Citation:

Rehman, Ateeq Ur, Nargis Tariq, Mian Ahmad Jan, Fazlullah Khan, Houbing Song, and Muhammad Ibrahim. "A Blockchain-Based Hybrid Model for IoMT-Enabled Intelligent Healthcare System." IEEE Transactions on Network Science and Engineering, 2024, 1–9. https://doi.org/10.1109/TNSE.2024.3376069.

DOI:

https://doi.org/10.1109/TNSE.2024.3376069

# A Blockchain-based Hybrid Model for IoMT-enabled Intelligent Healthcare System

Ateeq Ur Rehman+, *Member IEEE,* Nargis Tariq+, *Student Member IEEE,* Mian Ahmad Jan, *Senior Member IEEE,* Fazlullah Khan*, *Senior Member IEEE,* Houbing Song, *Fellow IEEE,* and Muhammad Ibrahim, *Member IEEE,*

**Abstract**—In recent years, the healthcare industry has undergone a digital transformation, making patient data publicly available and accessible. Healthcare units make a portion of the data public while keeping the rest private, necessitating various mechanisms for security and privacy. Blockchain technology has been widely adopted in the healthcare sector to secure data transactions. However, public blockchains face challenges in scalability and privacy, whereas private blockchains struggle with centralization, interoperability, and complexity. To address these challenges, we propose an Internet of Medical Things (IoMT)-based hybrid blockchain architecture. The proposed architecture combines the decentralized Ethereum and the centralized Hyperledger Fabric blockchain (Eth-Fab) using SQLite to leverage Ethereum smart contracts with the Hyperledger permission model. Moreover, we introduce access control strategies to enhance patient data authentication and authorization. We have employed machine learning algorithms to assist healthcare practitioners in accurately detecting diseases and making time-efficient decisions. Additionally, we modeled the proposed architecture using the M/M/1 queuing model and derived closed-form expressions for latency, throughput, and server utilization. The validity of these expressions was verified through Monte Carlo simulations. The results demonstrate that higher service times (block generation) yield better outcomes in terms of latency, throughput, and utilization, regardless of the arrival time, i.e., transactions in the mining pool.

**Index Terms**—Internet of Medical Things, Hyperledger Fabric, Ethereum, Security, privacy, Machine Learning, M/M/1 Queuing Model, Monte Carlo.

## 1 Introduction

INTERNET of Medical Things (IoMT) and machine learning (ML) have become integral in the healthcare industry, providing a sophisticated platform for the interaction of various entities. IoMT facilitates global access to hospitals for online healthcare services and data processing. The industry is anticipated to reach a market value of USD 390.7 billion by 2024 [?]. IoMT generates an immense volume of medical data, which can be collected and analyzed swiftly and cost-effectively using ML algorithms. Additionally, real-time data processing in IoMT is enhanced by the availability of smart wearables in the market [?]. However, concerns have been raised regarding the security of these wearables, as they may not offer sufficient protection for patients' health records [?]. Recent statistics from the United States have reported about 20,030 health-related data breaches, impacting approximately 199.3 million records [?]. These statistics highlight the significant risk to healthcare data, underscoring the necessity to investigate vulnerabilities and develop effective solutions.

In conventional IoMT-enabled healthcare systems, centralized server-based solutions are predominantly employed for storing and sharing health information, where security measures rely on the assumption that the IoMT server is trustworthy [?]. This approach, however, introduces a single point of failure; if compromised, it can jeopardize the integrity of the entire system. Blockchain technology has emerged as a prominent solution for securing and preserving healthcare data, with both public and private (permission-based) blockchains being widely adopted. Public blockchains, such as those based on Ethereum, offer transparency but face challenges in scalability and privacy due to their open network structure. Conversely, permission-based blockchains like Hyperledger offer a more controlled environment but struggle with issues of centralization and complexity.

Notable efforts in the literature include the work by the authors in [?] and [?] proposed models for secure access, storage, and sharing of patient health information using Ethereum and the Interplanetary File System (IPFS). Additionally, [?] utilized Ethereum, Django, and IPFS for managing medical records, while [?] proposed FedBlockHealth approach to preserve privacy and security by combining federated learning and Ethereum blockchain. Similarly, [?] combined real-time data and predictive analytics with Hyperledger Fabric to facilitate data processing, analysis, and storage for veterinary clinic data, ensuring access is limited to authorized participants. Despite these advancements,

*Ateeq Ur Rehman is with the Department of Computing, Staffordshire University, United Kingdom. E-mail:{ ateequr.rehman@staffs.ac.uk*
*N. Tariq and M. Ibrahim are with the Department of Information Technology, University of Haripur, Pakistan. E-mail:{nargis.tariq, m.ibrahim}@uoh.edu.pk*
*M. A. Jan is with the Department of Computer Science, College of Computing and Informatics, University of Sharjah, Sharjah, 27272, UAE Email: mjan@sharjah.ac.ae*
*F. Khan is with the School of Computer Science, Faculty of Science and Engineering, University of Nottingham Ninbo China, Ningbo 315104, Zhejiang, China. E-mail: fazl.ullahg@nottingham.edu.cn*
*Houbing Song is with the College of Engineering and Information Technology, University of Maryland, United States. E-mail: songh@umbc.edu*
*+ shows first-equal authors and * shows corresponding author.*

most existing solutions focus either on public or private blockchains, each with distinct limitations. For instance, private blockchains restrict data access to specific nodes, limiting participant numbers and potentially allowing for higher throughput and faster consensus. Public blockchains, however, suffer from slower transaction processing times, which may deter organizations from making their data publicly available [**?**]. Decentralized cryptographic primitives have been suggested to increase transaction speed while reducing costs, making them suitable for IoMT-enabled healthcare systems [**?**]. Merging public and permissioned blockchains could significantly enhance data integrity within the healthcare system.

In this paper, we propose an IoMT-based hybrid blockchain architecture, integrating the decentralized Ethereum with the centralized Hyperledger Fabric blockchain (Eth-Fab) using SQLite. This Eth-Fab model fetches the patient's iris image using IoMT sensors, stores it on IPFS, and secures its data hash using Ethereum. We employ IPFS, Ethereum, Hyperledger Fabric with SQLite, and hybrid access control policies to ensure patients can access their disease records using smart devices without needing to visit the hospital. Additionally, ML algorithms such as multi-layer perceptron (MLP), support vector machine (SVM), extra trees classifier (ETC), and stochastic gradient descent (SGD) are utilized for disease prediction, with the results securely stored on the blockchain, accessible to patients and partially to healthcare practitioners. Our proposed model not only secures health data but is also analytically modeled using the M/M/1 queuing model, deriving closed-form expressions for latency, transaction validation, throughput, and system utilization. This model achieves high transaction speed and low cost, validated through Monte Carlo simulation, showing decreased latency with increased service time and improved throughput with increased arrival and service times. Utilization decreases with the rise in arrival and service times, demonstrating the model's efficiency.

The primary contributions of our research are summarized as follows:

(i) We introduce a novel hybrid model for securing the Internet of Medical Things (IoMT) data that combines Ethereum and Hyperledger Fabric alongside SQLite. This model ensures patient privacy through a permissioned blockchain framework, enhanced by newly designed access control mechanisms. This approach enables secure patient registration via a modified smart contract.

(ii) Our architecture aims to safeguard the privacy and security of patient data by employing hybrid access control techniques. These techniques restrict data access to authorized personnel only, thereby mitigating the risk of data breaches and unauthorized access.

(iii) We leverage ML algorithms for disease prediction and then securely storing on the blockchain, ensuring data integrity and accessibility.

(iv) We adopt the M/M/1 queuing model to formulate our proposed architecture, from which we derive closed-form expressions for key performance metrics such as latency, throughput, and system utilization. The efficacy of these expressions is further validated through Monte Carlo simulation.

The rest of the paper is organized as follows. Section 2 describes the proposed system model followed by a proof of concept utilizing mathematical modeling in Section 3. Section 4 describes experimental assessment, and Section 5 brings the investigation to a conclusion.

## 2 System Model

The proposed model leverages various technologies to achieve its objectives. Ethereum is utilized for registration, enabling IoMT devices to optionally provide their public keys, passwords, document hashes, and EHRs Manager, among others. Hyperledger Fabric is employed for storing patients' data, while SQLite facilitates the connection between Hyperledger Fabric and Ethereum. In our hybrid blockchain, we employ Practical Byzantine Fault Tolerance (pBFT) consensus on the Hyperledger Fabric side to ensure trust and validation through known and authorized participants. Conversely, proof of work (PoW) is used on the Ethereum side as the underlying consensus mechanism, ensuring the network's decentralized and trustless nature. Through the integration of pBFT and PoW, private and public blockchains are merged, allowing miners to collect pending transactions and solve the complex PoW challenge. The first miner to solve the challenge broadcasts the solution and the new block of transactions to the network. Other network users verify the solution and transactions, embodying trust through competition. This integration of technologies ensures accurate disease prediction while maintaining safe and efficient data handling. Furthermore, ML algorithms are applied to predict illnesses based on the available data. Figure 1 illustrates the system architecture, which is elaborated upon in the subsequent subsections.

### 2.1 Biometric Iris Scanners

In the proposed model, IoMT-based iris scanners enhance patient identification security and robustness through the following steps:

- Eye Image Capture: The iris scanner captures eye images for biometric identification via iris recognition or eye-tracking technologies.
- Iris Segmentation: The scanner segments the iris for characterization, detecting the pupil.
- Iris Normalization: Focuses on the pupillary side of the eye.
- Feature Extraction: Segmentation and normalization are employed to augment a normal iris image, achieving a well-distributed picture.
- Data Encoding: Converts data types.
- InterPlanetary File System (IPFS): Stores encrypted data to generate a hash.

### 2.2 Proposed Working Principle of Ethereum

Ethereum facilitates patient registration through public key document hashes, passwords, and additional information (e.g., the EHR manager's desk number and name). To preserve patients' and practitioners' information, the following steps are undertaken:

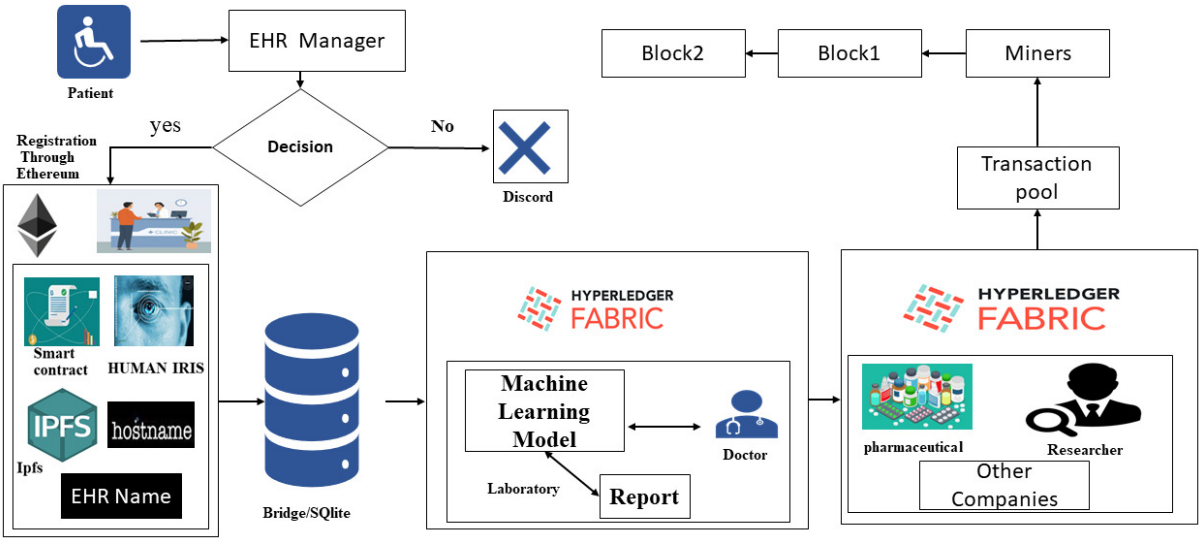


Fig. 1. System Architecture demonstrating the use of Ethereum and Hyperledger Fabric with SQLite to forecast disease data using ML

(i) Patient Registration: Let $P$ denote the set of patients, where each patient is assigned a public key, document hash $H$, password $PW$, manager desk number $MDN$, and name $MN$, represented as tuples $(H, PW, MDN, MN)$. The registration process, modeled as a function $R$, maps patient information to Ethereum addresses:

$$R : (H, PW, MDN, MN) \rightarrow E \tag{1}$$

where $E$ is the set of Ethereum addresses.

(ii) Smart Contract Automation: Let $C$ represent healthcare processes automated via smart contracts. Each process in $C$ accepts inputs $I$ and produces outputs $O$. Ethereum smart contracts execute these functions automatically, enhancing healthcare system efficiency.

$$M_L = \left(\frac{i}{R}\right) \times \sum(|IO|) \tag{2}$$

Here, $R$ extracts patient information from Eq. (1), with $i \geq 1$ denoting the number of patients. $M_L$ calculates the mean length of patient data for estimating blockchain-stored patient data.

## 2.3 Proposed Working Principle of Hyperledger Fabric

This section focuses on using Hyperledger Fabric to store patients' disease data $P_d$, granting patients more control over their medical information and enabling healthcare providers to make informed treatment decisions $T_d$. The following actions safeguard user data and patient health information, formulated in Eq. (3):

(i) HF Channels ($H_c$): Secure communication channels among patients, doctors, pharmaceuticals, and hospital managers are established.
(ii) Membership Service Provider (MSP) manages identities on the blockchain, authenticating new patients via the EHR manager.
(iii) Committing Nodes (CN) and Endorsing Nodes (EN) execute and verify transactions, respectively.
(iv) Ordering Nodes (ON) maintain transaction sequences.
(v) Chaincode (CC) implements smart contracts for interfacing with the distributed ledger.
(vi) Certificate Authority (CA) issues certificates, adding patient information to the blockchain.

Treatment decisions $T_d$ are derived as follows:

$$T_d = f(P_d, H_c, MSP, CN, EN, ON, CC, CA) \times C_l \tag{3}$$

The function $f$ processes and secures $P_d$ on Hyperledger Fabric, considering all input parameters. Commitment level $C_l$, ranging from 0 to 1, indicates data integrity and security levels.

### 2.3.1 *Transaction Flow*

Transactions involve two clients, the patient and the doctor, relying on each other for health assessments. An endorsement policy requires both parties to endorse any transaction, involving several stages for completion.

(i) The doctor initiates a transaction.
(ii) Endorsing peers verify the signature and transaction execution.
(iii) Proposal responses are inspected.
(iv) Target peers assemble endorsements into a transaction.
(v) Transactions are validated and committed.
(vi) Update the ledger.

## 2.4 Machine Learning-based Diseases Prediction

ML has emerged as a significant tool in healthcare, with the potential to transform medical practice and enhance patient outcomes. ML may be used in a variety of activities in healthcare, including illness detection, risk classification, therapy selection, and patient monitoring. One of the most significant advantages of ML in healthcare is its ability to manage large and complex datasets, which are commonly encountered in medical practice. Malaria, diabetes, impetigo, diabetes, AIDS, jaundice, chickenpox, and other health disorders have a significant influence on one's health and, if untreated, can lead to death [**?**]. In this situation, data mining approaches such as SVM, MLP, SGD, and ETC algorithms might be useful. We employed the aforementioned algorithms to detect the diseases of the patient.

### 2.5 Access Control Mechanism

Access control policies (ACP) are a collection of rules that determine who has access to resources, and what resources should they have access to. These ACPs are unique to the organization and must be kept private. Role Base Access Control (RBAC) and Attribute-based Access Control (ABAC) ACPs are fully role-centric and subject-centric. They do not enable management to monitor the user's compliance with work requirements or to assess the organization's aims [**?**]. Healthcare systems manage sensitive and personal data, which necessitates tight access restrictions to preserve patient privacy and data security. We proposed ABAC and RBAC policies for a healthcare system that uses hybrid blockchain technology, ML, and secure data access in this study.

#### *2.5.1 Attribute-based Access Control Model*

The ABAC policy considers user, resource, and environmental factors, defining access control rules through logical formulas. This policy facilitates fine-grained access control, adaptable to the diverse roles and requirements within the healthcare sector. Attributes of resources include their sensitivity level, data type, and location, while environmental attributes cover the time of day and network address. Access control rules are established using logical formulas, enabling precise control over access rights.

The design of the ABAC policy emphasizes flexibility, allowing it to be customized for different roles and needs within the healthcare system. Consequently, system elements such as users (U), objects (O), sessions (S), and actions (OP), along with user characteristics ($A_u$), object attributes ($A_o$), session attributes ($A_s$) and environmental attributes ($A_e$) follow mappings specified by NIST for subject, object, and environmental attributes, as detailed in the guide by [**?**]. Additionally, some definitions are inspired by [**?**], further enriching the framework's conceptual underpinnings.

$$A_{total} = A_u \cup A_o \cup A_s \cup A_e, \tag{4}$$

$A_{total}$ is the total set of attributes considered in the ABAC policy. This equation encapsulates the essence of ABAC by highlighting the importance of a multifaceted attribute set in the decision-making process for access control, ensuring that access rights are precisely granted based on a wide range of factors.

**Definition 1** (Range of Attributes)**.** *For a given property $a$ a within $A_e$, the set $V_a$ represents all allowable values in the system.*

**Definition 2** (Attribute Function)**.** *For any entity $e$ belonging to the set $A_e$, the function $f_{ae}$, known as the attribute function, assigns each entity to a distinct value within the attribute's scope. Precisely, $f_{ae}(e, a)$ yields the value corresponding to attribute $a$ of entity $e$.*

## EXAMPLES

2.5.1.1 Example 1: Doctors ($D_s$) may access patient records if those patients have provided permission. Only users with the necessary professional qualifications are allowed to access sensitive patient data.

2.5.1.2 Example 2: Nurses can only access patient records for the ward to which they are assigned. This means they cannot view records for patients not in their ward, nor can they view records for patients in other wards.

2.5.1.3 Example 3: Patients can only access their own records, and these records can only be accessed from authorized locations.

**Definition 3** (Attribute Expressions)**.** *Let $R_t$, $R_o$, and $P_r$ represent the sets of Resource types, User roles, and Patient Records, respectively, for access control based on attribute expressions. If $U = D_s$ and $R_t = P_r$, then access is allowed.*

**Definition 4** (Filter by Attribute)**.** *Let $N_s$, $A_w$, and $W_0$ represent the sets of Nurses, their assigned Wards, and Ward locations, respectively. If $U = N_s$ and $R_t = P_r$ and $A_w = W_0$, then access is allowed.*

**Definition 5** (Satisfaction with Attribute Filters)**.** *Let $P_a$, $R_o$, and $U_i$ represent the sets of Patients, Resource owners, and User IDs, respectively. If $U = P_a$ and $R_o = U_i$, then access is allowed.*

**Definition 6** (Satisfaction with a Relationship Condition)**.** *An entity $e \in A_e$ fulfills an attribute filter $F$, indicated by the symbol $e = F_{ae}$, if and only if*

$$\forall (a_i, v_i) \in F, f_{ae}(e, a_i) = v_i, \tag{5}$$

$$\textit{and} \quad \forall (a_i, \neg v_i) \in F, f_{ae}(e, a_i) \neq v_i. \tag{6}$$

*Eq.* (5) *and (6) collectively define the conditions under which an entity satisfies an attribute filter within the ABAC framework*

**Definition 7** (Access Request)**.** *It is denoted by the tuple $q = (u, o, s, op)$, where a user $u$ from the set $U$ requests the execution of operation $op$ from the set $OP$ on an object $o$ from the set $O$ during a session $s$ from the set $S$.*

**Definition 8** (Rule Satisfaction)**.** *An access request $q = (u, o, s, op)$ satisfies a rule $\rho$, symbolized by $q \models \rho$, if and only if*

$$(u, o, s) \in F \textit{ and } (u, o, s) \in R \textit{ and } op = op_\rho. \tag{7}$$

**Definition 9** (ABAC Policy)**.** *An ABAC policy is defined as the tuple $\pi = (E, OP, A, f_{ae}, P)$, where $E$ denotes the set of entities, $OP$ denotes the set of operations, $A$ denotes the set of attributes, and $P$ denotes the set of ABAC rules within the system. The function $f_{ae}$ represents the attribute function.*

**Definition 10** (ABAC Policy Decision)**.** *The decision $d\pi(q)$ regarding an access request $q$ under an ABAC policy $\pi$ is authorized (permitting access) if and only if*

$$\exists \rho \in P \textit{ such that } q \models \rho. \tag{8}$$

*If no such rule $\rho$ exists, the decision defaults to denying access. Authorization is thus contingent upon the access request meeting at least one rule within the policy; any request that fails to meet this criterion is denied.*

#### *2.5.2 Role-based Access Control Model (RBAC)*

The RBAC concept has three main components roles, permissions, and users. The EHR defines roles and permission for users to perform certain activities within the system. For example, a doctor may be assigned the role of "physician", which allows him to examine patient records. However,

a nurse may be assigned the role of "nurse", which allows her to view patient records who are under her care. RBAC has several advantages, including increased security, streamlined administration, and a lower chance of human mistakes.

**Theorem 1.** *Let $U$ denote the collection of system users, $O$ the set of objects (for example, patient records) that users may access, and $R$ the set of roles to which users can be assigned. We define a permission function $P : U \times O \rightarrow \{0, 1\}$, which maps each user and object pair to a binary value indicating whether the user has access to the object.*

*Proof.* Each user is allocated to one or more roles under RBAC management, with each role associated with a specific set of permissions. The system administrator defines these permissions for each role based on the users' job duties and their need-to-know basis. Access to objects is granted to users based on their assigned roles, rather than their individual identities. Representing this system as a graph where each node symbolizes a user, object, or role, and each edge denotes a permission relationship between a user and an object, or a role and an object, we utilize graph theory to demonstrate that RBAC can effectively restrict access to patient data, thereby safeguarding privacy and security. □

**Theorem 2.** *Let $G = (V, E)$ represent the access control system graph, where $V$ denotes nodes and $E$ denotes edges. A cut $(S, T)$ of the network is defined as a partition of the nodes into two disjoint sets $S$ and $T$, such that $S$ includes the source node (the user seeking access) and $T$ contains the destination node (the object being accessed). A cut is considered $(s, t)$-valid if $s \in S$, $t \in T$, and there exists at least one path in the graph from $s$ to $t$ that respects the permission constraints defined by the edges in $E$.*

*Proof.* In the context of RBAC, permission relationships between users and objects are represented as edges in the graph $G$. To ascertain whether a user $u$ is permitted access to an object $o$, it is necessary to determine if there exists an $(u, o)$-valid cut in the graph. The existence of such a cut implies a permissible path from $u$ to $o$ within $G$, thereby granting $u$ access to $o$. Conversely, the absence of an $(u, o)$-valid cut indicates that $u$ is not authorized to access $o$. This framework demonstrates the efficacy of RBAC in safeguarding patient data, ensuring privacy, and enhancing security. By structuring access permissions through roles and delineating access pathways within a graph model, healthcare systems can meticulously regulate access to sensitive information, effectively mitigating the risk of unauthorized disclosure. □

### 2.6 SQLite

SQLite is a self-contained library that implements a SQL database engine without requiring a server process, distinguishing it from most other SQL databases. It is renowned for its exceptional storage efficiency, rapid query execution, support for ACID (Atomicity, Consistency, Isolation, Durability) transactions, and minimal memory requirements. Furthermore, SQLite simplifies the deployment process by eliminating the need for setup or ongoing management [**?**]. In our project, SQLite serves as an intermediary facilitating the transfer of patients' public keys between Ethereum and Hyperledger Fabric. It is designed to temporarily store data, ensuring that any public key transmitted across these hybrid blockchains is automatically purged from SQLite upon power loss, thereby precluding the possibility of third-party intervention.

## 3 Proof of Concept using Mathematical Modelling

In this paper, we employed the M/M/1 Queuing Model as our analytical framework for investigating blockchain technology interfaces with machine learning.

### 3.1 M/M/1 Queuing-based Mathematical Modelling

To investigate the performance of queuing systems, the M/M/1 Queuing Model is a widely accepted and adopted mathematical model. Specifically, it is represented by the M/M/1, indicating a system characterized by Markovian (or memoryless) patterns in both arrival and service processes, coupled with the presence of a single server. This refers to systems with a single server in which both the customer arrival rate ($\lambda$) and the server service ($\mu$) rate stay constant throughout time. [**?**] The M/M/1 model performs the computation of various performance metrics, such as the average number of customers in the system, the average time customers spend waiting in the queue, the probability that the system is empty, and the probability that it is full. Key performance measures derived from the model include:

(i) The average number of customers in the system ($L$):

$$L = \frac{\lambda}{\mu - \lambda} \tag{9}$$

(ii) The average time a customer spends in the queue ($W$):

$$W = \frac{1}{\mu - \lambda}, \tag{10}$$

where ($\lambda$) shows the customer arrivals rate, while the service rate of the server (customers served / unit time) is $\mu$.

### 3.2 Enrollment Stage

In this paper, the proposed model aims to develop a secure and efficient platform to manage patient data with the help of blockchain, ML algorithms, and access control mechanisms. In the first phase, patient registration over Ethereum is performed, where patients create a unique identity for authentication. The public keys are used for the registration process to guarantee that only authorized persons can access the data.

#### *3.2.1 Operating Indicators*

The following operational probabilities for EHR systems are examined using queuing theory:

(i) The probability of EHRs being idle ($P_0$), busy ($P_b$) and $n$ number of EHRs ($P_n$) in a hospital may be expressed as.

$$P_0 = 1 - \frac{\lambda}{\mu}, \tag{11}$$

$$P_b = \frac{\lambda}{\mu}, \tag{12}$$

$$P_n = (P_b)^n \times P_0. \tag{13}$$

(ii) The probability that the number of Patients is greater than $K$, then we have,

$$P(n > K) = (P_b)^{K+1} \tag{14}$$

(iii) The probability of a nonempty queue may be expressed as:

$$P(n > 1) = 1 - P_0 - P_1 = (P_b)^2 \tag{15}$$

(iv) The probability of the average number of patients ($P_s$) residing in the system is expressed as:

$$P_s = \frac{\lambda}{\mu - \lambda} \tag{16}$$

(v) Now to find the average number of patients in the queue, $P_q$, we may have,

$$P_q = P_s - P_b = \frac{\lambda^2}{\mu(\mu - \lambda)} \tag{17}$$

(vi) The probability of average waiting time required for completion of Ethereum registration ($P_{we}$), can formulated as:

$$P_{we} = \frac{P_s}{\lambda} = \frac{1}{\mu - \lambda} \tag{18}$$

### 3.3 Transmit Stage

The second stage is to utilize Hyperledger Fabric to securely store patient disease data, which will only be accessible to authorized individuals, guaranteeing the privacy and security of patient information. Providers can connect to the hospital via the network and access the essential hospital data to deliver appropriate medication-related data. The solution uses Hyperledger Fabric to ensure that patient data is safe, immutable, and tamper-proof, while also allowing for easy access and exchange of data among healthcare practitioners. The probability of how long the average doctor spends ($L_n$) writing data on Hyperledger Fabric can be expressed as:

$$L_n = \frac{\mu}{\mu - \lambda} \tag{19}$$

Therefore, based on the above formulation, the average latency ($W$), The average throughput ($X$), and The system utilization ($\rho$) using the M/M/1 queuing model can obtained as:

$$W = \frac{1}{\mu - \lambda} \tag{20}$$

$$X = \lambda \times \left(1 - \frac{\lambda}{\mu}\right) \tag{21}$$

$$\rho = \frac{\lambda}{\mu} \tag{22}$$

**Theorem 3.** *Let us assume ($\lambda$) to be the patient arrival rate of registration requests, and $\mu$ represents the system's service rate. Then, to develop a safe and efficient platform for handling patient data, the proposed research needs to integrate blockchain technology, ML, and a secure data access mechanism through the M/M/1 queuing model, where*

- $\rho = \frac{\lambda}{\mu} < 1$ *represents the utilization of the system,*
- *if $\rho > 1$, the system is overloaded and the queue length will grow infinitely, and*
- *if $\rho < 1$, the system is underloaded and the average number of patients in the system will be less than 1*

*Proof.* Let $P_q$ represent the average number of patients in the queue and $L_{ss}$ the average number of patients serviced by the system. Let $W_q$ be the average queue waiting time and $P_{we}$ the average service time. Let $W$ represent the overall average time a patient spends in the system. Hence, to calculate the expression for $\lambda$ using the M/M/1 we have,

$$\lambda = \frac{P_s + P_q}{P_{we}} \tag{23}$$

Where $\lambda$ is derived from the M/M/1 model specification, indicating the arrival rate of patient registration requests.

The service rate $\mu$ is determined as:

$$\mu = \frac{1}{P_{we}} \tag{24}$$

By Little's Law, we find:

$$L = \lambda W \tag{25}$$

Where $L$ is the average number of patients in the system.

The average number of serviced patients $L_{ss}$ is influenced by the system's utilization $\rho$, thus:

$$L_{ss} = \rho L \tag{26}$$

The average queue length $P_q$ is then recalculated to reflect the time patients spend waiting minus the time being serviced, as:

$$P_q = \lambda(W - P_{we}) \tag{27}$$

Simplifying, we get:

$$P_q = \lambda(1 - \rho)\frac{1}{\mu} \tag{28}$$

For $W$, the average time a patient spends in the system, applying Little's Law yields:

$$W = \frac{L_{ss} + P_q}{\lambda} \tag{29}$$

Simplifying further:

$$W = \frac{\rho L + \lambda(1 - \rho)\frac{1}{\mu}}{\lambda} \tag{30}$$

This simplifies to:

$$W = \frac{1}{\mu} \tag{31}$$

indicating that the average time in the system is inversely related to the service rate.

Lastly, the average waiting time in the queue $W_q$ is:

$$W_q = \frac{P_q}{\lambda} \tag{32}$$

Leading to:

$$W_q = (1 - \rho)\frac{1}{\mu} \tag{33}$$

This demonstration of Little's Law for the M/M/1 queuing model underscores the relationship between system utilization, service rates, and waiting times, affirming the model's applicability to managing patient data in a blockchain-integrated healthcare system. □

# 4 Experimental Evaluation

The analytical approach integrates collected performance metrics into the Hyperledger Fabric blockchain framework. This process involves utilizing the M/M/1 queuing model, parameterized from the experimental testbed, to structure our model. To substantiate the proposed model's validity, we juxtapose the analytical outcomes with results derived from Monte Carlo simulations. This facilitates a comprehensive evaluation of the model's performance in simulating real-world operational scenarios.

## 4.1 Experimental Setup

We have successfully developed an Ethereum blockchain application using Solidity. For Hyperledger implementation, TypeScript, the Web3 API, and SQLite queries were employed, all integrated within a Node.js and JavaScript environment. Python was utilized for the system's ML components. The entire system operates efficiently on the Windows 11 operating system, supported by Docker Desktop and Visual Studio Code for development and testing.

## 4.2 Monte Carlo Simulation

Monte Carlo simulation is a computational technique used to model and simulate the behavior of complex systems or processes. It employs statistical methods, utilizing random sampling and probability distributions to estimate the potential outcomes of a system or process. A Monte Carlo simulation generates a huge number of random samples to represent many possible outcomes of the system or process being modeled. These samples are then used to compute the probability distribution of the outcomes, allowing an estimate of anticipated values and possible dangers associated with various situations [**?**].

### *4.2.1 Latency ($W_m$) Formulation*

The average latency can be calculated using the following equation,

$$W_m = \frac{1}{n}\sum_{i=1}^{n}(D_i - A_i), \tag{34}$$

where $n$ is the number of jobs, $A_i$ is the arrival time and $D_i$ is the departure time of the $i^{th}$ job, and $\sum_{i=1}^{n}(D_i - A_i)$ obtain the sum of the latencies for all jobs. The average latency is calculated by dividing the total by $n$.

### *4.2.2 Throughput ($X_m$) Formulation*

Throughput is formulated through Monte Carlo configuration as:

$$X_m = \frac{\mathrm{N_{tr}}}{\mathrm{Entry_{time}} + \mathrm{Ser_{time}}} \tag{35}$$

$\mathrm{N_{tr}}$ illustrates the number of trials run during the simulation, $\mathrm{Entry_{time}}$ represents the time when the first customer entered the system, while $\mathrm{Ser_{time}}$ denotes the total time spent serving customers.

### *4.2.3 Utilization ($\rho_m$) Formulation*

The utilization is calculated using the following equation,

$$\rho_m = \bar{\mu} \times \lambda \tag{36}$$

where $\bar{\mu}$ represents the average service rate and $\lambda$ is the arrival rate of jobs to the system.

## 4.3 Average Latency

This subsection elaborates on the average latency performance metric, analyzed through both analytical modeling using the M/M/1 model and simulation via the Monte Carlo method. Latency, or response time, includes the total time a customer spends in the system, encompassing both waiting in the queue and being serviced by the server. In our proposed model, latency specifically measures the duration required for a transaction to be processed and mined into a block. Fig. 2 illustrates that at $\lambda = 0.2$ and $\mu = 1.8$, the

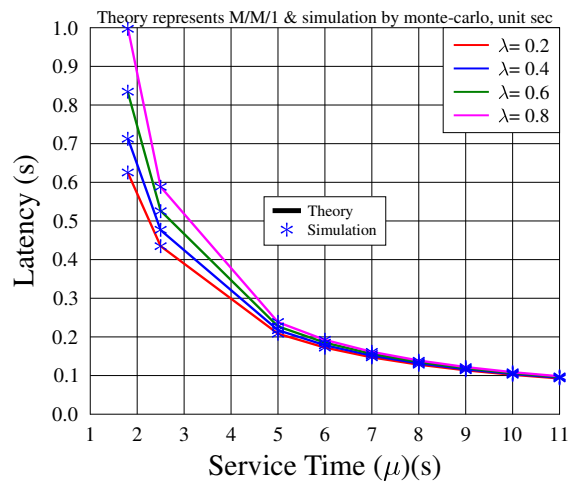


Fig. 2. Latency Performance in light of Service Time and Arrival Rate.

latencies for Monte Carlo and M/M/1 are 0.62561 seconds and 0.625 seconds, respectively. Similarly, at $\lambda = 0.8$ and $\mu = 11$, the Monte Carlo result is 0.098075513, and the M/M/1 result is 0.098039216. This validates the accuracy of the equations derived through M/M/1 for latency. Furthermore, the figure shows that increasing the $\mu$ decreases the average latency as a higher $\mu$ enables the server to process customers faster, reducing the queue length. However, latency begins to rise again as $\mu$ significantly exceeds $\lambda$, indicating server underutilization.

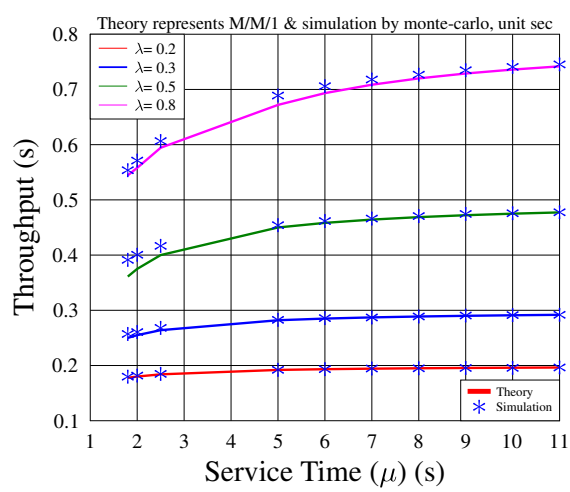


Fig. 3. Throughput Performance in Light of Service Time and Arrival Rate

### 4.4 Average Throughput

This subsection examines the throughput performance of our proposed model using both M/M/1 and Monte Carlo simulations. Throughput, the rate at which the system processes and completes customer requests, reflects the system's efficiency in handling transactions, equivalent to the customer arrival rate. Fig. 3 demonstrates that throughput increases with the $\lambda$ and $\mu$, showcasing the model's capability to process transactions faster as service time increases.

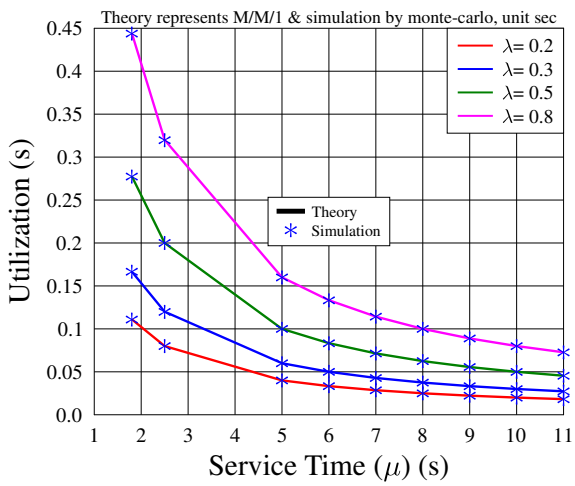


Fig. 4. Utilization Performance in Light of Service Time and Arrival Rate.

### 4.5 Average Utilization

Utilization performance, integrating both public and private blockchains, focuses on the fraction of time the server/miner is actively serving customers/creating blocks. It measures the workload of the server/miner. Fig. 4 reveals that utilization varies with $\lambda$ and $\mu$, impacting system performance. Higher $\lambda$ relative to $\mu$ increases system utilization, indicating full capacity operation, while higher $\mu$ values with lower $\lambda$ lead to underutilization.

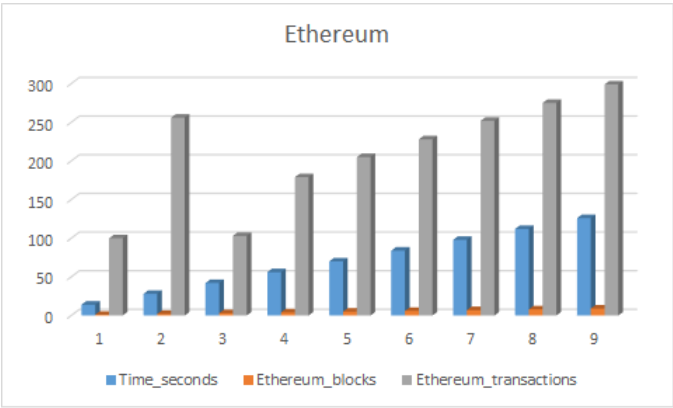


Fig. 5. Block Size of Ethereum Blockchain

### 4.6 Block Size of Hybrid Blockchain

Discussion on transactions and block size in hybrid blockchain environments highlights the efficiency of Ethereum and Hyperledger Fabric in processing transactions within specific time frames. Similarly, Fig. 6 depicts

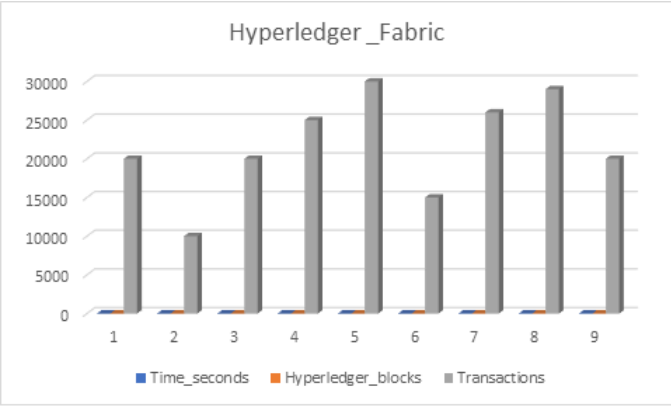


Fig. 6. Block Size of Hyperledger Fabric Blockchain

Hyperledger Fabric's transaction handling and block generation capabilities, underscoring the platforms' transaction storage efficiency depending on miner activity.

### 4.7 Disease Detection using Machine Learning

Machine learning algorithms employed for disease prediction include MLP, SVM, SGD, and extra trees classifiers. Model evaluation follows, utilizing various metrics to assess

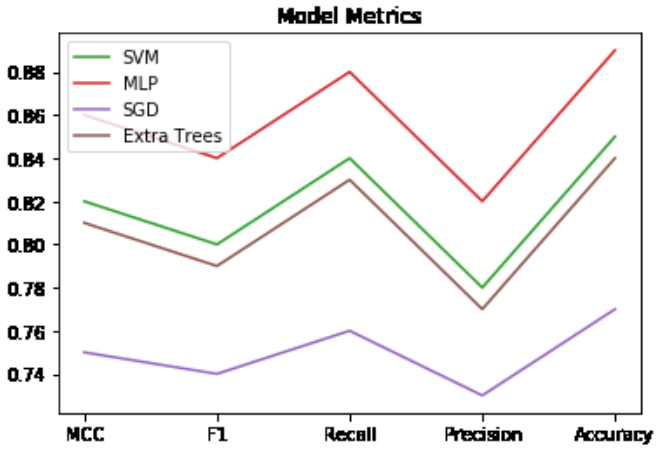


Fig. 7. Disease Detection using Machine Learning

performance. For SVM, the accuracy is 85%, precision is 0.78, recall is 0.84, F1 score is 0.80, and Matthews correlation coefficient (MCC) is 0.82. MLP shows an accuracy of 89%, precision of 0.82, recall of 0.88, F1 score of 0.84, and MCC of 0.86. SGD has an accuracy of 77%, precision of 0.73, recall of 0.76, F1 score of 0.44, and MCC of 0.77. Extra trees classifiers report an accuracy of 81%, precision of 0.79, recall of 0.83, F1 score

## 5 Conclusion

In the healthcare industry, it is common for different parties to share research data, medical information, and health records. Typically, this data is generated and transmitted by IoMT-based wearables, which need to be authenticated, and the data must be securely collected and transmitted.

To address this, we present a hybrid method for the secure authentication of patient data using a combination of public and private blockchain technology. The proposed system, built on Hyperledger Fabric, offers a solution for trustless communication among enterprises in the healthcare sector. Additionally, we have incorporated access policies to limit user access. We also employed various machine learning algorithms for predicting diseases and storing the predictions on the blockchain for auditing purposes. Therefore, we are developing 4 ML algorithms namely, SVM, MLP, SGD and ETC. In our system model and analysis, we presented stepwise operation of public and private blockchains and then modeled using M/M/1 queuing for deriving expressions for Latency, throughput, and systems utilization performance which are confirmed by Monte Carlo simulations. The suggested healthcare blockchain architecture offers a potential solution to the industry's challenges. It provides a secure and transparent system that ensures the integrity of patient data while also allowing for secure patient data access and exchange. This research illustrates how blockchain technology can enhance the security and transparency of private medical data and opens new avenues for further studying IoMT-based healthcare systems.

## ACKNOWLEDGEMENTS

This work was supported by the Ningbo Municipal Bureau of Science and Technology under Grant 2023J194.

**Ateeq ur Rehman** (Member, IEEE) is currently working as a Lecturer at the Department of Computing, Staffordshire University, UK. He received his Ph.D. degree from the University of Southampton, UK in 2017. His area of research includes cyber security, blockchain, and privacy-preserved machine learning, particularly in healthcare and smart cities.

**Nargis Tariq** got an MS degree in Computer Science from the University of Haripur, Pakistan. Her research interest preserving privacy, Blockchain, and Deep Learning.

**Dr. Mian Ahmad Jan** Assistant Professor at the Department of Computer Science, School of Computing and Informatics, University of Sharjah, UAE. He received his PhD from the University of Technology Sydney, Australia. His research interests include cybersecurity, energy-efficient and secured communication for Internet of Things. He has been among the World's Top 2% scientists for the years 2021 and 2022.

**Fazlullah Khan** (Senior Member, IEEE) got PhD in Computer Science from Abdul Wali Khan University Mardan, Pakistan in 2020. He is an Assistant Professor at the University of Nottingham Ningbo China, Zhejiang, China. His research interests are Security and privacy, the Internet of Things, Machine Learning, and Health Informatics. He has been among the World's Top 2% scientists for the years 2022, 2023.

**Houbing Herbert Song** (M'12–SM'14-F'23) is currently a Professor at the University of Maryland, Baltimore County (UMBC), Baltimore, MD. He serves as an Associate Editor in IEEE Transactions and Journals and received multiple best papers awards. His research interests include cyber-physical systems/internet of things, cybersecurity and privacy, and AI/machine learning/big data analytics.

**Muhammad Ibrahim** got Ph.D. in computer science from the Capital University of Science and Technology, Islamabad, in 2019. He is working at the University of Haripur. His research interests include large-scale network simulation and modeling in blockchain cloud computing, VM migration, and task scheduling in cloud computing.