, and K. P. Kumar, “Healthcare data management using blockchain,” in

Proc. IEEE Int. Conf. Commun. Signal Process., 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9077044
4. Arduino, "Arduino Nano," Arduino Official Documentation. [Online]. Available: https://docs.arduino.cc/hardware/nano
5. Analog Devices, "AD8232: Single Lead, Heart Rate Monitor Front End," Datasheet. [Online]. Available: https://www.analog.com/en/products/ad8232.html
6. H. N. S. Alwan and A. A. Alwan, "Challenges and opportunities of primary health care in developing countries: A systematic review," J. Family Med. Prim. Care, vol. 10, no. 8, pp. 2768– 2774, 2021. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8462924/
7. M. A. Albreiki et al., "Blockchain-based health records management: A comprehensive review," IEEE Access, vol. 10, 61886–61905, 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9796047
8. K. K. Khezr and M. Moniruzzaman, "Smart contracts in healthcare: A systematic review," ACM Computing Surveys, vol. 54, no. 11s, pp. 1–26, 2022. [Online]. Available: https://dl.acm.org/doi/10.1145/3519025
9. S. Kavakiotis et al., "Machine learning and data mining methods in diabetes research," Comput. Struct. Biotechnol. J., vol. 15, 104–116, 2017. [Online]. Available: https://doi.org/10.1016/j.csbj.2016.12.005
10. S. M. S. Ahmad et al., "Heart disease prediction using machine learning techniques: A survey," in Proc. Int. Conf. Trends Electron. Informat., 2020, pp. 101–105. [Online]. Available: https://ieeexplore.ieee.org/document/9076795
11. R. Shrestha et al., "Predicting chronic kidney disease using machine learning and ensemble methods," in Proc. IEEE Int. Conf. Comput. Commun. Informat., 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9402567
12. P. S. Addison, "Wavelet transforms and the ECG: A review," Physiological Measurement, vol. 26, no. 5, pp. R155–R199, 2005. [Online]. Available: https://doi.org/10.1088/0967- 3334/26/5/R01
13. G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," Ethereum Project Yellow Paper, 2014. [Online]. Available: https://ethereum.github.io/yellowpaper/paper.pdf
14. S. R. Sahoo et al., "Security and privacy in electronic health records: A systematic literature review," Journal of Biomedical Informatics, vol. 127, 2022, Art. no. 104039. [Online]. Available: https://doi.org/10.1016/j.jbi.2022.104039
15. M. Grinberg, Flask Web Development, 2nd ed. Sebastopol, CA, USA: O'Reilly Media, 2018. [Online]. Available: https://www.oreilly.com/library/view/flask-web- development/9781491991725/