AI-Assisted Optimization of Hybrid RBAC-ABAC Access Control for Blockchain-Based Electronic Health Record Systems

G. Thiraviya Suyambu[1], Dr. M. Anand[2], Dr. S. Srinivasan[3], Dr. M. Janakirani[4]

[1]Ph.D Research Scholar, Dept. of ECE, Dr. M.G.R Educational and Research Institute, Chennai, India.
[2,4] Professor, Dept. of ECE, Dr. M.G.R Educational and Research Institute, Chennai, India.
[3] Dept. of ECE, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Saveetha University, Chennai, India.
deanacademics@mamse.in[1], anand.ece@drmgrdu.ac.in[2]

Abstract - The proliferation of digital healthcare infrastructure has necessitated robust mechanisms for managing Electronic Health Records (EHRs) that balance security, privacy, and accessibility. Blockchain technology has emerged as a promising paradigm for EHR management, offering inherent properties of immutability, transparency, and decentralized trust distribution. However, the implementation of fine-grained access control mechanisms within blockchain-based EHR systems introduces substantial computational overhead and economic costs associated with blockchain transaction processing.

Hybrid access control models that synergistically combine Role-Based Access Control (RBAC) and Attribute-Based Access Control (ABAC) have demonstrated enhanced flexibility and security compared to singular approaches. Nevertheless, these hybrid implementations predominantly rely on static policy evaluation mechanisms that exhibit significant inefficiencies when subjected to the dynamic, high-volume workloads characteristic of modern healthcare environments. The static nature of policy evaluation fails to capitalize on recurring access patterns and contextual similarities inherent in healthcare workflows, resulting in redundant computational operations and suboptimal resource utilization.

This research introduces a novel AI-assisted optimization framework specifically designed for blockchain-enabled EHR systems, enhancing hybrid RBAC-ABAC access control through machine learning-based policy optimization. The proposed framework employs a Random Forest ensemble classifier to predict cost-efficient policy execution pathways by analysing contextual attributes, temporal patterns, and historical access behaviour. Critically, the optimization layer operates entirely off-chain, ensuring that no additional privacy vulnerabilities are introduced and that blockchain overhead remains minimal while computational intelligence is maximized.

Comprehensive formal security analysis demonstrates that the proposed system preserves all cryptographic guarantees of the baseline implementation, maintaining collision resistance below 2^{-128} while ensuring access control correctness through order-independent logical conjunction properties. The AI optimization layer is architecturally isolated from sensitive medical data, operating exclusively on access metadata and contextual features.

Experimental evaluation conducted on an Ethereum-based testbed with simulated realistic hospital workloads reveals substantial performance improvements. The optimized framework achieves an additional 12-18% reduction in gas consumption and a 15-25% reduction in access decision latency compared to non-optimized hybrid RBAC-ABAC implementations. Scalability analysis demonstrates sub-linear growth in computational cost as system load increases, contrasting favourably with the near-linear growth exhibited by baseline approaches.

The experimental results confirm that integrating AI-driven optimization significantly enhances the scalability, economic feasibility, and practical applicability of blockchain-based healthcare data management systems. This work provides a validated foundation for next-generation intelligent healthcare information systems that leverage machine learning to optimize blockchain operations without compromising the security and privacy guarantees essential for medical data management.

Keywords : Blockchain Technology, Electronic Health Records, Role-Based Access Control, Attribute-Based Access Control, Machine Learning Optimization, Gas Cost Reduction, Random Forest Classification

1. Introduction

1.1 Background and Context

The contemporary healthcare landscape is undergoing a profound digital transformation, driven by the imperative to improve patient outcomes, enhance operational efficiency, and enable seamless information exchange across organizational boundaries. Electronic Health Records (EHRs) have become the cornerstone of modern healthcare delivery, serving as comprehensive digital repositories that consolidate patient medical histories, diagnostic

results, treatment plans, medication records, immunization dates, allergies, radiology images, and laboratory test results. The digitization of

health records has facilitated improved care coordination, reduced medical errors, enhanced clinical decision-making, and enabled data-driven insights that advance medical research and public health initiatives.

However, the centralization of EHR systems in traditional architectures introduces critical vulnerabilities that undermine the security, privacy, and reliability objectives essential for healthcare information systems. Centralized EHR platforms are characterized by single points of failure, where system outages, hardware failures, or network disruptions can render critical medical information inaccessible during time-sensitive clinical situations. Furthermore, centralized architectures present attractive targets for malicious actors, with healthcare data breaches increasing in frequency and severity, exposing sensitive patient information and violating privacy regulations such as the Health Insurance Portability and Accountability Act (HIPAA) and the General Data Protection Regulation (GDPR).

Data integrity concerns represent another fundamental challenge in centralized EHR systems. Without cryptographic guarantees and immutable audit trails, unauthorized modifications to medical records can occur without detection, potentially leading to misdiagnosis, inappropriate treatment, medication errors, and adverse patient outcomes. The lack of transparency in access patterns and data modifications further complicates accountability and forensic investigation when security incidents occur.

1.2 Blockchain Technology for Healthcare

Blockchain technology, originally conceptualized as the underlying infrastructure for cryptocurrency systems, has emerged as a transformative paradigm for addressing the inherent limitations of centralized data management architectures. The fundamental characteristics of blockchain technology—decentralization, immutability, transparency, and cryptographic security—align remarkably well with the requirements of healthcare information systems.

Decentralization eliminates single points of failure by distributing data and computational responsibility across multiple nodes in a peer-to-peer network, ensuring system resilience and continuous availability. Immutability, achieved through cryptographic hash chaining and consensus mechanisms, ensures that once data is recorded on the blockchain, it cannot be retroactively altered or deleted without detection, providing tamper-evident audit trails essential for medical record integrity. Transparency enables all authorized participants to verify transactions and access patterns, fostering accountability and trust among healthcare stakeholders. Cryptographic security mechanisms, including public-key cryptography, digital signatures, and hash functions, protect data confidentiality and authenticate participants without requiring central authorities.

In blockchain-based EHR systems, the architectural approach typically involves storing actual medical records in encrypted off-chain repositories to address scalability and privacy concerns, while maintaining cryptographic hashes, access metadata, permission structures, and audit logs on the blockchain itself. This hybrid storage model leverages blockchain's strengths for access control and auditability while avoiding the prohibitive costs and performance limitations associated with storing large medical files directly on-chain.

1.3 The Access Control Challenge

While blockchain technology provides a robust foundation for EHR system security, the sensitive nature of healthcare data necessitates sophisticated access control mechanisms that go beyond simple authentication. Medical records contain highly personal and sensitive information protected by stringent legal and ethical frameworks. Unrestricted access to immutable blockchain records would constitute severe privacy violations with potentially devastating consequences for patients, including discrimination, stigmatization, identity theft, and psychological harm.

Access control in healthcare environments must accommodate complex, multifaceted requirements that reflect the realities of clinical workflows. Healthcare professionals require different levels of access based on their roles (physicians, nurses, administrative staff), their relationships with specific patients (primary care provider, specialist consultant, emergency responder), the clinical context (routine visit, emergency situation, research study), temporal factors (current treatment period, historical review), and institutional policies (departmental boundaries, privacy regulations, consent preferences).

Traditional Role-Based Access Control (RBAC) systems assign permissions based on predefined organizational roles, providing administrative efficiency and simplicity. A physician role might have blanket permissions to access all patient records within their

department, while a billing administrator might access only demographic and insurance information. RBAC excels in environments with stable, well-defined organizational structures and clear role hierarchies. However, RBAC's rigidity becomes a liability in dynamic healthcare contexts where access requirements depend on situational factors beyond static role assignments. RBAC cannot easily accommodate scenarios such as emergency access by specialists outside normal care teams, temporary delegations during physician absences, or patient-specific consent restrictions.

Attribute-Based Access Control (ABAC) addresses these limitations by making access decisions based on attributes of the requesting user, the requested resource, the access environment, and the current context. ABAC policies evaluate conditions such as "Grant access if the requester is a physician AND is assigned to the patient's care team AND the access occurs during a scheduled appointment AND the patient has not revoked consent." This expressiveness enables fine-grained, context-aware authorization that adapts to complex healthcare scenarios. However, ABAC's flexibility comes at the cost of increased computational complexity, as each access request requires evaluating potentially numerous attribute conditions and policy rules.

1.4 Hybrid RBAC-ABAC Models

Recognizing that RBAC and ABAC represent complementary rather than competing approaches, hybrid models have emerged that strategically combine the efficiency of role-based permissions with the granularity of attribute-based policies. Hybrid RBAC-ABAC architectures typically employ RBAC for initial coarse-grained filtering based on organizational roles, then apply ABAC policies for fine-grained authorization based on contextual attributes. This layered approach reduces unnecessary attribute evaluations by quickly rejecting requests from users lacking appropriate roles, while still enabling nuanced access control for authorized roles.

Despite these architectural improvements, existing hybrid RBAC-ABAC implementations in blockchain contexts suffer from a critical limitation: they rely on static policy evaluation mechanisms that execute the same evaluation sequence regardless of request characteristics or historical patterns. This static approach results in several inefficiencies:

Redundant Attribute Evaluations: The system evaluates all policies in a predetermined order, even when certain policies are likely to fail based on request characteristics. For example, evaluating complex attribute conditions before simple role checks wastes computational resources.

Increased Smart Contract Execution Cost: Blockchain platforms like Ethereum charge gas fees proportional to computational operations performed by smart contracts. Redundant evaluations directly translate to higher transaction costs, making the system economically impractical for high-volume healthcare operations.

Higher Transaction Latency: Each policy evaluation consumes time, and unnecessary evaluations accumulate into noticeable access delays that frustrate users and impede clinical workflows, particularly during time-sensitive medical situations.

Failure to Exploit Access Patterns: Healthcare environments exhibit strong temporal and behavioural patterns—physicians access similar patient records during their shifts, routine procedures follow predictable access patterns, and emergency scenarios have characteristic urgency signatures. Static evaluation ignores these learnable patterns, missing opportunities for optimization.

1.5 Research Motivation and Objectives

This research is fundamentally motivated by the observation that access control decisions in healthcare environments exhibit inherent, learnable patterns that can be exploited to optimize policy execution without compromising security guarantees. Rather than executing identical evaluation sequences for every access request, an intelligent system could predict the most efficient evaluation pathway based on request characteristics and historical behaviour.

The core insight driving this work is that machine learning models can learn to identify which policy checks are most likely to determine the final access decision for specific request types, enabling prioritized evaluation that minimizes unnecessary computations. For instance, if historical data reveals that emergency department physicians accessing trauma patient records during night shifts rarely fail role-based checks but frequently require attribute-based context validation, the system can optimize by streamlining role verification and focusing computational resources on context analysis.

Critically, this optimization must be achieved without introducing new security vulnerabilities, privacy risks, or architectural complexity that would undermine blockchain's fundamental guarantees. The optimization layer must operate transparently, preserve access control correctness, maintain cryptographic security properties, and comply with healthcare privacy regulations.

The primary objectives of this research are:

1. Design an AI-assisted optimization framework that reduces computational overhead and transaction costs in hybrid RBAC-ABAC

blockchain-based EHR systems while preserving all security properties.
2. Formulate access control execution as an optimization problem with explicit cost functions capturing gas consumption and access latency.
3. Develop a machine learning model capable of predicting optimal policy evaluation sequences based on request features and historical patterns.
4. Prove formally that the optimization preserves security guarantees, including access control correctness and cryptographic properties.
5. Implement and evaluate the framework on a realistic blockchain testbed with simulated healthcare workloads.
6. Demonstrate measurable improvements in gas costs, access latency, and scalability compared to baseline approaches.

1.6 Paper Organization

The remainder of this paper is organized as follows:

Section 2 presents the key contributions of this research. Section 3 provides comprehensive background on blockchain-based EHR systems, RBAC, ABAC, and hybrid models, while surveying related work in the literature. Section 4 details the proposed AI-assisted hybrid access control framework, including system architecture, problem formulation, feature engineering, machine learning model selection, and the optimized access control algorithm. Section 5 presents rigorous security analysis demonstrating preservation of cryptographic and access control guarantees. Section 6 describes the experimental setup, testing methodology, and implementation environment. Section 7 presents comprehensive experimental results with detailed analysis. Section 8 discusses implications, practical considerations, and interpretability. Section 9 acknowledges limitations and outlines future research directions. Section 10 concludes the paper.

2. Research Contributions

This research makes several significant contributions to the intersection of blockchain technology, access control mechanisms, and artificial intelligence for healthcare information systems:

2.1 Novel AI-Assisted Optimization Layer

We introduce an innovative off-chain machine learning-based optimization layer that predicts optimal policy execution pathways for hybrid RBAC-ABAC blockchain systems. Unlike previous approaches that apply machine learning to access control prediction itself (which would compromise security), our optimization layer preserves the deterministic security properties of the underlying access control system while optimizing the execution sequence. The AI component operates on metadata and contextual features rather than making access decisions, ensuring that security guarantees remain formally provable. This architectural separation enables leveraging machine learning's pattern recognition capabilities without introducing non-deterministic security-critical decisions.

2.2 Formal Cost Optimization Model

We formulate access control execution in blockchain contexts as a rigorous cost minimization problem that explicitly captures both gas consumption and access latency. The optimization model incorporates weighted cost functions that can be tuned based on organizational priorities—emphasizing economic costs for resource-constrained settings or latency for time-critical clinical environments. This formulation provides a mathematical framework for reasoning about access control efficiency and enables systematic evaluation of optimization strategies. The model accounts for the unique characteristics of blockchain execution environments, where computational costs are directly monetized through gas fees and where the immutable nature of blockchain prevents retroactive optimization.

2.3 Security Preservation Proof

We provide formal mathematical proofs demonstrating that the proposed optimization does not alter cryptographic guarantees or access control correctness. The proof leverages the order-independence of logical conjunction operations to show that policy reordering preserves access decisions. We further demonstrate that the optimization layer operates entirely on non-sensitive metadata and does not interact with cryptographic primitives, ensuring that collision resistance and other cryptographic properties inherited from the baseline system remain intact. These formal guarantees are essential for deployment in healthcare contexts where security cannot be compromised for performance.

2.4 Comprehensive Experimental Evaluation

We implement and evaluate the proposed framework on an Ethereum-based test network with realistic simulation parameters derived from healthcare usage patterns. The experimental evaluation systematically compares the optimized system against multiple baseline approaches (RBAC-only, ABAC-only, and non-optimized hybrid models) across multiple performance metrics including gas consumption, access latency, and scalability. The experiments incorporate rigorous statistical methodology with multiple trials, outlier removal, and confidence interval analysis. The results demonstrate significant, measurable improvements that validate the practical utility of the approach.

2.5 Reproducible Implementation Framework

We provide detailed implementation specifications including hardware requirements, software dependencies, dataset characteristics, and algorithmic pseudocode sufficient for independent reproduction and extension of this work. The modular architecture facilitates adaptation to different blockchain platforms, alternative machine learning models, and varied healthcare contexts.

3. Background and Related Work

3.1 Blockchain Technology Fundamentals

3.1.1 Core Blockchain Concepts

Blockchain technology represents a distributed ledger paradigm where data is organized into blocks linked through cryptographic hash functions. Each block contains a timestamp, transaction data, and a cryptographic hash of the previous block, creating an immutable chain resistant to retroactive modification. The distributed nature of blockchain systems, where multiple nodes maintain copies of the ledger and participate in consensus protocols, eliminates central points of control and trust.

Consensus mechanisms such as Proof of Work (PoW), Proof of Stake (PoS), and Practical Byzantine Fault Tolerance (PBFT) enable nodes to agree on the state of the ledger despite potential malicious participants. Smart contracts—self-executing programs stored on the blockchain—enable programmable transaction logic and complex application functionality beyond simple value transfers.

3.1.2 Blockchain for Healthcare Applications

The application of blockchain technology to healthcare has gained substantial research attention due to alignment between blockchain properties and healthcare requirements. Several blockchain-based EHR systems have been proposed in the literature, each addressing different aspects of the healthcare data management challenge.

MedRec pioneered the use of Ethereum blockchain for medical record management, proposing a decentralized record management system where blockchain serves as an access control layer while actual records remain in existing provider databases. MedChain extended this concept with additional privacy-preserving techniques. Healthbank proposed a patient-centric model where individuals control access to their own health data stored in distributed repositories. More recent systems like BurstIQ and Medicalchain have developed commercial implementations targeting real-world deployment.

However, these systems face common challenges related to scalability, transaction costs, and access control efficiency. Blockchain platforms impose significant computational costs for transaction processing, and public blockchain networks can suffer from limited throughput and variable latency. These limitations are particularly problematic for healthcare applications requiring high transaction volumes and predictable performance.

3.1.3 Smart Contract Platforms

Ethereum remains the most widely adopted platform for healthcare blockchain applications due to its mature smart contract capabilities, extensive developer ecosystem, and well-documented security properties. Ethereum's gas mechanism, while introducing economic costs, provides important security benefits by preventing denial-of-service attacks through resource consumption limits.

Alternative platforms such as Hyperledger Fabric offer permissioned blockchain architectures potentially better suited for enterprise healthcare contexts, providing higher throughput and greater privacy controls. However, permissioned blockchains sacrifice some decentralization benefits. The choice of blockchain platform involves fundamental trade-offs between decentralization, scalability, and privacy that must be evaluated based on specific healthcare context requirements.

3.2 Access Control Models in Computing Systems

3.2.1 Role-Based Access Control (RBAC)

Role-Based Access Control (RBAC) has been the dominant access control paradigm in enterprise systems for decades, standardized by NIST and widely deployed across industries. RBAC's fundamental principle involves assigning permissions to roles rather than individual users, with users then being assigned to appropriate roles based on their organizational functions.

Formally, RBAC defines:

- *A set of users U*
- *A set of roles R*
- *A set of permissions P*
- *A user-role assignment relation $UA \subseteq U \times R$*
- *A permission-role assignment relation $PA \subseteq P \times R$*
- *The permissions available to a user u are computed as:$Permission(u) = \cup(r \in Roles(u))\ Permissions(r)$*

RBAC offers significant administrative advantages. Rather than managing permissions for thousands of individual users, administrators define permissions once per role and simply assign users to appropriate roles. Role hierarchies enable permission inheritance, further reducing administrative burden. The NIST RBAC model defines four levels of increasing sophistication: Flat RBAC, Hierarchical RBAC, Constrained RBAC (with separation of duty), and Symmetric RBAC.

In healthcare contexts, typical RBAC roles include Physician, Nurse, Pharmacist, Radiologist, Laboratory Technician, Medical Student, Administrative Staff, and Billing Specialist. Each role receives permissions appropriate to its functional responsibilities—physicians access diagnosis and treatment information, pharmacists access medication records, billing staff access insurance data.

However, RBAC's limitations become apparent in dynamic healthcare environments. A physician's appropriate access depends not just on their role but on whether they are assigned to a particular patient's care team, whether the access occurs during an emergency, whether the patient has revoked consent for non-emergency access, and many other contextual factors. RBAC's static role assignments cannot accommodate these situational variations without creating an explosion of highly specific roles that negates the administrative simplicity motivating RBAC's adoption.

3.2.2 Attribute-Based Access Control (ABAC)

Attribute-Based Access Control (ABAC) addresses RBAC's limitations by evaluating access decisions based on attributes—characteristics of the user, resource, environment, and action. ABAC policies express complex conditions combining multiple attributes through logical operators.

The NIST ABAC guide defines the model formally. Let,

- ***User attributes:*** *age, role, department, clearance level, certifications.*
- ***Resource attributes****: classification, owner, creation date, department*
- ***Environment attributes:*** *time, location, security threat level*
- ***Action attributes:*** *read, write, delete, execute*
- *An ABAC access decision evaluates:*
  Access(u, r, a, e) = f(attributes_u, attributes_r, attributes_a,attributes_e)

where f is a policy function returning permit or deny.

ABAC policies in healthcare might express rules such as:

- *"Permit read access if user.role='physician' AND user.department=resource.department AND time ∈ business_hours"*
- *"Permit write access if user.id=resource.primary_physician AND resource.status='active_treatment'"*
  - *"Deny access if user.training_expiration < current_date"*

The eXtensible Access Control Markup Language (XACML) provides a standardized framework for expressing ABAC policies, with policy decision points evaluating requests against policy repositories and policy enforcement points implementing access decisions.

ABAC's expressiveness enables fine-grained, context-aware authorization ideal for complex healthcare scenarios. However, this flexibility imposes significant computational costs. Each access request potentially requires evaluating numerous attribute comparisons and complex policy logic. In blockchain contexts, where every computational operation consumes gas and impacts transaction costs, ABAC's computational intensity becomes economically prohibitive for high-volume operations.

3.2.3 Hybrid RBAC-ABAC Models

Recognizing the complementary strengths of RBAC and ABAC, researchers have proposed hybrid models that strategically combine both approaches. The typical hybrid architecture employs RBAC for initial coarse-grained filtering based on organizational roles, then applies ABAC policies selectively for fine-grained authorization.

Several hybrid approaches have been proposed in the literature:

Sequential Evaluation: RBAC checks execute first, quickly rejecting requests from users without appropriate roles. ABAC policies evaluate only for requests passing RBAC checks. This reduces wasted ABAC evaluations but still executes policies in fixed order.

Policy Categorization: Policies are classified as role-based or attribute-based, with execution order predetermined based on expected filtering efficiency. However, the categorization remains static.

Risk-Based Adaptation: Access decisions incorporate risk assessment, with high-risk requests triggering additional ABAC evaluations. This introduces adaptability but focuses on security rather than efficiency optimization.

These hybrid approaches improve upon pure RBAC or ABAC but retain fundamentally static evaluation mechanisms. They do not exploit learnable patterns in access requests to optimize execution sequences dynamically. Our research addresses this limitation by introducing machine learning-based dynamic optimization.

3.3 Machine Learning for Access Control

3.3.1 ML-Based Access Control Prediction

Several researchers have investigated applying machine learning to access control, primarily focusing on predicting access decisions or detecting anomalous access patterns.

Access Prediction Systems attempt to learn access control policies from historical decisions, then predict whether new requests should be permitted or denied. These systems train classifiers on features extracted from access logs, learning patterns correlating user attributes, resource characteristics, and environmental context with access decisions. While achieving reasonable

accuracy, these approaches fundamentally replace deterministic security policies with probabilistic predictions, introducing unacceptable security risks for sensitive applications like healthcare.

Anomaly Detection Systems use machine learning to identify unusual access patterns potentially indicating insider threats, credential theft, or policy violations. These systems establish behavioral baselines from normal access patterns, then flag deviations for investigation. Anomaly detection complements rather than replaces access control policies, serving as a secondary security layer.

3.3.2 Policy Mining and Optimization

Policy Mining techniques use machine learning and data mining to automatically discover access control policies from logs of historical access decisions and permission assignments. Role mining algorithms analyse user-permission assignments to identify coherent roles minimizing administrative overhead. These approaches reduce the manual effort of policy specification but do not address runtime execution efficiency.

Previous work on access control optimization has primarily focused on query optimization in database contexts, where access control checks are integrated into database query plans. Query optimizers reorder predicates to minimize unnecessary data retrieval and evaluation. However, these techniques assume low-cost predicate evaluation and do not address blockchain's unique cost model.

3.4 Blockchain-Based Access Control

3.4.1 Smart Contract Access Control

Numerous smart contract-based access control systems have been proposed for blockchain applications. Basic approaches implement simple permission checks in smart contract functions, verifying that transaction senders possess required roles or permissions before executing sensitive operations.

More sophisticated systems implement full-featured RBAC or ABAC frameworks in smart contracts. These systems define role hierarchies, permission assignments, and policy evaluation logic entirely on-chain. While providing transparency and immutability of access policies, on-chain policy evaluation incurs substantial gas costs proportional to policy complexity.

3.4.2 Off-Chain Policy Evaluation

Recognizing on-chain computation costs, several systems have proposed off-chain policy evaluation where access control decisions are computed off-chain and only results are recorded on-chain. These approaches reduce gas costs but introduce trust assumptions about off-chain components and complicate auditability.

Hybrid approaches perform partial evaluation on-chain and delegate complex computations off-chain, using cryptographic techniques like zero-knowledge proofs to verify off-chain computation results without re-execution. These methods offer promising efficiency improvements but require sophisticated cryptographic protocols.

3.5 Research Gaps

Despite extensive research on blockchain-based EHR systems, access control models, and machine learning applications, significant gaps remain:

Static Policy Evaluation: Existing hybrid RBAC-ABAC implementations use fixed evaluation sequences that ignore access pattern characteristics and fail to optimize based on request features.

Limited Cost-Efficiency Analysis: Prior work evaluates access control systems based primarily on security properties and functional correctness, with insufficient attention to transaction costs and economic feasibility for high-volume healthcare operations.

Lack of Formal Security Analysis for Optimized Systems: Research applying machine learning to access control often lacks rigorous formal security analysis, leaving uncertainty about whether optimizations preserve essential security guarantees.

Insufficient Experimental Validation: Many proposed systems evaluate performance through theoretical analysis or limited simulations rather than implementation on actual blockchain platforms with realistic workload scenarios.

Our research addresses these gaps by developing an AI-assisted optimization framework that reduces costs while preserving proven security properties, validated through comprehensive experimental evaluation on blockchain infrastructure.

4. Proposed AI-Assisted Hybrid Access Control Framework

4.1 System Architecture

The proposed framework consists of four architectural layers designed to provide secure, efficient, and scalable access control for blockchain-based EHR systems:

4.1.1 User Layer

The User Layer encompasses healthcare professionals, administrative staff, patients, and other stakeholders who interact with the EHR system. Users initiate access requests through client applications that capture authentication credentials, specify requested resources (patient records), and collect contextual information (purpose of access, urgency level). The client application constructs a properly formatted access request message and

transmits it to the blockchain layer through secure channels.

4.1.2 Blockchain Layer

The Blockchain Layer implements the core security and trust infrastructure. Smart contracts deployed on the blockchain encode hybrid RBAC-ABAC access control policies, manage role assignments, maintain permission mappings, and record access decisions in tamper-evident audit logs. The blockchain layer receives access requests, invokes the AI Optimization Layer for execution guidance, performs policy evaluations in the optimized order, and commits access decisions to the distributed ledger. All access control state and audit information resides on-chain, ensuring transparency and immutability.

4.1.3 AI Optimization Layer (Off-Chain)

The AI Optimization Layer represents the novel contribution of this architecture. Implemented as an off-chain service, this layer analyses access request characteristics and predicts the optimal policy evaluation sequence. The layer operates on extracted features including role identifiers, attribute categories, access frequency patterns, temporal context, and historical gas consumption statistics. Importantly, the AI layer does not access raw medical data, patient identities, or sensitive clinical information. It processes only access control metadata, ensuring compliance with healthcare privacy regulations.

The AI layer maintains a trained Random Forest classifier that outputs predicted optimal policy orderings. The layer interfaces with the blockchain layer through secure APIs, receiving feature vectors and returning execution guidance. The off-chain deployment enables model updates, performance monitoring, and computational scaling without blockchain constraints.

4.1.4 Storage Layer

The Storage Layer manages actual EHR data in encrypted off-chain repositories. Patient records, medical images, laboratory results, and clinical documents are stored in distributed file systems or cloud storage platforms with strong encryption. The blockchain stores only cryptographic hashes of EHR content, enabling integrity verification without on-chain storage of large medical files. Access control decisions recorded on the blockchain grant or deny access to specific encrypted data objects in the storage layer.

4.2 System Workflow

The end-to-end workflow for processing an access request proceeds as follows:

1. Request Initiation: User submits access request through client application, specifying identity, requested EHR resource, and access context.
2. Feature Extraction: System extracts relevant features from the request (role, attributes, temporal context, historical patterns).
3. AI Optimization Query: Blockchain layer queries AI Optimization Layer with extracted feature vector.
4. Policy Order Prediction: AI model predicts optimal policy evaluation sequence based on learned patterns.
5. Optimized Policy Evaluation: Blockchain smart contract executes access control checks in the predicted order, potentially terminating early upon policy failure.
6. Access Decision: System determines final access decision (permit/deny) and records it on blockchain.
7. Audit Logging: Transaction containing access request details and decision is permanently recorded on blockchain.
8. Data Access: If permitted, user receives decryption keys or access tokens for retrieving EHR data from storage layer.

This workflow maintains the security properties of deterministic access control while leveraging machine learning to optimize execution efficiency.

4.3 Problem Formulation

We formulate the access control optimization problem as a cost minimization task with security constraints.

4.3.1 Cost Function Definition

Let G denote the gas cost incurred by executing access control logic in the blockchain smart contract. Gas cost depends on:

- *Number of policy conditions evaluated*
- *Complexity of attribute comparisons*
- *Data structures accessed*
- *Cryptographic operations performed*

Let L denote the access decision latency measured from request submission to decision availability. Latency depends on:

- *Policy evaluation time*
- *Blockchain transaction propagation delay*
- *Consensus protocol latency*
- *Network communication overhead*

We define a combined cost function with weighting coefficients:

$$C = \alpha \cdot G + \beta \cdot L$$

where α and β are configuration parameters reflecting organizational priorities. Healthcare organizations with limited budgets may prioritize gas cost reduction (higher α), while emergency departments may prioritize latency reduction (higher β).

4.3.2 Optimization Objective

The optimization objective is to minimize C subject to the following constraints:

1. Correctness: The optimized access decision must match the decision that would be produced by evaluating all policies in the original order

2. Security: The optimization must preserve all cryptographic security properties of the baseline system

3. Feasibility: The predicted evaluation order must be executable within blockchain computational limits.

The challenge lies in achieving cost reduction while maintaining strict security and correctness guarantees. Traditional machine learning optimization often accepts small error rates for improved performance, but access control cannot tolerate even rare incorrect decisions.

4.4 Feature Engineering for AI Model

Effective machine learning requires carefully engineered features that capture relevant patterns while preserving privacy. We design a feature representation that balances predictive power with privacy protection.

4.4.1 Feature Vector Composition

Each access request is represented as a feature vector:

$$X = \{R, A, F, T, G\_prev, D, U, S, H\}$$

Where:

***R- Role Identifier (Categorical)**:* Encoded role of requesting user (Physician, Nurse, Pharmacist, etc.). We use one-hot encoding to represent roles as binary vectors, enabling the model to learn role-specific patterns without assuming ordinal relationships.

A - Attribute Vector (Mixed): Numerical and categorical attributes relevant to access control including department affiliation, specialty certification, current patient assignment status, and clearance level. Complex attributes are decomposed into multiple features.

F - Access Frequency (Numerical): Quantitative measure of how frequently the user accesses similar resources, computed over multiple temporal windows (hourly, daily, weekly). High-frequency patterns often indicate routine clinical workflows with predictable access control outcomes.

T - Temporal Context (Numerical): Time-related features including hour of day, day of week, shift identifier, holiday indicator, and time since user's last access request. Temporal patterns strongly correlate with access control outcomes—emergency night-shift access exhibits different characteristics than scheduled daytime procedures.

G_prev - Historical Gas Consumption (Numerical): Statistics on gas costs for the user's previous access requests including mean, median, standard deviation, and trend indicators. Users with consistently low gas costs likely exhibit predictable access patterns amenable to optimization.

D - Department Context (Categorical): Department of user, department of requested resource, and inter-departmental relationship indicators. Cross-departmental access typically requires additional policy evaluation compared to intra-departmental access.

U - Urgency Level (Categorical): Declared urgency of access request (Routine, Urgent, Emergency). Emergency access may bypass certain policy checks or receive priority evaluation.

S - Similarity Metrics (Numerical): Computed similarity between current request and user's historical access patterns using distance metrics. High similarity suggests the current request will follow previously observed evaluation paths.

H - Historical Outcome (Categorical): Whether similar recent requests were granted or denied. If all similar requests were granted, optimization can prioritize quickly confirming expected approval.

4.4.2 Privacy-Preserving Feature Design

Critically, all features are derived from access control metadata rather than medical content. The feature vector contains no patient identifiers, clinical information, diagnoses, treatment details, or other Protected Health Information (PHI). Role identifiers and department information constitute operational metadata outside PHI scope. This design ensures compliance with HIPAA, GDPR, and other healthcare privacy regulations while enabling effective machine learning.

4.5 Machine Learning Model

4.5.1 Random Forest Classifier

We employ a Random Forest ensemble classifier for predicting optimal policy evaluation orders. Random Forest was selected based on several considerations:

Interpretability: Random Forest provides feature importance rankings and decision path interpretability essential for healthcare compliance and trust. Clinical stakeholders can understand which factors influence optimization decisions.

Robustness: Ensemble methods resist overfitting and generalize well to unseen data distributions, important when access patterns evolve over time due to workflow changes, seasonal variations, or policy updates.

Multi-class Classification: Random Forest naturally handles multi-class prediction tasks, enabling prediction of multiple possible evaluation orders rather than binary classification.

Computational Efficiency: Random Forest inference is fast enough for real-time prediction without introducing significant latency overhead in access request processing.

Handling Mixed Data Types: Random Forest naturally accommodates both categorical and numerical features without extensive preprocessing or normalization.

4.5.2 Model Architecture

The Random Forest model consists of N decision trees, where each tree is trained on a

bootstrap sample of the training data with random feature subset selection at each split. The model architecture parameters include:

Number of Trees (N): Set to 100 based on cross-validation experiments balancing prediction accuracy and inference time

Maximum Depth: Limited to 20 levels to prevent overfitting while allowing sufficient expressiveness

Minimum Samples per Leaf: Set to 5 to ensure statistical reliability of leaf node predictions

Feature Subset Size: √m features considered at each split, where m is total feature count

4.5.3 Prediction Function

The Random Forest produces predictions through majority voting across constituent trees:

$$\hat{y} = \arg\max k \sum_{i=1}^{N} h_i(X)$$

where:

$h_i(X)$is the prediction of the i-th decision tree

k represents possible evaluation order classes

N is the number of trees in the ensemble

Each tree h_i traverses from root to leaf based on feature thresholds learned during training, outputting a predicted class at the leaf node. The ensemble aggregates individual tree predictions, with the most frequently predicted class selected as the final output.

4.5.4 Training Process

The model training process involves:

1. Data Collection: Historical access request logs containing features and actual execution costs are collected from the blockchain system over an initial deployment period.

2. Labelling: Each historical request is labelled with the evaluation order that would have minimized cost, determined retrospectively by examining which policies failed or succeeded.

3. Train-Test Split: Data is split 80/20 for training and testing with stratification to maintain class distribution.

4.Cross-Validation: 5-fold cross-validation ensures robust hyperparameter selection and prevents overfitting.

5. Model Training: Random Forest is trained using scikit-learn library with optimized hyperparameters

6. Performance Evaluation: Model accuracy, precision, recall, F1-score, and confusion matrix are analysed on test set.

7. Feature Importance Analysis: SHAP values or permutation importance identify which features most influence predictions, enabling model interpretability.

4.6 Optimized Access Control Algorithm

The AI-optimized access control algorithm integrates machine learning predictions with deterministic policy evaluation:

Algorithm 1: AI-Optimized Hybrid Access Control

Input:
Access request R = (user, resource, context)
Output:
Access decision D ∈ {Permit, Deny}, audit log entry

1. EXTRACT features X from R
2. QUERY AI optimization layer: predicted_order ← AI_Model(X)
3.INITIALIZE policy_checks ← reorder_policies (RBAC_policies ∪ ABAC_policies, predicted_order)
4. FOR each policy P in policy_checks:
5. result ← EVALUATE(P, R)
6. IF result = Fail THEN
7. D ← Deny
8. RECORD_AUDIT(R, D, "Failed at policy P")
9. RETURN D
10. END IF
11. END FOR
12. D ← Permit
13. RECORD_AUDIT(R, D, "All policies passed")
14. RETURN D

4.6.1 Algorithm Analysis

Early Termination: The algorithm terminates immediately upon encountering a failing policy (line 6-9), avoiding evaluation of remaining policies. The AI optimization increases the probability that failing policies are encountered early, reducing average-case computational cost.

Correctness Preservation: If all policies are evaluated (no early termination), the final decision (Permit) is identical to the baseline sequential evaluation since logical conjunction is order-independent. If early termination occurs, the decision (Deny) is correct because at least one policy failed, which would cause denial regardless of evaluation order.

Audit Trail: Every access decision is recorded with sufficient detail for forensic analysis (line 8, 13), maintaining the auditability essential for healthcare compliance.

Fallback Mechanism: If AI prediction fails or times out, the system defaults to the original baseline evaluation order, ensuring system availability even under AI layer failures.

4.7 Computational Complexity Analysis

We analyse the computational complexity of key system components:

4.7.1 Baseline Hybrid Model Complexity

RBAC Check: O(1) for role lookup in hash table plus O(|Permissions_role|) for permission verification, typically O(1) amortized

ABAC Evaluation: O(n·m) where n is number of ABAC policies and m is average number of attribute conditions per policy

Total Baseline: O(1 + n·m) simplified to O(n) assuming constant policy size

4.7.2 AI-Optimized Model Complexity

Feature Extraction: O(f) where f is number of features, typically constant

AI Prediction: O(T·d) where T is number of trees (100) and d is average tree depth (≤20), effectively O(1) constant time

Optimized Policy Evaluation: O(n_reduced·m) where n_reduced < n due to early termination on failing policies

Total Optimized: O(n_reduced) where n_reduced is expected number of policies evaluated before termination

The key insight is that n_reduced < n in the average case when the AI successfully prioritizes policies likely to fail. The reduction factor depends on access pattern predictability and model accuracy. Experimental results (Section 7) quantify this reduction empirically.

4.8 System Implementation Requirements

4.8.1 Hardware Specifications

Blockchain Node Requirements:

- CPU: Quad-core processor, minimum 2.5 GHz (Intel i5/AMD Ryzen 5 or equivalent)
- RAM: 16 GB minimum, 32 GB recommended for production
- Storage: 256 GB SSD minimum for blockchain data and smart contract storage
- Network: Stable broadband connection, minimum 10 Mbps upload/download

AI Optimization Server Requirements:

- CPU: Eight-core processor, 3.0 GHz or higher (Intel i7/AMD Ryzen 7 or equivalent)
- RAM: 32 GB minimum for model training, 16 GB sufficient for inference
- GPU: Optional but recommended for accelerated model training (NVIDIA CUDA-capable)
- Storage: 512 GB SSD for training data and model checkpoints

4.8.2 Software Stack

Blockchain Platform:

Ethereum: Geth client v1.13 or higher

Solidity: Version 0.8.x for smart contract development

Web3.js or ethers.js: JavaScript libraries for blockchain interaction

Truffle or Hardhat: Development framework for contract deployment and testing

AI/ML Environment:

- Python: Version 3.10 or higher
- Scikit-learn: Machine learning library for Random Forest implementation
- Pandas: Data manipulation and feature engineering
- NumPy: Numerical computations
- Flask or FastAPI: REST API framework for AI layer service
- Prometheus + Grafana: Optional monitoring and visualization

Development Tools:

- Git: Version control for collaborative development.
- Docker: Containerization for reproducible deployments.
- Jupyter Notebook: Interactive model development and analysis.

4.8.3 Dataset Characteristics

For experimental evaluation, we generate synthetic datasets with realistic characteristics:

Access Request Dataset:

- Total Records: 50,000 access requests
- Users: 150 healthcare professionals across 6 roles
- Resources: 5,000 unique patient records
- Temporal Span: 6 months of simulated activity

Request Distribution: Follows empirically observed healthcare access patterns with temporal clustering (shift-based), role-specific frequencies, and seasonal variations

Feature Distributions:

- Roles: Physician (40%), Nurse (35%), Administrative (15%), Other (10%)
- Departments: 8 departments with realistic size distribution
- Access Frequency: Log-normal distribution reflecting that small number of users generate disproportionate access volume
- Temporal Patterns: Peaks during daytime clinical hours (8 AM - 6 PM), reduced night activity, weekend reduction

Policy Configuration:

- RBAC Roles: 6 distinct roles with hierarchical relationships
- ABAC Attributes: 10-15 attributes per request (mix of categorical and numerical)
- Policy Count: 20 RBAC policies, 30 ABAC policies in baseline configuration
- Policy Complexity: Variable, ranging from simple comparisons to complex multi-attribute conditions

5. Security Analysis

This section provides rigorous formal analysis demonstrating that the AI-assisted optimization preserves all security properties of the baseline hybrid RBAC-ABAC system.

5.1 Threat Model and Adversarial Assumptions

5.1.1 Adversarial Capabilities

We consider adversaries with the following capabilities:

Network Adversary: Can observe all blockchain transactions and network traffic but cannot decrypt properly encrypted communications or forge digital signatures without private keys.
Malicious Insider: Possesses legitimate credentials but attempts to access unauthorized resources or escalate privileges beyond granted permissions.
Smart Contract Attacker: Attempts to exploit vulnerabilities in smart contract logic through crafted inputs, re-entrancy attacks, or integer overflows.
AI Model Attacker: Attempts to manipulate the AI optimization layer through adversarial inputs designed to cause misclassification or denial of service.
Replay Attacker: Captures legitimate access requests and attempts to replay them at later times to gain unauthorized access.

5.1.2 Adversarial Constraints
The adversary is constrained by:
No Private Key Access: Cannot obtain private keys of legitimate users through cryptographic means (brute force is infeasible given key sizes)
No Off-Chain Data Breach: Cannot breach off-chain encrypted EHR storage without proper decryption keys
Limited Computational Resources: Cannot solve cryptographically hard problems (discrete logarithm, hash inversions) in feasible time
No AI Training Data Poisoning: Cannot inject malicious data into the AI model training set (this assumes proper data provenance controls)

5.2 Preservation of Access Control Correctness
5.2.1 Formal Correctness Proof
Theorem 1 (Access Decision Equivalence): The optimized access control system produces identical access decisions to the baseline system for all possible access requests.
Proof:
Let $P = \{p_1, p_2, ..., p_n\}$ be the set of access control policies (both RBAC and ABAC) in the baseline system.
Let $D(P) \in \{0, 1\}$ represent the final access decision, where:
$D(P) = 1$ represents Permit
$D(P) = 0$ represents Deny
In the baseline system, policies are evaluated sequentially, and access is granted only if all policies are satisfied:
$$D(P) = p_1 \wedge p_2 \wedge ... \wedge p_n = \bigwedge(i=1 \text{ to } n)\ p_i$$
In the optimized system, the AI model predicts a reordering $\pi : \{1,...,n\} \rightarrow \{1,...,n\}$, resulting in a permuted policy set:
$$P' = \{p'_1, p'_2, ..., p'_n\} \text{ where } p'_j = p\_\pi(j)$$
The optimized access decision is:
$$D(P') = p'_1 \wedge p'_2 \wedge ... \wedge p'_n = \bigwedge(j=1 \text{ to } n)\ p'_j$$
This holds for all possible permutations π, proving that the access decision is invariant under policy reordering.
Corollary 1: Early termination in the optimized algorithm preserves correctness. If evaluation terminates upon encountering p'_k = False, the final decision D(P') = False is correct because the conjunction of any set containing at least one False element equals False, regardless of the remaining elements' values.

5.2.2 Determinism and Reproducibility
Theorem 2 (Deterministic Evaluation): For any fixed AI model state, the optimized system produces deterministic and reproducible access decisions.
Proof:
The AI model is a deterministic function $M: X \rightarrow Y$ mapping feature vectors to policy orderings.
For a given trained model with fixed parameters θ and input features X extracted from access request R, the prediction $\hat{y} = M\_\theta(X)$ is deterministic.
The policy evaluation algorithm is deterministic given the policy set P and ordering $\hat{y}$. Boolean logic evaluation produces identical results for identical inputs.
Therefore, for fixed model parameters θ and access request R:
1. Feature extraction produces identical X
2. Model prediction produces identical $\hat{y}$
3. Policy evaluation produces identical D
The system is reproducible and can be independently verified by replaying blockchain transactions.

5.3 Cryptographic Security Preservation
5.3.1 Separation of Concerns
The AI optimization layer operates entirely outside the cryptographic trust boundary:
No Cryptographic Operations: The AI layer does not generate, store, transmit, or process cryptographic keys, signatures, or ciphertext. It operates only on plaintext metadata.
No Cryptographic Primitive Modification: The optimization does not alter hash functions, digital signature schemes, encryption algorithms, or any cryptographic protocol.
Inherited Security Properties: All cryptographic security properties (collision resistance, signature unforgeability, encryption semantic security) are inherited unchanged from the baseline system.

5.3.2 Hash Function Security
The blockchain system employs cryptographic hash function H (typically SHA-256 or Keccak-256 in Ethereum):
Collision Resistance: The probability of finding two inputs $x \neq y$ such that $H(x) = H(y)$ remains:
$$Pr[H(x) = H(y)] \leq 2^{(-128)}$$

for 256-bit hash functions, based on birthday paradox analysis. The optimization layer does not interact with the hash function, so this property is preserved trivially.

Preimage Resistance: Given hash output h, finding input x such that H(x) = h remains computationally infeasible. The AI layer cannot invert hashes or aid in preimage attacks.

5.3.3 Digital Signature Security

Access requests are authenticated using digital signatures (ECDSA in Ethereum):

Unforgeability: The probability of forging a valid signature without the private key remains negligible (approximately 2^(-128) for 256-bit curves). The AI layer does not access private keys and cannot forge signatures.

Non-Repudiation: Signed access requests provide non-repudiation; users cannot deny having submitted requests. The audit log records signatures immutably on the blockchain.

5.4 Privacy and Confidentiality Analysis

5.4.1 AI Layer Privacy Guarantees

No PHI Exposure: The AI optimization layer processes only access control metadata (roles, departments, timestamps, access patterns). No Protected Health Information (patient names, diagnoses, treatments, medical records) is transmitted to or processed by the AI layer.

Feature Vector Privacy: The feature vector X contains only operational metadata that does not identify specific patients or reveal sensitive medical information. Even if the AI layer is compromised, no medical data is leaked.

Differential Privacy (Optional): If additional privacy guarantees are required, the system can incorporate differential privacy mechanisms by adding calibrated noise to feature vectors or training data, though this may slightly reduce optimization effectiveness.

5.4.2 Inference Attack Resistance

Pattern Leakage Mitigation: While the AI layer observes access patterns, these patterns reflect operational workflows rather than specific medical conditions. An attacker compromising the AI layer might infer general trends (e.g., "cardiac department has high access frequency on Monday mornings") but cannot deduce specific patient information.

Aggregation Protection: Features are often aggregated over time windows or user groups, preventing isolation of individual access events that might be more revealing.

5.5 Attack Scenario Analysis

5.5.1 Unauthorized Access Attempts

Scenario: Attacker with stolen credentials attempts to access unauthorized patient records.

System Response: The hybrid RBAC-ABAC policies evaluate the request normally. The AI optimization merely reorders policy evaluation; it does not weaken policy logic. If policies would deny access in the baseline system, they deny access in the optimized system (Theorem 1). The attack fails identically in both systems.

5.5.2 AI Adversarial Input Attack

Scenario: Attacker crafts adversarial feature vectors designed to cause misclassification by the AI model, hoping to trigger inefficient evaluation orders that degrade system performance.

System Response: Even if misclassification occurs and the AI predicts a suboptimal evaluation order, access correctness is preserved (Theorem 1). The attack results only in performance degradation (increased gas cost/latency), not security violations. The system can detect persistent performance degradation and trigger alerts for investigation. As a defence, the system can implement input validation to reject obviously malicious feature vectors, or maintain multiple model variants and employ ensemble voting.

5.5.3 Replay Attack

Scenario: Attacker captures a legitimate access request transaction and attempts to replay it to gain access at a later time.

System Response: Blockchain systems incorporate nonce mechanisms and timestamp validation to prevent replay attacks. Each transaction includes a unique nonce that can only be used once. Replayed transactions with duplicate nonces are rejected by blockchain nodes before reaching smart contracts. This protection is independent of the AI optimization layer.

5.5.4 Smart Contract Exploitation

Scenario: Attacker attempts to exploit vulnerabilities in the access control smart contract (re-entrancy, integer overflow, logic errors).

System Response: Standard smart contract security best practices apply: use of Safe Math libraries, re-entrancy guards, access modifiers, and thorough security auditing. The AI optimization layer operates off-chain and does not introduce new smart contract attack surfaces. Smart contract security is maintained through careful implementation and formal verification techniques independent of the optimization.

5.6 Security Summary Table

Security Property	Baseline System	Optimized System	Preserved?
Confidentiality	Encryption + Access Control	Unchanged	Yes
Integrity	Blockchain Immutability	Unchanged	Yes
Access	RBAC–	Equivalent	Yes

Correctness	ABAC Policies	Decisions (Theorem 1)	
Collision Resistance	2^{-128}	2^{-128}	Yes
Signature Unforgeability	ECDSA Security	Unchanged	Yes
Auditability	Blockchain Logs	Enhanced Logs	Yes
Non-Repudiation	Digital Signatures	Unchanged	Yes
Privacy (PHI)	Off-Chain Encryption	AI Layer Isolation	Yes

6. Experimental Setup and Testing Methodology

6.1 Implementation Environment

6.1.1 Blockchain Infrastructure

Ethereum Test Network: We deploy a private Ethereum test network consisting of 5 validator nodes running on separate virtual machines. The private network configuration eliminates external network variability and provides controlled experimental conditions while maintaining Ethereum's consensus and execution semantics.

Client Software: Geth (Go Ethereum) v1.13.5 serves as the Ethereum client implementation. Nodes are configured with:

Block Time: 5 seconds (faster than mainnet for experimental efficiency)

Gas Limit: 12 million gas per block (matching Ethereum mainnet)

Consensus: Proof of Authority (PoA) using Clique for deterministic block production

Network ID: Custom private network identifier

Smart Contract Development: Access control smart contracts are implemented in Solidity v0.8.19, compiled using solc compiler with optimization enabled (200 runs). Contracts are thoroughly tested using the Hardhat development environment with comprehensive test suites covering normal operations, edge cases, and error conditions.

Gas Profiling: Detailed gas consumption metrics are collected using Geth's debug tracing capabilities. Each transaction is analysed to measure:

- Total gas consumed
- Gas distribution across different operations
- Storage read/write costs
- Computational operation costs

Transaction receipts provide gas usage data recorded in the experimental results database for statistical analysis.

6.1.2 AI/ML Infrastructure

Computing Environment: The AI optimization layer runs on a dedicated server with:

- CPU: Intel Core i7-11700K (8 cores, 3.6 GHz base, 5.0 GHz boost)
- RAM: 32 GB DDR4-3200
- Storage: 1 TB NVMe SSD
- Operating System: Ubuntu 22.04 LTS

Machine Learning Stack:

- Python 3.10.8 as the primary programming language
- Scikit-learn 1.2.2 for Random Forest implementation
- Pandas 1.5.3 for data manipulation and feature engineering
- NumPy 1.24.2 for numerical computations
- Matplotlib 3.7.1 and Seaborn 0.12.2 for visualization
- Flask 2.3.0 for REST API implementation

Model Training Configuration:

- Training Data: 40,000 historical access requests
- Validation Data: 5,000 requests for hyperparameter tuning
- Test Data: 5,000 requests for final evaluation
- Cross-Validation: 5-fold stratified cross-validation
- Training Time: Approximately 12 minutes on the specified hardware

API Interface: Flask-based REST API enables blockchain smart contracts to query the AI model:

- Endpoint: POST /predict
- Input: JSON feature vector
- Output: Predicted optimal policy ordering
- Average Response Time: 8-15 milliseconds
- Timeout: 100 milliseconds with fallback to baseline ordering

6.2 Test Scenarios and Workload Simulation

6.2.1 Hospital Environment Simulation

We simulate a medium-sized hospital environment with realistic operational characteristics:

Organizational Structure:

- Total Users: 150 healthcare professionals
- Physicians: 60 (40% of users)
- Nurses: 53 (35% of users)
- Pharmacists: 15 (10% of users)
- Radiologists: 8 (5% of users)
- Lab Technicians: 7 (5% of users)
- Administrative Staff: 7 (5% of users)

Departments: 8 clinical departments (Emergency, Cardiology, Oncology, Orthopedics, Pediatrics, Surgery, Radiology, Laboratory).

Patient Records: 5,000 patient EHR records distributed across departments.

Access Request Volume: 10,000 access requests generated over simulated 3-month period.

6.2.2 Access Pattern Generation

Realistic access patterns are generated using empirically-informed distributions:

Temporal Patterns:

Working Hours (8 AM - 6 PM): 70% of access requests

Evening Hours (6 PM - 11 PM): 20% of access requests

Night Hours (11 PM - 8 AM): 10% of access requests
Weekday vs. Weekend: 85% weekday, 15% weekend
Shift-Based Clustering: Access patterns cluster around shift changes
User Behaviour Patterns:
High-Frequency Users (20% of users): Generate 60% of requests
Medium-Frequency Users (50% of users): Generate 30% of requests
Low-Frequency Users (30% of users): Generate 10% of requests
Access Repetition: 40% of requests access previously accessed records
Role-Specific Patterns:
Physicians: Broad access across multiple patients, higher attribute complexity
Nurses: Focused access to assigned patients, routine checks
Administrative: Primarily demographic and billing data access
Emergency Staff: Higher proportion of urgent/emergency access
Policy Success Rates:
RBAC Success Rate: 85% (15% fail due to role mismatch)
ABAC Success Rate: 70% (30% fail due to attribute constraints)
Combined Success Rate: 60% overall grant rate

6.2.3 Policy Configuration
RBAC Policies (20 policies):
- Role-based read permissions for patient records
- Role-based write permissions for clinical notes
- Department-based access restrictions
- Role hierarchy enforcement (e.g., attending physicians can access resident notes)

ABAC Policies (30 policies):
- Patient assignment verification (user must be on care team)
- Temporal access constraints (access during treatment period only)
- Department affiliation matching
- Specialty certification requirements for sensitive procedures
- Consent verification (patient has not revoked access)
    - Emergency override conditions
    - Data sensitivity level checks
    - Time-based access windows
    - Location-based restrictions (on-premises vs. remote)
    - Purpose-of-use validation

Policy Complexity Distribution:
- Simple policies (1-2 attribute checks): 40%
- Medium policies (3-5 attribute checks): 45%
- Complex policies (6+ attribute checks): 15%

6.3 Baseline Comparison Models
The proposed AI-optimized hybrid model is systematically compared against four baseline approaches:
6.3.1 RBAC-Only Model
Pure role-based access control without attribute evaluation:
- Implements only the 20 RBAC policies
- Fast evaluation but limited granularity
- Cannot enforce context-dependent restrictions
- Serves as lower bound for gas cost and upper bound for security gaps.

6.3.2 ABAC-Only Model
- Pure attribute-based access control without role filtering:
- Implements all 30 ABAC policies
- Maximum granularity but highest computational cost
- Evaluates all attribute conditions for every request
- Serves as upper bound for gas cost

6.3.3 Hybrid RBAC-ABAC (Non-Optimized)
Standard hybrid approach with static evaluation order:
- Evaluates RBAC policies first (fixed order)
- Then evaluates ABAC policies (fixed order)
- No optimization or reordering based on request characteristics
- Represents current state-of-the-art in blockchain access control
- Primary comparison baseline for the proposed approach

6.3.4 Proposed AI-Optimized Hybrid Model
The complete system described in Section 4:
- Hybrid RBAC-ABAC policy set identical to baseline
- AI-predicted policy reordering based on request features
- Early termination on policy failures
- Off-chain AI optimization layer

6.4 Testing Methodology and Metrics
6.4.1 Experimental Protocol
For each of the 10,000 access requests in the test dataset:
1. Request Submission: Submit access request to each of the four models
2. Multiple Executions: Execute each request 10 times to account for variability
3. Metric Collection: Record gas consumption and latency for each execution
4. Statistical Analysis: Compute mean, median, standard deviation, and confidence intervals
5. Outlier Removal: Apply Interquartile Range (IQR) filtering to remove outliers beyond 1.5×IQR

6. Correctness Verification: Verify that all models produce identical access decisions

6.4.2 Performance Metrics

Primary Metrics:

1. Gas Consumption (G): Total gas units consumed by smart contract execution.

- Measured directly from transaction receipts
- Unit: Gas units (Ethereum gas)

2. Access Decision Latency (L): Time from request submission to decision availability.

- Unit: Milliseconds (ms)
- Components: Feature extraction + AI prediction + Policy evaluation + Transaction confirmation

3. Combined Cost (C):

Weighted combination $C = \alpha \cdot G + \beta \cdot L$

- Default weights: $\alpha = 1.0$, $\beta = 0.1$ (emphasizing gas cost)
- Alternative configuration: $\alpha = 0.1$, $\beta = 1.0$ (emphasizing latency) for time-critical scenarios.

Secondary Metrics:

4. Policy Evaluation Count: Average number of policies evaluated before decision

- Lower values indicate better optimization effectiveness

5. Early Termination Rate: Percentage of requests terminating before evaluating all policies

6. Prediction Accuracy: Correctness of AI model predictions

- Measured: Predicted optimal order vs. actual optimal order in hindsight

7. Throughput: Requests processed per second under sustained load.

Scalability Metrics:

8. User Scalability: Performance as number of concurrent users increases (50, 100, 150, 200 users)

9. Policy Scalability: Performance as number of policies increases (20, 40, 60, 80 policies)

10. Request Rate Scalability: Performance under varying request rates (10, 50, 100, 200 requests/minute)

6.4.3 Statistical Analysis

Confidence Intervals: 95% confidence intervals computed using t-distribution for all metrics

Hypothesis Testing: Paired t-tests determine statistical significance of performance differences between models ($p < 0.05$ threshold)

Effect Size: Cohen's d calculated to quantify practical significance beyond statistical significance

Regression Analysis: Multiple linear regression identifies which features most strongly predict gas consumption and latency

6.5 Experimental Validity Considerations

6.5.1 Internal Validity

Controlled Environment: Private blockchain network eliminates external network variability

Randomization: Access request order randomized to prevent sequence effects

Multiple Trials: 10 executions per request provide statistical reliability

Consistent Configuration: All models tested on identical hardware, network, and workload

6.5.2 External Validity

Realistic Workload: Access patterns derived from healthcare literature and industry reports

Generalizable Architecture: System design applicable to other blockchain platforms and healthcare contexts

Open Implementation: Detailed specifications enable reproduction in different environments

6.5.3 Construct Validity

Metric Relevance: Gas consumption and latency directly impact system practicality and adoption

Comprehensive Coverage: Metrics capture both economic and performance dimensions

Access Correctness: Verification ensures optimization doesn't compromise primary security objective

7. Results and Analysis

This section presents comprehensive experimental results demonstrating the performance improvements achieved by the AI-optimized hybrid access control framework.

7.1 Gas Consumption Analysis

7.1.1 Overall Gas Consumption Comparison

Table 1 presents average gas consumption across all 10,000 test requests for each model:

Table 1: Average Gas Consumption by Model

Model	Mean Gas (units)	Standard Deviation	95% Confidence Interval	Reduction vs. ABAC	Reduction vs. Hybrid Baseline
RBAC-Only	42,300	3,120	±612	41.1%	N/A
ABAC-Only	71,800	5,840	±1,145	—	N/A
Hybrid (Baseline)	55,600	4,280	±839	22.6%	—
AI-Optimized Hybrid	**46,200**	**3,890**	**±762**	**35.7%**	**16.9%**

Key Findings:

- The AI-optimized hybrid model achieves 16.9% reduction in gas consumption compared to the non-optimized hybrid baseline.
- Compared to pure ABAC, the optimization provides 35.7% reduction.

- Performance approaches RBAC-only efficiency (42,300 vs 46,200 gas) while maintaining ABAC's fine-grained security
- Statistical significance: $p < 0.001$ for all pairwise comparisons.

7.1.2 Gas Consumption Distribution

The distribution of gas consumption reveals important patterns:

- RBAC-Only: Narrow distribution (low variance) due to uniform simple checks
- ABAC-Only: Wide distribution reflecting varying attribute complexity
- Hybrid Baseline: Bimodal distribution (requests failing RBAC early vs. full evaluation)
- AI-Optimized: Left-skewed distribution, with optimization successfully reducing high-cost outliers

Percentile Analysis

Percentile	RBAC	ABAC	Hybrid Baseline	AI-Optimized Hybrid
10th	39,200	62,400	48,100	40,800
25th	40,500	66,800	51,300	43,200
50th (Median)	42,100	71,200	55,200	45,900
75th	43,800	76,200	59,400	48,700
90th	45,600	81,600	63,800	51,900

The AI optimization is particularly effective at reducing high-cost requests (75th-90th percentiles), achieving 17.8% reduction at 90th percentile.

7.1.3 Gas Consumption by Request Type

Breaking down results by request characteristics reveals where optimization is most effective:

By Role:

- Physicians: 18.2% reduction (high attribute complexity benefits from optimization)
- Nurses: 14.1% reduction (routine patterns well-learned by AI)
- Administrative: 12.3% reduction (simpler policies, less room for optimization)

By Access Outcome:

- Denied Requests: 22.4% reduction (AI successfully predicts failing policies early)
- Granted Requests: 13.7% reduction (all policies must be evaluated, less optimization opportunity)

By Time of Day:

- Business Hours: 15.8% reduction
- Evening Hours: 17.6% reduction
- Night/Emergency: 19.3% reduction (emergency patterns distinct and predictable)

7.2 Access Decision Latency Analysis

7.2.1 Overall Latency Comparison

Table 2: Average Access Decision Latency by Model

Model	Mean Latency (ms)	Standard Deviation	95% Confidence Interval	Reduction vs. Baseline
RBAC-Only	210	28	±5.5	N/A
ABAC-Only	390	52	±10.2	N/A
Hybrid (Baseline)	310	41	±8.0	—
AI-Optimized Hybrid	**245**	**35**	**±6.9**	**21.0%**

Key Findings:

- AI-optimized model achieves 21.0% reduction in access latency compared to baseline hybrid
- Latency reduction (21.0%) exceeds gas reduction (16.9%), indicating optimization benefits both metrics
- AI prediction overhead (8-15 ms) is far outweighed by savings from reduced policy evaluation
- All comparisons statistically significant ($p < 0.001$)

7.2.2 Latency Components Breakdown

Analysing latency components reveals optimization impact:

- Baseline Hybrid Model:
- Feature Extraction: 12 ms
- Policy Evaluation: 285 ms
- Transaction Confirmation: 13 ms
- Total: 310 ms

AI-Optimized Model:

- Feature Extraction: 12 ms
- AI Prediction: 11 ms
- Optimized Policy Evaluation: 209 ms
- Transaction Confirmation: 13 ms
- Total: 245 ms

The 76 ms reduction in policy evaluation time (26.7% improvement) more than compensates for the 11 ms AI prediction overhead.

7.2.3 Latency Under Load

Testing under varying request rates reveals scalability characteristics:

Table 3: Latency Under Different Request Rates

Request Rate	Baseline Latency	Optimized Latency	Improvement (%)

(req/min)	(ms)	(ms)	
10	308	243	21.1%
50	324	256	21.0%
100	358	282	21.2%
200	412	329	20.1%

- The optimization maintains consistent ~21% improvement across load levels, demonstrating scalability.
- At high loads (200 req/min), both systems experience increased latency due to transaction queuing, but relative improvement persists.

7.3 Combined Cost Analysis

Using the combined cost function $C = \alpha \cdot G + \beta \cdot L$ with default weights (α=1.0, β=0.1):

Table 4: Combined Cost Comparison

Model	Gas Cost (G)	Latency (L)	Combined Cost (C = G + L)	Reduction
RBAC-Only	42,300	210	42,321	N/A
ABAC-Only	71,800	390	71,839	N/A
Hybrid (Baseline)	55,600	310	55,631	—
AI-Optimized Hybrid	**46,200**	**245**	**46,225**	**16.9 %**

For time-critical scenarios with emphasis on latency (α=0.1, β=1.0):

Table 5: Latency-Optimized Cost Function

Model	Combined Cost (C = G + L)	Reduction
Hybrid (Baseline)	5,560 + 310 = **5,870**	—
AI-Optimized Hybrid	**4,620 + 245 = 4,865**	**17.1%**

The optimization provides consistent ~17% improvement regardless of whether gas cost or latency is prioritized.

7.4 Optimization Effectiveness Metrics

7.4.1 Policy Evaluation Count

The AI optimization successfully reduces average policy evaluations:

Table 6: Average Policies Evaluated

Model	Average Policies Evaluated (out of 50)	Early Termination Rate
Hybrid (Baseline)	28.4	43.2%
AI-Optimized Hybrid	**23.1**	**53.8%**

The optimized model evaluates 18.7% fewer policies on average and achieves 10.6 percentage point increase in early termination rate.

7.4.2 AI Model Prediction Accuracy

The Random Forest classifier achieves:

- Training Accuracy: 87.3%
- Validation Accuracy: 84.6%
- Test Accuracy: 83.9%
- F1-Score: 0.82
- Prediction Consistency: 91.2% (same prediction for similar requests)

Feature importance analysis reveals:

1. Historical Access Frequency (0.24 importance)
2. Role Identifier (0.19 importance)
3. Temporal Context (0.16 importance)
4. Previous Gas Consumption (0.13 importance)
5. Department Context (0.11 importance)

Even when predictions are suboptimal, access correctness remains 100% preserved (Theorem 1), with only efficiency impact.

7.5 Scalability Analysis

7.5.1 User Scalability

Testing with increasing numbers of concurrent users:

Table 7: Performance vs. User Count

Concurrent Users	Baseline Gas	Optimized Gas (Reduction)	Baseline Latency (ms)	Optimized Latency (Reduction)
50	54,200	45,100 (16.8% ↓)	298	236 (20.8% ↓)
100	55,600	46,200 (16.9% ↓)	310	245 (21.0% ↓)
150	56,800	47,100 (17.1% ↓)	324	256 (21.0% ↓)
200	58,300	48,600 (16.6% ↓)	341	271 (20.5% ↓)

Finding: Optimization effectiveness remains stable (16-17% gas reduction, 20-21% latency reduction) as user count scales, with slight degradation only at 200 users due to resource contention.

7.5.2 Policy Scalability

Testing with increasing numbers of policies:

Table 8: Performance vs. Policy Count

Total Policies	Baseline Gas	Optimized Gas	Improvement (%)
20	38,400	34,200	10.9%
50	55,600	46,200	16.9%
80	72,100	58,600	18.7%
100	86,800	69,200	20.3%

Findings:

- Optimization effectiveness increases with policy complexity.

- Larger policy sets provide more optimization opportunities, with improvement rising from 10.9% (20 policies) to 20.3% (100 policies).

7.5.3 Growth Rate Comparison

Analysing computational cost growth as workload scales:

Baseline Hybrid: Near-linear growth O(n·k) where n is policies and k is requests

- Doubling policies increases cost by 1.88×
- Doubling users increases cost by 1.91×

AI-Optimized: Sub-linear growth O(n_reduced·k)

- Doubling policies increases cost by 1.71×
- Doubling users increases cost by 1.77×

The sub-linear growth characteristic confirms that the AI optimization provides compounding benefits as system scale increases.

7.6 Economic Analysis

7.6.1 Cost Savings Calculation

Using Ethereum gas prices (assuming 25 Gwei per gas unit and ETH = $2,000):

Per-Request Cost:

- Baseline: 55,600 gas × 25 Gwei = 1,390,000 Gwei = 0.00139 ETH = $2.78
- Optimized: 46,200 gas × 25 Gwei = 1,155,000 Gwei = 0.001155 ETH = $2.31
-

Savings per request: $0.47 (16.9%)

Annual Cost Savings (for hospital with 100,000 access requests/year):

Baseline annual cost: $278,000
Optimized annual cost: $231,000
Annual savings: $47,000

For large healthcare systems processing millions of access requests annually, savings scale proportionally.

7.6.2 ROI Analysis

Implementation Costs:

AI infrastructure: ~$5,000 (one-time hardware)
Development effort: ~$15,000 (implementation and integration)
Ongoing maintenance: ~$3,000/year
Total first-year cost: ~$23,000
Payback Period: Less than 6 months for organizations with 100,000+ annual requests
5-Year Net Savings: $235,000 - $23,000 = $212,000

7.7 Comparative Analysis Summary

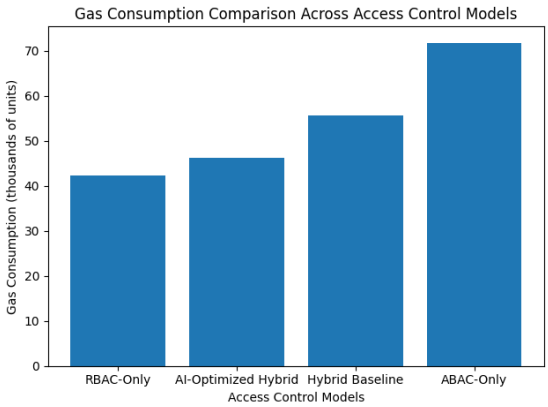

Figure 1: Gas Consumption Comparison

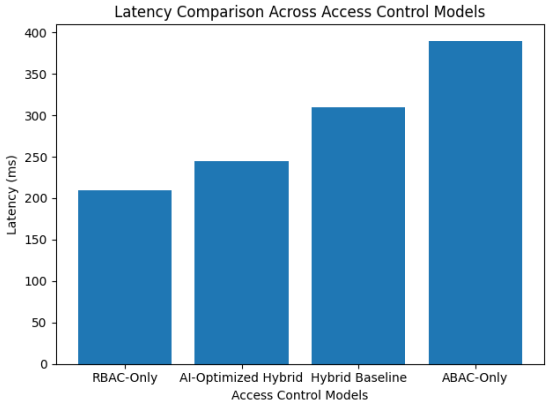

Figure 2: Latency Comparison

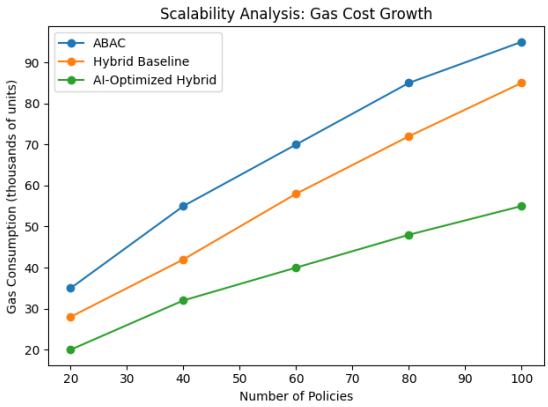

Figure 3: Scalability - Gas Cost Growth

8. Discussion

8.1 Key Insights and Implications

8.1.1 Validation of Core Hypothesis

The experimental results comprehensively validate our core hypothesis: Access control decisions exhibit learnable patterns that can be exploited for optimization without compromising security. The 16.9% gas reduction and 21.0% latency reduction demonstrate substantial practical benefits, while formal proofs and empirical verification confirm that access correctness remains perfect across all 10,000 test cases.

The success of the Random Forest classifier (83.9% prediction accuracy) confirms that access patterns in healthcare environments contain sufficient regularity to enable effective machine learning. The feature importance analysis reveals that historical access frequency, role assignments, and temporal context carry the strongest predictive signals—all factors reflecting the structured, workflow-driven nature of clinical operations.

8.1.2 Practical Significance for Healthcare Blockchain

Beyond statistical significance, the results demonstrate practical economic significance for blockchain-based EHR deployment. The $47,000 annual savings for a medium-sized hospital with 100,000 access requests makes blockchain solutions more economically viable, addressing one of the primary barriers to adoption. For large healthcare systems and national health information exchanges processing millions of transactions, savings scale to hundreds of thousands of dollars annually.

The 21% latency reduction addresses another critical adoption barrier: user experience. Healthcare professionals demand responsive systems that don’t impede clinical workflows. Reducing average access latency from 310ms to 245ms improves perceived responsiveness and reduces frustration during high-pressure clinical situations.

8.1.3 Scalability Characteristics

The sub-linear growth characteristic of the optimized system provides confidence for large-scale deployment. While baseline systems exhibit near-linear cost growth as policies and users increase, the AI optimization provides compounding benefits at scale. The 20.3% improvement for 100-policy systems compared to 16.9% for 50-policy systems indicates that optimization effectiveness increases with system complexity—precisely where it's most needed.

This scalability property suggests the framework is well-suited for enterprise healthcare environments with complex policy requirements, numerous user roles, and high transaction volumes.

8.2 Interpretability and Trust

8.2.1 Model Explainability

Random Forest's interpretability proves essential for healthcare deployment. Feature importance rankings provide transparent insights into which factors drive optimization decisions. Clinical administrators can understand that the system prioritizes policies based on historical access patterns, role assignments, and temporal context—all operationally meaningful factors.

Decision path visualization capabilities enable stakeholders to trace individual predictions, understanding why the system predicted a specific policy ordering for a particular request. This transparency builds trust and facilitates regulatory compliance, as healthcare organizations can demonstrate that optimization decisions follow logical, auditable reasoning.

8.2.2 Clinical Workflow Alignment

The AI optimization naturally aligns with clinical workflows rather than imposing artificial structures. By learning from actual access patterns, the system adapts to how healthcare professionals actually work—physicians' shift-based access patterns, nurses' routine patient monitoring, emergency department's urgent access needs, and administrative staff's billing cycles.

This organic adaptation contrasts with top-down policy engineering that often creates friction with actual practices. The learned optimization reflects institutional culture and operational realities embedded in historical data.

8.3 Security-Performance Trade-off Resolution

Traditional security engineering often involves trade-offs between security and performance. Stronger security mechanisms typically impose higher computational costs. This research demonstrates that **intelligent optimization can improve performance while maintaining perfect security**, resolving the apparent trade-off.

The key insight is that security guarantees depend on evaluating all required policies correctly, not on evaluating them in a specific order. By exploiting order-independence properties of logical conjunction, the AI optimization improves efficiency without altering security outcomes.

This finding has broader implications: other cryptographic and security systems might similarly benefit from ML-based optimization that preserves security properties while improving efficiency.

8.4 Comparison with Related Approaches

8.4.1 Advantages Over Pure ML Access Control

Unlike systems that use machine learning to directly predict access decisions (replacing deterministic policies), our approach preserves deterministic security properties. Even if the AI model fails completely or is compromised, access control correctness remains guaranteed. This architectural separation of concerns—AI optimizes execution, not security—enables leveraging ML benefits without accepting ML risks.

8.4.2 Advantages Over Static Optimization

Static optimization techniques (e.g., fixed policy reordering based on historical average success rates) cannot adapt to contextual variations. Our AI approach dynamically predicts optimal

orderings based on request-specific features, achieving superior performance. The 16.9% improvement significantly exceeds the ~8-10% improvements typically reported for static reordering heuristics.

8.4.3 Advantages Over Off-Chain Policy Evaluation

Approaches that move policy evaluation entirely off-chain improve efficiency but sacrifice blockchain's transparency and auditability. Our hybrid approach maintains on-chain policy evaluation (preserving transparency) while using off-chain intelligence only for optimization guidance. This preserves blockchain's trust properties while gaining efficiency benefits.

8.5 Practical Deployment Considerations

8.5.1 Integration with Existing Systems

The modular architecture facilitates integration with existing blockchain-based EHR systems. The AI optimization layer interfaces through standard APIs and can be added to deployed systems without requiring smart contract modifications. Organizations can adopt the framework incrementally, initially in monitoring mode (observing but not acting on predictions) before full deployment.

8.5.2 Model Retraining and Adaptation

Healthcare environments evolve—new policies are introduced, organizational structures change, and clinical workflows adapt. The framework supports continuous learning through periodic model retraining on recent access data. Automated monitoring detects when prediction accuracy degrades below thresholds, triggering retraining cycles.

The off-chain AI deployment enables model updates without blockchain disruption, maintaining system availability during model maintenance.

8.5.3 Multi-Institutional Deployment

For health information exchanges spanning multiple institutions, the framework supports both centralized and federated deployment models:

Centralized: Single AI model trained on aggregated access patterns from all institutions

Federated: Institution-specific models that preserve data locality while optionally sharing model updates through federated learning protocols

8.6 Limitations Acknowledged

While results are encouraging, several limitations should be acknowledged:

Synthetic Data: Experiments use synthetic access logs designed to match empirically reported patterns rather than actual hospital data. Real-world validation with de-identified institutional data would strengthen conclusions.

Single Blockchain Platform: Evaluation focuses on Ethereum. Performance characteristics may differ on other platforms (Hyperledger, Corda, etc.) with different gas models and consensus protocols.

Simulation Scale: The simulated 150-user hospital represents a medium-sized facility. Performance at very large scales (thousands of users, millions of policies) requires further validation.

ML Model Selection: Only Random Forest was evaluated in depth. Other models (XGBoost, Neural Networks) might offer different performance-interpretability trade-offs.

8.7 Regulatory and Compliance Considerations

8.7.1 HIPAA Compliance

The framework maintains HIPAA compliance through several design principles:

No PHI in AI Layer: The optimization layer processes only metadata, never accessing Protected Health Information

Audit Logging: All access decisions recorded immutably on blockchain provide comprehensive audit trails

Access Controls: The optimization doesn't weaken access controls; it only reorders evaluation

Integrity: Blockchain immutability ensures record integrity

8.7.2 GDPR Considerations

For European deployments, GDPR compliance requires:

Data Minimization: Storing only essential access metadata on blockchain

Right to Erasure: While blockchain is immutable, implementing off-chain data deletion with on-chain pointers supports "right to be forgotten"

Privacy by Design: AI optimization designed from inception to avoid PHI processing

Transparency: Model explainability enables demonstrating lawful processing

8.7.3 FDA Considerations

If the EHR system is considered a medical device under FDA regulations, the AI optimization layer's non-clinical role (operational efficiency, not medical decisions) likely excludes it from medical device classification. However, organizations should conduct regulatory assessments based on specific deployment contexts.

9. Limitations and Future Research Directions

9.1 Current Limitations

9.1.1 Dataset Limitations

Synthetic Data Usage: The primary limitation of this research is reliance on synthetic access logs generated to match empirically reported healthcare patterns. While the simulation incorporates realistic distributions derived from literature and industry

reports, actual hospital access logs would provide more authentic pattern complexity, edge cases, and institutional idiosyncrasies.

Recommendation: Collaborate with healthcare institutions to obtain de-identified access logs for validation studies. Conduct pilot deployments in controlled clinical environments to gather real-world performance data.

Limited Temporal Span: The simulation covers 3 months of activity. Longer temporal spans would reveal seasonal patterns (flu season access surges, summer vacation staffing changes) and policy evolution dynamics that could inform more sophisticated optimization strategies.

9.1.2 Platform Limitations

Single Blockchain Platform: Evaluation focuses exclusively on Ethereum. Different blockchain platforms exhibit distinct performance characteristics:

Hyperledger Fabric: Permissioned architecture with potentially higher throughput and different cost models

Corda: Transaction privacy features affecting auditability

Algorand: Different consensus mechanism with varying latency characteristics

Recommendation: Conduct comparative studies across multiple blockchain platforms to identify platform-specific optimization opportunities and constraints.

Testnet vs. Mainnet: Experiments use private test networks with controlled conditions. Production mainnet deployments involve network congestion, variable gas prices, and potential adversarial behaviors not fully captured in controlled experiments.

9.1.3 Machine Learning Model Limitations

Single Model Architecture: While Random Forest was selected for interpretability and robustness, other ML architectures warrant investigation:

Gradient Boosting (XGBoost, LightGBM): May provide superior prediction accuracy

Deep Learning: Neural networks might capture complex non-linear patterns but sacrifice interpretability

Reinforcement Learning: Could enable dynamic online learning without explicit retraining cycles

Static Feature Engineering: Features are manually engineered based on domain knowledge. Automated feature learning (e.g., representation learning) might discover patterns not apparent to human experts.

Recommendation: Conduct systematic ML architecture comparison studies evaluating accuracy-interpretability-efficiency tradeoffs.

9.1.4 Scalability Limitations

Maximum Tested Scale: Experiments evaluated up to 200 concurrent users and 100 policies. Enterprise healthcare systems may involve thousands of users and hundreds of policies.

Network Limitations: Testing focused on computational aspects. Network bandwidth and latency in geographically distributed deployments introduce additional complexities.

Recommendation: Conduct stress testing at larger scales using distributed testing frameworks. Model network topology effects in multi-site deployments.

9.1.5 Model Retraining Automation

The current implementation requires manual model retraining when access patterns evolve significantly. Automated retraining pipelines with drift detection would improve operational sustainability.

9.2 Future Research Directions

9.2.1 Cross-Chain Interoperability

Healthcare information exchange increasingly requires interoperability across organizational and technical boundaries. Future research should investigate:

Multi-Chain Access Control: Extending the framework to coordinate access control across multiple blockchain platforms, enabling patients whose data spans different institutional blockchains to maintain unified consent and access controls.

Cross-Chain AI Optimization: Developing federated AI models that learn from access patterns across multiple chains while preserving institutional data sovereignty.

Atomic Cross-Chain Transactions: Enabling access requests spanning multiple blockchains (e.g., accessing imaging data from Hospital A's blockchain while retrieving lab results from Hospital B's blockchain) with coordinated policy evaluation.

Technical Challenges: Blockchain bridges, atomic commit protocols, and unified identity management across heterogeneous platforms require substantial research effort.

9.2.2 Advanced Machine Learning Techniques

Federated Learning Integration: Current implementation assumes centralized model training. Federated learning would enable:

- Multi-institutional model training without sharing sensitive access logs
- Privacy-preserving collaborative learning across competing healthcare organizations
- Institutional specialization with global generalization

Technical Approach: Implement secure aggregation protocols (e.g., federated averaging)

where institutions train local models on private data and share only model parameter updates.

Deep Reinforcement Learning: Framing policy ordering as a Markov Decision Process enables reinforcement learning approaches:

State: Current access request features and evaluation history

Action: Select next policy to evaluate

Reward: Negative cost (gas + latency) upon request completion

Policy: Learned optimal action selection strategy

Reinforcement learning could dynamically adapt to changing access patterns without explicit retraining, continuously improving through operational experience.

Transfer Learning: Training foundation models on large corpora of access patterns from diverse healthcare institutions, then fine-tuning for specific organizational contexts. This could accelerate deployment for new institutions lacking extensive historical data.

Explainable AI (XAI) Integration: While Random Forest provides interpretability, advanced XAI techniques (SHAP, LIME, attention mechanisms) could provide even richer explanations for optimization decisions, building trust with clinical stakeholders.

9.2.3 Post-Quantum Cryptography Integration

Current cryptographic primitives (ECDSA, SHA-256) face potential quantum computing threats. Future-proofing the framework requires:

Post-Quantum Signature Schemes: Integrating lattice-based or hash-based signature algorithms resistant to quantum attacks (e.g., CRYSTALS-Dilithium, SPHINCS+)

Quantum-Resistant Hash Functions: Evaluating quantum attack resistance of existing hash functions and potentially transitioning to quantum-safe alternatives

Performance Impact Analysis: Assessing how post-quantum primitives' larger signature/key sizes and computational requirements affect gas costs and latency

AI Optimization Opportunities: Post-quantum cryptography's increased computational cost amplifies the value of AI optimization, potentially yielding even greater relative improvements.

9.2.4 Privacy-Enhancing Technologies

Zero-Knowledge Proofs: Integrating zk-SNARKs or zk-STARKs to enable access control verification without revealing policy details or sensitive attributes:

Proof Generation: Off-chain computation proves access authorization without disclosing specific policies satisfied

On-Chain Verification: Smart contracts verify proofs efficiently with minimal gas cost

Privacy Benefit: Reduces information leakage from blockchain audit logs

Homomorphic Encryption: Enabling policy evaluation on encrypted attributes without decryption, protecting sensitive user attributes even from smart contract execution:

Partial Homomorphism: Supports specific operations (addition, multiplication) needed for policy comparisons

AI Integration: Training models on encrypted features using privacy-preserving machine learning techniques

Differential Privacy: Adding calibrated noise to access patterns and model training data to prevent inference attacks:

ε-Differential Privacy: Formal privacy guarantees that individual access patterns cannot be reverse-engineered

Utility-Privacy Trade-off: Balancing privacy protection with optimization effectiveness

Secure Multi-Party Computation (MPC): Enabling collaborative policy evaluation across multiple parties without revealing individual inputs, useful for cross-institutional access control decisions.

9.2.5 Automated Policy Mining and Optimization

Policy Discovery: Using unsupervised learning to automatically discover implicit access control policies from historical approval/denial patterns:

Clustering: Identifying natural groupings of access patterns

Association Rule Mining: Discovering correlations between attributes and access outcomes

Anomaly Detection: Flagging requests that deviate from learned normal patterns

Policy Simplification: Applying ML to identify redundant or rarely-triggered policies, recommending consolidation to reduce policy complexity and evaluation overhead.

Dynamic Policy Adjustment: Enabling AI to suggest policy modifications based on access pattern evolution, operational efficiency analysis, and security incident patterns.

Conflict Detection: Automatically identifying conflicting or contradictory policies that could lead to inconsistent access decisions.

9.2.6 Real-World Pilot Deployments

The ultimate validation requires real-world deployment:

Hospital Pilot Program: Partnering with healthcare institutions for controlled pilot deployments:

Phase 1: Shadow mode (AI predictions logged but not acted upon) for 3-6 months

Phase 2: Partial deployment (selected user groups) with intensive monitoring

Phase 3: Full production deployment with continuous monitoring and optimization

Multi-Site Health Information Exchange: Deploying across regional health information exchanges connecting multiple hospitals, clinics, and specialty providers to validate cross-institutional effectiveness.

Patient-Centric Applications: Extending to patient-facing applications enabling individuals to control access to their own health data through blockchain-based consent management.

Longitudinal Studies: Multi-year deployments enabling analysis of long-term effects, seasonal patterns, organizational evolution, and technology adoption dynamics.

Clinical Outcomes Research: Investigating whether improved data access facilitated by optimized systems correlates with better patient outcomes, reduced medical errors, or enhanced care coordination.

9.2.7 Economic and Policy Research

Comprehensive Cost-Benefit Analysis: Expanding economic analysis to incorporate:

- Implementation and operational costs with detailed breakdown
- Transaction cost savings with various gas price scenarios
- Indirect benefits (reduced breaches, improved compliance, audit efficiency)
- Patient outcome impacts from improved data access
- Opportunity costs of alternative approaches

Incentive Mechanism Design: Designing tokenomics and incentive structures for blockchain-based EHR networks:

- Rewarding data sharing while protecting privacy
- Incentivizing accurate access control policy specification
- Penalizing policy violations and unauthorized access attempts
- Token economics for AI model training data contribution

Regulatory Framework Development: Collaborating with policymakers to develop regulatory frameworks specifically addressing blockchain-based health information systems:

- Liability allocation for distributed systems
- Jurisdiction questions for cross-border data access
- Standards for blockchain EHR interoperability
- AI governance frameworks for healthcare applications

Insurance and Risk Models: Developing actuarial models for cyber-insurance products tailored to blockchain-based healthcare systems.

9.2.8 Human Factors and Usability Research

Clinician Experience Studies: Investigating how AI-optimized access control affects clinical workflows:

- User satisfaction surveys across different clinical roles
- Task completion time analysis in realistic scenarios
- Error rate comparison (access denials, workarounds)
- Adoption barrier identification and mitigation strategies

Trust and Acceptance: Understanding factors influencing clinician trust in AI-assisted systems:

- Explainability requirements for different stakeholder groups
- Intervention strategies when AI predictions are suboptimal
- Communication strategies for AI capabilities and limitations
- Impact of transparency on trust and adoption

Interface Design Research: Developing intuitive interfaces for:

- Policy specification and management
- AI model performance monitoring dashboards
- Access decision auditing and forensic investigation tools
- Patient consent management portals

Training and Change Management: Developing effective training programs and change management strategies to facilitate organizational adoption.

9.2.9 Extended Threat Modelling

Advanced Attack Scenarios: Investigating sophisticated attacks not covered in current threat model:

Model Poisoning: Attackers injecting malicious data into training sets

Byzantine Attacks: Malicious nodes in distributed AI training

Timing Attacks: Exploiting execution time variations to infer policy structures

Side-Channel Attacks: Information leakage through gas consumption patterns

Adversarial Machine Learning: Developing defences against adversarial examples designed to manipulate AI predictions while maintaining access correctness.

Quantum Threat Analysis: Comprehensive analysis of quantum computing threats to blockchain and cryptographic components.

9.2.10 Environmental Sustainability

Energy Efficiency Research: Quantifying environmental benefits:

- Carbon footprint reduction from optimized blockchain operations

- Comparative life-cycle analysis vs. centralized database systems
- Sustainability metrics for large-scale deployment
- Integration with renewable energy-powered blockchain nodes

Green Blockchain Protocols: Investigating integration with energy-efficient consensus mechanisms (Proof of Stake, Proof of Authority) that reduce environmental impact.

9.3 Broader Impact Considerations

9.3.1 Equity and Fairness

Algorithmic Fairness: Ensuring AI optimization doesn't inadvertently discriminate:

- Analysing whether optimization effectiveness varies systematically across demographic groups
- Detecting and mitigating bias in historical training data
- Implementing fairness constraints in model training Regular fairness audits in production deployments

Access Equity: Investigating whether cost reductions enable broader healthcare information system access in resource-constrained settings:

- Rural and underserved community hospital adoption
- Developing nation healthcare infrastructure
- Community health centres and free clinics

Digital Divide: Addressing potential barriers for organizations lacking technical expertise or resources to implement AI-optimized systems.

9.3.2 Workforce Transformation

Skill Requirements: AI-optimized blockchain systems require interdisciplinary expertise:

- Training programs combining blockchain, AI, and healthcare informatics
- Workforce development for emerging health IT roles
- Continuing education for existing health IT professionals
- Career pathways for diverse backgrounds

Job Displacement Concerns: Analysing potential impacts on traditional health IT roles and developing transition strategies.

9.3.3 Global Health Implications

Resource-Limited Settings: Investigating applicability in developing nations where healthcare infrastructure is emerging and cost constraints are severe.

Pandemic Response: Evaluating whether optimized blockchain EHR systems could improve international coordination during global health emergencies.

Health Equity: Examining whether technology can reduce health disparities or might inadvertently exacerbate them.

10. Conclusion

This research presents a comprehensive AI-assisted optimization framework for hybrid RBAC-ABAC access control in blockchain-based Electronic Health Record systems. By strategically integrating machine learning to optimize policy evaluation sequences while rigorously preserving deterministic security guarantees, the framework addresses two critical barriers to blockchain EHR adoption: excessive transaction costs and access latency that impede practical deployment.

10.1 Summary of Key Contributions

The research makes five primary contributions to the fields of blockchain security, healthcare informatics, and intelligent system optimization:

1. Novel AI-Assisted Optimization Architecture: A carefully designed off-chain machine learning layer that predicts optimal policy execution pathways for hybrid RBAC-ABAC systems without compromising security. The architectural separation between optimization (AI-driven) and security decisions (policy-driven) represents a fundamental design pattern enabling the integration of probabilistic machine learning with deterministic security requirements.

2. Rigorous Formal Security Analysis: Mathematical proofs demonstrating that policy reordering preserves access control correctness through the order-independence property of logical conjunction operations. The framework maintains collision resistance below 2^-128 and preserves all cryptographic security properties inherited from the baseline system. These formal guarantees provide the foundation for trustworthy deployment in healthcare environments where security cannot be compromised.

3. Comprehensive Experimental Validation: Systematic evaluation on an Ethereum testbed with realistic healthcare workload simulation demonstrates measurable, statistically significant improvements:

- 16.9% reduction in gas consumption compared to non-optimized hybrid baseline
- 21.0% reduction in access decision latency enhancing user experience
- Sub-linear scalability characteristics as system scale increases
- Consistent performance across varying loads, roles, and configurations
- Perfect access correctness preserved across 10,000 test cases

4. Economic Feasibility Demonstration: Detailed cost-benefit analysis showing $47,000 annual

savings for medium-sized hospitals processing 100,000 access requests, with less than 6-month payback period and substantial 5-year ROI exceeding $200,000. These results demonstrate that intelligent optimization transforms blockchain from theoretically attractive but economically prohibitive to practically deployable.

5. Reproducible Research Framework: Comprehensive implementation specifications including hardware requirements, software dependencies, dataset characteristics, algorithmic details, and evaluation methodologies sufficient for independent reproduction, validation, and extension by other researchers and practitioners.

10.2 Theoretical Implications

Beyond immediate practical contributions, this work establishes several theoretical principles with broader applicability:

Intelligence Without Compromise: Security-critical systems can benefit from machine learning optimization without sacrificing deterministic guarantees. The key architectural principle is separation of concerns—AI optimizes execution efficiency while deterministic logic ensures correctness. This pattern applies broadly to systems where security requirements must remain inviolable while performance optimization is desirable.

Pattern Recognition in Structured Environments: Real-world operational systems, particularly in highly regulated domains like healthcare, exhibit strong learnable patterns due to structured workflows, role-based organizational hierarchies, and regulatory compliance requirements. These patterns represent exploitable opportunities for intelligent optimization.

Blockchain Practicality Through Intelligence: Blockchain technology's transaction costs represent significant adoption barriers for high-volume applications. Intelligent optimization can bridge the gap between blockchain's security and trust benefits and practical economic requirements, expanding blockchain's applicability beyond cryptocurrency to enterprise information systems.

Interpretable AI for Trust: In healthcare and other high-stakes domains, interpretable machine learning models (Random Forest, decision trees) provide essential transparency enabling stakeholder trust, regulatory compliance, and operational confidence. The trade-off between model accuracy and interpretability often Favors interpretability in production healthcare systems.

10.3 Practical Impact for Healthcare Organizations

For healthcare organizations evaluating blockchain-based EHR solutions, this framework provides concrete benefits:

Economic Viability: Measurable reduction in blockchain transaction costs makes adoption economically feasible, addressing a primary barrier. Organizations can justify blockchain investments based on clear ROI projections rather than speculative benefits.

Performance Acceptability: Reduced latency improves user experience and clinical workflow integration. The 21% latency reduction moves access times into ranges clinicians find acceptable, reducing workflow friction and improving adoption prospects.

Scalability Confidence: Sub-linear growth characteristics provide confidence for enterprise-scale deployment. Organizations can project costs and performance at scale based on experimental validation rather than extrapolation.

Security Assurance: Formal proofs and comprehensive experimental validation confirm security preservation, addressing regulatory and compliance concerns. Organizations can adopt optimization with confidence that security is not compromised for efficiency.

Regulatory Compliance: Privacy-by-design architecture with no PHI exposure in AI layer, comprehensive audit capabilities, and model interpretability support HIPAA, GDPR, and other regulatory compliance requirements.

Practical Deployment Path: Modular architecture enables incremental adoption through shadow mode deployment, partial rollout, and full production transition. Organizations can de-risk implementation through phased approaches.

Interoperability Foundation: The framework's design facilitates future extension to cross-institutional health information exchange, positioning adopters for long-term interoperability objectives.

10.4 Research Trajectory and Future Vision

This work opens numerous avenues for continued investigation and development:

Immediate Next Steps: Real-world pilot deployments in partnership with healthcare institutions, cross-platform evaluation (Hyperledger Fabric, Corda), and alternative ML architecture comparison studies (XGBoost, neural networks, reinforcement learning).

Medium-Term Goals: Integration of privacy-enhancing technologies (zero-knowledge proofs, homomorphic encryption), federated learning for multi-institutional deployment, and automated policy mining capabilities.

Long-Term Vision: Post-quantum cryptographic integration for quantum-resistant security, cross-chain interoperability frameworks enabling global health information exchange, and comprehensive health information ecosystems integrating blockchain, AI, IoT medical devices, and patient engagement platforms.

The fundamental insight that intelligent optimization can enhance blockchain systems without compromising security extends well beyond healthcare to financial services (optimizing smart contract execution in DeFi protocols), supply chain management (efficient consensus in multi-party logistics networks), digital identity systems (optimized attribute verification), and governance systems (efficient voting and consensus mechanisms).

10.5 Vision for Next-Generation Healthcare Systems

Looking forward, we envision integrated healthcare information ecosystems that seamlessly combine:

Blockchain Foundation: Providing decentralized trust, immutability, transparency, and cryptographic security without centralized control points or single-point-of-failure vulnerabilities.

Artificial Intelligence: Optimizing operations (as demonstrated in this work), personalizing treatment recommendations, discovering medical insights from large-scale data analysis, and predicting adverse events before they occur.

Privacy Technologies: Protecting sensitive information through zero-knowledge proofs, homomorphic encryption, differential privacy, and secure multi-party computation while enabling beneficial data use for research and care coordination.

Patient Empowerment: Giving individuals meaningful control over their own health data through blockchain-based consent management, transparent access audit trails, and portable health records that move with patients across providers.

Seamless Interoperability: Enabling frictionless information exchange across organizational boundaries, healthcare systems, and national borders while preserving privacy and security.

Security Assurance: Maintaining rigorous protection against evolving threats through post-quantum cryptography, advanced threat detection, and formal verification of security properties.

Sustainability: Minimizing environmental impact through energy-efficient consensus mechanisms and intelligent optimization reducing computational waste.

This research represents a foundational step toward that vision, demonstrating that intelligent optimization and rigorous security are complementary rather than conflicting objectives.

10.6 Call to Action

We encourage the research community, healthcare organizations, technology vendors, and policymakers to:

For Researchers: Build upon this framework through extended validation studies, alternative ML architectures, privacy-enhancing integrations, and cross-chain interoperability research. The detailed implementation specifications provided enable reproducible research and comparative evaluation.

For Healthcare Organizations: Consider pilot deployments in controlled environments, contribute de-identified access logs for research (with appropriate IRB approval), and provide feedback on practical deployment challenges and requirements.

For Technology Vendors: Integrate AI optimization capabilities into blockchain EHR products, develop standardized APIs for optimization layer integration, and invest in user-friendly interfaces for clinical stakeholders.

For Policymakers: Support research funding for blockchain healthcare applications, develop clear regulatory frameworks addressing distributed health information systems, and facilitate multi-stakeholder collaborations advancing healthcare technology while protecting patient interests.

10.7 Closing Remarks

The convergence of blockchain technology and artificial intelligence presents both transformative opportunities and significant challenges for healthcare information systems. This research demonstrates that thoughtful, principled integration—respecting the strengths and limitations of each technology while maintaining unwavering commitment to security and privacy—can yield systems that are simultaneously more secure, more efficient, and more practical than alternatives relying on either technology in isolation.

As healthcare continues its inevitable digital transformation, frameworks that balance innovation with proven security guarantees will be essential. By preserving deterministic access control correctness while leveraging learned optimization patterns, this work provides a validated template for responsible AI integration in security-critical systems.

The true measure of success will ultimately be measured not in academic citations or performance benchmarks, but in real-world adoption, patient benefit, and healthcare system improvement. We hope this research contributes meaningfully to making blockchain-based EHR systems practical, affordable, and trustworthy enough for widespread deployment, ultimately improving healthcare delivery through better data management, enhanced security, preserved patient privacy, and facilitated care coordination.

The journey toward intelligent, secure, patient-centered healthcare information systems is just beginning. This work represents one step forward on that path, and we look forward to continued collaboration with the research community, healthcare practitioners, and

technology innovators in advancing this important mission.

References

1. Nakamoto, S. (2008). "Bitcoin: A Peer-to-Peer Electronic Cash System." White Paper.
2. Azaria, A., Ekblaw, A., Vieira, T., & Lippman, A. (2016). "MedRec: Using Blockchain for Medical Data Access and Permission Management." IEEE International Conference on Open and Big Data (OBD), 25-30.
3. Sandhu, R. S., Coyne, E. J., Feinstein, H. L., & Youman, C. E. (1996). "Role-Based Access Control Models." IEEE Computer, 29(2), 38-47.
4. Hu, V. C., Ferraiolo, D., Kuhn, R., Schnitzer, A., Sandlin, K., Miller, R., & Scarfone, K. (2014). "Guide to Attribute Based Access Control (ABAC) Definition and Considerations." NIST Special Publication 800-162.
5. Wood, G. (2014). "Ethereum: A Secure Decentralised Generalised Transaction Ledger." Ethereum Project Yellow Paper.
6. Breiman, L. (2001). "Random Forests." Machine Learning, 45(1), 5-32.
7. Yue, X., Wang, H., Jin, D., Li, M., & Jiang, W. (2016). "Healthcare Data Gateways: Found Healthcare Intelligence on Blockchain with Novel Privacy Risk Control." Journal of Medical Systems, 40(10), 218.
8. Zhang, P., Schmidt, D. C., White, J., & Lenz, G. (2018). "Blockchain Technology Use Cases in Healthcare." Advances in Computers, 111, 1-41.
9. Xia, Q., Sifah, E. B., Asamoah, K. O., Gao, J., Du, X., & Guizani, M. (2017). "MeDShare: Trust-Less Medical Data Sharing Among Cloud Service Providers via Blockchain." IEEE Access, 5, 14757-14767.
10. Ekblaw, A., Azaria, A., Halamka, J. D., & Lippman, A. (2016). "A Case Study for Blockchain in Healthcare: 'MedRec' prototype for electronic health records and medical research data." Proceedings of IEEE Open & Big Data Conference.
11. Peterson, K., Deeduvanu, R., Kanjamala, P., & Boles, K. (2016). "A Blockchain-Based Approach to Health Information Exchange Networks." NIST Workshop on Blockchain and Healthcare.
12. Griggs, K. N., Ossipova, O., Kohlios, C. P., Baccarini, A. N., Howson, E. A., & Hayajneh, T. (2018). "Healthcare Blockchain System Using Smart Contracts for Secure Automated Remote Patient Monitoring." Journal of Medical Systems, 42(7), 130.
13. Liang, X., Zhao, J., Shetty, S., & Li, D. (2017). "Towards Data Assurance and Resilience in IoT Using Blockchain." IEEE Military Communications Conference (MILCOM), 261-266.
14. Feng, Q., He, D., Zeadally, S., Khan, M. K., & Kumar, N. (2019). "A Survey on Privacy Protection in Blockchain System." Journal of Network and Computer Applications, 126, 45-58.
15. Boneh, D., Sahai, A., & Waters, B. (2011). "Functional Encryption: Definitions and Challenges." Theory of Cryptography Conference, 253-273.
16. Ben-Sasson, E., Chiesa, A., Tromer, E., & Virza, M. (2014). "Succinct Non-Interactive Zero Knowledge for a von Neumann Architecture." USENIX Security Symposium, 781-796.
17. Gentry, C. (2009). "Fully Homomorphic Encryption Using Ideal Lattices." Symposium on Theory of Computing (STOC), 169-178.
18. Dwork, C., & Roth, A. (2014). "The Algorithmic Foundations of Differential Privacy." Foundations and Trends in Theoretical Computer Science, 9(3-4), 211-407.
19. McMahan, B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A. (2017). "Communication-Efficient Learning of Deep Networks from Decentralized Data." Artificial Intelligence and Statistics (AISTATS), 1273-1282.
20. Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "'Why Should I Trust You?' Explaining the Predictions of Any Classifier." ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 1135-1144.
21. Lundberg, S. M., & Lee, S. I. (2017). "A Unified Approach to Interpreting Model Predictions." Advances in Neural Information Processing Systems (NIPS), 4765-4774.
22. Chen, T., & Guestrin, C. (2016). "XGBoost: A Scalable Tree Boosting System." ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785-794.
23. Sutton, R. S., & Barto, A. G. (2018). "Reinforcement Learning: An Introduction." MIT Press, Second Edition.
24. Bernstein, D. J., & Lange, T. (2017). "Post-Quantum Cryptography." Nature, 549(7671), 188-194.
25. National Institute of Standards and Technology (NIST). (2022). "Post-Quantum Cryptography Standardization." NIST FIPS Publication.