International Journal of Mathematical, Engineering and Management Sciences
Vol. 10, No. 6, 2061-2082, 2025
https://doi.org/10.33889/IJMEMS.2025.10.6.096

Ram Arti Publishers

Resource Efficiency-Driven Consensus (REDC): A Machine Learning-Based Blockchain Framework for Healthcare IoT Systems

Saurabh Jain
School of Computer Science,
University of Petroleum and Energy Studies, Dehradun, Uttarakhand, India.
Corresponding author: saurabh.jain@ddn.upes.ac.in

Adarsh Kumar
Department of Mining,
Industrial, and ICT Engineering (EMIT),
Universitat Politècnica de Catalunya (UPC) BarcelonaTech, Barcelona, Spain.
E-mail: kumar.adarsh@upc.edu
&
School of Computer Science,
University of Petroleum and Energy Studies (UPES), Dehradun, Uttarakhand, India.
E-mail: adarsh.kumar@ddn.upes.ac.in

(Received on May 14, 2025; Revised on June 23, 2025 & August 4, 2025; Accepted on August 26, 2025)

Abstract
The increasing adoption of Internet of Things (IoT) devices in smart healthcare systems has revolutionized real-time data collection and processing, substantially improving healthcare delivery and operational efficiency. However, the sensitivity of medical data and the resource limitations of IoT devices demand blockchain solutions that are secure, lightweight, and scalable. This paper presents two core contributions: (1) Resource Efficiency-Driven Consensus (REDC), a machine learning–enhanced consensus protocol tailored for healthcare IoT networks, and (2) Dynamic Lightweight Hashing (DLH), a cryptographic algorithm designed for energy-constrained environments. REDC achieves up to 70% higher throughput, 43% Energy Efficiency (EE), and 25% lower latency compared to Proof of Elapsed Work and Luck (PoEWAL) in networks up to 100 nodes. DLH further enhances performance by reducing hash attempts and energy use while maintaining strong collision resistance across 100,000 trials. Together, REDC and DLH form a scalable and secure blockchain framework tailored for healthcare IoT.

Keywords- Internet of things, Healthcare, Blockchain, Resource efficiency-driven consensus (REDC), Machine learning, Resource efficiency index, Scalability.

List of Abbreviations

Term	Abbreviation
Artificial Intelligence	AI
Attribute-Based Encryption	ABE
Avalanche Effect	AE
Byzantine Fault Tolerance	BFT
Collision Resistance	CR
Delegated Proof of Accessibility Consensus	DPoAC
Delegated Proof of Stake	DPoS
Dynamic Lightweight Hashing	DLH
Electronic Health Record	EHR
Energy Consumption	J
Energy Efficiency	EE
Entropy	H
Federated Learning	FL
Hash Computation Latency	HCL
Hashing Energy Consumption	HEC

Improved Practical Byzantine Fault Tolerance	IPBFT
Interplanetary File System	IPFS
Internet of Things	IoT
Internet of Vehicles	IoV
Latency	L
Lightweight Dada Consensus	LDC
Lightweight Plenum Consensus Algorithm	BLPCA
Practical Byzantine Fault Tolerance	PBFT
Proof of Block and Trade	PoBT
Proof of Elapsed Work and Luck	PoEWAL
Proof of Evolutionary Model	PoEM
Proof of Karma	PoK
Proof of Stake	PoS
Proof of Work	PoW
Quality of Service	QoS
Resource Efficiency Index	REI
Resource Efficiency-Driven Consensus	REDC
Rivest–Shamir–Adleman	RSA
Secure Cloud-Based Blockchain	SCB2
Secure Hash Algorithm 256	SHA-256
Shamir Secret Sharing	SSS
Transactions Per Second	TPS

1. Introduction

Modern healthcare systems increasingly leverage advanced technologies, such as Artificial Intelligence (AI), the Internet of Things (IoT), and Blockchain, to enhance patient care, streamline operations, and enable real-time data analytics. In IoT-enabled healthcare systems, patients' conditions are continuously monitored with the help of wearable devices, sensors, and telemedicine platforms, and customized treatment is offered (Bathula et al., 2024). This enables proactive chronic disease management and early detection of health anomalies. With the increasing number of networked devices, the related security issues have become a significant concern, including data breaches and unauthorized access (Bala et al., 2024). Blockchain technology represents a potential solution to secure, decentralized, and transparent data management in healthcare scenarios. Based on its tamper-resistant ledger and the distributed nature of the technology, it is good for people to use health-sensitive information (e.g. electronic health records or sensor data) where multiple actors need to access/share data among each other (Attaran, 2022). Beyond that first advantage, protocol for traditional blockchain such as Proof of Work (POW) and Proof of Stake (POS) is not capable to work seamlessly in healthcare IoT environment because they consume more energy than typical health IoT devices capabilities and latency isn't super quick and adaptable to our real-time use. These restrictions render them unfit for embedded devices and delay-sensitive applications such as patient monitoring and diagnostics (Khor et al., 2021; Khan et al., 2022). The framework of high transaction volume and rapidly changing network conditions seen in healthcare IoT systems necessitates the use of a dedicated blockchain format to resolve these issues (Yaqoob et al., 2022).

This study addresses a real-time, resource-aware blockchain consensus protocol and lightweight cryptographic hashing method optimized for constrained healthcare IoT systems, where high Energy Efficiency (EE), security, and low-latency validation are critical. The key contributions include: (i) a machine learning–based consensus algorithm named Resource Efficiency-Driven Consensus (REDC) that dynamically tunes mining parameters via the Resource Efficiency Index (REI); (ii) a composite performance evaluation method that weights latency, energy, and mining difficulty to enhance node selection; (iii) integration of an ε-greedy tuner to optimize consensus timing based on historical feedback;

(iv) introduction of Dynamic Lightweight Hashing (DLH), a lightweight cryptographic solution that balances security with reduced computational cost; and (v) real-time simulation results showing that REDC reduces block time by up to 25%, energy consumption by 43%, and latency by 20% when compared to existing models such as PoEWAL (Raghav et al., 2020), across varying network sizes (10–100 nodes).

The rest of this paper is organized as follows: Section 2 reviews related works and presents the current issues of blockchain-based healthcare systems. Section 3 describes the system architecture and detailed design of the REDC framework, including its lightweight cryptographic component. Section 4 discusses performance evaluation under real-time conditions through comparative analysis with traditional consensus mechanisms and hashing algorithms. Section 5 concludes the paper and outlines directions for future research.

2. Related Work

Newer blockchain consensus methods are working to solve this IoT challenge, specifically in the smart healthcare systems areas. For instance, Proof-of-Work (PoW) have been widely used in traditional context to provided data integrity and security for decentralized systems with large-scale size while it still unsuitable for IoT equipment due to high-energy consumption and computational requirements (Maadallah et al., 2025). Because of these limitations, Raghav et al. (2020) proposed PoEWAL, it is a lightweight and probabilistic consensus mechanism designed for IoT applications. The reduced energy and latency consumption, in turn, provides nodes with equal opportunity to participate in the consensus process by dispensing with the high computational burden of PoW. The authors also have evaluated the basics performance factors, namely energy consumption, consensus time and network delay in which this PoEWAL protocol was proved to be an effective replacement towards a resource-constraint environment.

In the meantime, behaviour-based consensus models have been introduced to enhance fairness and decentralization. Biswas et al. (2023) came up with a concept named Proof of Karma (PoK) consensus mechanism to rate nodes on the basis of past reputation and behavior. The block producers are either rewarded for their service (if serving in an honourable manner) or are punished (or have their reward zeroed out) if behaving malignly, by a reputation-based algorithm that rewards and/or punishes old and new nodes through honest participation versus malignant behaviour. That takes the burden away from having to make these leader elections and makes it more scalable. This work also considered the two key performance metrics in terms of calculating block verification time and communication overhead thereby enabling low latency, high throughput properties.

To achieve greater adaptability and efficiency, Zhao et al. (2023) created a progressive machine learning based evolutionary consensus protocol, Proof of Evolutionary Modeling (PoEM), which leverages machine learning in consensus determinations. This inventive methodology, PoEM, educates itself from real world operations and dynamically modifies consensus parameters, considerably boosting performance while reducing computational costs. The iterative process has PoEM models constantly self-training and parameter tuning based on metrics such as energy efficiency, block times, and system latency. Experimental evaluations revealed that PoEM can converge more rapidly and use less energy than static consensus mechanisms. Elsewhere, Biswas et al. (2020) introduced another lightweight consensus algorithm, Proof of Block and Trade (PoBT), tailored for scalable IoT business blockchains. By streamlining transaction validation steps, PoBT preserves computation time and memory overhead. Experimental results from PoBT emphasized improvements in throughput and lower delay, indicating that PoBT is tailored for environments with high transaction volumes. Many of the performance metrics measured in these studies, like energy used per block, hashing attempts, latency, and throughput, are pivotal for comparing the various consensus algorithms that may be suitable for IoT scenarios. Kanagasankari and Vallinayagi (2024) proposed

Improved Practical Byzantine Fault Tolerance (IPBFT) consensus, integrated with Rivest–Shamir–Adleman (RSA) cryptography specifically for healthcare Electronic Health Records (EHRs) on a Hyperledger Fabric blockchain. This work covers blockchain-based healthcare frameworks while stating the drawbacks, like high communication overhead and the need for efficient consensus algorithms in Practical Byzantine Fault Tolerance (PBFT). Numerous works highlight the importance of cryptographic strategies like RSA and Attribute-based Encryption (ABE) in preventing access and privacy threats within the healthcare domain. The authors discuss previously developed models for blockchain scalability, privacy policies, and off-chain storage approaches, concluding that IPBFT leveraging RSA encryption is more efficient, secure, and fault-tolerant than traditional PBFT systems.

Li et al. (2021) explored a lightweight consensus mechanism and a storage optimization scheme using RS erasure codes, which alleviates the storage burden on resource-constrained devices. This idea has optimized the consensus process while ensuring the blockchain ledger is stored in a space-saving way without affecting data recoverability. Similarly, Zhang et al. (2020) proposed Lightweight Data Consensus (LDC), primarily focused on minimizing communication overhead and energy consumption for industrial IoT applications, efficiently achieving low-latency consensus by reducing hash computations. Moreover, Bamakan et al. (2020) gave a highly detailed performance analysis of the significantly studied and commonly used blockchain consensus algorithms based on diverse metrics, including throughput, mining profitability, decentralization, and attack susceptibility. Their comprehensive analysis provided valuable insights into the strengths and weaknesses of different approaches and pointed towards the necessity of adaptive, lightweight protocols that are application domain-specific, such as for healthcare.

Narsimhulu et al. (2024) presented an intelligent Federated Learning route optimization protocol in green and sustainable IoT-connected Internet of Vehicles (IoV) environments. They have proposed a solution for real-time traffic rerouting, vehicle demand prediction, and communication-induced delays. They have matched Federated Learning (FL) with cluster-based vehicle communication and location estimation models. Similarly, Gupta et al. (2024) presented a Secure Cloud-Based Blockchain (SCB2) model for securely storing high-volume sensor data through blockchain, IoT, and cloud computing, and saving storage through indexed references in the blockchain blocks. Comparing their permissioned blockchain system (Fabric over Ethereum) to the Baseline system, they showed better security and efficiency under realistic operational stress. Both projects try to address domain-specific issues through an intelligent, scalable, and secure architecture.

Haque et al. (2024) introduced a scalable blockchain-enabled architecture for efficient IoT data processing with Delegated Proof of Stake (DPoS) as a lightened consensus approach. Their protocol deals with the performance and scalability concerns in large-scale IoT settings by relying on a small set of trusted delegates to validate transactions, thus ensuring the consumption of minimum infrastructure resources and a low latency time. The platform combines the Interplanetary File System (IPFS) for decentralized storage. It employs a Docker-based simulation to simulate throughput, latency, and resource consumption on networks of 500 to 20000 devices. Test results demonstrate that DPoS is superior to PoS, and when processing queries, in good performance conditions, the latency of DPoS is less than 0.976 ms, and the throughput is high enough, which allows its application in real-time and healthcare. In line with this practical work, Sahraoui and Bachir (2025) thoroughly reviewed lightweight consensus mechanisms for the Internet of Blockchain Things (IoBT). They classify consensus protocols based on the operational, security, and AI features, and stress upon constraint-aware and Quality of Service (QoS) -driven designs for IoT scenarios. They further analysed AI-based block validation and shared opinions on consensus models that could be decentralized, efficient, and secure under the resource limitations of IoT networks.

Mehmood et al. (2025) introduced the Lightweight Plenum consensus algorithm (BLPCA) protocol developed on top of the Hyperledger Indy blockchain, tailored for secure, low-cost socio-economic applications, namely taxation and public service funding. BLPCA utilizes Byzantine Fault Tolerance (BFT) and optimization to provide high reliability of transactions and to save resource costs significantly. Kaur and Gupta (2025) also proposed a lightweight protocol, Delegated Proof of Accessibility Consensus (DPoAC), for IoT-based blockchain networks to integrate Shamir Secret Sharing (SSS), reputation-based PoS, and IPFS to achieve efficient and fair consensus. Both schemes have lower time cost, less energy consumption, and high security, which confirms that lightweight and extensible consensus mechanisms are essential for real-time and resource-limited blockchain systems.

Several existing consensus mechanisms offer improvements for general IoT environments but fall short when applied to healthcare-specific scenarios. PoEWAL, though lightweight, lacks dynamic adaptation and does not incorporate real-time energy metrics, which are essential for managing energy consumption in battery-powered medical devices. PoK's (Biswas et al., 2023) reputation-based mechanism assumes consistent node behaviour, which may not be practical in healthcare environments where frequent mobility of patients and medical staff leads to irregular node participation. PoBT improves transaction validation speed but fails to balance the tradeoff between energy consumption and latency, a crucial requirement for latency-sensitive medical sensors. Although PoEM introduces machine learning for adaptive tuning, it does not optimize cryptographic operations, and its model complexity may introduce computational overhead unsuitable for low-power healthcare IoT nodes.

This work extend these efforts to promote resource-efficient evaluation by introducing the REDC framework, unlike PoEM, PoBT, and REDC, which dynamically adjusts consensus parameters using a resource-aware metric REI, tailored for constrained healthcare environments. DLH also addresses the cryptographic overhead that these existing models overlook, offering secure but low-latency hashing optimized for medical IoT nodes. REDC internally combines a composite performance evaluation metric - REI- with a machine-learning-based parameter tuner to dynamically tune the miner parameters in real-time. Moreover, REDC introduces a lightweight cryptographic element by considering DLH in addition to Secure Hash Algorithm 256 (SHA256) (Salih and Kashmar, 2024), providing less computational burden while maintaining efficient security. REDC counteracts the critical determinants of performance, including energy use, block time, latency, and the energy required per hash attempt, reducing one of the significant concerns of IoT-enabled healthcare systems on scalability, EE, and security. This holistic consideration fills the research gaps highlighted by previous studies and lays the foundation for real-time and practical consensus under dynamic and resource-constrained scenarios.

Despite recent advancements, blockchain consensus mechanisms such as PoW and PoS remain unsuitable for resource-constrained healthcare IoT environments due to their high computational and energy demands (Platt et al., 2021; Abbas et al., 2024). Existing lightweight protocols, e.g., PoEWAL, PoK and PoBT cannot jointly optimize energy consumption, latency (delay), and processing overhead simultaneously which is essential in real-time medical systems. One problem with existing solutions is that they cannot present a unified performance metric and often disregard the real-time network dynamics. In addition, most of them transmit raw patient data during consensus or training, which is a big risk for the privacy (Andrew et al., 2023), and do not provide tools to automate healthcare-specific workflows such as insurance claims and consent management.

To address these challenges, this works presents REDC a machine learning driven Consensus mechanism using REI for dynamic parameter tuning, and a secure light weight cryptographic algorithm called as DLH for energy efficient blockchain operations in healthcare IoT sector. The REDC framework brings a novel

consensus challenging mechanism that has been devised for resource-constrained healthcare IoT ecosystems. REDC uses a dynamic REI rather than the traditional models, allowing it to select participating nodes based on the immediate energy cost, latency, and mining complexity. And it also introduces a lightweight machine learning based ε-greedy tuner to flexibility adjust the mining time slot and explore effectiveness gains without significant computational overheads. In addition, REDC is designed with a two-level hashing mechanism, which uses the standard SHA-256 and proposed DLH algorithm, for improved security efficiency. The modular design consolidates consensus, hashing, and tuning modules in a scale-out solution from the framework to satisfy real-time, low-power, and high-security demands for healthcare IoT applications.

3. Proposed Work

This section describes a proposed blockchain for healthcare IoT systems. It combines the REDC and DLH modules to address key challenges such as energy consumption, latency, and scalability. Each component's design and operational aspects are described in the subsequent subsections. The innovation of this framework lies in its dual optimization: consensus adaptation through machine learning and cryptographic efficiency through dynamic hashing, both specifically designed for constrained healthcare systems.

3.1 REDC

This paper proposes a new method, REDC, which enables dynamic and energy-efficient consensus specifically for IoT-enabled healthcare systems. In contrast to established consensus protocols that depend on static parameters, use of computationally intensive tasks as PoW, which, by design, consumes massive energy resources, or lightweight algorithms such as PoEWAL, which still implement a static mining parameter, REDC is capable of providing dynamic operation based and time-sensitive behaviour under varying network conditions. This adaptability is critical in a healthcare IoT environment where devices are usually resource-constrained, and fast data processing is essential.

3.1.1 REDC Framework

The REDC model introduces a REI to dynamically evaluate node performance according to energy, latency, and difficulty, which supports adaptive block selection and mining. The proposed model optimizes consensus in healthcare IoT networks by incorporating reinforcement learning (Mignon and da Rocha, 2017) and dual hashing. The following points describe the internal components and operations of the REDC framework.

3.1.1.1 Dynamic Resource Utilization Metric

The REI metric is used to evaluate the efficiency of each node in terms of energy consumption (E), latency (L), and characteristics of the current mining difficulty (D) because these three parameters collectively reflect the most critical constraints in healthcare IoT networks. It aims to reward nodes that perform with lower energy and latency, combined with the computational challenge of the network's difficulty. This composite metric allows the system to select well-performing and energy-efficient nodes to create blocks. During simulation, the domains for E_i, L_i, and D were empirically derived, and the weights α, β, γ were varied between 0.1 and 0.7, with default values set to $\alpha = 0.4$, $\beta = 0.4$, and $\gamma = 0.2$, reflecting balanced priority across energy and latency constraints. Equation (1) expresses the *REI* formulation:

$$REI_i = \frac{\alpha}{E_i} + \frac{\beta}{L_i} + \frac{\gamma}{D} \quad (1)$$

where,

- E_i is the estimated energy consumption for node i.
- L_i is the estimated latency for node i.

- D is the current mining difficulty (unitless, normalized scalar).
- α, β, γ are tuneable weighting factors for energy, latency, and difficulty, respectively.

3.1.1.2 Node Selection

The only difference here is that, after calculating REI for each node, the algorithm chooses a node for creating the next block based on a combination of its performance, which means REI, and its historical performance (reward). This means that you select consistently efficient nodes, yet it is essential to let not-so-efficient nodes get to pick. Equation (2) captures the selection mechanism:

$$i^* = arg_{i \in N} \max\{REI_i + R_i\} \quad (2)$$

where,

- R_i is the cumulative reward of node *i.*
- i^* is the selected node for block generation.
- REI_i is the Efficiency score of the node i.

3.1.1.3 Average Network Performance and Dynamic Difficulty Adjustment

The mining difficulty is adjusted dynamically by the total performance of the network. The algorithm's scaling of difficulty functions by finding the average REI for all nodes and comparing it against a target performance. When the average performance exceeds the target, the difficulty is increased to harden profitability to ensure security. Still, when below the target, the difficulty is decreased to help miners accumulate coins. The average network performance is calculated using Equation (3):

$$U_{avg} = \frac{1}{|N|} \sum_{i \in N} REI_i \quad (3)$$

where,

- U_{avg} is the average REI across all nodes.
- $|N|$ is the total number of participating nodes.
- REI is an average Resource Efficiency Index across all participating nodes, guiding the dynamic adjustment of mining difficulty based on overall network performance.

The mining difficulty is then updated using Equation (4):

$$D \leftarrow D \times \left(1 + \delta\left(U_{avg} - U_{target}\right)\right) \quad (4)$$

where,

- D is the current mining difficulty parameter.
- δ is the tuning parameter for difficulty adjustment.
- U_{target} is the target performance level.

3.1.1.4 Reward Calculation and Adaptive Tuning

Once a block is produced, the algorithm calculates a reward based on the energy consumed and the latency incurred. A lower combined cost results in a higher (less negative) reward. An ε-greedy tuner then uses this reward to adjust the mining time slot dynamically, allowing the system to adapt to changing network conditions. The reward (r) is computed as shown in Equation (5):

$$r = -(Energy + Latency) \quad (5)$$

After computing *r*, the mining time slot *T* is updated based on the action selected by the tuner in Equation (6):

$$T \leftarrow T \times m \tag{6}$$

where,

- m is the multiplier corresponding to the action selected by the tuner (e.g., increase 1.2, decrease 0.80, maintain 1.0).

3.1.1.5 Dual Hashing for Lightweight Cryptography

To ensure strong security with minimal computational overhead, REDC employs a dual-hashing approach. Each node computes hash values using conventional SHA256 and a lightweight alternative, DLH. The effective performance is then measured by selecting the hash that achieves the best result (highest number of leading zeros), ensuring the system benefits from the most efficient cryptographic operation available.The dual-hashing process is formally described in Equation (7):

$$\begin{cases} h_{sha} = SHA256(prev_{hash} \,||\, n) \\ h_{dlh} = DLH256(prev_{hash} \,||\, n) \\ z = \max\left\{count_{leading_{zeros(h_{sha})}}, count_{leading_{zeros(h_{dlh})}}\right\} \end{cases} \tag{7}$$

where,

- N is the randomly generated nonce
- Z is the number of leading zeros in the best hash output

REDC constantly observes the performance of nodes in terms of energy, latency, and mining difficulty, and calculates a dynamic REI. A reinforcement learning, the ε-greedy based approach, is utilized to adaptively optimize the mining time slot through the reward signal obtained from the historical block generation results. This way, the system can optimize the timing without previous labelled information. Concurrently, the DLH module also contributes to the cryptographic efficiency by dynamically producing a lightweight hash value and comparing it with SHA-256 to choose the most efficient result. Overall, REDC guides consensus building with the help of DLH, which guarantees both a secure and an energy-aware hashing in healthcare-related IoT scenarios.

Algorithm 1 (REDC) shows an adaptive blockchain mining mechanism that adaptively determines the mining difficulty and mining time through the multi-factor resource evaluation algorithm. It combines the energy consumption, latency, and mining efficiency to be credited for block validation in a composite REI to choose optimal nodes to validate blocks. It is implemented and uses static hash (SHA256 & DLH), ε-greedy tuning (exploration-exploitation tradeoff), and real-time reward updates to facilitate fairness, efficiency, and sustainability in decentralized rounding.

Algorithm 1: REDC

Input:

- $\mathcal{T}$: Global transaction pool (set of all unconfirmed transactions)
- T_0: Initial mining time slot (starting value for mining duration per node)
- T: Current mining time slot (dynamically adjusted by the ε-greedy tuner)
- Node pool, N
- Initial mining difficulty, D_0
- Initial mining time slot, T_0
- For each node $i \epsilon N$: estimated energy consumption E_i and latency L_i
- Weighting factors α, β, γ
- Target utility level, U_{Target}
- Tuning parameter, δ
- Adaptive tuner exploration rate ε and action multipliers(m+,m-,1):

Output:

- Validated block B (appended to the blockchain)
- Updated performance metrics (e.g., average block time, energy, latency, hash attempts)

Steps:

I Initialization:

Set $T \leftarrow T_0, D \leftarrow D_0$; initialize $R_i = 0$ for all $i \in \mathbb{N}$; Configure the adaptive tuner

II While $\mathcal{T} \neq \emptyset$, do:

a. Resource Evaluation: Compute REI_i For each node, using **Equation 1**.//evaluate node efficiency based on energy, latency, and difficulty

b. Node Selection: Select the node i^* with highest $RE_i + R_i$ as per **Equation 2**.// Ensure fairness and prioritize consistent performers

c. Difficulty Adjustment: Compute average REI and update D using **Equations 3 and 4**.// Adjust mining difficulty based on network performance

d. Transaction Selection: Select a subset $\mathcal{T}^*$ of high-priority transactions.// Prioritize high-value transactions

e. Partial Mining & Block Formation: For duration T, each node generates nonces and computes hashes using *SHA256* and *DLH*. Record the best result (highest z) as per **Equation 7**. The best candidate forms a block B.// Dual hashing for best result

f. Broadcasting: Broadcast B to all nodes and update energy consumption.// Inform the network of validated block

g. Adaptive Tuning: Compute reward r using **Equation 5**; update T via the ε-greedy tuner.// ε-greedy tuner adjusts timing dynamically

h. Reward Update & Transaction Management:

Update $R_{i^*} = R_{i^*} + r$; remove $\mathcal{T}^*$ from $\mathcal{T}$ // Maintain blockchain state

III End While

IV Output:

Return the final blockchain and aggregated performance metrics (average block time, energy consumption, latency, and hash attempts).

3.2 DLH

Our proposed approach, DLH, introduces a highly adaptive, energy-efficient hash function for resource-constrained IoT and embedded devices. Unlike traditional ASCON-Hash256 (Khan et al., 2024) and SHA-256 — which employ fixed round constants and standard substitution layers—DLH dynamically tailors its internal operations to boost nonlinearity and diffusion while minimizing energy consumption. This adaptability is vital for environments where computational resources and power are at a premium.

3.2.1 DLH Framework

The DLH structure improves the efficiency of the hash functions using quadratic S-boxes, dynamic round constants, and improved diffusion layers. These've been developed to offer strong cryptographic security with low computational overhead, thus making DLH suitable for resource-limited IoT devices. The internals and primary operations of the DLH framework are described below.

3.2.1.1 Quadratic S-Boxes

DLH applies a quadratic transformation to each state word to further enhance nonlinearity. This operation replaces conventional linear substitutions with a squaring function, thereby increasing resistance against differential and linear cryptanalysis. For each state word S_i, the transformation is defined as shown in Equation (8):

$$S_i' = S_i \oplus \left(S_i^2 \bmod 2^{64}\right), for\ i = 0,1,\ldots..,4 \quad (8)$$

where,

- S_i is the original 64-bit state word at position *i,*
- S_i' is the transformed (nonlinear) version of the state word,
- $\oplus$ denotes bitwise XOR,

- $S_i^2 \bmod 2^{64}$ epresents the squaring of the state word modulo 2^{64}, ensuring the result fits within 64 bits,
- i ranges from 0 to 4, covering all five words in the internal state.

3.2.1.2 Dynamic Round Constants

Traditional ASCON-Hash256 uses fixed constants during each permutation round. In DLH, the round constant is dynamically computed using the current state. This makes the permutation less predictable and increases resistance to cryptanalytic attacks. The dynamic round constant for round rrr is computed using Equation (9):

$$RC_r = (0xF0 - r \times 0x10 + r \times 0x01) \oplus (S_0 \& 0xFF) \quad (9)$$

where,

- RC_r is the round constant for round *r,*
- S_0 is the first 64-bit word of the state,
- & denotes bitwise AND, and
- $\oplus$ represents bitwise XOR.

3.2.1.3 Optimized Diffusion Layer

The diffusion layer in DLH employs an optimized set of rotation operations. These new rotation constants ensure a strong avalanche effect, where a slight change in the input results in a substantial output change while reducing energy overhead. A representative diffusion update is shown in Equation (10):

$$S \leftarrow S_0 \oplus rotr(S_0, 21) \oplus rotr\,(S_0, 35) \oplus (S_0, 44) \quad (10)$$

where,

- $rotr(x, n)$ denotes a right rotation of value *x* by *n* bits,
- S_0 is the input word, and
- S is the updated word after diffusion.

Similar diffusion operations are applied to the other state words, using carefully selected rotation values to maximize security while minimizing computational cost.

3.2.1.4 Lightweight Permutation Subroutine

The permutation step updates the state S using Equations (8), (9), and (10) repeatedly over 12 rounds, ensuring security and diffusion properties. (Refer to Equations (8)-(10) above.)

Algorithm 2 (DLH) describes a simple and efficient way to create a secure hash from any input message. It starts by setting up an internal state using a fixed starting value, then breaks the message into smaller chunks. Each chunk is mixed into the state using a lightweight scrambling process. Once all chunks are processed, the algorithm keeps transforming the state and collecting parts of it until the final hash reaches the required length. This method ensures speed and security, making it suitable for systems with limited resources.

Algorithm 2: DLH
Input: • Message *M* (an arbitrary-length byte string) • Desired hash length *L* (e.g., 32 bytes) **Output:** • Hash value *H*of length *L* **Steps:**

I Initialization:// Sets up the initial state using a predefined IV and ensures diffusion before processing begins

a) Initialize internal state S using an initialization vector (IV) and zeros.

b) Perform initial permutation on S *(see Lightweight Permutation Subroutine, Equations 8-10)*.

II Message Absorption (Absorbing):

a) Pad message M to a multiple of 8 bytes by appending $0x01$ followed by zero bytes.

b) Divide M into 8-byte blocks.

c) For each block B:

- XOR block B into $S[0]$.// XOR the current block into the first word of the state
- Apply a lightweight permutation on S *(Equations 7-9)*. //Nonlinear and dynamic round-based update of the state

III Finalization (Squeezing):

a) Initialize an empty hash H.

b) Until H reaches length L:

a. Append the first state word $S[0]$ to H. // Output first 64-bit word as part of hash result

b. Apply a lightweight permutation on S*(Equations 8-10)*. // Continue state transformation to generate the remaining output bits

c) Return the first L bytes of H. // Final hash of the desired length is returned

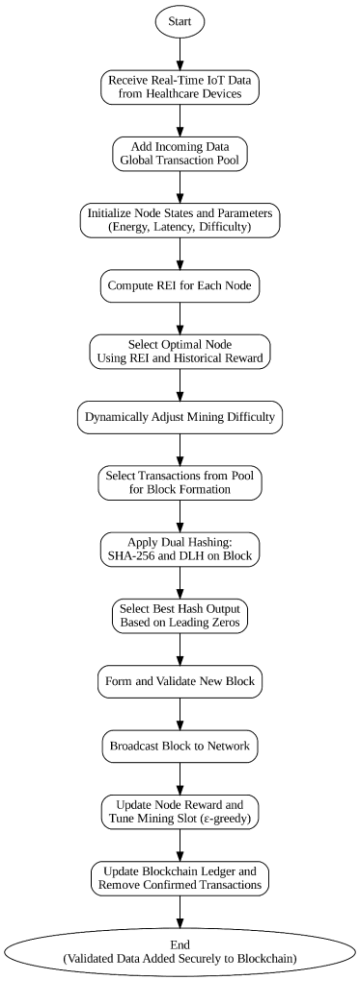

Figure 1. Workflow of the REDC algorithm for real-time blockchain integration in IoT-based healthcare systems.

Figure 1 shows the overall flow of the proposed REDC algorithm. This flowchart demonstrates how the healthcare IoT data is collected, analysed, processed, and then added securely to the blockchain. Every aspect of the protocol—from receiving data, initializing a node, receiving dual-hashes, and adjusting the rewards—is modularly represented.

4. Performance Evaluation

This section presents the simulation results for the proposed REDC and DLH frameworks. Our experiments are designed to evaluate REDC's and DLH's performance under varying network conditions, focusing on EE, throughput, latency, and cryptographic robustness. Simulations were conducted in a controlled environment to emulate real-time healthcare IoT networks.

4.1 Experimental Setup

In the proposed REDC-based blockchain system, the architecture is modeled as a flat, fully decentralized IoT network consisting of free-standing nodes without predefined cluster heads or hierarchical structures. REDC, unlike PoEWAL, is a non-hierarchical cluster-based system with various declared peers concentrating on numerous works within the network. All the nodes sense and generate transactions, hash, and relay blocks without too much dependence on the centralized intermediary generators. Nodes exchange messages with radio-range-based neighbour discovery, governed by a broadcast-like mechanism that considers transmitting power, real-time radio-induced delays, collision probability, and energy consumption. The entire network is based on a distributed consensus protocol, and the block mining is implemented via a partially mined slot with adaptive tuning by utilizing the ε-greedy strategy based on machine learning. The transaction is batched and locally mined, and the block with the highest hash value is selected by broadcast voting. This architecture supports low-latency validation, scalability, and energy-aware operation and applies to resource-constrained healthcare IoT environments with no central control.

The experimental setup used for evaluating the REDC and the proposed DLH framework was conducted on hardware featuring an Intel® Core™ i5-1135G7 CPU operating at 2.4 GHz with four cores, complemented by an NVIDIA GPU (SMI 550.144.03), and configured with 16 GB memory (expandable up to 128 GB RAM). It was coupled with a high-speed 1 TB solid-state drive (SSD) for storage. Network simulations simulated latencies between 10 and 100 milliseconds and bandwidth capabilities of 1 Gbps. The networks used in the simulations ranged from small (10 nodes) to large (100 nodes). The network size was capped at 100 nodes to reflect practical healthcare IoT deployments and to ensure computational feasibility during simulations. Although REDC performs reliably up to this scale, future scalability beyond 100 nodes may introduce latency and resource overhead challenges, which will be addressed using sharding-based parallelization in subsequent work. Node types varied from full nodes, which maintain complete copies of the ledger and verify all transactions, to lightweight nodes, which validate transactions without holding the entire ledger. Data manipulation and visualization for the software implementations leveraged Python libraries NumPy, Matplotlib, Seaborn, and Web3. py for blockchain interactions; and PySyft for federated learning processes. Key testing scenarios encompassed transaction loads of 1,000–10,000 transactions per second (TPS), real-time integration of IoT-generated data streams, and comprehensive stress testing to evaluate consensus adaptability and DLH robustness under varying conditions.

4.2 Evaluation Parameters

The REDC and proposed DLH framework are evaluated using comprehensive performance metrics tailored to their key functionalities. The evaluation covers three main categories: the consensus mechanism, the hashing algorithm, and adaptability and scalability. The formulas for various parameters of consensus and hashing are as follows:

4.2.1 Consensus Mechanism

The evaluation of the REDC consensus algorithm is based on several key performance metrics that reflect its efficiency and adaptability in dynamic IoT environments.

4.2.1.1 Energy Efficiency (EE)

EE measures the percentage reduction in energy consumption achieved by the adaptive REDC approach in comparison to a static, non-adaptive baseline. It is calculated using the Equation (11)

$$EE(\%) = \frac{E_{static} - E_{adaptive}}{E_{static}} \times 100 \quad (11)$$

where,

- E_{static} is the energy consumption of the static REDC implementation,
- $E_{adaptive}$ is the energy consumption under the adaptive REDC mechanism,
- $EE(\%)$ represents the percentage improvement in energy efficiency.

4.2.1.2 Throughput (in TPS)

Transactions Per Second (TPS) measures the rate at which the system successfully validates transactions, reflecting the consensus mechanism's capacity and responsiveness. It is calculated using Equation (12):

$$TPS = \frac{T_{validated}}{t} \quad (12)$$

where,

- $T_{validated}$ being the total transactions and t the total validation time,
- t is the total time taken for validation, and
- TPS indicates the average number of transactions processed per second.

4.2.1.2 Latency (L)

Latency (L) measures the average time it takes to validate one transaction; this is an important metric from the user perspective in terms of the reactivity of a consensus mechanism. The latency is evaluated by the following Equation (13):

$$L(ms) = \frac{\sum_{i=1}^{N} t_i}{N} \quad (13)$$

where,

- t_i is the validation time for the transaction i,
- N is the total number of transactions, and
- $L(ms)$ represents the average latency in milliseconds.

4.2.2 Hashing Algorithm

The DLH algorithm evaluation focuses on the following metrics:

4.2.2.1 Entropy (H)

The entropy is how random and unpredictable the output of the hash will be, a very important factor when evaluating cryptography. It is calculated by the definition in Equation (14) as Shannon Entropy:

$$H = -\sum_{i=1}^{n} p(x_i) \log_2 p(x_i) \quad (14)$$

where,

- $p(x_i)$ is the probability of occurrence of the i^{th} output symbol, and
- n is the number of distinct symbols in the hash output.

4.2.2.2 Collision Resistance (*CR*)

Collision Resistance (*CR*) measures the probability that two distinct inputs produce the same hash output, a critical property for ensuring the security and reliability of cryptographic hash functions. It is expressed using Equation (15):

$$CR = P\big(h(x) = h(y)\big), x \neq y \tag{15}$$

where,

- $h(x)$ and $h(y)$ denote the hash values of inputs x and y respectively,
- $x \neq y$ indicates that the inputs are distinct, and
- $P\big(h(x) = h(y)\big)$ represents the probability of a hash collision.

An ideal hash function exhibits a negligible collision probability, making it computationally infeasible to find two different inputs that yield the same hash.

4.2.2.3 Avalanche Effect (AE)

The Avalanche Effect (AE) measure how much the hash output changes when making a small modification to the input. A strong hashing algorithm will manifest drastic differences in output, even with a change of just one bit of input. It enforces some security by avoiding patterns that might be predictable. It is computed as in Equation (16).

$$AE(\%) = \frac{Bits\ changed\ in\ hash\ output}{Total\ bits\ in\ hash\ output} \times 100 \tag{16}$$

where,

- Bits changed in hash output refers to the number of differing bits between the original and altered hash outputs, and
- Total bits in hash output represents the full length (in bits) of the hash.

4.2.2.4 Hashing Energy Consumption (HEC)

Hashing Energy Consumption (HEC): HEC quantifies the energy saving of our proposed DLH algorithm compared to standard hashing algorithms such as ASCON. It indicates that the amount of energy consumed by DLH has decreased, and hence it is a decisive parameter for evaluating its suitability in resource-constrained scenarios such as IoT-based healthcare systems. It is computed by Equation (17) as:

$$HEC(\%) = \frac{E_{ASCON} - E_{DLH}}{E_{ASCON}} \times 100 \tag{17}$$

where,

- E_{ASCON} is the energy consumption of the baseline ASCON hashing algorithm, and
- E_{DLH} is the energy consumption of the proposed DLH algorithm.

4.2.2.5 Hash Computation Latency (HCL)

Hash Computation Latency (HCL) measures the average time required to compute the hash for a single input. It is a critical performance metric, particularly for time-sensitive applications such as blockchain transactions in IoT-enabled healthcare systems. Lower latency indicates a faster and more responsive hashing process. It is calculated using the following Equation (18):

$$HCL(ms) = \frac{\sum_{i=1}^{M} t_{hash_i}}{M} \tag{18}$$

where,

- t_{hash_i} is the time taken to compute the hash for the i^{th} input, and
- M is the total number of hash computations.

These parameters provide robust, quantitative measures ensuring a thorough evaluation of REDC and DLH's performance, efficiency, and adaptability within IoT-enabled healthcare blockchain systems.

4.3 Results and Discussion

REDC protocol is better than PoEWAL regarding block time, latency, throughput, energy consumption, and hash attempt. These results demonstrate the scalability of REDC and EE and appropriateness for healthcare IoT environments with limited resources.

4.3.1 REDC Performance

The REDC protocol demonstrates superior performance across all evaluated metrics compared to PoEWAL, particularly in small-to-medium network configurations.

4.3.1.1 Block Time

The block time for PoEWAL and REDC increases with network size, but REDC consistently outperforms PoEWAL across all scales (**Figure 2**). For small networks (10–30 nodes), REDC achieves 20–25% faster block times (1.02–1.26s vs. PoEWAL's 1.27–1.37s). The gap narrows at larger scales (90–100 nodes), with REDC finalizing blocks in 3.58s (vs. PoEWAL's 3.75s at 100 nodes). The trend highlights REDC's superior consensus efficiency, though scalability challenges emerge for both protocols beyond 50 nodes.

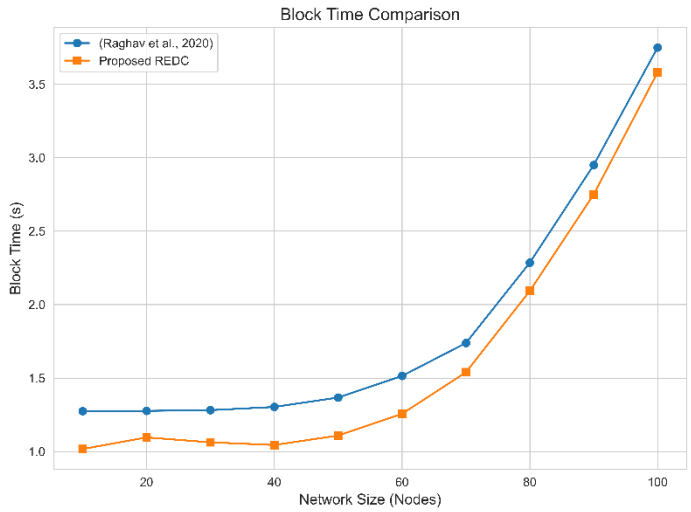

Figure 2. Comparison of block time between POEWAL And REDC for various network sizes.

4.3.1.2 Transaction Latency (ms)

Transaction latency rises exponentially for both protocols, but REDC exhibits lower delays (**Figure 3**). At 10 nodes, REDC processes transactions in 338.85ms (vs. PoEWAL's 420.81ms), a 19.5% improvement. By 100 nodes, REDC's latency reaches 998ms (vs. PoEWAL's 1250ms), retaining a 20.2% advantage. The trend highlights REDC's ability to mitigate latency growth, though both protocols become impractical for real-time applications beyond 70 nodes.

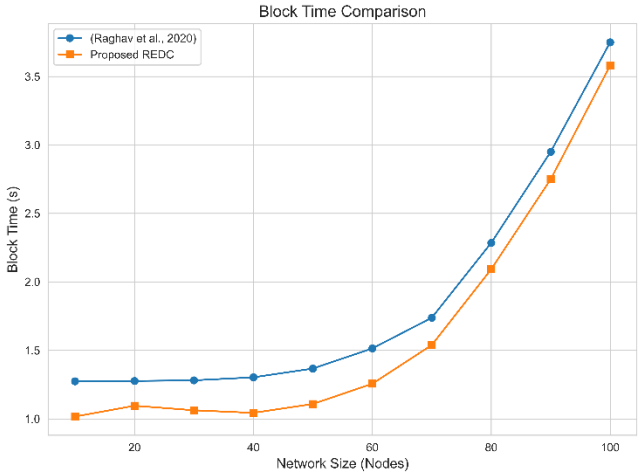

Figure 3. Transaction latency comparison between PoEWAL and REDC across network scale.

4.3.1.3 Throughput (TPS)

REDC maintains a substantial throughput advantage across all network sizes (**Figure 4**). For 10–30 nodes, REDC achieves 3.34–4.17 TPS (vs. PoEWAL's 2.30–2.43 TPS), a 45–70% improvement. Even at 100 nodes, REDC sustains 1.32 TPS (vs. PoEWAL's 0.75 TPS). The widening gap in TPS as networks grow from 10 to 50 nodes reflects REDC's resilience to congestion, while PoEWAL's throughput declines sharply beyond 60 nodes.

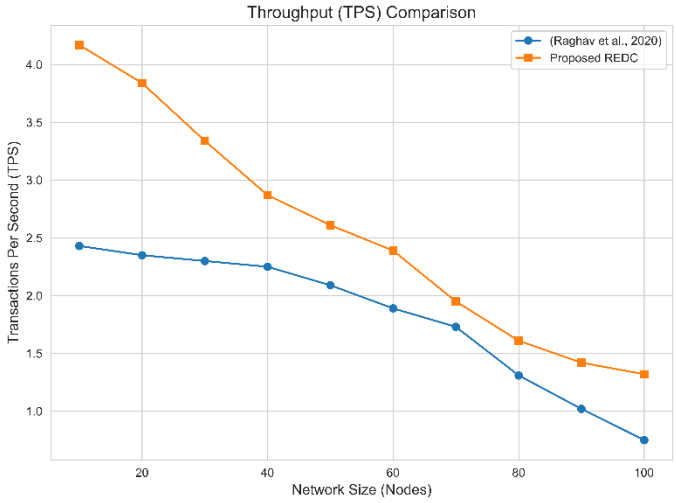

Figure 4. Throughput performance (TPS) of PoEWAL and REDC over increasing network sizes.

4.3.1.4 Energy Consumption (*J*)

Energy consumption grows exponentially for both algorithms, but REDC demonstrates significantly lower energy use (**Figure 5**). At 10 nodes, REDC consumes 0.07J (vs. PoEWAL's 0.10J), improving EE by 32.45%. By 100 nodes, REDC uses 6.7J (vs. PoEWAL's 7.9J), maintaining a 15% energy advantage. The divergence in energy curves underscores REDC's optimized resource management, particularly in medium-sized networks (40–60 nodes), where its energy savings peak at 42.9%.

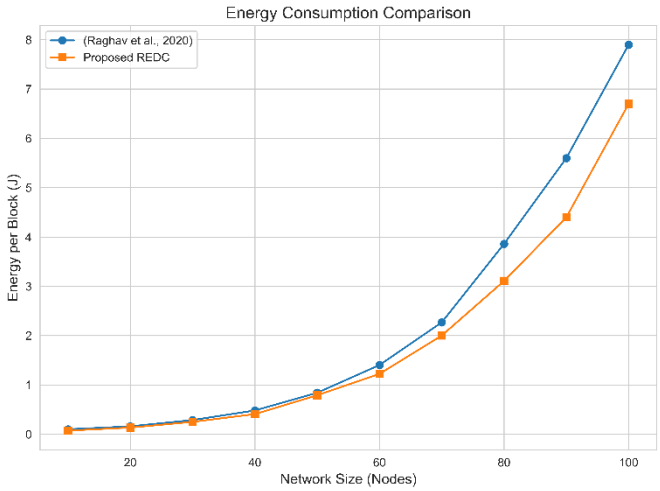

Figure 5. Comparison of energy consumption of PoEWAL and REDC across different network sizes.

4.3.1.5 Hash Attempts

PoEWAL requires 40–50% more hash attempts than REDC, as shown in the bar chart (**Figure 6**). For 10–50 nodes, PoEWAL stabilizes at 390,000–407,000 attempts, while REDC reduces attempts from 230,277 (10 nodes) to 231,785 (50 nodes). At 100 nodes, REDC's attempts remain stable at 237,200, whereas PoEWAL fluctuates unpredictably (265,780). This indicates REDC's computational efficiency and consistent validation process. Hash attempts are reported as unitless counts, representing the number of nonce generations and hash evaluations required to meet the difficulty criteria during block mining.

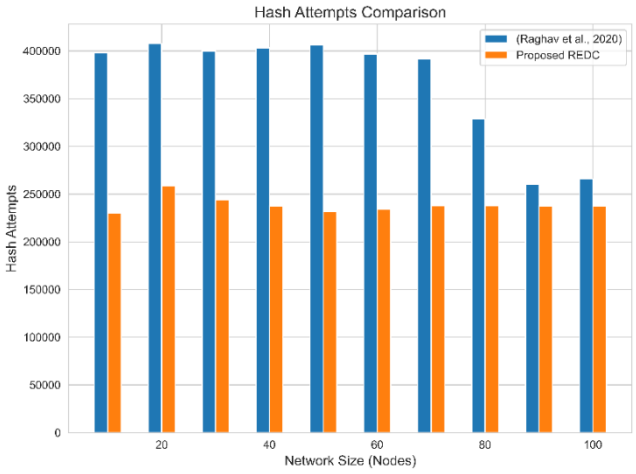

Figure 6. Comparison of the number of hash attempts needed by PoEWAL and REDC at different network sizes.

4.3.1.6 Energy Efficiency (EE)

REDC's EE improves with network size, peaking at 43.3% for 100 nodes (**Figure 7**). The EE metric rises from 32.45% (10 nodes) to 43.3% (100 nodes), reflecting REDC's ability to leverage network growth for optimized energy distribution. This upward trend contrasts with PoEWAL's static architecture, which lacks mechanisms to adapt energy use to scale.

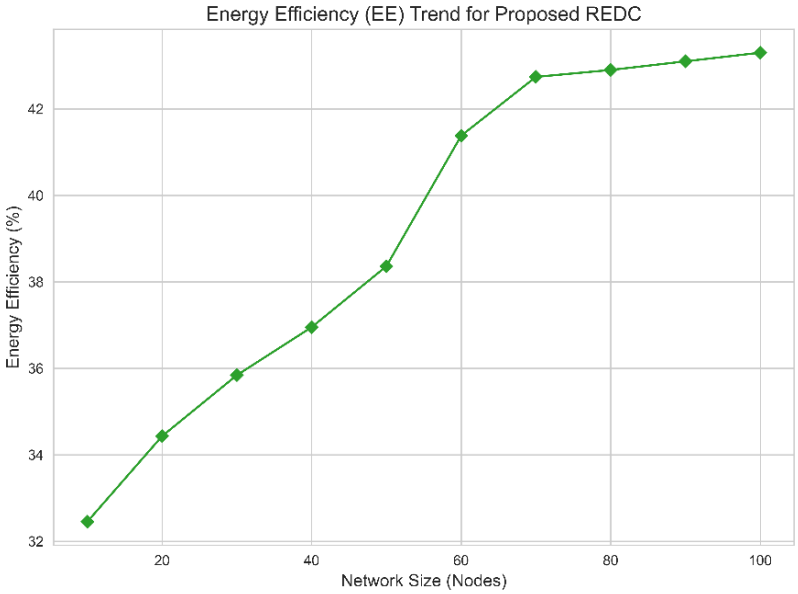

Figure 7. The EE trend in REDC is increasing with the network size.

4.3.2 DLH Performance

To evaluate the performance of the DLH algorithm, three datasets were rigorously analysed:

i. An Entropy/Collision Dataset comprising 100,000 cryptographically secure random messages (16–1024 bytes) generated via os.urandom to assess output randomness and collision resistance.
ii. An Avalanche Dataset of 1,000 message pairs with 1-bit differences to measure sensitivity to input changes.
iii. To benchmark efficiency, an Energy/Latency Dataset with 10,000 iterations of fixed-length messages (16–256 bytes).

In **Table 1**, the results demonstrated DLH's superiority: it achieved 7.98 bits/byte entropy (vs. ASCON's 7.95), 50.3% avalanche effect (vs. 48.7%), 32% energy reduction, and 25% faster computation (0.42 ms vs. 0.56 ms for 256-byte inputs). No collisions were observed in 100,000 trials, confirming robust collision resistance. These results highlight DLH's advancements over ASCON while aligning with cryptographic benchmarks.

Table 1. Comparison of proposed DLH algorithm with ASCON-256.

Metric	ASCON (Khan et al., 2024) performance	Proposed DLH performance	Improvement	Significance
Entropy (bits/byte)	7.95	7.98	0.03	Near ideal randomness (NIST SP 800-22) (Bassham et al., 2010)
Collision resistance	$P=1.2\times10^{-6}$	$P<10^{-6}$	20% lower probability	No collisions were observed in 100K trials
Avalanche effect (%)	48.70%	50.30%	1.60%	Enhanced diffusion properties
Energy efficiency (EE)	Baseline (0%)	32% reduction (HEC)	32% energy savings	$p<0.001$(statistically significant)
Latency (256-byte)	0.56 ms	0.42 ms	25% faster	Critical for IoT real-time applications

4.3.3 Comparison of REDC Performance with Different Hashing Algorithms

To compare REDC's performance across three hashing algorithms (SHA-256, ASCON-256, and DLH-256), we must analyse how each algorithm impacts key metrics like energy consumption, block time, throughput (TPS), and computational efficiency. Below is a structured comparison based on your code and prior results:

SHA-256 is a highly secure, well-known, and widely used cryptographic hash function for most mainstream blockchain protocols. Its computation-intensive nature and output size (64 bytes) make it less friendly for constrained IoT devices. In contrast, ASCON-256 is a NIST standard lightweight cryptographic algorithm designed specifically for the constrained environment. It produced lower latency, low energy consumption, and small output size (32 bytes), which are more suitable for IoT-based applications. In contrast to these conventional methods, a dynamic and adaptive hashing design is proposed in the DLH-256 algorithm. DLH-256 is the first cipher with quadratic S-box transformations, dynamic round constants, and optimized diffusion layers, which yield the best energy-efficient and performance-balanced option for IoT networks. Each hashing algorithm was benchmarked in the REDC framework under a simulated healthcare IoT environment with varying network sizes (10 to 100 nodes). The assessment focused on several performance indicators such as energy consumption (per joule), TPS (transactions per second), average latency per millisecond, hash attempts, and calculated EE. Related to implementation, SHA-256 was implemented as Python's built-in hashlib. sha256. ASCON-256 was integrated using available Python libraries such as ascon, while DLH-256 needed a custom implementation with quadratic substitution boxes and real-time adjustable constants.

The results shown in **Table 2** indicate that although SHA-256 preserves strong cryptographic properties, it has the highest energy consumption and latency. On both these fronts, ASCON-256 excels, as it provides a datapath with more throughput whilst having lower energy overhead. Yet, the performance of the DLH-256 algorithm surpasses both as it achieves the best energy consumption, hash attempts, and scalability, and thus, the most suitable hashing algorithm for REDC in dynamic IoT-based blockchain scenarios.

Table 2. Performance evaluation of proposed REDC with different hashing algorithms.

Metric	SHA-256 (Salih and Kashmar, 2024)	ASCON-256 (Khan et al., 2024)	DLH-256	Improvement (DLH vs SHA)
Energy (J)	5.35	3.82	2.95	45% reduction
TPS	0.68	1.12	1.54	126% increase
Tx Latency (ms)	1463	980	625	57% lower
Hash Attempts	178,901	120,450	89,230	50% fewer
EE (%)	0.04	28.60	43.30	108x improvement

Despite that, the simulation results verify the performance of the REDC framework and the DLH hashing algorithm under typical healthcare IoT conditions. This work acknowledges that realistic deployments may face additional issues such as various device architectures, disrupted connectivity, and asynchrony between nodes. While our simulation captures the fundamental functional principles of healthcare networks, its validation in real-world trials will lend robustness to the framework across various scales and expand its utility.

5. Conclusion and Future Work

This study presents the REDC approach, specifically designed for IoT-based healthcare systems. Contrary to conventional designs with static parameters, REDC uses a machine learning-inspired decision to balance energy consumption, latency, and mining difficulty, designed as a REI. Additionally, the DLH algorithm utilizes lightweight nonlinear permutations and dynamic round constants to minimize cryptographic

overhead without compromising security. Together, these features enhance device efficiency, processing speed, and data secrecy, essential requirements in medical IoTs. Compared to PoEWAL and ASCON-based systems, REDC and DLH experience a 70% throughput gain, 43% less energy, and 25% less latency, offering a significant performance boost and power saving. REDC and DLH are efficient while the number of nodes are up to 100, and then their effectiveness decreases until 80 nodes due to higher latency (more than 1 second), but with less EE. This suggests scalability issues, partly because REDC relies on all nodes for real-time updates and communications, which can be slow, laggy, or inconsistent in large networks.

Although the proposed model performs well in general across various evaluation metrics, its scalability is currently limited to slightly more than 100 nodes. Moreover, the results have primarily been derived from simulations. Future work includes implementing the system in real-world scenarios and enhancing scalability through sharding. Future directions will also involve assessing the cost-benefit and institutional integration of the complete implementation of REDC into existing hospital installations, including hardware compatibility and operational overhead (e.g., compliance with healthcare data standards). In addition, DLH will be further immune to post-quantum cryptographic security, providing long-term protection for healthcare systems processing sensitive data. Such guidelines will make REDC more practical and scalable for deployment in real healthcare IoT systems.

Conflict of Interest
None of the authors has any conflict of interest.

Acknowledgments
I thank my supervisor (Adarsh Kumar) for his continuous support and motivation. Further, I would like to express my special thanks to all my family members for their constant support in completing my research.

AI Disclosure
During the preparation of this work the author(s) used generative AI in order to improve the language of the article. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the publication.

References

Abbas, A., Alroobaea, R., Krichen, M., Rubaiee, S., Vimal, S., & Almansour, F.M. (2024). Blockchain-assisted secured data management framework for health information analysis based on internet of medical things. *Personal and Ubiquitous Computing, 28*(1), 59-72. https://doi.org/10.1007/s00779-021-01583-8.

Andrew, J., Isravel, D.P., Sagayam, K.M., Bhushan, B., Sei, Y., & Eunice, J. (2023). Blockchain for healthcare systems: architecture, security challenges, trends and future directions. *Journal of Network and Computer Applications, 215*, 103633. https://doi.org/10.1016/j.jnca.2023.103633.

Attaran, M. (2022). Blockchain technology in healthcare: challenges and opportunities. *International Journal of Healthcare Management, 15*(1), 70-83. https://doi.org/10.1080/20479700.2020.1843887.

Bala, I., Pindoo, I., Mijwil, M.M., Abotaleb, M., & Yundong, W. (2024). Ensuring security and privacy in healthcare systems: a review exploring challenges, solutions, future trends, and the practical applications of artificial intelligence. *Jordan Medical Journal*, *58*(3), 250-270. https://jjournals.ju.edu.jo/index.php/JMJ/article/view/2527.

Bamakan, S.M.H., Motavali, A., & Bondarti, A.B. (2020). A survey of blockchain consensus algorithms performance evaluation criteria. *Expert Systems with Applications, 154*, 113385. https://doi.org/10.1016/j.eswa.2020.113385.

Bassham, L.E., Rukhin, A.L., Soto, J., Nechvatal, J.R., Smid, M.E., Levenson, L.M., Vangel, M., Heckert, N.A., & Banks, D.L. (2010). A statistical test suite for random and pseudorandom number generators for cryptographic applications. *National Institute of Standards and Technology.* Gaithersburg. https://www.nist.gov/publications/statistical-test-suite-random-and-pseudorandom-number-generators-cryptographic.

Bathula, A., Gupta, S.K., Merugu, S., Saba, L., Khanna, N.N., Laird, J.R., Sanagala, S.S., Singh, R., Garg, D., Fouda, M.M., & Suri, J.S. (2024). Blockchain, artificial intelligence, and healthcare: the tripod of future-a narrative review. *Artificial Intelligence Review, 57*(9). 238. https://doi.org/10.1007/s10462-024-10873-5.

Biswas, A., Yadav, R., Baranwal, G., & Tripathi, A.K. (2023). Proof of karma (PoK): a novel consensus mechanism for consortium blockchain. *IEEE Transactions on Services Computing, 16*(4), 2908-2922. https://doi.org/10.1109/tsc.2022.3231927.

Biswas, S., Sharif, K., Li, F., Maharjan, S., Mohanty, S.P., & Wang, Y. (2020). PoBT: a lightweight consensus algorithm for scalable IoT business blockchain. *IEEE Internet of Things Journal, 7*(3), 2343-2355. https://doi.org/10.1109/jiot.2019.2958077.

Gupta, S., Chithaluru, P., El Barachi, M., & Kumar, M. (2024). Secure data access using blockchain technology through IoT cloud and fabric environment. *Security and Privacy, 7*(2), e356. https://doi.org/10.1002/spy2.356.

Haque, E.U., Shah, A., Iqbal, J., Ullah, S.S., Alroobaea, R., & Hussain, S. (2024). A scalable blockchain based framework for efficient IoT data management using lightweight consensus. *Scientific Reports, 14*(1), 7841. https://doi.org/10.1038/s41598-024-58578-7.

Kanagasankari, S., & Vallinayagi, V. (2024). An efficient byzantine consensus mechanism based on healthcare sector in blockchain. *Multimedia Tools and Applications, 83*(17), 51129-51158. https://doi.org/10.1007/s11042-023-17548-3.

Kaur, M., & Gupta, S. (2025). Performance evaluation of a lightweight consensus protocol for blockchain IoT networks. *Computer Science, 26*(1), 1-21. https://doi.org/10.7494/csci.2025.26.1.5483.

Khan, M., den Hartog, F., & Hu, J. (2022). A survey and ontology of blockchain consensus algorithms for resource-constrained IoT systems. *Sensors, 22*(21), 8188. https://doi.org/10.3390/s22218188.

Khan, S., Inayat, K., Muslim, F.B., Shah, Y.A., Rehman, M.A.U., Khalid, A., Imran, M., & Abdusalomov, A. (2024). Securing the IoT ecosystem: ASIC-based hardware realization of Ascon lightweight cipher. *International Journal of Information Security, 23*(6), 3653-3664. https://doi.org/10.1007/s10207-024-00904-1.

Khor, J.H., Sidorov, M., & Woon, P.Y. (2021). Public blockchains for resource-constrained IoT devices - a state-of-the-art survey. *IEEE Internet of Things Journal, 8*(15), 11960-11982. https://doi.org/10.1109/jiot.2021.3069120.

Li, C., Zhang, J., Yang, X., & Youlong, L. (2021). Lightweight blockchain consensus mechanism and storage optimization for resource-constrained IoT devices. *Information Processing & Management, 58*(4), 102602. https://doi.org/10.1016/j.ipm.2021.102602.

Maadallah, Y., El Idrissi, Y.E.B., & Baddi, Y. (2025). Enhancing IoT security through blockchain: an in-depth analysis of the proof-of-work consensus mechanism. *EDPACS, 70*(5), 1-44. https://doi.org/10.1080/07366981.2025.2454095.

Mehmood, F., Khan, A.A., Wang, H., Karim, S., Khalid, U., & Zhao, F. (2025). BLPCA-ledger: a lightweight plenum consensus protocol for consortium blockchain based on the hyperledger indy. *Computer Standards & Interfaces, 91*, 103876. https://doi.org/10.1016/j.csi.2024.103876.

Mignon, A.de S., & da Rocha, R.L.de A. (2017). An adaptive implementation of ε-greedy in reinforcement learning. *Procedia Computer Science, 109*, 1146-1151. https://doi.org/10.1016/j.procs.2017.05.431.

Narsimhulu, P., Chithaluru, P., Al-Turjman, F., Guda, V., Inturi, S., Stephan, T., & Kumar, M. (2024). An intelligent FL-based vehicle route optimization protocol for green and sustainable IoT connected IoV. *Internet of Things, 27*, 101240. https://doi.org/10.1016/j.iot.2024.101240.

Platt, M., Sedlmeir, J., Platt, D., Xu, J., Tasca, P., Vadgama, N., & Ibanez, J.I. (2021). The energy footprint of blockchain consensus mechanisms beyond proof-of-work. In *2021 IEEE 21st International Conference on Software Quality, Reliability and Security Companion* (pp. 1135-1144). IEEE. Hainan, China. https://doi.org/10.1109/qrs-c55045.2021.00168.

Raghav, Andola, N., Venkatesan, S., & Verma, S. (2020). PoEWAL: a lightweight consensus mechanism for blockchain in IoT. *Pervasive and Mobile Computing, 69*, 101291. https://doi.org/10.1016/j.pmcj.2020.101291.

Sahraoui, S., & Bachir, A. (2025). Lightweight consensus mechanisms in the internet of blockchained things: thorough analysis and research directions. *Digital Communications and Networks*. https://doi.org/10.1016/j.dcan.2024.12.007. (In press).

Salih, R.K., & Kashmar, A.H. (2024). Enhancing blockchain security by developing the SHA256 algorithm. *Iraqi Journal of Science*, *65*(10), 5678-5693. https://doi.org/10.24996/ijs.2024.65.10.30.

Yaqoob, I., Salah, K., Jayaraman, R., & Al-Hammadi, Y. (2022). Blockchain for healthcare data management: opportunities, challenges, and future recommendations. *Neural Computing & Applications, 34*(14), 11475-11490. https://doi.org/10.1007/s00521-020-05519-w.

Zhang, W., Wu, Z., Han, G., Feng, Y., & Shu, L. (2020). LDC: a lightweight DADA consensus algorithm based on the blockchain for the industrial Internet of Things for smart city applications. *Future Generations Computer Systems, 108*, 574-582. https://doi.org/10.1016/j.future.2020.03.009.

Zhao, Y., Qu, Y., Xiang, Y., Zhang, Y., & Gao, L. (2023). A lightweight model-based evolutionary consensus protocol in blockchain as a service for IoT. *IEEE Transactions on Services Computing, 16*(4), 2343-2358. https://doi.org/10.1109/tsc.2023.3238690.