Proof of Physical Activity (PoPA): A Novel Blockchain Consensus Algorithm Based On Physical Work

by

Rodel Advan
22341047
Md. Mashfiqun Nabi
22141027
Faiaj Sahib
24141168
Aarshi Durriya Raihan
22101546

A thesis submitted to the
Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
BRAC University
January 2026

Declaration

It is hereby declared that

1. The thesis report submitted is our own original work while completing degree at BRAC University.
2. The report does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The report does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Rodel Advan
22341047

Md. Mashfiqun Nabi
22141027

Faiaj Sahib
24141168

Aarshi Durriya Raihan
22101546

Approval

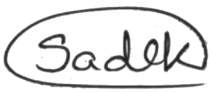
The thesis report titled "Proof of Physical Activity (PoPA): A Novel Blockchain Consensus Algorithm Based On Physical Work" submitted by

1. Rodel Advan (22341047)
2. Md. Mashfiqun Nabi (22141027)
3. Faiaj Sahib (24141168)
4. Aarshi Durriya Raihan (22101546)

Of Fall, 2025 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on January 28, 2026.

Examining Committee:

Supervisor:
(Member)



Dr. Md Sadek Ferdous, PhD
Professor
Department of Computer Science and Engineering
BRAC University

Thesis Coordinator:
(Member)

Dr. Md. Golam Rabiul Alam, PhD
Professor
Department of Computer Science and Engineering
BRAC University

Head of Department:
(Chair)

Dr. Sadia Hamid Kazi, PhD
Chairperson and Associate Professor
Department of Computer Science and Engineering
BRAC University

Abstract

Blockchain technology, which has persisted for a considerable period and has achieved remarkable advancements since the emergence of Bitcoin, still faces a challenge in normalizing cryptocurrency mining using smartphones for general users. This difficulty stems from the complexity of certain algorithms, such as Proof of Work (PoW) and Proof of Stake (PoS), which smartphones cannot effectively handle. Despite numerous proposals for smartphone-based consensus algorithms, integrating various Internet of Things (IoT) accessories, like smartwatches, into the smartphone consensus for enhanced processing capabilities remains uncommon. To address this research gap, we present a novel smartphone-based blockchain consensus protocol called Proof of Physical Activity (PoPA). PoPA leverages human physiological data collected from smartwatch sensors to participate in the mining process. This approach not only enhances the mining efficiency but also serves as a proof of physical fitness, supporting various applications.

Keywords: Blockchain; Consensus; Protocol;

Acknowledgement

Thank you to Dr. Md Sadek Ferdous, PhD sir for his supervision in producing this report. His guidance, feedback, and encouragement were invaluable throughout the research and writing process. We also extend our heartfelt gratitude towards Mr. Md. Yeasin Ali for his consistent guidance throughout the research process. Additionally, we would also like to thank our friends and family for their support and understanding during this time. Finally, we acknowledge the resources and facilities provided by BRAC University that facilitated our research work.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Blockchain became popularized by the rise of Bitcoin in 2008 and thus most people equate Blockchain to be only related to cryptocurrencies. However, it emerged other influential technologies like the World Computer of Ethereum, various NFTs, Web 3.0 and so much more. The main advantage of blockchain is that it offers strong unalterable transparent transactions while ensuring pseudo-anonymity of any user involved in the system. It utilizes a complex data structure to form a secure database that is decentrally stored across the globe.

Thus, to utilize these benefits, many organizations and individuals made a lot of small blockchain based technologies where vulnerabilities slowly began to pop up. It all started with individuals trying to use Proof of Work's (PoW) vulnerabilities to mine the most bitcoins for themselves. PoW mostly depends on a computer's computational capabilities, so someone with a more expensive and a more highly powered CPU can easily have an upper hand on other users and can end up doing most of the mining. Furthermore, due to its operations requiring a considerable amount of computational power, PoS can consume a lot of electricity. The algorithm also does not penalize any nodes that act maliciously. To solve these problems, Ethereum came up with a Proof of Stake (PoS) system and a lot more additional functionalities in their blockchain which again brought in more vulnerabilities to explore with Smart contracts on the rise. PoS, as the name suggests, requires users to put an amount of money into an escrow account as their stake (based on which they will be rewarded or penalized) in order to become a miner. So, a user already needs to have a certain amount of coins to be able to participate in the mining process, which gives already rich users an unfair advantage and nodes with a lower amount of coins, no opportunities to participate in the process.

1.1 Background

As the blockchain ecosystem expands beyond finance into new areas such as health care and fitness, the importance of energy efficient, decentralized, and secure consensus mechanisms increases. Traditional mining is inaccessible to many users due to its high demand of computational resources, thus creating a barrier to entry that hinders wide-spread participation. Over the last couple of years, researchers have explored multiple alternatives to traditional mining consensus mechanisms to address the three main problems that have plagued previous implementations: excessive energy consumption, centraliza-

tion, and limited scalability. The focus of research development in this area has included using blockchain technology with Internet of Things (IoT) devices to enhance data security through the use of permissioned networks, and incorporating machine learning into blockchain systems to support customized services. A recent trend among researchers has been to explore ways to link real world activity (i.e., physical exercise or scientific computations), with the process of validating blockchain consensus by leveraging this approach as an alternative to heavy computation models that consume large amounts of energy. While there is increasing attention given to this area of study, most current solutions do not address the issue of how users can interact with blockchain based systems using their daily usage of smart phones and wearables to validate consensus. Additionally, many solutions currently employ token-based incentives or traditional validation mechanisms to validate consensus, rather than utilizing the actual physical activity of users to validate consensus. On the practical side, various mobile applications and fitness platforms provide rewards to individuals for engaging in physical activity; however, the majority of these systems do not utilize secure decentralized consensus mechanisms and face limitations related to scalability. Proof of Physical Activity (PoPA) addresses the challenges associated with these limitations by providing a lightweight, low-energy consensus mechanism that utilizes physiological data collected from smartwatches and/or mobile phone sensors as input to allow every-day users to participate in blockchain networks and promote a healthy lifestyle.

1.2 Motivation

The recent upswing in wearable and portable IoT device usage like smartwatches creates a good opportunity to look into more sustainable and accessible ways validate blocks for a blockchain network. This research introduces a novel consensus algorithm, Proof of Physical Activity (PoPA), uses real-life physiological data (such as heart rate and steps count) to confirm that a transaction occurred and to provide rewards to users. Instead of using computer processing power to confirm a block or financial investment in mining such as the way other blockchain validation methods currently do, PoPA will use the level of physical activity a user performs as measured by their wearable IoT device to decide if they can be allowed to confirm a block. In doing so, the goal of this system is to encourage users with high levels of physical activity to continue to use the system to improve their health; and to further, identify a potential area of integration of physiological data with a blockchain - something that is largely unexamined in current mainstream blockchain research.

1.3 Problem Statement

Current studies and established consensus show a disconnect limiting blockchain's potential to incentivize real-world, wellness-focused behavior. To address this, we propose Proof of Physical Activity (PoPA) - a novel consensus model that validates blocks based on verified physical activity bridging blockchain and health while promoting energy-efficient consensus.

1.4 Methodology in Brief

We use physiological information (i.e., activity data) collected via smartwatches and smartphones to verify blockchain transactions; therefore, we adapted the design science methodology of Wieringa [4] to meet the needs of our research. Our design science methodology includes developing an architecture for collecting and verifying physical activity data from wearable devices in a secure manner, and transmitting this data securely into the blockchain network. To accommodate individuals with different levels of physical activity, a low-difficulty adjustment mechanism was developed so that all users have the capability to perform the mining process. Rewards for participating in the consensus mechanism are determined by each user's level of physical activity thereby creating an incentive for a healthier lifestyle while maintaining the integrity and security of the blockchain. We also included fraud countermeasures to ensure that the data used within the blockchain will be accurate and un-manipulated/spoofed. A simulator/testbed was designed to evaluate the feasibility and efficiency of the proposed consensus mechanism using either synthetic or real smartwatch data to evaluate the effectiveness of the proposed consensus mechanism.

1.5 Objectives

- **RO1:** To critically analyze the current state-of-the-art research and systems within the scope of movement-based blockchain consensus mechanisms.
- **RO2:** Design a consensus mechanism that incorporates smartwatches and makes use of movement data to validate blockchain transactions, which includes an incentivization model through which participants can be rewarded for a healthy lifestyle.
- **RO3:** Propose fraud countermeasures in terms of data integrity to avoid fraudulent data reporting against vulnerabilities such as manipulation or spoofing.
- **RO4:** Establish various strengths and weaknesses of the movement-based model relative to other sustainable blockchain solutions by conducting comparative analyses with existing approaches.
- **RO5:** Design a simulator or testbed that can validate the proposed consensus mechanism using real or synthetic smartwatch data to ensure its practical feasibility and effectiveness.

1.6 Scope and Challenges

This study operates at the consensus protocol level regardless of whether the blockchain is public or private; however, it could be used to support any of those types of blockchain networks. Its main area of focus will be integrating the physical activity derived from both the smartphone and the smartwatch into the validation process of a distributed network. It will also provide for the secure, transparent and incentivized validation of this physical activity data.

There are several key challenges that need to be addressed. They include synchronizing

the data being reported by heterogeneous devices and the blockchain, securely propagating the data in real-time, and validating the physical activity data as authentic while protecting the users' privacy. Lastly, developing a system that is scalable and provides a balance between how efficiently a consensus can occur, and the integrity of the health data that is collected is a major challenge.

1.7 Structure

The research is structured as follows: Chapter 2 describes the background and terminology associated with blockchain and the current state of existing blockchain (review). Chapter 3 outlines the needs, implications and limitations of the study with the project management plan. Chapter 4 outlines the methodology for the study which includes specifications of the design, the system architecture, and the protocols. Chapter 5 outlines how the proposed system was put into practice. Chapter 6 summarizes the analysis and offers a future plan of action for this research via concluding comments in chapter 7.

Chapter 2

Literature Review

The chapter focuses on the foundational concepts essential to understanding blockchain consensus mechanisms and explores existing literature relevant to health integrated blockchain systems. It begins with technical preliminaries, reviews prior research, and concludes with a synthesis of key insights that shape the direction of this study.

2.1 Preliminaries

Here, we discuss the preliminary knowledges that are required to understand the contents of the research.

2.1.1 Blockchain

As discussed by Golosova and Romanovs [8], blockchain is a decentralized database that adds new data with growing ordered blocks in a list ensuring transparent and secure transactions. The most popular description of the time to describe this technology was given by Don and Alex Tapscott: "The blockchain is an incorruptible digital ledger of economic transactions that can be programmed to record not just financial transactions but virtually everything of value" [6].

So, blockchain basically has a global digital ledger where each block contains n transactions in a sequence. Miners have to solve a complex mathematical algorithm using a large processing power to add a block to the blockchain. In return, they receive a small fee from each transaction. However, only a single miner can add a block to a blockchain and thus everyone is competing to add the block [6], [8].

Blockchain uses Asymmetric Public Key Cryptography identifying users using their Public Key and uses their digital signature through private key to ensure integrity and anonymity during transactions. Most commonly, RSA is used for this.

The descriptions above explain how the technology works. But we need to understand a few more concepts to understand the vulnerabilities that it might present.

2.1.2 Consensus Protocols

A consensus protocol is the method by which participants in a distributed network-like blockchain system agree on the state of the network and validate the transactions going through it. It ensures that all nodes (participants) reach a common agreement on the validity of data, even if some nodes fail or act maliciously. In blockchain system, the most common consensus protocols are the following:

Proof of Work (PoW): As per Antonopoulos and Harding [2], the PoW consensus protocol utilizes a complex algorithm to make a special mathematical problem that is related to the next block's header and other information attached with it. Miners compete to calculate the number using their maximum processor power and that process is called mining. Bitcoin uses PoW as the main mining protocol. The biggest block which is mined with the fastest calculation is added to the blockchain and the miner is rewarded a hefty sum of fee from many transactions [5].

Proof of Stake (PoS): As per Golosova and Romanovs [8], the PoS consensus protocol is an improved PoW version for a system with many users where mining is done for an easier mathematical problem through a group of miners working together and the other miners in the system vote for the valid block together to get it into the blockchain. The reward is then distributed among the miners.

Here, to participate in the mining, miners need to put a huge amount of crypto coins as Stake. If miners are misbehaving in the system, their stake is restricted from them [18]. The problem with this protocol, however, is that it does injustice to the decentralized structure of the blockchain system to an extent [30].

2.1.3 Other Important Terminologies:

Ethereum and Smart Contract: As per Tikhomirov [10], Ethereum is a blockchain platform just like Bitcoin, however it utilizes the PoS algorithm and utilizes the concept of blockchain to implement a world computer. This world computer is like a virtual machine where we can put Smart Contracts as users (which are like coded functions in a computer) and anyone in the world will be able to use it.

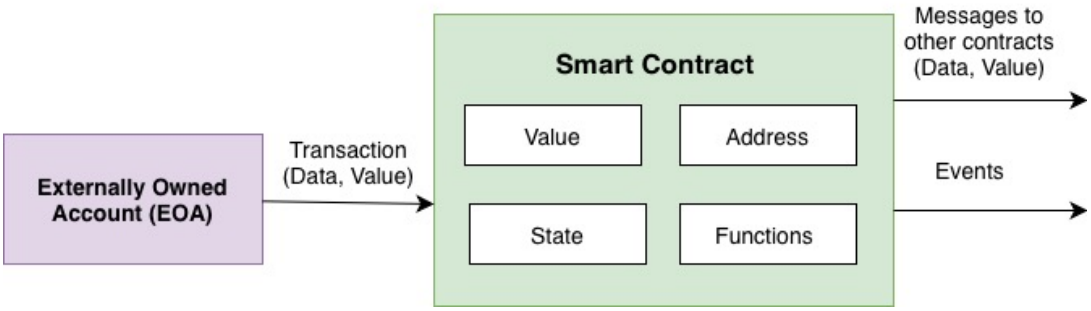


Figure 2.1: Smart Contracts [3]

As seen in Figure 2.1, to use or insert a smart contract, there is a fee given through a transaction and all logs are mined into the blockchain system present. They are self-executing contracts with the terms directly written into code. Vulnerabilities in these contracts can be exploited to manipulate the contract's behavior [17].

Fork: Due to the decentralized nature of blockchain, forks are two or more competing blockchain database instances which emerge due to a mass disagreement in consensus in blockchain protocols between nodes making a tree-like structure instead of a single chain. Attackers may try to hinder a system by creating a fork and thus making two separate chains independent of one another making the transactions in a fork block be invalidated [36].

2.1.4 Misconceptions regarding Blockchain technology:

Blockchain's rapidly growing popularity have seen a large population adapt to cryptocurrencies and the technology itself due to the complex underlying mechanisms. This has lead to some myths and misconceptions about the technology. Clarifying some of them is essential to understand the true potentials and challenges.

Immutability: Many people consider everything inside a blockchain system to be immutable. However, that is only true for the blockchain or transaction data. But, states and smart contracts are continuously changing many things of a system [10].

Blockchain can store Large Scale Data: Although possible, it is impractical, costly, and inefficient. Such systems are more suitable for systems where minimum data is being stored or transacted. Due to blockchain being an add only data structure, there is more data proportional to the time. So, storage is largely increased [25].

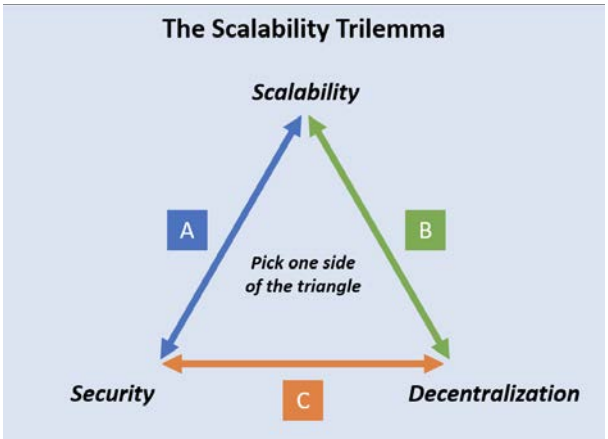


Figure 2.2: The Scalability Trilemma [11]

Anonymity: As per the Trilemma of Scalability from Figure 2.2, security, decentralization, and scalability, only two of these properties can be properly achieved in a system. Blockchain's security is not exactly anonymous, but pseudo anonymous. Identities are protected through public keys. However, if any identity is revealed publicly through

tracking the public transactions (which can be done through dusting attacks), then the identity is forever disclosed [30].

High power consumption: High electricity is only needed for PoW based systems, but not for PoS. However, PoS also causes the decentralized nature of Blockchain to be a pseudo decentralized system like the previous anonymity issue [19].

2.1.5 Security Overview:

Blockchain is a fairly recent yet booming technology with lots of future potential that is taking over the market. With the introduction of Bitcoin followed by Ethereum, the industry value has been rising ever since.

However, with new technology, comes new security risks. Although the technology itself is very secure, it has its weaknesses too that is already being exploited everyday. With development in the Quantum Computing sector, the risks to this technology's very existence due to the security issues are at question.

The report discusses various types of security vulnerabilities of blockchain networking systems and applications and what might be the future implications for it. Initially, the report discusses descriptions of blockchain technology and the misconceptions surrounding it followed by security issues like transaction censoring attack, 51% attack, smart contract exploitation, double-spending attack, selfish mining attack, Sybil attack, dusting attacks and DDoS attack. This part will also shortly explore the impact of Quantum Computing technology on existing blockchain technology.

Security Vulnerabilities:

Blockchain was initially developed to ensure a secured platform in a decentralized nature. However, as per the trilemma of security vs decentralization vs scalability which we already discussed, blockchain systems try to implement all of the parts to an extent bringing the modernized version of blockchain to a gray area in terms of both security and decentralization [30]. The technology has greatly scaled through ethereum and bitcoin, yet, it faces many threats due to its complex nature trying to balance many factors.

Repudiation: This refers to the dispute that arises among the sender and the recipient of a transaction when either of the parties deny sending or receiving signed and encrypted messages. Non-repudiation is when it is computationally feasible to solve such disputes, via the use of RSA, logging mechanisms in communications that are web-based, digital signatures and blockchain [20]. In blockchain technology specifically, once transactions are recorded, they are immutable, making transactions indisputable and undeniable, and providing a solution for repudiation.

51% Attack: When blockchain security issues are discussed, 51% attacks are always one of the most discussed vulnerabilities of such systems. If any attacking entity somehow manages to hold more than 50% of the network's hash rate (in PoW) or staking power

(in PoS), they can reorganize the blockchain and may also create a fork if they wish to [16].

Although very hard to implement due to the resources necessary, this has been a common attack in short scaled blockchain systems and also has been seen in Ethereum. In 2019, the Ethereum Classic network experienced a 51% attack, where a hacker reorganized the blockchain and double-spent over $1 million worth of ETC by renting high computational power to match the hash rate to become greater than 50% when Ethereum Classic utilized the PoW consensus protocol [16]. Similar attacks followed in Ethereum based platforms in 2020 and up to this day more and more such attacks are still seen with the rise of cryptocurrency around the world. But, the most common way this is utilized is through Double Spending attack as an attacker can spend the same cryptocurrency more than once by exploiting weaknesses in the consensus protocols or transaction malleability. This also enables attackers to commit transaction censoring attacks by choosing not to include some specific transactions in a block.

For PoW based attacks, it is recommended to have a very high hashrate so such attacks do not occur. PoS is more secure to these attacks as getting control of stakeholder accounts for voting and adding the block is much harder.

Sybil Attack: As per Platt and McBurney [29], Sybil attacks are when an attacker creates multiple fake nodes in a network to gain control of a network's operation by trying to skew the consensus or spamming the network with multiple requests. This is mainly done in low hash rate PoW systems or done through holding high stakes in a PoS system to hold maximum votes for asserting a block into a blockchain. This is similar to 51% attack and like that, it allows attackers for potential double spending and transaction censoring attacks.

The PoS system tries to prevent it by ensuring a very high value of stake to be held. However, PoW systems can be easily abused here and thus give a centralized priority towards the controlling attacker if the attacker has good processing power.

DDoS (Distributed Denial of Service) and DoS attacks: According to Greene and Johnstone [9], overwhelming the network with excessive requests or transactions, making it slow or unusable for legitimate users is a DDoS attack. Although blockchain networks are generally resilient, they can still suffer from certain types of DDoS attacks, especially if their design isn't robust against such overloads. Their research focuses on Ethereum nodes deployed on Microsoft's Azure Cloud and examines the impact of flood and bandwidth depletion attacks on the system. Through various experiments, the findings reveal that Ethereum Transaction Servers are vulnerable to both types of attacks, whereas Ethereum Mining Servers demonstrate resilience against bandwidth depletion attacks. In 2016, a DDoS attack on Ethereum overwhelmed the server by spamming low-cost, resource intensive transactions.

Moreover, the study highlights that in blockchain systems, particularly private consortia, DoS attacks can disrupt availability, even if the blockchain is generally resilient to such attacks in public settings. The transaction servers (handling front-end interactions) are especially vulnerable, unlike the decentralized mining nodes.

2.2 Review of Existing Research

Xie et al. [27] explore different existing blockchain consensus algorithms by delving into how they work and what pros and cons they have over each other, and overall. It is important to understand how consensus works and what features would make consensus validation the most efficient because consensus is essentially the backbone of blockchain technology. This survey provides an in depth look into the most recent innovations in consensus mechanisms and identifies potential challenges or shortcomings that new algorithms such as Proof of Physical Activity (PoPA) could solve. Our research topic introduces a novel consensus algorithm that makes use of physical work as proof for block validation. This approach addresses many issues that arise while using traditional mechanisms, especially ones pertaining to energy efficiency, decentralization and scalability.
The paper first dive into five different categories of consensus mechanisms: Proving-based, Voting-based, Alternating-based, VRF-based (Verifiable Random Function-based) and Hybrid-based. Proving-based mechanisms include Proof of Work (PoW) and Proof of Stake (PoS) algorithms. PoW is the most recognizable and traditional consensus algorithm and it ensures decentralization. However, it is prone to 51% attacks and consumes a large amount of energy, which is further increased by the competition between nodes to become a miner. The inefficiency of PoW called for an algorithm which is less energy-intensive. This is where PoS was introduced. It is much more energy efficient but it is vulnerable to centralization because it requires users to already own a certain amount of currency in order to participate as a validator. Our proposed mechanism can address both of these issues by redirecting energy consumption to purposeful and tangible physical work, something practically anyone can do for free. Alternating-based models include Delegated Proof of Stake (DPoS), which employs a voting system to choose a validator. Like PoS, it is efficient but is susceptible to centralization. As the name suggests, Hybrid models combine the stronger aspects of different algorithms. For example, Decred is composed of a combination PoW and PoS, which makes it secure and decentralized.
Our proposed PoPA algorithm could similarly incorporate elements from other mechanisms to become the best version of itself. The authors explain that blockchain systems typically have to compromise between scalability, decentralization and security. By determining blockchain consensus using physical work, PoPA can solve this issue by enhancing energy efficiency while simultaneously not trading-off decentralization. Security can be ensured by using a permissioned blockchain model that only authorized users can access, as fitness or health data is very sensitive. In recent years, more state-of-the-art technology is being explored to further solidify blockchain consensus protocols. For instance, the use of VRF-based and quantum-resistant algorithms in consensus provide increased security, and incentives for honest participation and penalties for malicious behavior ensure all users follow protocol. Furthermore, there is a growing interest regarding real-world activities that can be utilized as consensus inputs shvhs as Proof of Driving and Proof of Contribution. This perfectly aligns with our proposed algorithms, which also seeks to use real-world activity such as physiological data from IoT-enabled fitness devices as the major consensus factor. The authors provided a comprehensive analysis of existing consensus mechanisms and a detailed overview of their performance metrics by evaluating energy consumption and efficiency of each mechanism. However, their discussion surrounding consensus protocols that leverage human activities is limited and there is no insight regarding incorporating physical proof with blockchain. Despite its limitations, the paper plays an important part in understanding popular consensus algorithms and

the challenges they present. The paper's critique of existing protocols highlights the need for new protocols that address these challenges.

Blockchain technology has been revolutionary in the financial industry because of its core features which include decentralization, transparency, security and immutability. However, as the technology is becoming more and more prevalent, so is its use in various other industries, such as healthcare and fitness. Its use has been most relevant in IoT-enabled fitness systems, which employ the use of wearable devices equipped with sensors that can help generate huge amounts of data. As this data contains personal health-related information, it requires secure processing and a secure storage system. Blockchain, with its core features ensuring privacy, security and integrity of data and its model as a distributed ledger can be a promising solution to processing and storing such sensitive data. In [21], Jamil et al. introduce the concept of a secure IoT based fitness framework that utilizes blockchain technology and integrates ML (machine learning) to provide users with fitness and health recommendations that are specifically tailored to them. The framework consists mainly of two modules: a blockchain-based IoT network for secure data collection and storage, and a smart contract based inference engine for real-time data analysis. The inference engine uses data from Internet of Things (IoT) devices including, for instance, your smartwatch, to create customized workout plans and diet suggestions. The authors explain that, utilizing a permissioned blockchain model, the system will only permit authorized individuals to view the data. In addition to the above, the authors discuss current non-permissioned consensus algorithms; namely, Proof of Work (PoW), Proof of Stake (PoS), and Delegated Proof of Stake (DPoS).
Consensus algorithms have a direct impact on the integrity of the blockchain, yet, all of the above algorithms have their own sets of limitations. One limitation of PoW is its significant resource utilization in terms of energy. Likewise, due to the fact that they are open-source blockchains, there are significant levels of latency, and therefore, low transaction throughput in open-source blockchains such as Ethereum. Finally, the authors note that prior mechanisms rarely linked the actual validation of consensus to real world, tangible activities; hence limiting their utility. According to research, past applications of blockchain-based fitness platforms include Sweatcoin and Run2Play. These platforms use models like "proof-of-fitness" to reward users based on physical activity, which are tracked by wearable mobile devices like smartwatches. These systems issue tokens for physical activity (like the number of steps walked in a single day) which are tracked by their smartwatches, thus incentivizing healthy behaviors. However, as promising as these platforms are, they are prone to suffering from limited scalability, high computational costs, and a dependency on cryptocurrency based incentives, and ultimately, they do not fully utilize blockchain's potential for real-time data analysis and decision-making capabilities. To address the above mentioned limitations, the authors developed a novel blockchain architecture, which was designed to optimize performance for IoT fitness applications. The proposed architecture utilizes Hyperledger Fabric, which enables a lightweight and scalable blockchain network that can process an increasing number of transactions per second without a decrease in performance. This proposed blockchain architecture is a permissioned blockchain, which increases user data security. It provides easy integration with IoT devices and enables real-time recommendations via smart contracts. The authors' "Proof of Physical Activity" represents a conceptual extension of this research, as it connects consensus validation of a blockchain to physical activity. The purpose of our project is to utilize physical activity measured by IoT fitness devices, such as steps

taken or calories burned, to determine consensus. Unlike traditional mining methods, where consensus is determined by computational power, or distributed in a manner that is unfair, we propose to determine consensus by something tangible, which is physical activity. The proposed framework offers a reference point for integrating blockchain with IoT fitness devices. By combining high-level machine learning models with the proposed framework, our system can verify users' physical activity, and predict their customized workout and dietary plans. Also, because our system utilizes real-time inference engines, users are receiving continuous and real-time values. While it is true that existing blockchain-based fitness systems show great potential for integrating technology with wellness, they fail to address scalability issues and energy efficiency. The proposed Proof of Physical Activity algorithm aims to overcome these limitations and more, by directly linking consensus validation to real-word physical activities and incentivising a healthier lifestyle for its users.

PoX, as proposed by Shoker [7], is an innovative approach to fix the energy-inefficient blockchain technology and also addresses environmental concerns. It shifts the focus from computational power to solving scientifically meaningful exercises, hence providing a sustainable alternative to traditional Proof-of-Work. The workflow of PoX involves the employer-an example can be any scientific research entity-delegating a computationally intensive problem, termed "eXercise", to miners. This problem is securely stored, and its hash digest and access details are made available on a public platform (XBoard). To ensure fairness, the exercise's digest is randomized to prevent collusion among miners. The employer also makes a blockchain deposit to guarantee the availability of the exercise throughout the process. Miners then bid randomly to solve the exercise, committing to a deposit to obtain access rights. Miners, having solved the problem, store the solution and create a hash of the publication for verifiers to check. Verification in PoX is probabilistic. Verifiers cross-check parts of the solution collectively to ensure that it is correct. The verifiers are also committed to storing the data they verified to enable auditing later on. Sufficient confirmation from the number of validators will confirm the solution, and the PoX is added to the blockchain, which then is linked with the block header digest to prevent reuse in multiple blocks. The PoX, after being validated, enhances security and accountability but may not be valid if the stored data gets lost when the storage time expires. PoX provides the added benefits of energy efficiency, usefulness of solving real problems, and high security through randomization and probabilistic verification. Some other challenges it faces are regarding data integrity on storage and scalability for mass applications.

Bandara et al. proposed a green blockchain consensus protocol called Proof-of-Pedal. It targets the promotion of human physical activity, such as cycling, to create an incentive by considering huge energy consumption problems of traditional PoW protocols [28]. Making use of burned calories and carbon footprint reduction as a stake inside the PedalChain blockchain, PoP replaces electricity-driven computational work with human energy. In the process, it incentivizes users-participants and block miners-with a cryptocurrency called Pedalcoin towards a sustainable outcome that also improves physical health. Suitable for both public and permissioned blockchain systems, the protocol also introduces a deterministic asynchronous consensus mechanism in permissioned systems because of the challenges brought by nondeterminism. To operationalize PoP, the mobile application Self-Sovereign Identity-enabled ClimGuards was built for measuring users'

caloric expenditure and carbon emissions reduced while cycling. The ClimGuards can be integrated into PedalChain, which is well-known as the "Greenchain" for ensuring data integrity while being completely transparent. The ClimGuards do not only address the environmental footprint of blockchain technology; it encourages cycling as an active way to improve overall health and as a means of transportation. As such, the ClimGuards are aligned with both sustainability and public health goals. PoP combines determinism, user incentives and real-time monitoring to offer a unique solution to the energy consumption of consensus-based blockchain protocols, which is superior to alternative energy-efficient mechanisms, such as Proof-of-Useful-Work and Proof-of-Exercise.

The rapid development of Internet of Things (IoT) applications has led to a large number of interconnected devices, which generates a significant amount of transaction data requiring robust security and privacy mechanisms [31]. However, traditional centralized authority-based security mechanisms have been shown to be inadequate in terms of their performance capabilities due to bottlenecked performance and the presence of single point failures. While blockchain technologies have contributed much to gaining prominence in the decentralized solutioning of securing IoT networks, this is probably due to their characteristics of transparency and immutability. Smart surveillance, biometric authentication, and access control are some ideas which are being explored using blockchain security frameworks. On the other side, integrating blockchain with IoT systems presents challenges in designing scalable and lightweight consensus mechanisms. Existing consensus protocols, such as Proof-of-Work, are effective against Sybil attacks but computationally intensive, hence energy-consuming and not scalable. Mechanisms based on Byzantine Fault Tolerance, such as Practical Byzantine Fault Tolerance, provide higher throughput but suffer from network scalability issues due to reliance on identity authentication. Abbasi et al. propose Proof-of-Resource (PoR), a novel consensus mechanism tailored for resource-constrained IoT environments. Unlike PoW, PoR inherently uses computational resources in IoT devices, memory, and energy for consensus with the least possible energy overhead while maintaining the security and reliability of the transactions. Having the least computation load in IoT, it could turn out highly useful and apply even in the scenarios concerning the decentralized security model for IoT. The research objectives involve devising a scheme with a low power-consuming consensus mechanism, study of performance evaluation, and evaluation for its respective security features. The approach is aligned with the increasing demand for sustainable and scalable blockchain solutions that can enable seamless IoT integration, while also addressing security and resource constraints. Thus, this study represents an important contribution to the advancement of decentralized security frameworks and will help bridge the gap between blockchain and IoT applications to support a strong, resilient, and efficient digital system.

In their study, Yin et al. [34] conducted a thorough evaluation of the body of literature on HAR (Human Activity Recognition) as well as its applicability to mobile devices. With the rapid growth of our blockchain technologies, there have been many ways to accomplish consensus. The two most popular methods are: Proof of Work (PoW), and Proof of Stake (PoS). These consensus mechanisms require so much processing power to operate effectively, that they are not suitable for devices with limited processing capability, such as smartphones and smartwatches. The consensus mechanism Proof of Physical Activity (PoPA) has provided the basis for an innovative method by combining Human Activity Recognition (HAR) and blockchain technology to overcome these barriers. The focus

of the work was to highlight the important advancements in the area of HAR methods, particularly to take advantage of the embedded sensors in smartphones and smartwatches to determine user activity. Various sensors such as accelerometer, gyroscope, heart rate monitors are being used in HAR to detect activities like walking, running and in some cases minute activities such as finger gestures can also be detected. The sensors are capable of generating accurate data which can infer human physiological activity with great precision. The integration of such data with consensus algorithms can be used as a foundation for validating physical activity in the PoPA consensus algorithm. The paper addresses several challenges of HAR: Mobile devices experience environmental interference such as noise, unexpected movements of the device, making it hard to detect the exact human activity. The limited computational power and finite life of the battery present in the devices restrict the devices to process HAR data extensively locally, which is why lightweight algorithms are very important. Also the recognition of activities that have similar sensory patterns remain a challenge (such as walking and climbing stairs). The challenges are directly related to the technical hurdles PoPA faces with accuracy and energy-efficiency prioritized to validate consensus participation. The article discusses several forms of activity recognition as well, for instance, data-driven techniques (machine learning), and knowledge-driven techniques that may be able to analyze data generated by sensors to determine what an individual is doing. Machine learning models such as decision trees and deep neural networks were found to be highly accurate at identifying complex activities, and thus, can be used in the design of PoPA systems, where the mechanism to verify activity must be both robust and scalable. The study also heavily emphasizes the combination of multimodal sensor data. In order to function within blockchain systems, that require precise measurement of activity, many HAR systems must filter through sensor data to remove noise, and extract useful data. Use of multiple sensors (such as accelerometers and heart rate monitors) provide additional reliability to PoPA by providing assurance that the collected data accurately reflects human physical activity. The findings from this study support that advances in HAR can significantly aid in the advancement of developing smartphone based blockchain systems. Through application of existing HAR methodology, PoPA can assure accuracy by utilizing advanced data preprocessing with multiple sensors to confirm activity, utilizing light weight models that will not overburden smartphones when participating in consensus, and integration of IoT devices (i.e., smart watches) to further enhance user engagement and improve mining efficiency on the network.

Lopez-Barreiro et al. [23] conducted a comprehensive review of the possible uses of blockchain tech in physical fitness, healthcare, and sports. It was informative regarding how the decentralized nature of blockchain could be applied to these industries by leveraging its immutable characteristics. Use of blockchain also offers a secure platform for managing physiological information collected from wearable devices (such as smartwatches) as these information are considered to be very sensitive. The study also talks about several important applications such as introducing platforms that would incentivize physical activity through cryptocurrency rewards, storing health records electronically and securely and active aging, through which people can maintain their health and well-being regardless of their age. These applications align with the vision of PoPA, where participants of the network would contribute verified physical activity data to participate in mining and transacting. However, the paper also mentions challenges that are also related to the PoPA algorithm. Data spoofing risk leads at the top, where users may sub-

mit false information and fake data to gain rewards. Another risk is the privacy concern of individuals as physiological data will be stored on the blockchain. Their work also puts emphasis on the existing systems offered by the advanced features of blockchain, namely smart contracts and oracles. These features could be used heavily to integrate real world activity data to blockchain and also for automated reward distribution. PoPA would also utilize these crucial aspects to ensure that the system is accurate, scalable and fair to every participant of the network. At the same time, the traditional blockchain consensus mechanisms are considered as barriers due to their high complexity, computational costs and energy consumption. This paves the way for lightweight consensus algorithms like PoPA which is designed to offer a solution to those challenges with the use of lightweight consensus methods. PoPA has the capability to utilize blockchain's strengths in areas like maintaining immutability, data transparency, and security while avoiding the energy-intensive operations that are typically seen in traditional consensus mechanisms. The paper emphasizes the high potential that blockchain holds for revolutionizing not just healthcare but also fitness and wellness activities. By integrating gamification and predictive analytics, users are not only engaged more effectively but are also motivated to achieve healthier outcomes through consistent participation. PoPA takes advantage of these insights to address data integrity concerns, privacy risks, energy efficiency, and overall system accuracy, therefore leveraging blockchain's unique strengths and creating a more accessible, secure, and user-friendly consensus framework which encourages both physical activity participation and network security.

The authors Y. Sang and L. Wang explore how blockchain technology and IoT can be utilized together for real-time data analysis and collection of fitness data [24]. While the authors' primary focus was on the application of these technologies to improve health results among college students, they did use IoT devices along with blockchain technology for data collection and safe storage of data. They have established a foundation for the collection and analysis of physical fitness metrics including lung capacity, standing long jump, and running performance. Sang and Wang discuss the advantages of using blockchain, specifically its immutability and decentralization, and how these advantages support their research. Additionally, they discuss the use of cryptographic techniques, including public key cryptography, elliptical curve algorithms, and hash functions to safely store and transmit data. Their results indicate that their proposed framework works well as they were able to find significant differences between males and females during different testing sessions. The findings help participants to better understand their current fitness levels and develop personalized strategies to improve their fitness over time. This research supports PoPA's vision by proving IoT and blockchain technologies can be successfully combined to securely process and protect physical activity data. Their findings identify the scalability concern and suggest additional research to support this area within similar educational settings. It offers real-world recommendations on developing data acquisition systems and blockchain solutions to keep data safe and accurate, which is very critical for PoPA implementation. These findings suggest that similar approaches could be employed to protect fitness records in PoPA while maintaining transparency through blockchain. The effective integration of the system would promote health among the individuals while incentivizing them along the way.

Lopez-Barreiro et al. [32] investigate blockchain integration with gamification models to deliver healthy habits for individuals. The study confirms blockchain's power for improv-

ing the transparency and reliability along with traceability controls in promoting habits that improve physical and mental health through incentives. The research work experimented through a blockchain-based decentralized application (dApp) prototype that implemented blockchain smart contracts to conduct automated operations while providing users with transparent control of "challenge" monitoring and reward distribution. The Sepolia testnet allowed the research team to come up with a real-world implementation of the blockchain-based gamification system that received excellent usability ratings from participants. The gamified model functioned through leaderboards along with progress monitoring and automatic cryptocurrency rewards (ETH on the Sepolia testnet) which maintained system data accuracy and delivered personalized motivation to the participants. These research results can be applied to make the PoPA model useful, due to the fact that it shows how blockchain technology can provide authentication of an individual's participation in physical activities, and protect any related data. Blockchain technology is used to track the actions and pay out rewards through a smart contract, so the participant has a direct incentive with Proof of Physical Activity protocols by validating a participant's physical activity while offering incentives for steady engagement. The paper also discusses the barriers which include maintaining user retention along with IoT implementation challenges, and finding appropriate GDPR-level compliance solutions for successful deployment. In addition, the authors point out that the unique benefits in promoting fitness among the masses yet studies show current implementations fall short in unlocking blockchain's capabilities such as interoperability and real-time data validation. This research provides PoPA with foundational insights to build an active feedback framework which combines custom challenge models and automated reward systems to deliver the motivation for physical exercise without compromising transparency, security and data integrity.

We have also looked at other smartphone based consensus algorithms to get a glimpse of how other researchers approached this topic. Kim et al. proposed the "Proof of Phone" (PoP) consensus mechanism which introduces a novel blockchain platform based on the HELO (High Entry cost and Low Operation cost) concept, aimed at addressing the inefficiencies of conventional mechanisms like Proof of Work (PoW) and Proof of Stake (PoS) [14]. Similar to how we are thinking of approaching the topic, we can take inspiration on how the mechanisms can be optimized for a small processing unit like our phone can be seen here. For "Proof of Phone", by utilizing smartphones integrated with Authenticated Mining Units (AMUs) to impose high entry costs while lowering operational expenses, the platform ensures cost-effectiveness and maintains network integrity. The PoP method uses existing hardware to minimize cost to 1.8%-98.2% compared to traditional blockchains; and minimizes miner competition by eliminating "mining" for all but a few blocks at most. Therefore, this novel approach fits perfectly into our research goals since PoP has lower operating costs and smartphone compatibility than other blockchain-based decentralized identity systems, and therefore offers lower barriers to entry for the large-scale use of identity verification systems. However, the research gap may be viewed as the fact that the authors do not present any security mechanisms they will implement to protect against attacks, nor have they demonstrated the feasibility of their proposed mining protocol in practice.

In addition, we will be using smart watch data to further improve the security of the blockchain identity verification process by providing an additional source of dynamic in-

put (e.g., accelerometer and gyroscope readings) from mobile phones and smart watches to verify identities. This approach is consistent with the work done by Rawat et al., where data from various wearable devices including smart watches and health apps was analyzed to estimate stress and physical activity levels [33]. Their results indicated that the heart rate and stress condition varied greatly throughout the day and could be used by our system to dynamically mine in response to real world changes. By adding these dynamic inputs into our blockchain architecture, we will be able to build a decentralized identity system that can adapt to the real world to improve its practical applicability.

Additionally, as Chowdhury et al. [13] has explored how trust can be achieved in smart wearables using blockchain technology, we see the presence of a trust model built upon blockchain technology, which ensures the integrity of data collected from wearable devices. This model addresses the issues of data privacy, security, and trustworthiness in environments where individuals share and use their personal health information. Therefore, this trust model, along with our use of smartwatches, will assist us in developing PoPA by demonstrating some of the ways we can ensure the authenticity and integrity of the physiological data we collect and use for the mining process. It utilizes mechanisms like Reputation-Based Trust Model where users gain more credibility over time, Data Integrity Verification through maintaining a blockchain based ledger, Anomaly Detection for Fraud Prevention and much more. By adopting similar trust mechanisms, PoPA can enhance the reliability of physical activity data, ensuring that the consensus algorithm accurately reflects genuine user activity.

Again, if we take a look at another consensus algorithm Proof of Burn, we can analyze what type of security measures we might need to take to ensure a secure system for all users [22]. The Proof of Burn consensus was implemented in the system alongside PoS in a hybrid manner and the whole system was simulated in SimBlock, a blockchain network simulator. We also plan to implement a similar approach based on further requirement analysis and simulate the environment on a large scale through the network simulators. This will allow usu to test the environment for any security issues and resolve them accordingly based on our non functional requirement analysis. Similar to the discussed paper, we will take similar approaches to test and ensure security for our PoPA consensus algorithm for selfish mining, sybil attacks, and 51% attack.

A strong relationship between equitably measured physical activity and long-term health outcomes can be seen in recent observed evidence. Specifically, in a large prospective cohort study, a strong inverse relationship was found between volume of daily steps and overall mortality among 16,741 older women whose steps counts were recorded using accelerometers [15]. As step counts increased, mortality risks started declining significantly. Even moderate activity like 4400 steps a day showed great benefits, with results plateauing at around 7500 steps a day. However, it was noted that once step volume was accounted for, walking intensity had insignificant association with mortality, highlighting that overall activity is far more important than activity intensity for health benefits. This discovery challenges conventional fitness standards such as the daily 10,000 steps benchmark. It validates achievable daily activity levels by illustrating that they can produce beneficial physiological outcomes. Such evidence is particularly useful for systems that use metrics that were derived from physical activity like step count, such as PoPA. It supports the idea that verified step volume is a good indicator of physical activity

and thus, can act as a reliable consensus input. Furthermore, it is useful for defining participation thresholds used to become a validator. Finally, since the focus is not on step intensity, these metrics are not disadvantageous to participants who cannot perform intensive physical activity.
Through recent research, the usage of activity metrics derived from wearables has been expanded beyond simple step counts by the integration of physiological response with the volume of movement [35]. Daily heart rate per step (DHRPS) is a composite metric that is calculated by dividing the average daily heart rate by steps per day. Here, heart rate and step counts are derived from longitudinal Fitbit data. It was observed that even after adjusting for step count and heart rate individually, higher DHRPS values were consistently associated with higher rates of multiple cardiovascular diseases, including type 2 diabetes, hypertension, heart failure, stroke and coronary atherosclerosis. Notably, when compared to just steps per day or heart rate independently, DHRPS showed a stronger relationship with cardiovascular fitness and disease risk. Further, this metric was also more strongly correlated with maximum metabolic equivalents measured during exercise stress testing. Overall, these findings suggest that values recorded from wearables which capture both activity volume and physiological response can provide medically relevant insights. For PoPA, this study reinforces the use of wearable derived inputs as meaningful indicators of real world physical effort.

2.3 Summary of Key Findings

The existing research research is foundational and contextual for the PoPA consensus protocol.

First, the foundational work done on Proof-of-Work (PoW), and Proof-of-Stake (PoS) consensus mechanisms – in particular how they are implemented in Bitcoin and Ethereum – shows the trade-offs between the energy consumption needed for computational security and the need for an efficient use of energy. These research results support why PoPA was developed to redirect this energy usage from being used in mining to being used through positive health behaviors - using these behaviors as resources to support consensus instead of brute computational power.

Second, research on hybrid and reputation based consensus algorithms supports PoPA's dual weighted validator model. The research suggests that using historical consistency (reputation) can improve resilience against manipulation and short term Sybil attacks. While there is no prior research on using activity as a measure of reputation; the PoPA uses consistent activity to assign a higher weighting to validators - supporting reputation with consistent healthy behaviors.

Third, research on collecting and tracking health/fitness related data through wearable devices show the technical feasibility and ubiquity of real time authentication of physical activity data - providing evidence for the technical feasibility of using metrics such as heart rate, steps taken, and intensity of movement as secure and non-falsifiable inputs to the consensus algorithm.

Fourth, research on developing incentive structures to promote healthy behaviors through

smart contracts and DApps provide evidence of user acceptance and scalability in real world health contexts. Although many applications and middleware have been developed to support healthy behavior through incentives - none of them include a health centric validation at the consensus level. PoPA addresses this gap by including a health centric validation at the consensus level.

Lastly, research on blockchain for public goods and decentralized identities demonstrates the social relevance of PoPA. Research advocates for the ability of blockchain to be used for more than just financial transactions - particularly in promoting equitable access to digital services, digital trust, and non-monetary incentives - all of which align with the goals of PoPA to create a fair, health driven consensus layer that promotes community well-being above pure capital accumulation.

2.3.1 Positioning Relative to Existing Literature

Table 2.1 summarizes the key difference in scope between our work and the existing literature. Prior work primarily handles physical activity at the application or smart-contract layer, whereas our approach integrates activity validation directly into the blockchain consensus protocol.

Table 2.1: High-level comparison between existing literature and PoPA

Aspect	**Existing Literature**	**PoPA (Our Work)**
Use of Physical Activity	Physical activity is recorded or rewarded at the application or smart-contract level	Physical activity is used directly within the consensus protocol
Validation of Activity Data	Relies on application logic or trusted smart contracts, often assuming honest input	Activity submissions are validated by the protocol
Role of Smart Contracts	Smart contracts are the primary mechanism for enforcing logic and rewards	Smart contracts are not required for consensus; validation occurs at the protocol level

Chapter 3

Requirements, Impacts and Constraints

This chapter provides a comprehensive overview of the system specifications, threat modeling, environmental and social impacts, ethical considerations, economic analysis, project management plan, and risk management strategies for the Proof of Physical Activity (PoPA) consensus algorithm. Each section delves into critical aspects that ensure the robustness, sustainability, and ethical integrity of the PoPA system.

3.1 Threat Modeling & Analysis

Initially, we identified potential threats and vulnerabilities that could compromise the security and integrity of the theorized PoPA system. For each identified threat, we analyzed its potential impact on the system and developed mitigation strategies to address these vulnerabilities using the threat model called STRIDE. The STRIDE threat model [26], is a security threat classification framework that categorizes potential risks in software systems in the following categories. Not all STRIDE categories are relevant to PoPA. Elevation of Privilege is excluded since the protocol does not rely on role-based access control:

3.1.1 Spoofing

In a spoofing attack, the attacker impersonates another identity in order to gain unauthorized access. Usually, they use the alternate identity to attain confidential information or get into a secure portal. The attack can be as straightforward as using a fake email to trick the recipient into giving out sensitive information. [26].

- **Sybil Attacks:** In a Sybil attack, the attacker creates multiple fake nodes and tries to manipulate the consensus, or to spam the network with multiple requests. This may grant the attacker majority influence over the system [29].

3.1.2 Tampering

The attacker or hacker changing the data in a system through removal, modification, or manipulation, with the goal of harming the system. This type of attack targets the integrity of the data in a network [26].

- **Data Tampering:** Physical activity data can be altered during transmission to misrepresent actual performance.
- **Fake Physical Activity:** Users might stimulate their IoT devices to manipulate their physical activity data. This might trick their IoT devices to produce false values. For example, a user can shake their device to produce a higher value for steps taken on a particular day.

3.1.3 Repudiation

Repudiation occurs when a malicious node attacks a system and does not take accountability for its actions. In most cases, the system is unable to reliably pinpoint the perpetrator of the attack [26].

- **Transaction Denial:** Attackers may claim that they never performed a fraudulent transaction.
- **Data Non-repudiation Issues:** Users may deny generating a specific physical activity record used within the system.

3.1.4 Information Disclosure

This security issue deals with the unauthorized publishing of private information [26].

- **Privacy Breaches:** Sensitive health and fitness data is leaked or sold.
- **Sensor Data Linkage and Re-identification:** Data from available external datasets in social media, or location metadata can be cross-referenced with anonymized physiological data (e.g., heart rate, GPS) and used to re-identify users [20].

3.1.5 Denial of Service

DoS (Denial of Service) attacks aim to overpopulate a system with excessive traffic, causing it to cease functioning at its normal capacity. A system being inactive can be very costly for the participants in it and can result in great losses [26].

- **Resource Exhaustion:** In resource exhaustion attacks, attackers crash a computer program by taking advantage of its vulnerabilities [1]. Attackers can spam the system with invalid transactions to overwhelm validation nodes, and cause it to become unresponsive.

3.2 Design Specifications

The Proof of Physical Activity (PoPA) consensus algorithm introduces a sustainable alternative to traditional mining systems. It defines the system architecture, validator design, core algorithms, and protocol logic for secure block generation, reward distribution, and validator fairness in a human-centric blockchain ecosystem. So, as per our threat modeling and analysis, we have defined the following design specifications for the PoPA system.

3.2.1 Key Design Specifications

From the literature review, several specifications were identified as essential for a successful implementation of PoPA:

1. **Integration with IoT Devices:** Seamless integration with smartwatches, smartphones, and similar IoT devices to collect accurate physical activity data.
2. **Data Verification Mechanisms:** Robust validation of collected activity data to detect and prevent fraudulent or spoofed inputs.
3. **Consensus Protocol:** A protocol that validates blocks based on verified physical activity data rather than computational work or monetary stake.
4. **Incentivization Model:** A reward model that aligns incentives with user activity contributions and promotes healthy behavior.
5. **Security Measures:** Comprehensive protections against threats and vulnerabilities across the data, network, and device layers.

3.2.2 Functional Requirements based on Requirements Analysis

Requirement analysis is a critical aspect of designing a system as it defines the scope, objectives, functionalities that the system should have. It is imperative to analyze the functional, security and performance requirements so that the system works in the proper way. Functional requirements work as an outline that defines the core capabilities the system must have to fulfil its purpose. This speeds up the development process and is crucial for user interaction. From the design specifications mentioned above, we have identified the following functional requirements for the PoPA system:

- **User Access:** Users must be able to securely access their wallet through their private and public keys.
- **Physical Activity Data Collection:** The system must collect user activity data from IoT devices (e.g., smartwatches and smartphones) through sensors.
- **Activity Verification Mechanisms:** The system must verify the authenticity of physical activity data to prevent cheating by using multiple validation methods such as similar data submissions from different devices of a single owner and comparing them.
- **Blockchain Transaction Generation:** Verified physical activity data must be hashed and stored as transactions on the blockchain.
- **Consensus Mechanism Execution:** The PoPA algorithm must validate and approve blocks based on user activity contributions instead of computational work.
- **Multi-Device Compatibility:** The system must support different devices, including different versions of Android in various devices, wearables, etc.
- **Real-Time Activity Monitoring:** The system should allow users to track their activity status in real-time and notify them when their activity meets consensus requirements.

3.2.3 Security Requirements

Security requirements define the necessary measures to protect the system and its data from threats, ensuring integrity, confidentiality of the data from attacks.

- **Data Encryption for Privacy:** Extracted physical variables will be hashed for ensuring security of data. This will mitigate Information Disclosure and Repudiation.
- **Security Attack Preventation:** Sybil, DDoS. 51%, selfish mining attacks etc. needs to be prevented through additional mechanisms like cryptographic signatures, verification of IoT data through third party application and more. This will prevent Spoofing, Tampering, and Denial of Service.
- **Reputation system that punishes bad behavior:** Any suspicious activity of nodes acting in the system will be blocked from transactions to prevent bad behavior. This will mitigate Spoofing, Repudiation and Tampering.
- **Ensuring IoT Security:** Ensure IoT is not prone to network attacks and build the consensus accordingly. This will mitigate Spoofing, Tampering and Elevation of Privilege.

3.2.4 Performance Requirements

Performance requirements address the expected responsiveness and efficiency in terms of speed, scalability and resource utilization to ensure the system performs smoothly.

- The system should verify and validate physical activity data with low latency to ensure a smooth user experience.
- System should be scalable and support many IoT devices without performance issues
- The system should handle Fault Tolerance and thus continue functioning effectively even when some nodes or devices fail or disconnect.
- The consensus mechanism should not require high computational power, allowing IoT devices to participate efficiently.
- Ideally, storage should be optimized too.

3.3 Environmental and Social Impact

Health Chain's architecture has created a new paradigm for "Resource Regeneration" in Consensus Models, as opposed to the "Resource Extraction" found with traditional consensus models. In doing so, it provides specific ecological and social advantages.

3.3.1 Environmental Advantages

Unlike traditional "Proof of Work" blockchains, which consume over a trillion watts of electrical energy per year and contribute to the massive amount of electronic waste produced by obsolete Application Specific Integrated Circuit (ASIC) mining units, the PoPA algorithm is a "Green Blockchain." Through the use of "Parasitic Computing," (i.e., utilizing the existing background processing power and sensor capabilities of smartphones that users already have and are charging), the system has achieved a near zero marginal carbon footprint. As such, the validation process is transformed from an industrial-scale, energy-consuming operation to a sustainable, utility-based background process; effectively decoupling blockchain security from electrical energy consumption.

3.3.2 Social Implications/Public Health

The project utilizes the principles of Behavioral Economics to create a global solution to the crisis of sedentary lifestyles. By creating a monetary incentive for physical exercise, the system establishes a "Merit-Based Economy" in which human effort is the scarce resource, not capital or hardware. By providing an additional motivation to meet the World Health Organization (WHO) activity guidelines, this gamification of physical activity can lead to reduced long-term societal costs associated with non-communicable diseases (such as cardiovascular disease and diabetes). Additionally, the system has a very low barrier to entry and therefore democratizes access to being a validator, enabling individuals from economically challenged communities to utilize their standard smartphone to act as validators, thereby contributing to both economic inclusion and improved physical well-being.

3.4 Ethical Issues

A novel challenge presented by a health-based consensus algorithm extends professional responsibilities in software development beyond standard ethics.

3.4.1 Fairness and Non-Discrimination

One of the most important ethics issues raised by this model is ableism; the "Activity Score" threshold assumed as an activity baseline for all users excludes users with disabilities and/or elderly users from participating in reward generation when the algorithms favor high-intensity activity. Professional obligations mandate the development of Adaptive Baselines in which users' specific physical limitations are normalized to their ability to exert energy in order to provide fairness and inclusiveness to the network.

3.4.2 Data Privacy and User Autonomy

Health data is private information. Although the data is hashed to verify its integrity, there will always be some form of exposure of physical characteristics (e.g., active hours, type of workout) through the validation of the hash. There exists a professional obligation to support Data Minimization; i.e., only the mathematical proof of work should be broadcasted to the network, while raw biometric data will remain encrypted on the

user's device. This will prevent the public ledger from being used as a surveillance tool to monitor user behavior.

3.5 Economic Analysis

The Health Chain (PoPA) Blockchain Ecosystem introduces a transformative economic model that redefines the cost structure and incentive mechanisms of blockchain consensus. By leveraging physical activity as a resource for block validation, PoPA presents a sustainable alternative to traditional Proof-of-Work (PoW) systems, which are characterized by high energy consumption and significant capital expenditures.

3.5.1 Resource Allocation and Efficiency

The Health Chain (PoPA) Blockchain Ecosystem has the potential to transform the economics of consensus through its elimination of the capital intensive barriers found within the traditional Proof-of-Work (PoW) systems. As opposed to Bitcoin, PoPA will leverage "parasitic computing," which utilizes the underutilized processing capabilities and sensors (accelerometers and gyroscopes) in a user's smartphone, instead of the expensive and energy intensive application-specific integrated circuits (ASICs) and large scale industrial electricity inputs used in Bitcoin mining. Therefore, the capital expenditures (CapEx) for network validators in the PoPA system will be effectively zero, as the software is operating off an existing consumer owned device. Additionally, the "stateless http" architecture of the system will minimize operational expenditures (OpEx) as it requires virtually no bandwidth (<50mb/month) and battery consumption when compared to the high maintenance cooling and electricity requirements needed for mining rigs.

3.5.2 Cost Benefit Estimation

The financial benefits of the PoPA blockchain model compared to traditional blockchain models, including reduced environmental impacts, demonstrate a clear advantage. While estimates indicate that the cost of securing a single block of Bitcoin is in the thousands of dollars of electricity, the "cost" of validating a block in PoPA is derived solely from the calories burned during the execution of the task, thereby creating a positive economic cycle for the reduction of long term healthcare costs associated with sedentary lifestyles. In addition to the economic benefit, the PoPA blockchain model introduces a "merit-based economy" where the distribution of rewards is directly related to health outcomes as opposed to capital staking. This allows for a direct ROI for third party stakeholders, such as insurance providers, to fund block rewards and have those rewards correlated with measurable reductions in the risk of cardiovascular disease among their users.

3.6 Project Management Plan

The research will follow a structured methodology divided into three main phases:

- **Phase 1 (P1):** Literature Review and Theoretical Framework Development. This phase will involve an extensive review of existing literature on blockchain consensus mechanisms, particularly those utilizing physical activity data. The goal is to establish a strong theoretical foundation for the proposed PoPA mechanism.

- **Phase 2 (P2):** Design and Development of PoPA. In this phase, the PoPA consensus mechanism will be designed, focusing on integrating smartwatch-derived physical activity data into the blockchain validation process. This includes developing fraud prevention mechanisms to ensure data integrity and designing an incentivization model to reward users for their physical activity.
- **Phase 3 (Defense):** Implementation and Evaluation. The final phase will involve implementing the PoPA mechanism in a simulated environment or testbed. The system will be evaluated based on its performance, security, and effectiveness in promoting physical activity compared to other consensus mechanisms.

3.7 Risk Management

We have assessed all of the potential technical, operational, and regulatory risks that exist in the Health Chain system, including how those risks can be mitigated through appropriate architectural solutions to provide the long-term stability required to protect the integrity of the Health Chain system. The most significant technical vulnerabilities are related to Sybil and replay attacks which allow attackers to generate fake data in an effort to obtain rewards. These threats are reduced by implementing two primary mechanisms. First, a strict similarity detection mechanism is used to reject any duplicate payloads that may be sent into the system, thereby preventing any attacker from sending in duplicate payments for rewards. Second, a deterministic method is used to select validators so as to prevent the occurrence of a blockchain fork , which would potentially result in conflicting views of the state of the blockchain among validators and subsequently the participants in the Health Chain system. Operationally, the Health Chain system is dependent upon third party APIs (e.g., Google Health Connect), such as APIs that provide access to medical records, etc. As such, there exists a risk of either having no access to certain data due to the unavailability of APIs or API schemas changing at some point in time. A number of strategies were used to mitigate these risks. For example, a modular data layer was designed into the system, so that the health chain system could seamlessly switch between different data sources if needed. In addition, the health chain system utilizes asynchronous fetching with a "grace period", to enable the health chain system to avoid delays in the communication with other nodes in the network caused by network latency, while maintaining consensus within the system. Finally, to address both privacy and regulatory issues associated with the use of biometric data in the Health Chain system, the Health Chain system has been designed to operate in accordance with data minimization principles, i.e., the system processes raw sensor data locally, and therefore, although the proof of physical work is verifiable on the public ledger, sensitive personal health information is maintained in a private and encrypted format, consistent with applicable data protection standards.

Chapter 4

Proposed Methodology

Traditional blockchain consensus mechanisms such as Proof of Work (PoW) and Proof of Stake (PoS) are critical components of the integrity of blockchain technology, but both can present major drawbacks. For example, Proof of Work needs a very high level of energy consumption while Proof of Stake might create centralization issues. In the last few years there has been rapid development of portable IoT wearables like smart watches providing the perfect opportunity to explore new sustainable methods that allow all participants to participate in a blockchain network's consensus.
This research introduces a novel consensus algorithm, *Proof of Physical Activity (PoPA)*, which utilizes a user's real world physiological data (heart rate, step count, and active minutes) to propose blocks and reward users. It uses an *Activity Score* to determine a quantifiable amount of physical activity done by the user, and if the score passes a set threshold, the user successfully mines a block. The Activity Score is accompanied by a *Reputation Score*. The more blocks a user mines (the more physically active the user is), the greater their Reputation Score will be. A higher Reputation Score will give the user access to higher incentives.
This approach will reward users to exercise more consistently and lead to a healthier, more active lifestyle. This method will provide a fairer way for all users to participate and receive rewards proportional to their efforts. Additionally, this system requires significantly lower amounts of energy than existing CPU intensive mechanisms.

4.1 Methodology Overview

For this research, we are following a modified Design Science methodology framework by Wieringa [4] to suit our research work. This methodology involves designing a system that securely collects and verifies physical activity data from user's wearable devices. This data is then securely transmitted to the blockchain network. A lower entry barrier is implemented to ensure that users with varying levels of physical activity can participate in the mining process. The consensus mechanism rewards users based on their physical activity levels, promoting a healthier lifestyle while maintaining the integrity and security of the blockchain. Fraud countermeasures are incorporated to ensure data integrity, preventing manipulation or spoofing of physical activity data. A simulator or testbed is designed to validate the proposed consensus mechanism using real or synthetic smartwatch data, ensuring its practical feasibility and effectiveness.
From our literature review, we see that there is a high variance of smart watch data that can be utilized in our consensus algorithm for mining. So, for our simulated environment

before implementing our finalized proof of concept, we ensure that any variable accessible to the smartphone device, be it weather, location, gyroscope and accelerometer parameters, are utilized by the consensus algorithm to create block headers and announce that all nodes have to calculate to mine the block. Furthermore, we will have to ensure that any mathematical data that is being used in the process are verified to be true through a trust based model connecting phone and the smartwatch. So, any participatory and transactional data received from our smart watches will be added to a blockchain instantly and consistently, to ensure no manipulation of data is done. Later on, we will simulate the whole consensus protocol in SimBlock, a blockchain simulator [12], to understand how its behaving and adjust accordingly on the final proof of concept. Figure 4.1 presents the above methodology in a diagram.

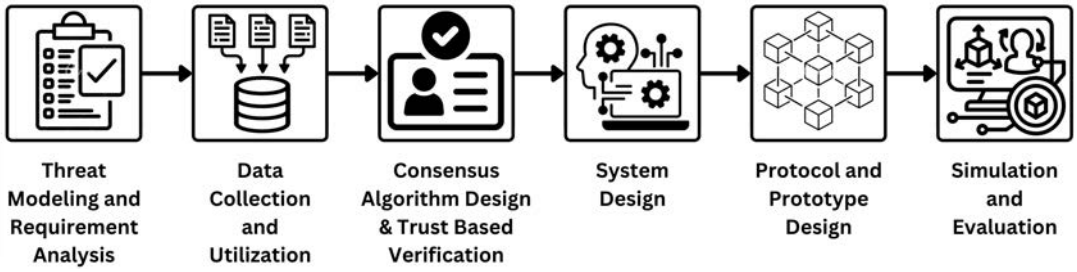


Figure 4.1: Methodology of this research

The planned system will utilize a systematic approach to designing and evaluating the system so as to integrate physical activity metrics into a blockchain consensus in a safe and reasonable manner:

- **Threat modeling and requirement analysis:** As detailed in section 3.1 of chapter 3, this phase involves identifying potential vulnerabilities of the system and determining what are the fundamental characteristics of a health-based blockchain consensus. Through this analysis we can ensure that the architecture of our framework meets the privacy, security, and fairness of rewards concerns.

- **Data collection and utilization:** Utilizing real world activity parameters (heart rate active minutes, step count) as the primary metric for activity, we collect, preprocess and normalize these parameters to generate reliable activity scores through the use of the available sensor technology on smartwatches as well as the Health Connect SDK on Android devices.

- **Consensus algorithm design and trust-based verification:** The PoPA consensus mechanism combines reputation tracking based on actual activity metrics in order to create a consensus mechanism where block rewards represent fair and verifiable contributions to the network.

- **System design:** System modules (validator nodes, full nodes, block creation, activity hash creation etc.) are designed to be modular and separate concerns in order to provide for future extensibility.

- **Protocol and prototype development:** Logic of the core system, data structures used within the simulation (blocks and transactions), reward calculation formulas were all formalized and encoded in the simulation using a modified version

of the SimBlock framework and have been developed into a production ready prototype using python full nodes and an android application.

- **Simulation and evaluation:** The PoPA protocol was simulated and tested across different numbers of nodes and thresholds of activity levels. Block propagation, reward fairness were measured to verify how the consensus performed under specific conditions.

4.2 System Architecture

As shown in Figure 4.2, PoPA is structured into three tiers: edge devices (smartwatches), smartphones (validators), and the blockchain network. Physical activity data is collected at the edge, computed and validated at the mobile layer, and finalized into blocks by the blockchain validator nodes.

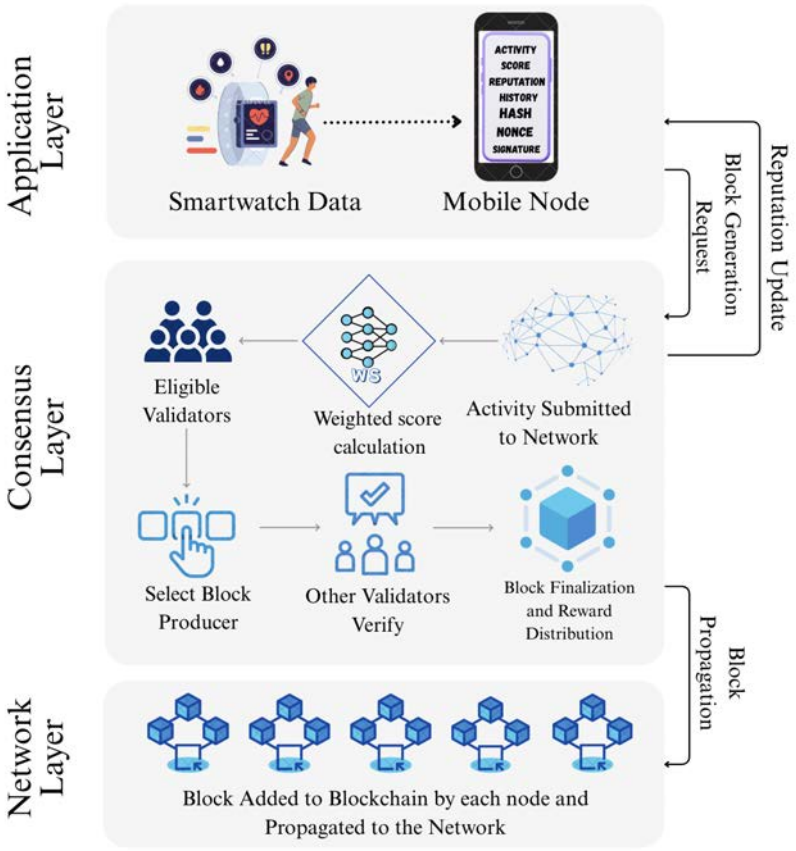


Figure 4.2: PoPA System Architecture

4.2.1 Architectural Layers:

As seen from Figure 4.2, the overall system architecture is a layered architecture with three layer and three properties. The layers include the Application layer (which includes the Edge Devices and Smartphones), the Network layer (which includes the Full Nodes) and the Consensus layer (which includes the Validator & Participatory Nodes). The

architecture structurally contains three category of devices whose properties define how nodes interact and participate in the PoPA consensus protocol.
This pipeline is illustrated in Figure 4.2 containing the following structures in our proposed environmental setup:

- **Edge Devices (Smartwatches)** collect raw sensor data which sends and synchronizes with the Health API in the edge device. These devices capture physical activity metrics such as step count, heart rate, and active minutes.
- **Smartphones** compute Activity Score (AS), sign submissions, and maintain wallets. These smart phones are user devices that collect raw physical activity data from the smart watch sensors (gyroscope, accelerometer) and health APIs, including step count, heart rate, and active minutes. These devices generate activity proofs that are transmitted to the validator network. These nodes can also transact on the blockchain as wallet nodes.
- **Full Nodes** maintain the distributed ledger, validate blocks, and enforce consensus rules. These nodes form the core blockchain network, maintaining a full copy of the ledger, validating incoming blocks, and enforcing the PoPA consensus rules. They participate in block validation and voting processes.

We can also see two types of properties in the nodes participating in the consensus.

- **Validator & Participatory Nodes:** nodes submit activity in fixed epoch windows and compete to produce a block. Submissions that fail verification checks are rejected. From Table 4.2, each validator computes a weighted score $WS = 0.7 \cdot AS_{norm} + 0.3 \cdot RH_{norm}$ (scores normalized to $[0, 1]$) and participates in consensus if the threshold is met. The selected validator is chosen *deterministically at random* from the eligible set using a seed derived from the epoch and previous block hash, while others verify and vote. Here, AS denotes a node's recent validated activity score (displayed on a 0–100 scale but normalized internally), and RH represents its historical reliability in submitting valid activity data. Validators maintain synchronization by following Unix epoch time.
- **Full Nodes:** maintain a complete copy of the blockchain and the global state. These nodes play the important role of helping block and transaction propagation, enforce consensus rules, and maintain overall network robustness.

These nodes combined with our consensus mechanism form the backbone of the PoPA blockchain network.
Consensus Mechanism: PoPA employs epoch-based participation with strict timing windows. The validators with adequate activity scores form the *eligible validators set*. A single validator is selected *deterministically* from this pool using a seed based on the previous block hash and current epoch. The other non-selected validators verify the block using majority voting.

The underlying mechanism uses a fixed threshold-based eligibility check and the reward is calculated from the weighted score. Validators must meet minimum thresholds (AS ≥ 0.3 and RH ≥ 0.1) to become an eligible validator. Among those eligible validator

nodes, selection is random (but deterministic to avoid chain forks) with equal probability (1/N), ensuring no validator can monopolize the network. However, block rewards are proportional to weighted score ($R = WS \times r_0$ where $WS = 0.7 \cdot AS + 0.3 \cdot RH$), incentivizing both physical activity and consistent participation. Reputation increases by 0.05 per successful block proposal and decays exponentially at 0.999845 per epoch when inactive (approximately 20% daily decay), with a minimum floor set at 0.1. This exponential decay mechanism encourages consistent activity submission as to get higher rewards validators must maintain a high reputation score. Voting-based mechanisms may be introduced which would allow even better governance of the network.

4.2.2 Block and Reward Properties

The **genesis block** sets the initial configuration and acts as the root of the chain, the first parent. Following the example of Ethereum, we have not capped the token supply. The name of the token for our consensus is PhysiCoin. Block rewards are distributed as:

$$R = WS \times r_0$$

where $WS = 0.7 \cdot AS_{norm} + 0.3 \cdot RH_{norm}$ is the weighted score combining activity (70% weight) and reputation (30% weight), and r_0 is the base reward rate per block which is set to 10 PhysiCoins in our example. Rewards are proportional to the weighted score to incentivize activity and consistency. The activity score is computed on a 0–100 scale for easier reference and normalized to $[0, 1]$ for consensus calculations.
The timeframe of producing a block, the **Block time** has been set by fixed-length epochs of 60 seconds with phased windows for submission, validation and selection. Activity submissions are accepted during $t \in [0, 40]$ seconds, followed by a gossip/grace period until $t = 55$ seconds for synchronizing the eligible validator's list among the full nodes. The selected validator creates and broadcasts the block at $t \approx 55$ seconds to allow propagation before the epoch ends at $t = 60$ seconds.

4.3 Protocol Flow

The multi-phase protocol suite for PoPA's sophisticated production implementation ensures that submissions to activities are secure; that validators are chosen deterministically; that consensus is Byzantine fault-tolerant; and that it can prevent Sybil attacks. This section introduces the formal data models, notations, and protocols used in PoPA.

4.3.1 Data Model

Table 4.1 lists the variables and mathematical symbols that have been used throughout the PoPA protocol. These definitions are crucial throughout block generation, validation, and the reward component of the proposed system.
The formal specification of PoPA entities and variables is presented in Table 4.2. Each component contributes to the overall consensus logic, validator scoring, and block construction.

Table 4.1: Formal Notation Used in the PoPA System

Symbol	Description
N	A general node in the network
V	A validator node that is eligible for block proposal
FN	Full node which maintains the blockchain state and participate in consensus
AS	Activity Score calculated on 0–100 scale; normalized internally to $[0, 1]$
RH	Reputation History; a score reflecting past validation behavior and Sybil penalties (if any)
WS	The Weighted Score which is used to determine the reward for block production: $WS = 0.7 \cdot AS_{norm} + 0.3 \cdot RH_{norm}$
R	Reward received for successful block proposal
r_0	Base reward rate per block (10 PHY)
P	Raw physical activity data package: (Steps, HeartRate, ActiveMinutes, Epoch, DeviceID)
$H(P)$	Activity proof hash: SHA256(Steps ‖ HeartRate ‖ ActiveMinutes ‖ Epoch)
Sig_N	ECDSA signature of node N using SECP256k1
TS	Timestamp in milliseconds (Unix epoch)
$Epoch$	60-second consensus round number
N_A	Wallet address: first 20 bytes of SHA256(PubKey) as 40-character hex string
$PoPA\_Block$	Block structure: (index, previous_hash, timestamp, transactions[], validator, activity_proof, AS, RH, WS, reward, epoch, signature)
R_{tot}	Total cumulative reward earned by a validator
$B[i]$	The i-th block in the blockchain
TX	Transaction: transfer of tokens between addresses
$TxID$	Transaction ID: SHA256(Sender ‖ Recipient ‖ Amount ‖ Fee ‖ TS ‖ PubKey)
Fee_{tx}	Transaction fee: 0.01 PHY for user transfers, 0.0 PHY for system rewards
R_{fees}	Total transaction fees collected in a block, awarded to block validator

4.3.2 Protocol Specifications

This subsection provides the detailed description of the multi phase protocol suite used in PoPA. These protocols ensure that activity submission is secure, validator selection is fair, the system is fault and fraud tolerant and that blocks are proposed and added in a structured manner.

Table 4.3: Activity Submission Protocol Messages (M_1-M_8)

Msg	Flow	Description
M_1	$N \rightarrow FN$	[`ActivityData`$(Steps, HeartRate, ActiveMinutes)$, $AS, H(P), TS, N_A]_{\text{signed}}$
M_2	$FN \rightarrow N$	[`submissionAck`(accepted/rejected, reason)]
M_3	$FN_i \rightarrow FN_j$	[`reconcileRequest`$(\{H(\text{activity}_1), H(\text{activity}_2), \ldots\})$]
M_4	$FN_j \rightarrow FN_i$	[`reconcileResponse`({missing_hashes})]
M_5	$FN_i \rightarrow FN_j$	[`syncActivities`({full_activity_data})]
M_6	$FN_j \rightarrow FN_i$	[`syncAck`(count_added)]

Continued on next page

Table 4.3 – *Continued from previous page*

Msg	Flow	Description
M_7	FN (internal)	`runSybilDetection`({all_raw_submissions}) → {accepted, rejected, banned}
M_8	FN (internal)	`getEligibleValidators`() → $\{V_1, V_2, \ldots, V_m\}$ where $AS \geq 0.3, RH \geq 0.1$

Table 4.3 illustrates the activity submission protocols. The detailed protocol flow is described below.

Protocol 1: Multi-Phase Activity Submission Protocol (M_1-M_8)

Unlike traditional single-message submission protocols, PoPA implements a four-phase activity submission process (each with a specific time window) with P2P reconciliation and Sybil detection to ensure Byzantine fault tolerance and prevent duplicate submissions also known as replay attack.

Phase 1 (t=0–40s): Collection Phase

Validators (from the mobile app) start the activity submission by sending signed activity data to one or more full nodes they are connected to. These data packages contain the step count, heart rate, active minutes, locally computed activity score (AS), activity proof hash $H(P)$, timestamp (TS), and address of the validator node N_A (message M_1). After receiving all the submissions, full nodes perform validation checks including timestamp freshness (5-minute window), replay attack detection using transaction IDs. During this collection phase, full nodes store the actual raw submissions without applying Sybil filtering (`skip_sybil_detection=True`), allowing all activity data to be collected before the Sybil analysis. The full node responds with a submission acknowledgment indicating whether the submission was accepted or rejected, along with a reason if rejected (message M_2) back to the eligible node who sent the activity.

Phase 2 (t=40–45s): P2P Reconciliation Phase (Gossip Anti-Entropy)

After the collection window closes, full nodes start gossipping to ensure network-wide node consistency. Each full node FN_i broadcasts a reconciliation request to the alive peer nodes FN_j containing a list of activity hashes it has received (message M_3). The receiving node FN_j compares these hashes against its own database and determines if the it is missing any submissions or not. It then responds with a list of missing hashes (message M_4). Node FN_i then sends the complete activity data for the missing submissions (message M_5), and node FN_j acknowledges receipt with a count of newly added submissions (message M_6). This bidirectional exchange continues among all full nodes until all the nodes have the exact raw datasets, preventing the chance of chain forks before Sybil detection.

Phase 3 (t=45–50s): Sybil Detection Phase

After reconciliation is complete (at t=45-50s), the full nodes independently executes a three-layer Sybil detection algorithm (message M_7) in their local environment. The first layer detects exact duplicates by computing the activity proof hash $H(P) = \text{SHA256}(Steps \parallel HeartRate \parallel ActiveMinutes \parallel Epoch)$ for each submission. When duplicates are identified, a deterministic winner is selected using this formula:

$$\text{Winner} = \text{sorted\_wallets}[\text{SHA256}(H(P) \parallel PreviousBlockHash) \mod n]$$

All losing wallets are permanently banned to prevent further fraudulent activity. The second layer enforces device binding, ensuring only one wallet per device ID ($DeviceID$)

Table 4.2: Formal Data Model Definitions in PoPA

Symbol	Definition
$Validator_{PoPA}$	$\triangleq$ Identified by wallet address N_A (40-char hex); maintains $(AS, RH, WS, R_{tot}, DeviceID)$
AS	$\triangleq$ Activity score: $0.8 \times steps_{norm} + 0.1 \times hr_{norm} + 0.1 \times activeMins_{norm}$ (Determines eligibility for block proposal)
$steps_{norm}$	$\triangleq$ Normalized steps: $\min(steps/7500, 1.0)$
hr_{norm}	$\triangleq$ Normalized heart rate: Piecewise function: 1.0 (optimal range, 60-100 BPM), 0.8 (acceptable, 10 BPM deviation from optimal), 0.5 (sub-optimal, > 0 BPM), 0.0 (no data)
$activeMins_{norm}$	$\triangleq$ Normalized active minutes: $\min(activeMins/60, 1.0)$
RH	$\triangleq$ Reputation score: updated via $+0.05$ (successful block), -0.10 (failed block), $\times 0.999845^{epochs}$ (decay)
WS	$\triangleq$ Weighted score: $0.7 \cdot AS + 0.3 \cdot RH$ (determines the reward)
$Transaction$	$\triangleq$ (tx_id, sender, recipient, amount, fee, timestamp, signature, public_key, type)
$Block$	$\triangleq$ (index, previous_hash, timestamp, transactions[], validator, activity_proof, AS, RH, WS, reward, nonce, hash, signature, epoch)
`ActivitySubmission`	$\triangleq$ Signed message: [ActivityData(Steps, HeartRate, ActiveMinutes), AS, $H(P)$, TS, N_A]
`submissionAck`	$\triangleq$ Response: (accepted/rejected, reason)
`BlockProposal`	$\triangleq$ Broadcast by selected validator: signed PoPA_Block
`blockAck`	$\triangleq$ Response: (validity, added)
`reconcileRequest`	$\triangleq$ P2P gossip: set of activity hashes for anti-entropy synchronization
`reconcileResponse`	$\triangleq$ Response: set of missing hashes
`syncActivities`	$\triangleq$ P2P data transfer: full activity data for missing submissions
`getEligibleValidators`	$\triangleq$ Internal computation: returns validators where $AS \geq 0.3$ and $RH \geq 0.1$
`runSybilDetection`	$\triangleq$ Internal process: three-layer detection (exact duplicates, device binding, 95% similarity)

per epoch can participate. Duplicates are rejected. The third layer applies similarity detection using Euclidean distance in normalized 3D space (steps, heart rate and active minutes): $d = \sqrt{(s_1 - s_2)^2 + (h_1 - h_2)^2 + (a_1 - a_2)^2}$. If similarity exceeds 95%, the hash-based selection from Layer 1 is applied to choose a winner, losers are banned.

Phase 4 ($t \geq 55$s): Validator Eligibility Computation

After the Sybil detection has concluded, full nodes compute the eligible validator list by applying activity and reputation thresholds (message M_8). The eligibility function returns $\{V_1, V_2, \ldots, V_m\}$ where each validator satisfies the condition: $AS \geq 0.3$ (30% activity threshold) and $RH \geq 0.1$ (10% reputation threshold). Validators with Sybil penalties below the reputation floor are excluded. Timing enforcement ensures this computation occurs only after $t = 55$s to accommodate delayed submissions. This eligible validator set is later used in the deterministic selection process.

Table 4.4: Validator Selection Protocol Messages (M_9-M_{11})

Msg	Flow	Description
M_9	FN (internal)	$seed \leftarrow \text{SHA256}(CurrentEpoch \parallel PreviousBlockHash)$
M_{10}	FN (internal)	$selected\_index \leftarrow \text{hexToInt}(seed[0:16]) \mod \lvert\{\text{eligible\_validators}\}\rvert$
M_{11}	FN (internal)	$V_{selected} \leftarrow \text{sortedEligibleValidators}[selected\_index]$

Table 4.4 summarizes the validator selection protocols. Next, we describe the detailed protocol flow.

Protocol 2: Deterministic Validator Selection Protocol (M_9-M_{11})

As stated earlier, PoPA uses deterministic random selection. It uses a seed to prevent fraud and ensure all nodes agree on one single validator. All validator nodes and full nodes independently compute the same validator deterministically after $t \geq 55$s. The deterministic seed made from the current epoch and previous block hash is used (message M_9). This seed is then used to calculate the index of the selected validator using modulo arithmetic (message M_{10}). The validator is selected by the generated index from the alphabetically sorted eligible validator pool (message M_{11}).

Table 4.5: Block Proposal Protocol Messages (M_{12}-M_{14})

Msg	Flow	Description
M_{12}	$V_{selected}$ (internal)	$block \leftarrow$ `createBlock`$(index, prev\_hash, TS, \{TX\}, ActivityProof, AS, RH, WS, R, Sig_V)$
M_{13}	$V_{selected} \rightarrow$ All Nodes	$[$`blockProposal`$(PoPA\_Block)]_{\text{signed}}$
M_{14}	$FN \rightarrow V_{selected}$	[`blockAck`(validity, added)]

Table 4.5 summarizes the block proposal protocol messages. We describe the detailed protocol flow below.

Protocol 3: Block Proposal and Finalization Protocol (M_{12}-M_{14})

The selected validator creates and broadcasts a block before $t \geq 60$s with strict validation requirements. The selected validator $V_{selected}$ constructs a block using the `createBlock` function, which includes the block index, previous block hash, timestamp (TS), transaction set $\{TX\}$, activity proof, activity score (AS), reputation (RH), weighted score (WS), reward (R), and validator signature Sig_V (message M_{12}). Then the validator broadcasts the signed block proposal to all nodes in the network that it knows of (message M_{13}). The full nodes receive the block proposal and perform the required validation checks. They verify that the block is from the selected validator and not anyone else, the previous hash matches the last block, ensure this is the only block for the epoch, confirm that heart rate data is present to avoid fake step counts, validate $AS \geq 0.3$, check the timestamp $TS \geq$ epoch start $+50s$, ensure no future epochs, and verify ECDSA signature validity. After the checks have passed successfully, full nodes respond with an acknowledgment indicating whether it was added to the chain (message M_{14}).

The reward distribution mechanism allocates block rewards based on these formulae:

$$R_{activity} = WS \times r_0 \quad \text{where } r_0 = 10 \text{ PHY}$$
$$R_{fees} = \sum_{tx \in block.transactions} tx.fee$$
$$R_{total} = R_{activity} + R_{fees}$$

The validator receives both the activity-based reward and all transaction fees from included transactions.

Table 4.6: Transaction Protocol Messages (M_{15}-M_{18})

Msg	Flow	Description
M_{15}	$N \rightarrow FN$	[`Transaction`$(TxID, Sender_A, Recipient_A, Amount, Fee, TS, Type)]_{\text{signed}}$
M_{16}	$FN \rightarrow N$	[`txAck`(accepted/rejected, reason)]
M_{17}	$FN_i \rightarrow FN_j$	[`txBroadcast`$(Transaction)$]
M_{18}	$FN_j \rightarrow FN_i$	[`txReceived`(accepted)]

Table 4.6 summarizes the transaction protocols. Below we describe the detailed protocol flow.

Protocol 4: Transaction Protocol (M_{15}-M_{18})

Transactions support wallet-to-wallet transfers, validator rewards with fee payment and security checks. A mobile node initiates a transaction by submitting a signed data package to full nodes containing the transaction ID (TxID), sender address ($Sender_A$), recipient address ($Recipient_A$), amount, fee, timestamp (TS), and transaction type (message M_{15}). The transaction ID is computed as $TxID = \text{SHA256}(Sender \parallel Recipient \parallel Amount \parallel Fee \parallel TS \parallel PubKey)$. The full nodes receive the transactions and perform five critical security checks. First, timestamp freshness is verified to ensure $currentTime - TS < 300000$ ms (5-minute window). Second, replay prevention checks that $TxID \notin used\_nonces$, ensuring each transaction is processed only once. Third, ECDSA signature verification confirms `ECDSA_VERIFY`$(PubKey, TxID, Signature) = \text{true}$. Fourth, balance verification ensures $Balance[Sender] \geq Amount + Fee$ to prevent overdraft. Fifth, address derivation verifies `pubkey_to_address`$(PubKey) = Sender_A$, confirming the sender controls the claimed address. If all checks pass, the full node responds with an acceptance acknowledgment; otherwise, it returns a rejection with a specific reason (message M_{16}).
The system supports three transaction types, each with distinct fee structures. Standard wallet transfers (`transfer`) facilitate peer-to-peer value exchange with a fee of 0.01 PHY. Activity rewards (`activity_reward`) are system-generated transactions distributing rewards for validated physical activity submissions with zero fees. Validator rewards (`validator_reward`) compensate block producers for securing the network, also with zero fees. After initial validation, transactions are propagated through the mempool using gossip-based broadcasting. Full node FN_i broadcasts the transaction to its peer FN_j (message M_{17}), and FN_j responds with an acknowledgement (message M_{18}).

Table 4.7: Wallet Operation Protocol Messages (M_{19}-M_{24})

Msg	Flow	Description
M_{19}	N (internal)	$(PrivKey, PubKey) \leftarrow$ `generateKeyPair`()
M_{20}	N (internal)	$N_A \leftarrow$ `hex`(SHA256($PubKey$)[0 : 20])
M_{21}	$N \rightarrow FN$	[`queryBalance`(N_A)]
M_{22}	$FN \rightarrow N$	[`balanceResponse`($Balance, PendingTx$)]
M_{23}	$N \rightarrow FN$	[`queryReputation`(N_A)]
M_{24}	$FN \rightarrow N$	[`reputationResponse`($RH, SybilPenalty, IsBanned$)]

Table 4.7 summarizes the wallet operation protocols. The detailed protocol flow is described in the below paragraphs.

Protocol 5: Wallet Operations (M_{19}-M_{24})

Wallets use SECP256k1 elliptic curve cryptography with SHA256-based hexadecimal address encoding. A public (PubKey) and private key (PrivKey) pair is generated by executing `generateKeyPair()` (message M_{19}). From the public key, the a wallet address N_A is derived by the formula `hex`(SHA256($PubKey$)[0 : 20]), yielding a 40-character hexadecimal string from the first 20 bytes of the SHA256 hash (message M_{20}). Message signing follows ECDSA standards, mobile nodes sign messages using $Sig_N =$ `ECDSA_SIGN`($PrivKey$, SHA256($message$)).
For balance queries, nodes send a request to a full node containing their wallet address N_A (message M_{21}) The full node then respond with the current balance and list of pending transactions (message M_{22}). Similarly, for reputation queries the reputation status is fetched by sending the wallet address to full nodes (message M_{23}). Any Sybil penalties applied, and whether they are banned from the network are also available (message M_{24}). These allow users to monitor the status of their wallet.

Table 4.8: P2P Network Protocol Messages (M_{25}-M_{37})

Msg	Flow	Description
M_{25}	$N \rightarrow Bootstrap$	[`getPeers`()]
M_{26}	$Bootstrap \rightarrow N$	[`peerList`($\{FN_1, FN_2, \ldots\}$)]
M_{27}	$N \rightarrow FN_i$	[`handshake`(`GET /node_status`)]
M_{28}	$FN_i \rightarrow N$	[`nodeStatus`($chainLength, version, peers$)]
M_{29}	$FN_i \rightarrow FN_j$	[`POST /p2p/reconcile/request`($\{activity\_hashes\}$)]
M_{30}	$FN_j \rightarrow FN_i$	[`reconcileResponse`($\{missing\_hashes\}$)]
M_{31}	$FN_i \rightarrow FN_j$	[`POST /p2p/reconcile/sync`($\{full\_activities\}$)]
M_{32}	$FN_j \rightarrow FN_i$	[`syncAck`($count\_added$)]
M_{33}	$N \rightarrow FN$	[`GET /chain`]
M_{34}	$FN \rightarrow N$	[`chainResponse`($length, blocks$[])]
M_{35}	N (internal)	if $FN.chainLength > local.chainLength$: `replaceChain`($FN.blocks$)
M_{36}	$FN_i \rightarrow$ All Nodes	[`POST /activity/reject`($WalletID, Reason, RelatedWallet$)]
M_{37}	$FN_j \rightarrow FN_i$	[`rejectionAck`(processed)]

Table 4.8 summarizes the P2P network protocol messages. The description of these are given below.

Protocol 6: P2P Network Protocols (M_{25}-M_{37})

The P2P network implements automatic peer discovery, health monitoring, and synchronization. Nodes discover peers via a hardcoded bootstrap node by sending a `getPeers()`

request (message M_{25}), which responds with a list of active full nodes $\{FN_1, FN_2, \ldots\}$ (message M_{26}). The requesting node then performs handshakes with each peer by issuing `GET /node_status` requests (message M_{27}), receiving responses containing chain length, version, and peer lists (message M_{28}). Health checks occur every 60 seconds with a failure threshold of 3, automatically removing unresponsive peers from the active set.

Activity reconciliation employs gossip anti-entropy to ensure all full nodes maintain identical activity datasets before Sybil detection. Full node FN_i starts the reconciliation process by sending a POST request containing hashes of all activities that it received (message M_{29}). Its peer FN_j then compares these hashes against its own database and responds with a list hashes it is missing (message M_{30}). Node FN_i then transmits the complete activity data for missing submissions (message M_{31}), and FN_j acknowledges with the count of newly added entries (message M_{32}).

Block synchronization in PoPA follows Bitcoin's longest chain rule, where nodes periodically query peers for chain status through `GET /chain` requests (message M_{33}). Full nodes respond with their chain length and complete block list (message M_{34}). If the peer's chain length exceeds the local chain length, the requesting node replaces its entire chain with the longest valid chain (message M_{35}). This ensures eventual consistency across the network. When Sybil detection identifies duplicates, full nodes coordinate rejection decisions by broadcasting rejection messages to all nodes, containing the wallet ID, rejection reason, and related wallet addresses (message M_{36}). Receiving nodes acknowledge and maintain the consistent sybil detection state (message M_{37}).

Table 4.9: Sybil Detection and Reputation Update Protocol Messages (M_{38}-M_{46})

Msg	Flow	Description
M_{38}	FN (internal)	`recordSybilOffense`($WalletID, AttackType, RelatedWallet, Evidence$)
M_{39}	FN (internal)	$SybilScore \leftarrow \{flags : count, penalty : -0.1 \times count, isBanned : count \geq 5\}$
M_{40}	FN (internal)	$RH_{new} \leftarrow \min(RH + 0.05, 1.0)$ (successful block)
M_{41}	FN (internal)	$RH_{new} \leftarrow \max(RH - 0.10, 0.1)$ (failed block)
M_{42}	FN (internal)	$RH_{new} \leftarrow \max(RH \times 0.999845^{epochs\_inactive}, 0.1)$ (decay)
M_{43}	$Admin \rightarrow FN$	[`GET /admin/sybil/flagged`]
M_{44}	$FN \rightarrow Admin$	[`flaggedList`($\{submissions\}$)]
M_{45}	$Admin \rightarrow FN$	[`POST /admin/sybil/ban`($WalletID$)]
M_{46}	$FN \rightarrow Admin$	[`banAck`(banned)]

Table 4.9 summarizes the Sybil detection and reputation update protocol messages. Next, we describe the detailed protocol flow.

Protocol 7: Sybil Detection and Reputation Update Protocol (M_{38}-M_{46})

The Sybil tracker maintains persistent state across epochs using SQLite databases. When Sybil attacks are detected, full nodes record detailed information using `recordSybilOffense`, storing the wallet ID, attack type (exact duplicate, device binding violation, or similarity), related wallet addresses, and cryptographic evidence (message M_{38}). The system computes a Sybil score containing the count, reputation penalty ($-0.1 \times count$), and ban status (message M_{39}). The progressive ban system implements increasing penalties: wallets accumulating five or more Sybil flags are permanently banned while wallets with fewer flags receive proportional reputation penalties of -0.1 per flag.

Reputation updates are carried out in three ways. After successful block proposals, validators receive a reputation increase of 0.05, capped at 1.0 to prevent score inflation (message M_{40}). Failed block proposals incur a penalty of -0.10 until it has reached the minimum reputation of 0.1 (message M_{41}). Inactive validators are penalized exponential decay at a rate of $0.999845^{epochs\_inactive}$, also floored at 0.1 (message M_{42}). This decay formula produces approximately 20% daily reduction, forcing participants to maintain regular activity to preserve higher rewards.
For administrative oversight, the system provides dedicated endpoints enabling human review of flagged submissions. This is mostly helpful for development purposes. Administrators can query flagged submissions via `GET /admin/sybil/flagged` (message M_{43}). The endpoint returns a comprehensive list of suspicious activities (message M_{44}). After manual review, administrators can issue permanent bans by sending `POST /admin/sybil/ban` by including the wallet ID (message M_{45}), and the full node acknowledges the ban operation (message M_{46}). This prevents any sort of sophisticated Sybil attacks that may bypass the automatic checks.

4.4 Core mechanisms

In this section we provide details of the core mechanisms in the PoPA consensus protocol, including validator selection, the balance between reputation and activity, and the evolution of our system from simulation in early days to a production implementation today.

4.4.1 Validator Selection Mechanism

The Activity Score *(AS)* is computed on a 0–100 scale from three components: steps (80%), heart rate (10%), and active minutes (10%). Steps and active minutes are normalized against research-backed thresholds (e.g., 7500 steps/day, 60 active minutes), while the contribution of heart rate is determined via range-based scoring (optimal/acceptable/suboptimal).
To ensure fairness and efficiency, a node becomes *eligible* when its activity score meets the minimum threshold (30%) to participate.
From among all eligible validators, the selected validator is chosen *deterministically at random* using a seed derived from the epoch and previous block hash. Other validators serve as verifiers of the block.
This approach ensures network efficiency while maintaining decentralization, as any participating node can be deemed eligible if its AS exceeds the minimum threshold.

4.4.2 Reputation vs Activity

Our motivation behind this consensus algorithm is to promote sustained physical activity among participants. And with this goal in mind we agreed that nodes demonstrating consistent behavior should receive greater rewards than those with sporadic but intense activity. This means, the entry barrier is low, nodes with activity score and reputation that meet the eligibility thresholds (AS ≥ 0.3, RH ≥ 0.1) can participate in block production, but those with higher weighted scores (WS $= 0.7 \cdot AS + 0.3 \cdot RH$) earn proportionally higher block rewards.
This system greatly increases the value of long-term and consistent participation within the network. Thus, the activity score (AS) reflects recent physical activity, while the

reputation history (RH) captures the node's track record of consistent engagement. The 24-hour sliding window applies to step count aggregation, while reputation undergoes exponential decay at a rate of 0.999845 per epoch (equivalent to 20% daily decay). From the code snippets shared in Algorithm 4 and Algorithm 6, we see that the AS is calculated based on daily step count, heart rate, and active minutes, while the RH is updated based on successful block proposals and consistent activity submissions.
During Phase-1, we initially prioritized nodes whose *Weighted Score (WS)* regularly exceeded the mining threshold. In the final implementation, validator selection became deterministically randomized among eligible nodes to prevent monopolization, while rewards remained proportional to WS, thereby continuing to encourage deterministic consistency in participation.

4.4.3 Evolution of Validation procedure

Initially, for our simulation phase, we used Weighted Score to select validators in the validation procedure. However, after running multiple tests, we observed that it would prioritize super athletes in a system or would fail to deterministically output similar results from full nodes for synchronization. This pattern showed a bias in the selection process, which could potentially lead to centralization and unfair advantages for certain nodes which we can refer to as the *(*Super Athlete Problem). To address this issue, we modified the selection algorithm to use a purely random selection mechanism, independent of the activity and weighted scores. So, the weighted score has been put to use in the reward calculation (Algorithm 5) only while the activity score determines eligibility. The results we see in the graph confirm that with the new random selection method, validator selection is evenly distributed among all nodes. This change enhances the fairness and decentralization of the PoPA blockchain network.

4.4.4 From Simulation to Implementation: A Two-Phase Algorithmic Evolution

Following an iterative approach, the evolution of PoPA's algorithms were from the first simulations of design to the final production-ready implementation of the same. In the next section, I will show you examples of each phase to illustrate the effect of real world constraint and limitation on the PoPA's final implementation.

Phase 1: Designing the Simulation (SimBlock)

PoPA's consensus was simulated and validated by using SimBlock, a highly used but somewhat less supported blockchain simulator. Phase 1's focus was to create the main concepts of PoPA's consensus with many simplifications as possible concerning the timing, network condition, and behavior of validators. The simulation algorithms provided validation of a proof-of-concept, which indicated the design parameters would require additional development to be applicable in a real world scenario.
Characteristics of the Simulation:

- **Timing Model:** An average of dynamic block intervals equaling two seconds per block was the targeted time model.
- **Activity Metrics:** Three components, steps (40%), heart rate (40%), motion intensity or speed (20%) made up the activity metrics.

- **Validator Selection:** A single validator was selected per round based on its highest weighted score.
- **Network Model:** Synchronous idealized communication was modeled without delay or message loss.
- **Difficulty Adjustment:** Dynamic activity difficulty tuning was used.

The pseudo-code of Algorithm 12 demonstrates the PoPA Consensus and Block Creation pipeline implemented within SimBlock. As shown in Table 5.1, the metrics and variables mentioned above were used in Algorithm 12. Once a block is generated through the Algorithm 12 pipeline, the block is validated through Algorithm 14, prior to being propagated to the chain through Algorithm 13.
The simulation difficulty adjustment of Algorithm 15 dynamically adjusted mining targets based upon the activity of the validators and the timing of the blocks:

$$\delta = (V_{AS} - V_{AS_{prev}}) \cdot (\Delta t - 2000)$$

$$\text{If } \delta \geq 0, \text{ decrease difficulty by } 5\%, \text{ else increase it.}$$

Phase 2: Production Implementation (Live POC)

After validating the simulation, we implemented a production ready proof-of-concept that would solve the many real world problems associated with network synchronization issues, Byzantine failures, the limitations of mobile devices, and security risks. This also included a large scale algorithmic improvement as a result of testing the new algorithm with real world android devices and health connect sdk integration.
Implementation Characterstics:

- **Timing Model:** Timing model was completely changed. It used fixed 60 second epochs with deterministic phase boundaries (activity submission: 0-40 seconds; grace period: 40-55 seconds; selection: 55 seconds; start of block creation at 55 seconds so that all nodes have time to propagate the information prior to the end of the epoch).
- **Activity Metrics:** Activity metrics were scored using research validated scoring techniques for heart rate range and active minutes (weights are: steps = 80%; heart rate = 10%; active minutes = 10%)
- **Validator Selection:** Validators selected using deterministic random selection from the eligible validator pool with threshold-based filtering.
- **Network Model:** Network model is redundant multi-node design using a 2/3 vote with automatic peer discovery in an ideal p2p network.
- **Security:** Sybil detection, replay protection, activity staleness to clean out, transaction fees.
- **Resilience:** Caching of reputation to avoid rate limiting, http retry with multi node fail over, timestamp normalization.

Changes to Key Algorithms from Simulation to Production

Table 4.10 shows the primary changes to the key algorithms between the simulation and the two production implementations along with our reasoning behind each change.

Table 4.10: Algorithmic Evolution: Simulation vs. Implementation

Component	**Simulation Design**	**Production Implementation**	**Reason**
Epoch Timing	Dynamic averaging (2s target)	Fixed 60s with phased structure	Network asynchrony; mobile device latency
Activity Scoring	3 metrics: steps (40%), HR (40%), motion (20%)	3 metrics: steps (80%), HR (10%), active minutes (10%)	Medical research (JAMA); Health Connect API compatibility
Validator Selection	Highest WS wins directly	Threshold filtering + deterministic random from eligible pool	Fairness; prevents monopolization
Network Architecture	Single-node submission	Multi-node with 2/3 majority	Eliminate single point of failure
Sybil Defense	Basic duplicate detection	Hash-based selection, permanent blacklisting, 95% similarity threshold	Production attacks observed; address grinding prevention
Reputation Decay	Not implemented	20% daily decay (0.999845 per epoch)	Reward sustained activity over sporadic bursts
Transaction Model	Reward-only	Dual fees: 0.01 PHY user tx + validator rewards	Spam prevention; incentivize tx inclusion
Timing Enforcement	Flexible submission	Activity window closure (t=40s) + grace period (40-55s)	Fork prevention; P2P gossip propagation
Resilience	Assumed reliable network	Cached fallback, HTTP retry, 20s read timeout	Mobile network unreliability; server load
Peer Discovery	Static node list	Dynamic bootstrap-based discovery with refresh	Real-world node churn

4.4.5 Production Algorithm Suite

The algorithms listed below represent the complete production implementation deployed in the live PoPA proof-of-concept. We can see the developments from phase-1 simulation to phase-2 production in the structured timing, multi-node redundancy, Sybil defenses, and deterministic validator selection. The pointers of the updates are summarized in Table 4.10.

Consensus Algorithms

Main Consensus Loop The production consensus loop (Algorithm 1) implements a strict phased timing model with multi-node redundancy, replacing the simulation's flexible timing approach. Key enhancements include: (1) activity submission window enforcement (0-40s), (2) 15-second grace period for P2P gossip (40-55s), (3) majority voting for eligible validator list, and (4) deterministic validator selection.

Algorithm 1: PoPA Consensus Process Flow (Production)

Require: $walletAddress$, $privateKey$, $nodeUrl$
Ensure: Continuous consensus participation

Constants:
$EPOCH\_DURATION \leftarrow 60000$ ms ▷ 60 seconds
$ACTIVITY\_WINDOW \leftarrow 40000$ ms ▷ Activity submissions close at t=40s
$ELIGIBLE\_FETCH\_TIME \leftarrow 55000$ ms ▷ Fetch eligible list at t=55s
$SUBMISSION\_TIMES \leftarrow [0, 15000, 30000]$ ms ▷ Submit at t=0s, 15s, 30s

while $isRunning$ **do**
    $currentEpoch \leftarrow getCurrentEpoch()$
    $epochStartTime \leftarrow getEpochStartTime()$

    Step 1: Clear previous epoch data
    $activitySubmissions \leftarrow \emptyset$
    $blockVotes \leftarrow \emptyset$
    $proposedBlock \leftarrow NULL$

    Step 2: Activity Submission Phase (t=0s, 15s, 30s)
    for $targetTime \in SUBMISSION\_TIMES$ **do**
        $elapsed \leftarrow currentTime() - epochStartTime$
        wait until $elapsed \geq targetTime$

        ▷ Check still within activity window (t ¡ 40s)
        if $elapsed < ACTIVITY\_WINDOW$ **then**
            $AS \leftarrow calculateActivityScore()$
            $RH \leftarrow getReputation(walletAddress)$
            $myRole \leftarrow determineRole(AS)$ ▷ ELIGIBLE, PARTICIPANT, or INACTIVE

            if $myRole \in \{ELIGIBLE\_VALIDATOR, PARTICIPANT\}$ **then**
                $submission \leftarrow createActivitySubmission(walletAddress,$

```
                AS, currentEpoch)
            submitToAllFullNodes(submission)  ▷ Multi-node submission with 2/3 threshold
        end if
    else
        log("Submission window closed, skipping")
    end if
end for

Step 3: Wait until t=55s to fetch eligible validators
elapsed ← currentTime() − epochStartTime
waitTime ← ELIGIBLE_FETCH_TIME − elapsed
if waitTime > 0 then
    wait waitTime ms          ▷ Grace period for P2P gossip (t=40s to t=55s)
end if

Step 4: Fetch eligible validators with majority voting
eligibleCandidates ← fetchEligibleWithMajorityVote()      ▷ Query all known full nodes
if |eligibleCandidates| = 0 then
    continue                                  ▷ No eligible validators, skip epoch
end if

Step 5: Determine if I am selected validator
previousBlock ← getLatestBlock()
selectedValidator ← selectValidator(eligibleCandidates,
      currentEpoch, previousBlock.hash)

Step 6: If selected, create and submit block immediately (at t=55s)
if selectedValidator = walletAddress then
    log("I AM SELECTED! Creating block immediately...")
            ▷ Create block at t=55s to allow 5s propagation before epoch ends
    pendingTx ← getPendingTransactions()
    activitySubs ← getActivitySubmissions()
    block ← proposeBlock(walletAddress, privateKey,
          currentEpoch, pendingTx, activitySubs)
    broadcastBlockToAllNodes(block)
    updateReputation(walletAddress, SUCCESSFUL_BLOCK)

                                  ▷ Wait for block to propagate and epoch to end
    elapsed ← currentTime() − epochStartTime
    remainingTime ← EPOCH_DURATION − elapsed
    if remainingTime > 0 then
        wait remainingTime ms
    end if
else
                ▷ Not selected, wait for block from winner (10s for propagation)
    wait 10 seconds
```

76: **end if**
77: **end while**

Validator Selection Production validator selection (Algorithm 2) uses deterministic randomness seeded by previous block hash and epoch, preventing predictable patterns while ensuring all nodes agree on the winner. This replaces simulation's deterministic "highest score wins" approach.

Algorithm 2: Deterministic Random Validator Selection (Production)

Require: $eligibleCandidates$, $epoch$, $previousBlockHash$
Ensure: $selectedValidator$ chosen deterministically

if $|eligibleCandidates| = 0$ **then**
  return $NULL$ ▷ No eligible validators
end if

if $|eligibleCandidates| = 1$ **then**
  return $eligibleCandidates[0]$ ▷ Only one candidate
end if

Step 1: Create deterministic seed
$seedData \leftarrow epoch \parallel previousBlockHash$ ▷ Concatenate
$seed \leftarrow SHA256(seedData)$

Step 2: Sort candidates by wallet address
$sortedCandidates \leftarrow sort(eligibleCandidates, key = walletAddress)$
▷ Ensures all nodes have same ordering

Step 3: Convert seed to integer for selection
$seedHex \leftarrow seed[0:16]$ ▷ First 16 hex characters
$seedInt \leftarrow hexToInt(seedHex)$

Step 4: Select using modulo
$selectedIndex \leftarrow seedInt \mod |sortedCandidates|$
$selectedValidator \leftarrow sortedCandidates[selectedIndex]$

return $selectedValidator$

Eligible Validator Filtering This algorithm (Algorithm 3) implements strict timing enforcement and multi-layered validation absent from simulation, including Sybil score checking, staleness detection, and grace period enforcement.

Algorithm 3: Get Eligible Validators (Production)

Require: $activitySubmissions$, $currentEpoch$
Ensure: $eligibleValidators$ list

Constants:
$ACTIVITY\_THRESHOLD \leftarrow 0.3$ ▷ 30% minimum

```
MIN_REPUTATION ← 0.1                                   ▷ 10% minimum
MAX_ACTIVITY_AGE ← 300000 ms                           ▷ 5 minutes
ELIGIBLE_FETCH_TIME ← 55000 ms                         ▷ Must wait until t=55s

eligibleValidators ← []
currentTime ← getCurrentTimeMs()
epochStartTime ← getEpochStartTime()
elapsed ← currentTime − epochStartTime

Timing Enforcement: Must wait for grace period to end
if elapsed < ELIGIBLE_FETCH_TIME then
    error("Tooearlytocomputeeligiblelist(t = " + elapsed/1000 + "s, needt ≥ 55s)")
end if

for walletId, submission ∈ activitySubmissions do

    Check 1: Sybil Score
    sybilScore ← getSybilScore(walletId)
    if sybilScore.isBanned then
        continue                                       ▷ Skip banned wallets
    end if

    Check 2: Staleness (2-epoch timeout)
    activityAge ← currentTime − submission.timestamp
    if activityAge > MAX_ACTIVITY_AGE then
        continue                                       ▷ Skip stale submissions
    end if

    Check 3: Activity Threshold
    AS ← submission.activityScore
    if AS < ACTIVITY_THRESHOLD then
        continue                                       ▷ Below threshold
    end if

    Check 4: Reputation (with Sybil penalty)
    baseReputation ← reputation[walletId]
    finalReputation ← baseReputation − sybilScore.reputationPenalty
    if finalReputation < MIN_REPUTATION then
        continue                                       ▷ Reputation too low
    end if

    Add to eligible list
    eligibleValidators.append({walletId, AS, finalReputation})
end for

Fallback: If no eligible validators, use all participants
if |eligibleValidators| = 0 and |activitySubmissions| > 0 then
    for walletId, submission ∈ activitySubmissions do
```

if $activityAge < MAX\_ACTIVITY\_AGE$ **then**
$eligibleValidators.append(\{walletId, submission.AS, reputation[walletId]\})$
end if
end for
end if

return $eligibleValidators$

Activity Score Calculation

Algorithm 4: Calculate Activity Score (AS)

Require: $healthData$ containing $steps$, $heartRate$, $activeMinutes$
Ensure: $AS \in [0, 100]$ representing activity level on 0–100 scale

Constants:
$MAX\_STEPS \leftarrow 7500$ ▷ Research-backed threshold (JAMA study)
$OPTIMAL\_HR\_MIN \leftarrow 60$
$OPTIMAL\_HR\_MAX \leftarrow 100$
$MAX\_ACTIVE\_MINUTES \leftarrow 60$
$STEPS\_WEIGHT \leftarrow 0.8$ ▷ 80% weight
$HR\_WEIGHT \leftarrow 0.1$ ▷ 10% weight
$ACTIVE\_WEIGHT \leftarrow 0.1$ ▷ 10% weight

Step Score Calculation (80% weight):
$S_{normalized} \leftarrow \min\left(\frac{steps}{MAX\_STEPS}, 1.0\right)$
$stepsScore \leftarrow S_{normalized} \times STEPS\_WEIGHT \times 100$

Heart Rate Score Calculation (10% weight):
$baseScore \leftarrow HR\_WEIGHT \times 100$
if $OPTIMAL\_HR\_MIN \leq heartRate \leq OPTIMAL\_HR\_MAX$ **then**
$HR_{score} \leftarrow baseScore$ ▷ Optimal range: 100% of base
else if $(OPTIMAL\_HR\_MIN - 10) \leq heartRate \leq (OPTIMAL\_HR\_MAX + 10)$ **then**
$HR_{score} \leftarrow baseScore \times 0.8$ ▷ Acceptable: 80% of base
else if $heartRate > 0$ **then**
$HR_{score} \leftarrow baseScore \times 0.5$ ▷ Suboptimal: 50% of base
else
$HR_{score} \leftarrow 0.0$ ▷ No data - smartwatch required
end if

Active Minutes Score Calculation (10% weight):
$AM_{normalized} \leftarrow \min\left(\frac{activeMinutes}{MAX\_ACTIVE\_MINUTES}, 1.0\right)$
$activeMinutesScore \leftarrow AM_{normalized} \times ACTIVE\_WEIGHT \times 100$

Total Activity Score (0–100 scale):
$AS \leftarrow stepsScore + HR_{score} + activeMinutesScore$
return AS ▷ Returns 0–100

4.4.6 Weighted Score Calculation

Algorithm 5: Calculate Weighted Score (WS)

Require: AS (Activity Score on 0–100 scale), RH (Reputation Score on 0–100 scale)
Ensure: $WS \in [0, 1]$ representing combined score for validator selection

Normalize scores to 0–1 scale:
$AS_{norm} \leftarrow AS/100$
$RH_{norm} \leftarrow RH/100$

Weighted Score Formula:
$WS \leftarrow 0.7 \times AS_{norm} + 0.3 \times RH_{norm}$

▷ 70% weight on recent activity, 30% on historical reputation; both normalized
return WS ▷ Returns 0–1

4.4.7 Reputation Update

Algorithm 6: Update Validator Reputation

Require: $walletId$, $event \in \{SUCCESSFUL\_BLOCK, FAILED\_BLOCK, INACTIVE\}$
Ensure: Updated reputation score $RH \in [0.1, 1.0]$

Constants:
$SUCCESSFUL\_REWARD \leftarrow 0.05$ ▷ +5% for successful block
$FAILED\_PENALTY \leftarrow 0.10$ ▷ -10% for failed block
$DECAY\_RATE \leftarrow 0.999845$ ▷ 20% daily decay
$MIN\_REPUTATION \leftarrow 0.1$ ▷ 10% floor (minimum allowed)
$INITIAL\_REPUTATION \leftarrow 0.5$ ▷ 50% starting value for new wallets
$MAX\_REPUTATION \leftarrow 1.0$ ▷ 100% ceiling

$RH \leftarrow reputation.get(walletId, INITIAL\_REPUTATION)$ ▷ Get current or default to 50%

if $event = SUCCESSFUL\_BLOCK$ **then**
    $RH \leftarrow \min(RH + SUCCESSFUL\_REWARD, MAX\_REPUTATION)$
else if $event = FAILED\_BLOCK$ **then**
    $RH \leftarrow \max(RH - FAILED\_PENALTY, MIN\_REPUTATION)$
else if $event = INACTIVE$ **then**
    $epochsInactive \leftarrow currentEpoch - lastActiveEpoch[walletId]$
    $RH \leftarrow RH \times DECAY\_RATE^{epochsInactive}$
    $RH \leftarrow \max(RH, MIN\_REPUTATION)$
end if

$reputation[walletId] \leftarrow RH$
return RH

Important: New wallets start at $INITIAL\_REPUTATION = 0.5$ (50%), providing a neutral starting point. The $MIN\_REPUTATION = 0.1$ (10%) is the floor that reputation can never fall below, not the starting value.

Security Algorithms (Production Enhancements)

The new implementation introduced strong security measures that were fully absent from the simulation. This addressed increasing threats discovered in our real-world testing and threat modeling.

Sybil Attack Detection Algorithm 7 shows the multi-layered Sybil defense that has been implemented through (1) hash-based conflict resolution and (2) preventing address grinding attacks. This was critical for production deployment as the simulation assumed that all the participants of the network were honest which is not the case in real-world scenarios.

Algorithm 7: Detect Sybil Attacks (Production)

Require: $walletId$, $activityScore$, $deviceId$, $steps$, $heartRate$, $activeMinutes$, $currentEpoch$
Ensure: $isAccepted \in \{true, false\}$

Constants:
$ACTIVITY\_WINDOW \leftarrow 40000$ ms ▷ Submissions close at t=40s
$MAX\_ACTIVITY\_AGE \leftarrow 300000$ ms ▷ 5 minutes

Security Check 1: Timestamp Verification
$currentTime \leftarrow getCurrentTimeMs()$
$epochStartTime \leftarrow getEpochStartTime()$
$elapsed \leftarrow currentTime - epochStartTime$

▷ Check if within activity submission window
if $elapsed > ACTIVITY\_WINDOW$ **then**
    log("$Activityrejected : Submissionwindowclosed$")
    return $false$ ▷ Activity window closed at t=40s
end if

if $\neg verifyTimestampInCurrentEpoch(currentTime)$ **then**
    return $false$ ▷ Outside epoch window
end if
if $\neg verifyTimestampFreshness(currentTime, MAX\_ACTIVITY\_AGE)$ **then**
    return $false$ ▷ Too old (¿5 minutes)
end if

Security Check 2: Replay Attack Detection
if $walletId \in submissionTimestamps$ **then**
    $lastTimestamp \leftarrow submissionTimestamps[walletId]$
    if $currentTime = lastTimestamp$ **or** $currentTime < lastTimestamp$ **then**
        $flagSubmission(walletId,$"$replay\_attack$"$)$
        return $false$ ▷ Replay detected
    end if
end if

Security Check 3: Duplicate Health Data (Hash-Based Selection)
$fingerprint \leftarrow SHA256(steps \parallel heartRate \parallel activeMinutes \parallel currentEpoch)$

```
if fingerprint ∈ healthDataFingerprints then
    existingWallet ← healthDataFingerprints[fingerprint]
    if existingWallet ≠ walletId then
        Hash-based random selection (prevents address grinding):
        duplicateWallets ← [existingWallet, walletId]
        sort(duplicateWallets)                          ▷ For consistency across all nodes

                                  ▷ Use previous block hash for unpredictable randomness
        previousHash ← lastBlock.hash
        seedData ← fingerprint ‖ previousHash
        seed ← SHA256(seedData)
        seedInt ← hexToInt(seed[0 : 16])
        selectedIndex ← seedInt  mod |duplicateWallets|

        winner ← duplicateWallets[selectedIndex]
        losers ← [w | w ∈ duplicateWallets, w ≠ winner]

                                                ▷ Blacklist ALL losers (permanent ban)
        for loser ∈ losers do
            banWallet(loser)                                          ▷ Permanent ban
            deleteSubmission(loser)
            flagSubmission(loser, "exact_duplicate_hash", relatedWallet = winner)
            broadcastRejection(loser, winner, "hash_selection", selectedIndex)
        end for

        if walletId ∈ losers then
            return false                                  ▷ This wallet lost hash selection
        end if
    end if
end if

Security Check 4: Device ID Binding
if deviceId ≠ NULL then
    if deviceId ∈ deviceIdWallets then
        existingWallet ← deviceIdWallets[deviceId]
        if existingWallet ≠ walletId then
            flagSubmission(walletId, "duplicate_device",
                                originalWallet = existingWallet)
            return false                                ▷ Same device, different wallet
        end if
    end if
end if

Security Check 5: Similarity Detection (95% threshold)
for existingWallet, existingSub ∈ activitySubmissions do
    if existingWallet = walletId then
        continue                                                      ▷ Skip self
    end if
```

```
83:
84:     similarity ← calculateSimilarity(steps, heartRate, activeMinutes,
85:                                      existingSub.steps, existingSub.heartRate,
86:                                      existingSub.activeMinutes)
87:
88:     if similarity > 0.95 then                          ▷ 95%+ similar - likely Sybil
89:         Hash-based random selection (same as Check 3)
90:         similarWallets ← [existingWallet, walletId]
91:         sort(similarWallets)
92:         seed ← SHA256(similarityFingerprint ‖ previousHash)
93:         selectedIndex ← hexToInt(seed[0 : 16]) mod |similarWallets|
94:         winner ← similarWallets[selectedIndex]
95:         losers ← [w | w ∈ similarWallets, w ≠ winner]
96:                                                  ▷ Ban losers and accept winner
97:         for loser ∈ losers do
98:             banWallet(loser)
99:         end for
100:         if walletId ∈ losers then
101:             return false
102:         end if
103:     end if
104: end for
105:
106: All checks passed - Accept submission
107: activitySubmissions[walletId] ← submission
108: healthDataFingerprints[fingerprint] ← walletId
109: deviceIdWallets[deviceId] ← walletId
110: submissionTimestamps[walletId] ← currentTime
111:
112: return true
```

Health Data Similarity Calculation Algorithm 8 uses Euclidean distance in normalized 3D space to detect near-duplicate submissions.

Algorithm 8: Calculate Health Data Similarity

Require: $steps_1$, hr_1, am_1, $steps_2$, hr_2, am_2
Ensure: $similarity \in [0, 1]$ where 1.0 = identical, 0.0 = completely different

```
1:
2: Constants:
3: MAX_STEPS ← 30000
4: MAX_HR ← 180
5: MAX_AM ← 300
6:
7: Normalize to [0, 1] scale:
8: s1 ← steps1/MAX_STEPS
9: s2 ← steps2/MAX_STEPS
10: h1 ← hr1/MAX_HR
11: h2 ← hr2/MAX_HR
```

$a_1 \leftarrow am_1/MAX\_AM$
$a_2 \leftarrow am_2/MAX\_AM$

Calculate Euclidean distance in 3D space:
$distance \leftarrow \sqrt{(s_1 - s_2)^2 + (h_1 - h_2)^2 + (a_1 - a_2)^2}$
$maxDistance \leftarrow \sqrt{3}$ ▷ Maximum possible distance

Convert distance to similarity:
$similarity \leftarrow 1.0 - \frac{distance}{maxDistance}$

return *similarity*

Network Resilience Algorithms (Production)

The production implementation revealed that there are tons of network unreliability issues absent in simulation. Through the following algorithms, we tried to mitigate these issues as much as possible to ensure an operational blockchain network.

Multi-Node Peer Discovery Dynamic peer discovery has been implemented through Algorithm 9, replacing the simulation's static node list.

Algorithm 9: Discover Full Node Peers

Require: *bootstrapNodeUrl* (hardcoded trusted node)
Ensure: *knownNodes* list populated

Constants:
$BOOTSTRAP\_NODE \leftarrow$ BOOTSTRAP_NODE_URL ▷ Trusted bootstrap node endpoint
$REDISCOVERY\_INTERVAL \leftarrow 300000$ ms ▷ Re-discover every 5 minutes

Step 1: Query bootstrap node for active peers
$knownNodes \leftarrow []$

try:
    $response \leftarrow HTTP\_GET(BOOTSTRAP\_NODE + "/peers/active", timeout = 5s)$

▷ Add bootstrap node itself
    $knownNodes.append(\{url : BOOTSTRAP\_NODE, version : "bootstrap"\})$

▷ Add discovered peers

for $peer \in response.peers$ **do**
        $knownNodes.append(\{url : peer.url, version : peer.version, chainLength : peer.chainLength\})$

end for

```
    log("Discovered" + |knownNodes| + "fullnodes")

except (timeout, error):
                                          ▷ Fallback: Use only bootstrap node
    knownNodes ← [{url : BOOTSTRAP_NODE, version : "bootstrap"}]
    log("Peerdiscoveryfailed, usingbootstrapnodeonly")
end try

Step 2: Schedule periodic re-discovery
scheduleTask(discoverPeerNodes, REDISCOVERY_INTERVAL)

return knownNodes
```

Majority Voting for Eligible List Algorithm 10 implements Byzantine fault-tolerant consensus over eligible validators, critical for production security.

Algorithm 10: Fetch Eligible Validators with Majority Vote

Require: *knownNodes*, *epoch*
Ensure: *eligibleCandidates* list with majority consensus

```
if |knownNodes| = 0 then
    discoverPeerNodes()
end if

responses ← []

Step 1: Fetch from all nodes in parallel
parallel for node ∈ knownNodes:
    try:
        service ← createHttpClient(node.url)
        response ← GET(service + "/consensus/validators", timeout = 5s)
        responses.append(response)
    except (timeout, error):
        log("Failedtofetchfrom" + node.url)
    end try
end parallel for

Step 2: Find majority consensus
if |responses| = 0 then
    return []
end if
                                          ▷ Group responses by eligible list content
groupedResponses ← {}
for response ∈ responses do
    walletIds ← sort([validator.wallet_id | validator ∈ response.eligible_validators])
    key ← join(walletIds, ",")
    if key ∉ groupedResponses then
```

$groupedResponses[key] \leftarrow []$
end if
$groupedResponses[key].append(response)$
end for

▷ Find the most common response (majority)
$majorityGroup \leftarrow \arg\max_{g \in groupedResponses} |groupedResponses[g]|$
$majorityCount \leftarrow |groupedResponses[majorityGroup]|$
$requiredCount \leftarrow \lceil |knownNodes| \times \frac{2}{3} \rceil$

Step 3: Verify 2/3 majority
if $majorityCount < requiredCount$ **then**
error("$Nomajorityconsensus - Networkmaybepartitioned!$")
return []
end if

log("$Majorityconsensus:$"+$majorityCount$+"/"+$|knownNodes|$+"$nodesagree$")

$consensusResponse \leftarrow groupedResponses[majorityGroup][0]$
$eligibleCandidates \leftarrow [validator \mid validator \in consensusResponse.eligible\_validators]$

return $eligibleCandidates$

Transaction Processing with Fees (Production)

Production implementation introduced dual incentive model: activity rewards + transaction fees. This addresses spam prevention absent from simulation.

Algorithm 11: Process Block Transactions (Production)

Require: *block* containing transactions and validator address
Ensure: State updated with balances and fees collected

Constants:
$TRANSACTION\_FEE \leftarrow 0.01$ PHY ▷ Standard user transaction fee
$SYSTEM\_TX\_FEE \leftarrow 0.0$ PHY ▷ System reward transactions have NO fee

Step 1: Give activity reward to validator
$reward \leftarrow block.activityScore \times 10.0$ ▷ Up to 10 PHY per block
$balances[block.validator] \leftarrow balances[block.validator] + reward$

Step 2: Process transactions and collect fees
$totalFees \leftarrow 0.0$

for $tx \in block.transactions$ **do**
if $tx.type \notin \{$"$validator\_reward$","$activity\_reward$"$\}$ **then**
▷ Regular user transfer transaction (has fee)
$txFee \leftarrow tx.fee$ ▷ Default 0.01 PHY
$balances[tx.sender] \leftarrow balances[tx.sender] - (tx.amount + txFee)$
$balances[tx.recipient] \leftarrow balances[tx.recipient] + tx.amount$

```
        totalFees ← totalFees + txFee
    else
                                          ▷ System reward transaction (fee = 0.0)
        balances[tx.recipient] ← balances[tx.recipient] + tx.amount
    end if
end for

Step 3: Give all transaction fees to block validator
if totalFees > 0 then
    balances[block.validator] ← balances[block.validator] + totalFees
end if

                    ▷ Total validator earnings = Activity Reward + Transaction Fees
```

Simulation Algorithms (Phase 1 - SimBlock)

For comparison and completeness, the algorithms Algorithm 12, Algorithm 13, Algorithm 14 and Algorithm 15 represent the initial simulation implementation using SimBlock. These were used for a "proof of concept" to validate our idea; however, they did not meet any production requirements such as security hardening, network redundancy, or real-world constraints of devices. The simulation showed us the overall concept of our ideas which we later validated and developed from our testing and experiments.
Algorithm 13 uses the block creation algorithm to create a new block with the necessary metadata. This includes an increment of the block number from the last block created in the block chain, a link to the previous blocks hash, and recording of the current time stamp. The miner/validator will populate the block with all pending transaction from the network. In addition to the validators proof of physical work (derived from their activity hash), the block will be signed by the validator. The block producing validator's identity is established in the committee selection phase prior to each consensus round. The selected block proposer will embed their identity into the block via their digital signature. Finally, the protocol calculates and assigns the validator's reward for block creation before adding the block to the blockchain.
The validation algorithm in Algorithm Algorithm 14 performs three critical checks on a proposed block. When a candidate block is proposed by the the validator, Algorithm 14 is executed by the remaining validators in the committee to validate the block before acceptance. Firstly, it verifies the cryptographic integrity of the block by validating its hash. Secondly, it authenticates the block's origin by checking the validator's signature. Finally, it ensures all contained transactions are individually valid according to the network's consensus rules. Only blocks passing all three checks simultaneously are accepted into the blockchain.
The difficulty adjustment algorithm, Algorithm 12 dynamically modifies mining targets based on validator activity trends and block timing. Starting with the parent block's target value, it calculates the difference in activity scores between consecutive blocks and compares it against the expected time window (on average 2000 ms). The expected block interval is set to 2000 ms based on common design considerations in distributed consensus systems. This value serves as a reference point for illustrating protocol behavior and is not fundamental to the correctness of PoPA. Furthermore, A positive correlation triggers a 5% difficulty increase to maintain network stability, while negative correlation decreases

Algorithm 12 PoPA Consensus Process Flow

Require: Validator list V, Target Score Threshold T_{ws}
Ensure: New block added to blockchain if valid

Initialize Blockchain $\leftarrow \emptyset$
Initialize TransactionPool $\leftarrow \emptyset$
$X \leftarrow 2$
if Last block is not Genesis **then**
    $X \leftarrow \text{Average}(X, \text{Timestamp}_n - \text{Timestamp}_{n-1})$
end if
loop every X milliseconds
    for all deviceInput $\in$ incomingSmartWatchData **do**
        if validateSignature(deviceInput) **then**
            activityHash $\leftarrow$ extractHash(deviceInput)
            updateValidatorActivity(deviceInput.validatorID, activityHash)
        else
            rejectInput ("Invalid device data")
        end if
    end for
    for all $v \in V$ **do**
        $v.WS \leftarrow 0.7 \cdot v.AS + 0.3 \cdot v.RH$
    end for
    $C \leftarrow \text{filter}(V,\ v \rightarrow v.WS \geq T_{ws})$
    validator $\leftarrow$ **argmax**$(v \in C)$ of $v.MT$
    if validator $\neq$ null **then**
        block $\leftarrow$ **createBlock**(validator)
        if validateBlock(block) **then**
            Blockchain.append(block)
            TransactionPool.clear()
            log("Block added: " + block.number)
        else
            reject("Invalid block")
        end if
    else
        adjustDifficulty(block)
    end if
end loop

Algorithm 13 Block Creation Code

function CREATEBLOCK(validator)
    newBlock.number $\leftarrow$ Blockchain.last().number +1
    newBlock.previousHash $\leftarrow$ Blockchain.last().hash
    newBlock.timestamp $\leftarrow$ currentTime()
    newBlock.transactions $\leftarrow$ getPendingTransactions()
    newBlock.proofOfPhysicalWork $\leftarrow$ validator.activityHash
    newBlock.signature $\leftarrow$ sign(validator, newBlock)
    newBlock.reward $\leftarrow$ calcReward(validator)
    return newBlock
end function

Algorithm 14 Block Validation Code

function VALIDATEBLOCK(block)
  return $isHashValid(block) \wedge verifySignature(block) \wedge$
    $all\_tx\_valid(block.transactions)$
end function

difficulty by 5%. The chosen percentage is for demonstrative purposes and may be tuned in alternative implementations. The final target value is clamped between 0.1 and 1.0 to prevent extreme highs and lows to achieve balanced participation.

Algorithm 15 Adjust PoPA Mining Difficulty

procedure ADJUSTDIFFICULTY(parentBlock)
  $newTarget \leftarrow parentBlock.$`nextTarget`
  $activityDelta \leftarrow this.activityScore - parentBlock.activityScore$
  $timeDelta \leftarrow this.time - parentBlock.time - 2000$
  $\delta \leftarrow activityDelta \times timeDelta$
  if $\delta \geq 0$ **then**
    $newTarget \leftarrow newTarget \times 0.95$
  else
    $newTarget \leftarrow newTarget \times 1.05$
  end if
  return $\max(0.1, \min(1.0, newTarget))$
end procedure

4.4.8 Verifiable PoPA Block Structure

A `PoPA_Block` is the fundamental unit of data stored in the PoPA blockchain, encapsulating both physical activity proof and consensus-relevant metadata. Each block contains a sequential index, a cryptographic link to the previous block via `previous_hash`, and a list of validated transactions. The validator's address identifies the block proposer, while their Activity Score (AS), Reputation Score (RH) justify their selection.

$$PoPA\_Block \triangleq (\texttt{index}, \texttt{previous\_hash}, \texttt{timestamp}, \texttt{transactions[]}, \texttt{validator}, AS, RH, WS, R, \texttt{nonce}, \texttt{hash}, \texttt{Sig}_N, \texttt{epoch})$$

The reward R reflects the total incentive earned (activity-based reward plus transaction fees). A nonce provides uniqueness, the epoch number ties the block to a specific consensus round, and the validator's digital signature (Sig_N) ensures authenticity. The block hash is calculated over these fields (excluding the hash itself and signature) to create an unique cryptographic commitment that is immutable once stored on chain. The whole process makes the structure tamper-proof.

Table 4.11: Structure of a PoPA Block

Field	Description
`index`	Sequential block number (0 for genesis, increments by 1)
`previous_hash`	SHA256 hash of the previous block (creates immutable chain)
`timestamp`	Unix timestamp (seconds) when block was created
`transactions[]`	List of validated transactions included in this block
`validator`	Address of the selected validator who proposed this block
AS	Activity Score (0-1 normalized) of the validator at block creation
RH	Reputation Score (0-1 normalized) of the validator
WS	Weighted Score: $WS = 0.7 \cdot AS + 0.3 \cdot RH$
R	Total reward: $R_{activity} + R_{fees}$ where $R_{activity} = WS \times r_0$
`nonce`	Random value for uniqueness and tie-breaking
`hash`	SHA256 hash of this block (calculated over fields below)
$\texttt{Sig}_N$	ECDSA signature of the validator over the block hash
`epoch`	Epoch number (60-second rounds) when block was created

Block Hash Calculation: The block hash is computed using SHA256 over a JSON representation of the block fields, excluding the `hash` and `signature` fields themselves. Specifically, the hash includes: `index`, `previous_hash`, `timestamp`, `transactions` (as deterministically sorted dictionaries), `validator`, `activity_score`, `reputation_score`, `weighted_score`, `reward`, and `nonce`. Fields excluded from the hash calculation are: `hash` (self-referential), `signature` (applied after hashing), `activity_proof` (raw data stored separately), and `epoch` (derivable from timestamp).

4.5 Transaction and Activity Submission Structures

PoPA distinguishes between two types of data submissions: **financial transactions** (token transfers between wallets) and **activity submissions** (physical activity proofs for consensus participation). Both are cryptographically signed but serve different purposes.

4.5.1 Financial Transaction Structure

A financial transaction $T_{financial}$ represents the transfer of PHY tokens between wallet addresses. These transactions are included in blocks and incur a standard fee.

$$T_{financial} \triangleq \langle \texttt{tx\_id},\ \texttt{sender},\ \texttt{recipient},\ \texttt{amount},\ \texttt{fee},\ \texttt{timestamp},\ \texttt{signature},\ \texttt{public\_key},\ \texttt{type} \rangle$$

Field	Description
`tx_id`	Transaction ID (SHA256 hash of transaction fields)
`sender`	Wallet address of sender (40-character hex)
`recipient`	Wallet address of recipient (40-character hex)
`amount`	Amount of PHY tokens to transfer
`fee`	Transaction fee (0.01 PHY for user transfers, 0.0 for system rewards)
`timestamp`	Unix timestamp in milliseconds
`signature`	ECDSA signature over transaction hash
`public_key`	Sender's public key (64-byte hex, uncompressed)
`type`	Transaction type: `transfer`, `activity_reward`, `validator_reward`

Transaction ID Calculation:

$$TxID = \text{SHA256}(\texttt{sender} \parallel \texttt{recipient} \parallel \texttt{amount} \parallel \texttt{fee} \parallel \texttt{timestamp} \parallel \texttt{public\_key}) \tag{4.1}$$

4.5.2 Activity Submission Structure

An activity submission $T_{activity}$ represents the contribution of physical activity data for consensus participation. It is not a financial transaction but rather a proof of physical work.

$$T_{activity} \triangleq \langle \texttt{wallet\_id},\ AS,\ H(P),\ TS,\ \texttt{device\_id},\ \texttt{Sig}_N \rangle$$

Field	Description
`wallet_id`	Wallet address of the participant
AS	Activity Score (0-100 scale, normalized to 0-1 for consensus)
$H(P)$	Hash of physical proof: SHA256(Steps ‖ HeartRate ‖ ActiveMinutes ‖ Epoch)
TS	Timestamp of submission (Unix milliseconds)
`device_id`	Unique device identifier (for Sybil prevention)
$\texttt{Sig}_N$	ECDSA signature of the participant

4.5.3 Cryptographic Field Calculations

To ensure cryptographic integrity and non-repudiation, the following cryptographic operations are used:

For Financial Transactions

- **Transaction ID:** Deterministic hash of transaction fields ensures uniqueness and prevents replay attacks.

$$TxID = \text{SHA256}(\texttt{sender} \parallel \texttt{recipient} \parallel \texttt{amount} \parallel \texttt{fee} \parallel \texttt{timestamp} \parallel \texttt{public\_key}) \tag{4.2}$$

- **Signature:** ECDSA signature over the message bytes (UTF-8 encoding of TxID) using SECP256k1 curve.

$$\texttt{signature} = \text{ECDSA\_SIGN}(PrivateKey, \texttt{TxID.toBytes(UTF8)}) \tag{4.3}$$

- **Verification:** Full nodes verify that the signature is valid and that the sender address matches the public key.

$$\begin{aligned} \text{Valid} \iff & \text{ECDSA\_VERIFY}(PublicKey, TxID, Signature) \\ & \wedge (SenderAddress = \\ & \quad \text{hex}(\text{SHA256}(PublicKey))[0:20]) \end{aligned} \tag{4.4}$$

For Activity Submissions

- **H(P) (Health Data Fingerprint):** A cryptographic commitment to the raw sensor data, used for Sybil detection by comparing fingerprints.

$$H(P) = \text{SHA256}(Steps \parallel HeartRate \parallel ActiveMinutes \parallel Epoch) \tag{4.5}$$

- **Activity Score:** The activity score is calculated locally on the mobile device using the formula defined in Algorithm 4.

$$AS_{0-100} = 0.8 \cdot S_{score} \cdot 100 + 0.1 \cdot HR_{score} \cdot 100 + 0.1 \cdot AM_{score} \cdot 100 \tag{4.6}$$

  The scores are normalized to [0,1] before scaling to 0-100 for display to humans.

- **Signature:** Activity submissions are signed to prove authenticity and prevent tampering.

$$Sig\_N = \text{ECDSA\_SIGN}(PrivateKey, H(ActivityData)) \tag{4.7}$$

Chapter 5

Implementation

In this chapter, we detail the implementation of the Proof of Physical Activity (PoPA) consensus algorithm as described in Chapter 4. We discuss the tools and technologies used, the two-phase implementation approach (simulation and live proof-of-concept), and provide insights into the challenges faced and solutions devised during development.

5.1 Tools for Implementation

Our implementation has been divided into two phases. For our initial feasibility analysis, we implemented the algorithm in the SimBlock blockchain simulator [12] (Phase 1). For the production proof-of-concept (Phase 2), we developed a mobile application that runs on Android devices and communicates with full nodes over a publicly reachable network. So, in phase-2, the application is developed using Android Studio with Java/Kotlin for mobile nodes. The networking layer uses Retrofit for RESTful API communication. Cryptographic functions (hashing, digital signatures) are implemented using Bouncy Castle libraries. Google Health Connect SDK is integrated for standardized access to physiological data. The blockchain data structure and consensus logic are custom implemented based on the algorithms defined in Chapter 4.

5.2 Phase 1: Simulation (Feasibility Analysis)

To validate core feasibility before a live rollout, we implemented the PoPA consensus in SimBlock [12]. Using the simulation consensus flow in Algorithm 12, we reproduced end-to-end block production driven by activity score AS and reputation history RH, and verified reward assignment and dynamic thresholding.
In a representative run, the simulator produced a chain of **10 blocks** across **10 initialized nodes** (as shown in Figure 5.2 and Figure 5.3). This confirmed that nodes with higher *Weighted Score* $WS = 0.7 \cdot AS + 0.3 \cdot RH$ are preferentially selected to propose blocks once they pass the mining threshold.
The implementation of the block structure in the SimBlock Blockchain Simulator (Phase 1 simulation only) is shown in Table 5.1. Note that the simulation used simplified metrics (motion intensity) different from the production implementation (active minutes) described in Table 4.11.
Each node was initialized with randomized activity and a fixed reputation score, and block creation proceeded based on their computed *Weighted Score (WS)* in a working

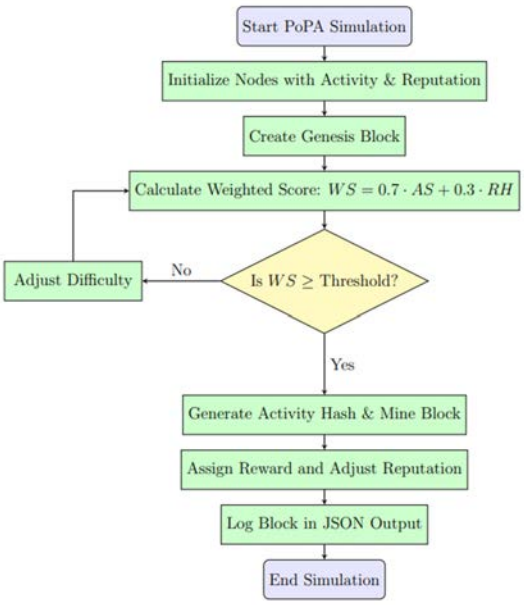


Figure 5.1: Simblock Simulation Flow

process similar to the flow in Figure 5.1. The block structure was implemented with slight variation as given by Table 5.1.

```
> Task :simulator:run
[START] PoPA Simulation...
[INIT] Node 1 | Activity=0.748 | Reputation=1.000
[INIT] Node 2 | Activity=0.788 | Reputation=1.000
[INIT] Node 3 | Activity=0.759 | Reputation=1.000
[INIT] Node 4 | Activity=0.731 | Reputation=1.000
[INIT] Node 5 | Activity=0.752 | Reputation=1.000
[INIT] Node 6 | Activity=0.751 | Reputation=1.000
[INIT] Node 7 | Activity=0.767 | Reputation=1.000
[INIT] Node 8 | Activity=0.737 | Reputation=1.000
[INIT] Node 9 | Activity=0.761 | Reputation=1.000
[INIT] Node 10 | Activity=0.734 | Reputation=1.000
Genesis 0
[GENESIS] Block by Node 1
[START] Mining blocks...
chain 1
Next target0.6649999999999999
```

Figure 5.2: Simblock Node Initiation

The simulation log confirms that eligible nodes were selected in a distributed manner, and each mined block triggered a consistent adjustment of the *difficulty/target threshold*, verifying the proper execution of dynamic tuning based on activity trends and timing. Concretely, the simulation uses the adjustment logic in Algorithm 15, where

$$\delta = (V_{AS} - V_{AS_{prev}}) \cdot (\Delta t - 2000) \ ,$$

and increases/decreases target by 5% depending on the sign of δ, with clamping to avoid oscillations (see Chapter 4).
However, several limitations are inherent to this simulation:

Table 5.1: PoPA Block Structure in SimBlock Implementation

Field	Code Variable	Description
`BlockID`	`block.getId()`	Unique identifier of the block (inherited from the parent `Block` class).
`H(P)`	`activityHash`	SHA-256 hash of activity proof generated from activity score, device ID, and timestamp.
`AS`	`activityScore`	Normalized physical activity score based on synthetic heart rate, steps, and motion intensity.
`RH`	`reputationScore`	Validator's reputation score, adjusted up/down based on mining success or failure.
`WS`	`0.7 * AS + 0.3 * RH`	Weighted score used to determine eligibility to mine a block.
`R`	`reward = AS * baseRewardRate`	Block reward calculated as a function of the activity score and fixed rate per unit score.
`Nonce`	`nonce`	Random value generated per block to simulate uniqueness and variability.
`TS`	`System.currentTimeMillis()`	Timestamp at which the block was created.

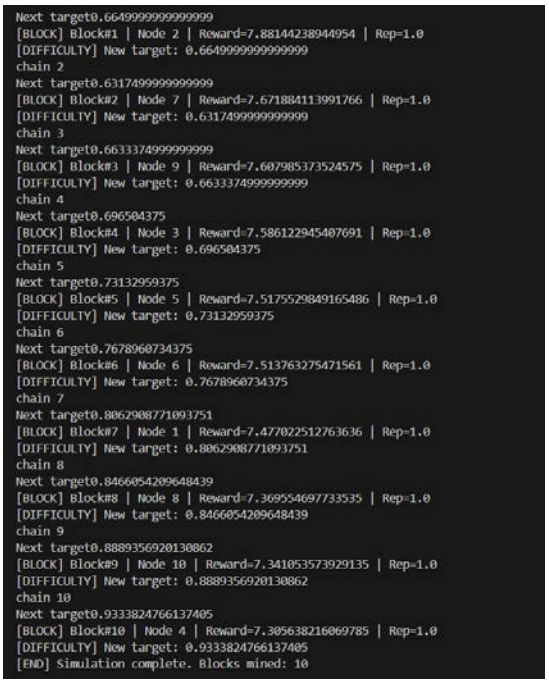


Figure 5.3: Simblock PoPA Simulation

- **No transaction propagation:** SimBlock currently does not support the inclusion or validation of actual transactions in blocks. As a result, there are several important performance metrics that are still untested, including network throughput, mempool behavior, and congestion handling.

- **There is no modeling of network delay:** The time it takes for blocks to propagate through the network, the time it takes to resolve forks and achieve consensus are all abstracted away. Latency and bandwidth can have significant impacts on both the efficiency and security of the consensus process in actual systems.
- **Simplified validator and network behavior:** Although the simulation has captured the core of the PoPA mechanism (activity, reputation, reward), it has not yet incorporated devicespecific metadata (e.g. age, medical condition), nor has it incorporated the health factors, necessary to ensure fair incentives across a large scale deployment.

Although the simulation has demonstrated the possibility of PoPA's mechanism in a lightweight blockchain simulator, the block creation logic, reward structure, and difficulty calibration all function as intended in the proof of concept. The production version (Phase 2) of the PoPA implementation uses a static activity threshold (30%) rather than a dynamic difficulty adjustment, to prioritize predictability and fairness over adaptability and complexity. The component-level architecture of the PoPA implementation is depicted in Figure 5.4.

5.3 Phase 2: Proof of Concept on a Publicly Accessible Network

Using the simulation as a base, we deployed a live proof-of-concept over a publicly accessible PoPA test network consisting of multiple full nodes. This phase implemented the production versions of the algorithms described in Chapter 4, and addressed real-world limitations imposed by asynchronous networking, device-specific timing and adversary behavior.

5.3.1 Application Module

The user-facing mobile application will be the primary interface for users to engage with the PoPA consensus process. The mobile application will be developed utilizing Android Studio and Java/Kotlin, and will leverage the Google Health Connect SDK to allow secure access to physiological data. The app collects step count and active minutes from connected smartwatches and smartphones, computes the Activity Score (AS), and prepares activity proofs for submission to the validator network.
The app features a user-friendly interface (Figure 5.5) that displays real-time activity metrics, submission status, and rewards earned. It securely signs activity proofs using the device's private key before transmitting them to the network. During the launch of the application, it asks for necessary permissions to access health data and network connectivity that allows for the bridge with the Health Connect SDK as seen from Figure 5.6.
It provides a few options for logging in as the user can either create a new wallet, import an existing wallet from the phone's cache memory, or login using the private key directly as illustrated in Figure 5.7.
Inside the app, as seen from Figure 5.5 we see any live balance of PhysiCoin (POC) tokens the user has, along with the total number of blocks mined by the user as shown in Figure 5.9. Recent block history is also displayed, showing block hashes, timestamps, and

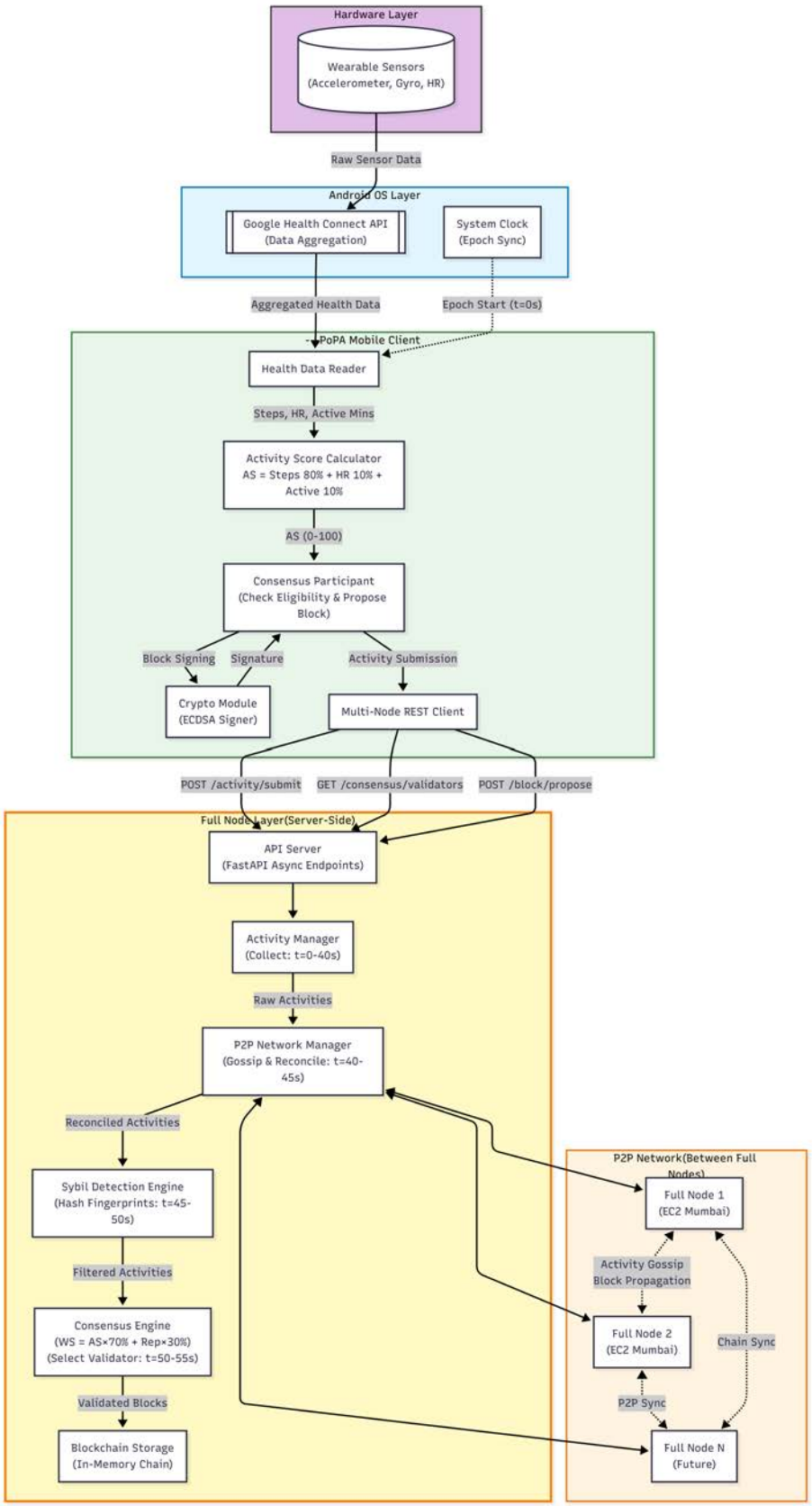


Figure 5.4: Component-Level Architecture of PoPA Implementation

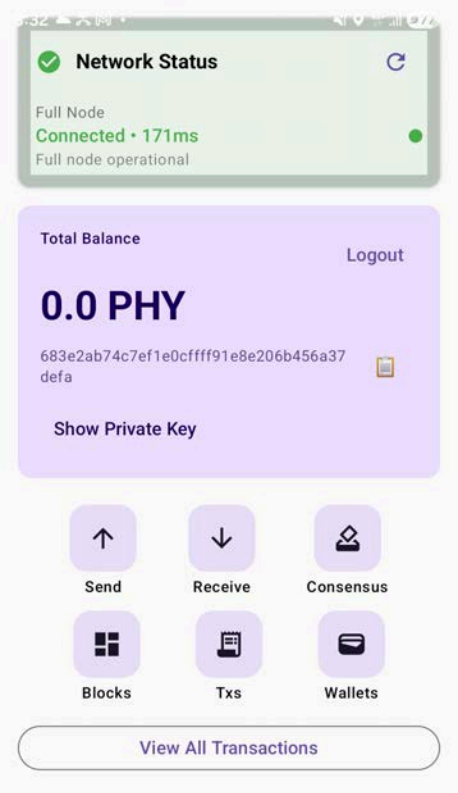


Figure 5.5: PoPA Mobile Application Interface

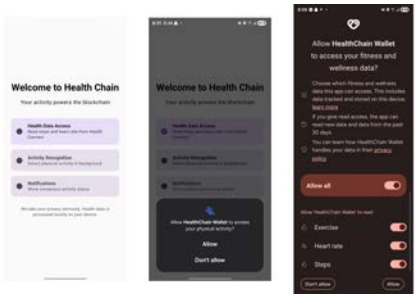


Figure 5.6: PoPA App Permissions

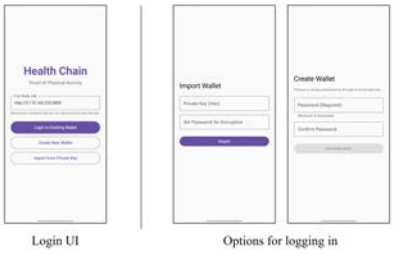


Figure 5.7: PoPA App Login Options

rewards earned. The app periodically fetches updated balances and block data from the network to keep users informed of their participation status. It utilizes the notification system to run in background to consistently send activity data to the network even when the app is not actively open on the screen to participate in the consensus. The app sends data every 20 seconds to the full nodes in the network as shown in Figure 5.8. However, if no Health Connected SDK supported smartwatch is connected to the phone, the app fails to collect the heart rate of the user and thus cannot compute the activity score to send to the network. So, the data sent is not accepted by the full nodes in the network for validation.

Figure 5.8: PoPA App Background Data Submission

5.3.2 Network Module

The network module consists of multiple full nodes deployed on cloud infrastructure (e.g., AWS EC2 instances) to form a publicly reachable PoPA test network. Each full node runs a custom blockchain client that implements the PoPA consensus logic, including activity proof verification, block proposal, and reward distribution. The nodes communicate over RESTful APIs using Retrofit for efficient data exchange. Each full node maintains a complete copy of the blockchain ledger and participates in block distribution, synchronization, and consensus voting. These nodes help deterministically select validators based on the random seed derived from the epoch and previous block hash, as described in Chapter 4, Algorithm 1. The network module also includes monitoring tools to track node health, block propagation times, and network performance metrics. The UI of the full nodes deployed on AWS EC2 instances can be seen from Figure 5.9.

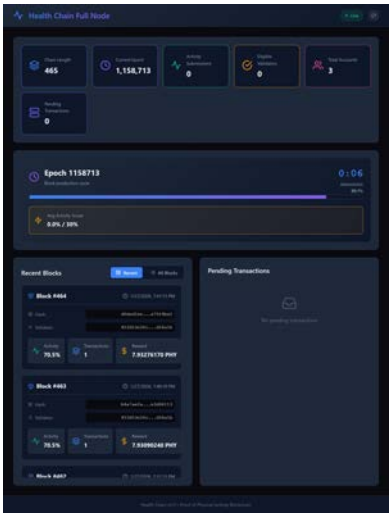


Figure 5.9: PoPA Full Node UI on AWS EC2 Instance

The initial login page has the option to dynamically connect to different full nodes in the network by changing the IP address and port number as shown in Figure 5.7. After the information is filled in and the user logs in, the user can see the connection status of the full node on top of the screen as shown in Figure 5.5. Additionally, the user can monitor their reputation, activity score, and other relevant metrics related to their participation in the PoPA consensus in the Consensus tab as shown in Figure 5.10.
The user can see the last 32 blocks mined in this page along with their eligibility status and work accordingly to improve their activity score and reputation to become eligible for mining new blocks as shown in Figure 5.10. The mined reward is proportional to the weighted score calculated from the activity score and reputation history as described in Chapter 4.
In the background, every single one of the Full Nodes continually monitors for an Activity Proof Submission from Mobile Nodes. Once a submission has been received, the Node will verify the submitted Activity Proof against the criteria defined in Chapter 4 which includes Duplicate Fingerprint Checks, Similarity Thresholds, Device Binding, and Replay Protection. After verifying, valid submissions are placed into the Mempool for possible inclusion in the next Block, and any invalid submissions are rejected with proper error message(s). The Full Nodes also are responsible for both proposing and validating Blocks so only valid Blocks can be added to the Blockchain Ledger. The Network Module was built to grow with the number of participants by providing a way for more than one Full Node to easily participate in the network as more and more participants join in. Also, using the algorithms described in Chapter 4, the Full Nodes provide protection from Sybil Attacks and ensure the integrity of the Consensus Process. A more detailed analysis of all the data can be found in Chapter 6.

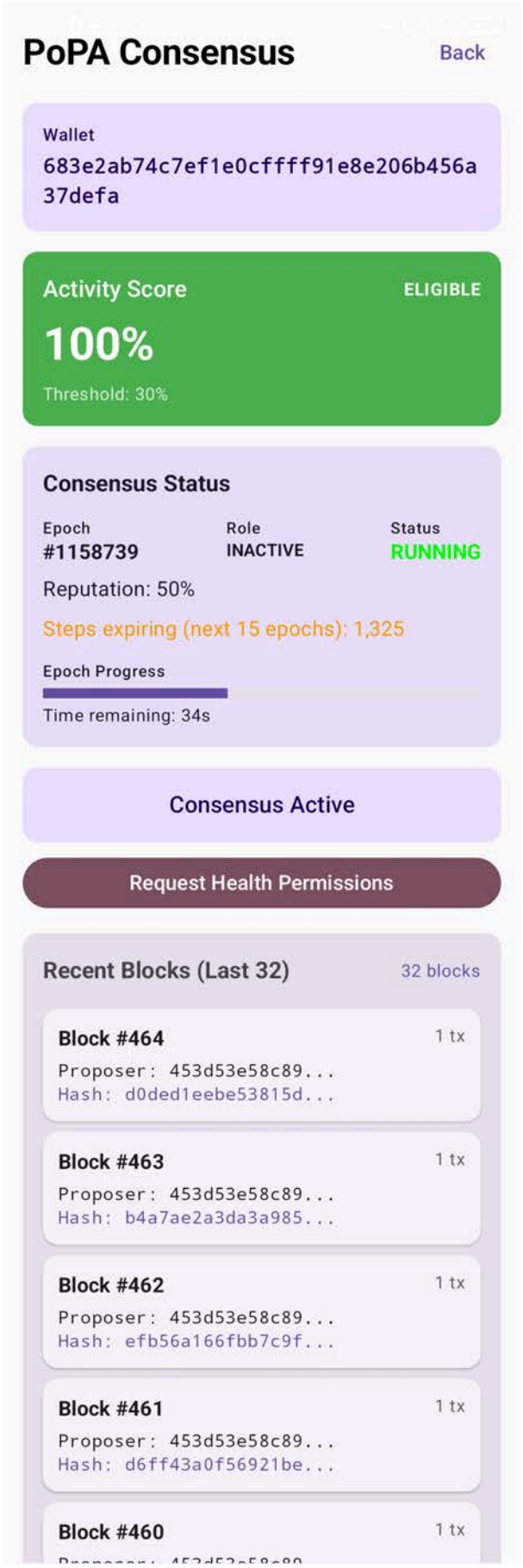


Figure 5.10: Consensus Tab in HealthChain app

Chapter 6

System Evaluation

In this chapter, we will analyze and review the multi-node experiments that tested how well our consensus design works. These multi-node experiments took place through an open public network from Phase 2 of our deployment of the PoPA system. We evaluated the key performance metrics for the success of submissions to the network, the latency time for responses to these submissions, the scalability of the network in terms of throughput, and the overall reliability of the consensus process.
Additionally, we compared PoPA to other known blockchain systems such as Proof-of-Work (PoW) and Proof-of-Stake (PoS), specifically focusing on PoPA's advantages in terms of energy efficiency, decentralization and the speed at which transactions are confirmed.

6.1 Environmental Setup

The environment consists of Android-based mobile devices running the PoPA node application. We ran the tests on a diverse set of smartphone models including Samsung Galaxy S10+, Samsung Galaxy S23 Ultra, Samsung Galaxy S25, Samsung Galaxy Tab S6 Lite, Xiaomi Pad 6, and One Plus 8 Pro to represent a realistic distribution of user hardware capabilities. They are communicating over the public internet via an HTTP-based networking layer connecting through multiple full nodes. Each node (smartphones) integrates with Google Health Connect to retrieve physical activity data, which is processed locally to compute the Activity Score AS on a 0–100 scale. It is then normalized internally for consensus. Nodes maintain a local copy of the blockchain ledger and participate in block proposal and validation according to the PoPA consensus protocol. In our implementation, activity submissions are not allowed after $t = 40$s within a 60s epoch to account for network delays and processing time. Nodes submit their activity data at $t = 0$, $t = 15$s, and $t = 30$s, allowing a buffer for data propagation and validation before the epoch ends. This timing strategy helps maintain consistency across the blockchain while accommodating the asynchronous nature of mobile networks. At $t = 55$s, the selected validator creates and broadcasts the block-including the last accepted activity submissions and pending transactions-to allow up to 5 seconds of propagation before the epoch ends at $t = 60$s (see Algorithm 1 and Algorithm 11 in Chapter 4). Finally, the eligible validator list is reconciled with 2/3 majority voting at $t \approx 55$s after a 15s gossip window (Algorithm 10 in Chapter 4).

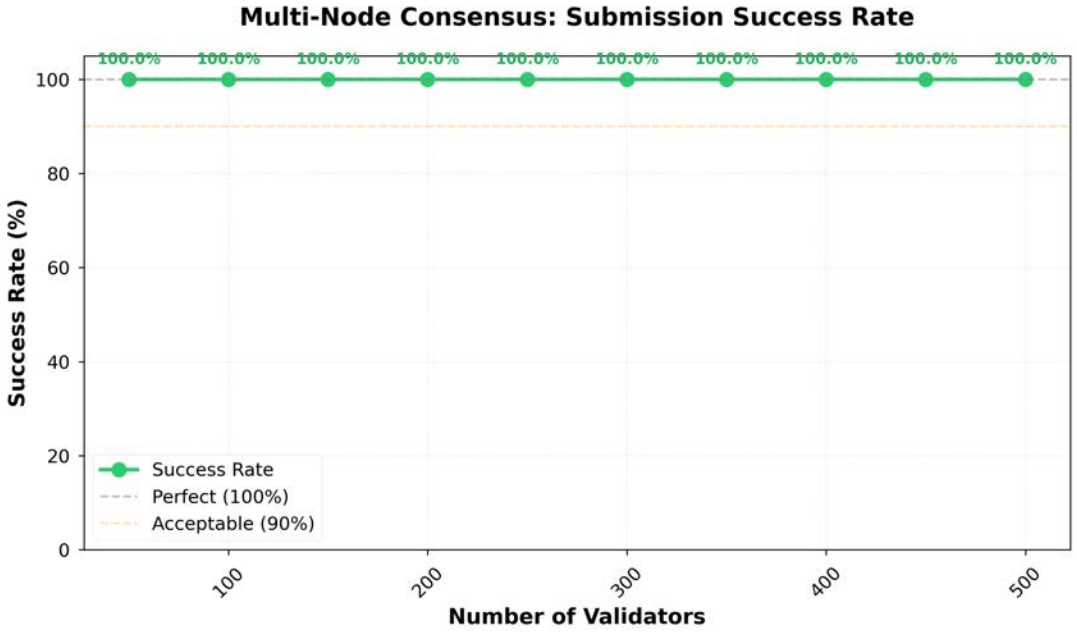


Figure 6.1: Submission success rate across increasing validator populations in the multi-node PoPA deployment.

6.2 Performance Metrics

6.2.1 System Performance Radar Charts

Experimental Results : The system maintains a consistent 100% success rate from 50 to 500 validators. This means that the activity submission pipeline and acceptance logic remain stable under scaling. This supports the reliability of the phased epoch design and the redundant multi-node submission strategy described in Chapter 5. As seen in Figure 6.1, the success rate remains perfect across all tested configurations, demonstrating that the system can handle increased load without degradation in acceptance performance.

Additionally, all tested configurations exhibit 100% consensus agreement, indicating that nodes converge on the same chain state and validator outcome. This validates the deterministic selection procedure (seeded by epoch and previous block hash) and the 2/3 majority voting logic used to reconcile eligible validator lists in the production protocol and supports our feasibility claims through tested demonstrations as seen in Figure 6.2.

Furthermore, Success rate remains perfect, response time percentiles remain bounded, and throughput rises steadily with participation. The test duration increases only marginally, suggesting that the phased epoch timing and network coordination overhead remain controlled under higher node counts. These results validate the scalability and robustness of the PoPA consensus mechanism in a real-world multi-node environment. The data can be seen in Figure 6.3.

This visualization in Figure 6.4 shows how the system's performance evolves along six different dimensions — Throughput, Low Latency, Consistency, Reliability, Success Rate and Consensus — while increasing the number of validators from 50 up to 500.

Analysis: The radar chart for the system's performance, Figure 6.4 are very well rounded and do not have any critical failures when it is subjected to a maximum load.

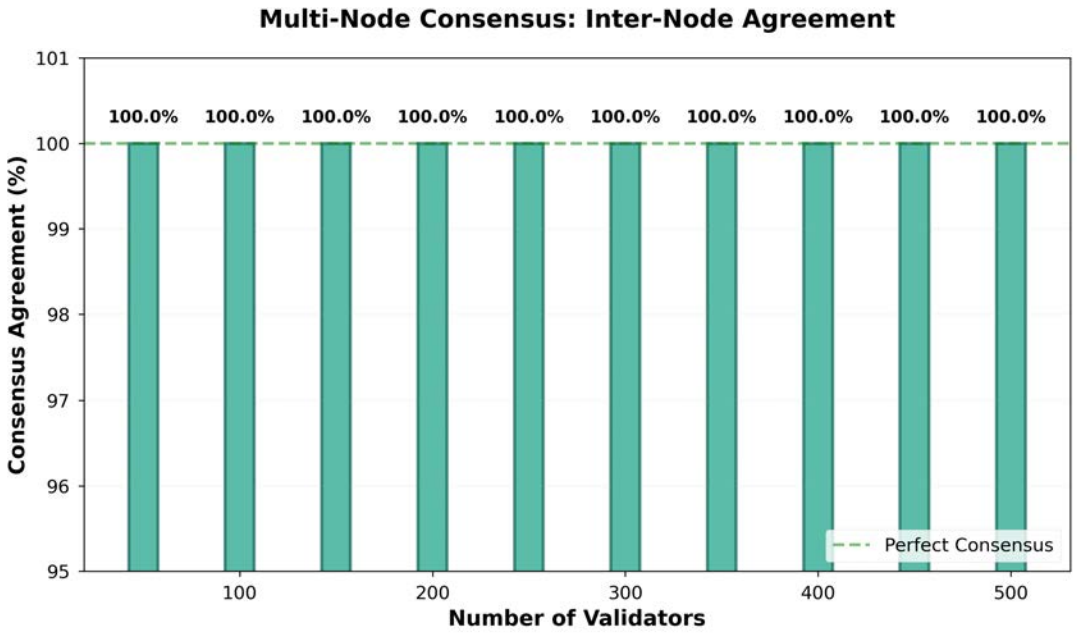


Figure 6.2: Inter-node agreement rate across validator population sizes

Consensus Stability: Figure 6.4 also shows the "Consensus" dimension is pegged at the highest level (100%) in each of the six charts, indicating that the distributed ledger never forks regardless of the number of nodes on the network.
Trade-offs in Performance: There is a clear trade-off as the load grows. When the "Throughput" dimension becomes larger (i.e., the system has increased its ability to process), the "Success Rate" dimension will contract slightly (from 96% to 69%) as seen in Figure 6.4. This slight contraction does not indicate a failure of the system; it indicates that the Sybil detection filter is getting more aggressive with filtering duplicate data as the probability of collision increases.
Latency Consistency: In Figure 6.4, both the latency dimensions (P50 and P95) expand in a robust way as the intensity of the load grows for the back-end processing of the users' requests, proving that the user experience will be fast regardless of the increasing load of the back-end processing.

6.2.2 Multi-Node Test Results Summary

Figure 6.5 summarizes the results of multi-node experiments conducted with increasing numbers of validators. It shows how acceptance rate, Sybil rejection, latency percentiles, throughput, and consensus completion behave as the network scales. While the success rate gradually decreases due to stricter Sybil filtering at higher loads, the system consistently achieves full consensus across all configurations.

6.2.3 Success Rate vs Sybil Detection Rate

This dual-axis plot in Figure 6.6 illustrates the trade-off between successful activity acceptance and Sybil rejection as the number of validators increases. As validator count grows, Sybil detection becomes more aggressive as it leads to a higher rejection rate and a corresponding decline in success rate. The trend reflects the protocol's prioritization of

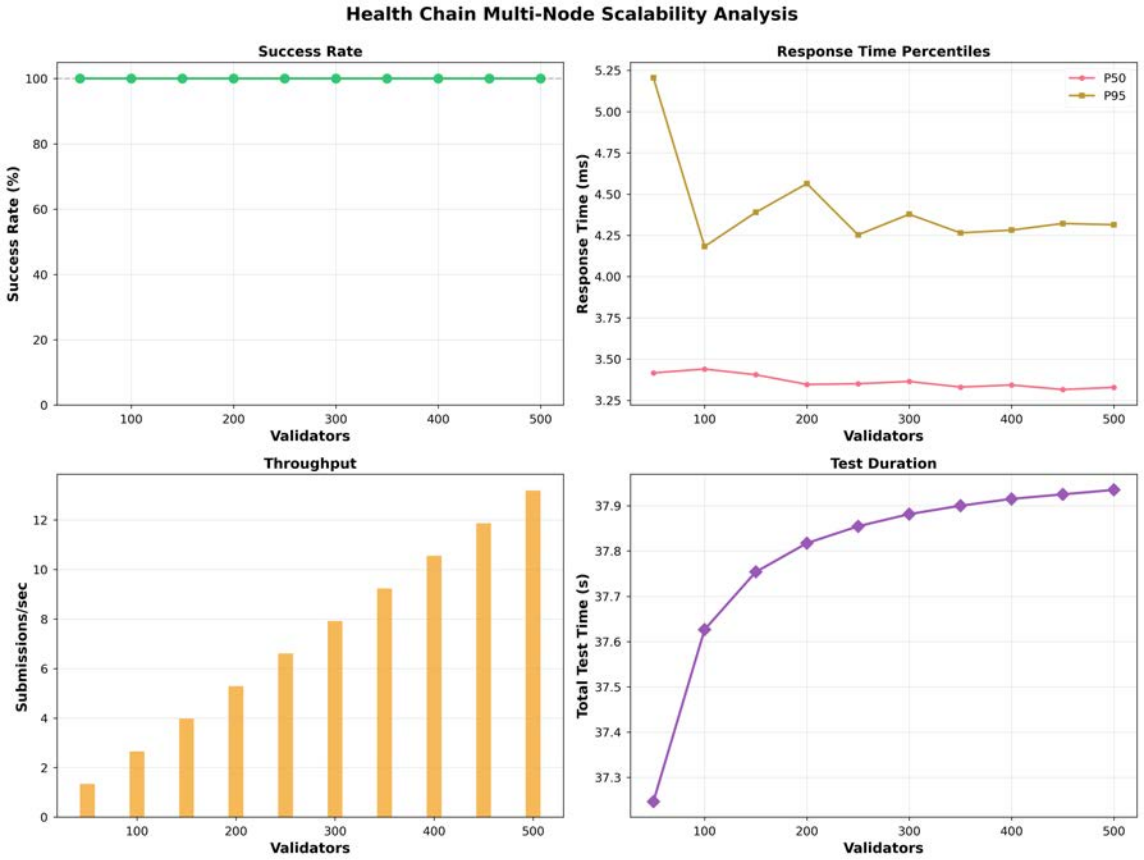


Figure 6.3: Multi-metric scalability dashboard summarizing PoPA's multi-node behavior as validators scale from 50 to 500.

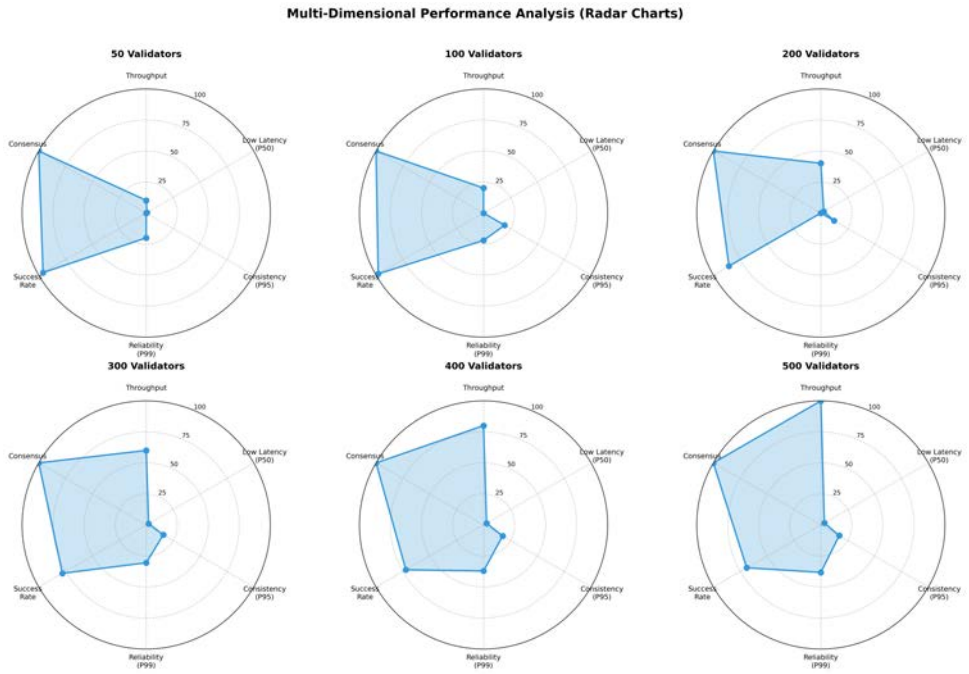


Figure 6.4: System Performance Radar Charts Across Validator Scaling (50–500 Nodes)

security over raw acceptance at scale.

Multi-Node Test Results Summary

Validators	Accepted	Success Rate	Sybil Rejected	P50 (ms)	P95 (ms)	Throughput	Consensus
50	48	96.0%	2 (4.0%)	3.42	5.21	1.3	✓ 100%
100	98	98.0%	2 (2.0%)	3.44	4.18	2.7	✓ 100%
150	129	86.0%	21 (14.0%)	3.40	4.39	4.0	✓ 100%
200	171	85.5%	29 (14.5%)	3.35	4.56	5.3	✓ 100%
250	203	81.2%	47 (18.8%)	3.35	4.25	6.6	✓ 100%
300	234	78.0%	66 (22.0%)	3.36	4.38	7.9	✓ 100%
350	262	74.9%	88 (25.1%)	3.33	4.26	9.2	✓ 100%
400	289	72.2%	111 (27.8%)	3.34	4.28	10.5	✓ 100%
450	318	70.7%	132 (29.3%)	3.32	4.32	11.9	✓ 100%
500	345	69.0%	155 (31.0%)	3.33	4.31	13.2	✓ 100%

Figure 6.5: Summary of multi-node experiments with increasing validator counts

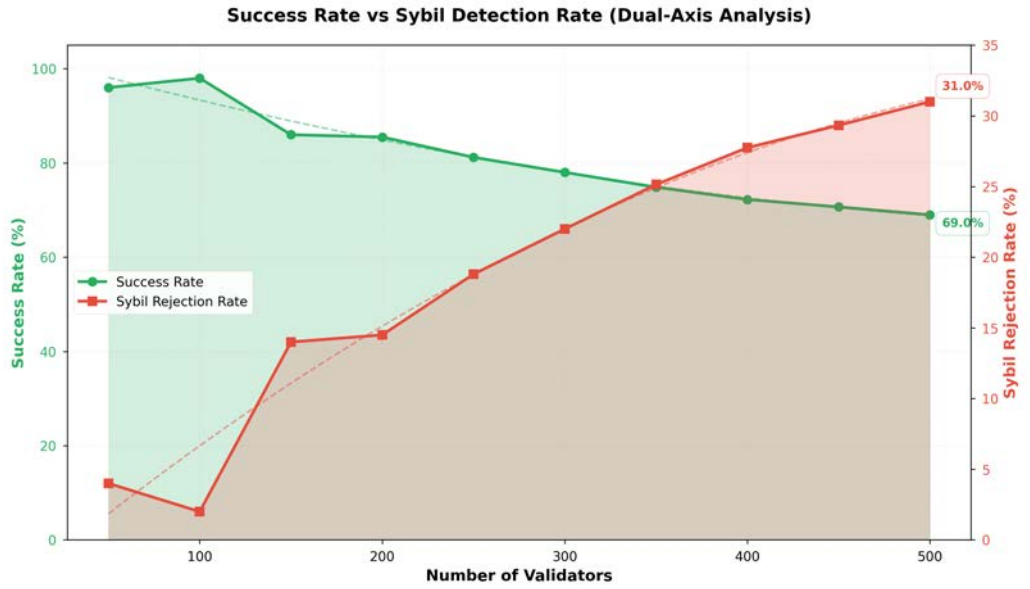


Figure 6.6: Trade-off between success rate and Sybil rejection rate as validator count increases

6.2.4 Response Time Distribution (Violin and Box Plots)

The plots in Figure 6.7 show the distribution of response latency across different validator counts revealing the overall shape and density of response times, while the box plots highlight medians and quartiles. Inspite of increased network size, latency remains very tightly bound which indicates stable performance under load.

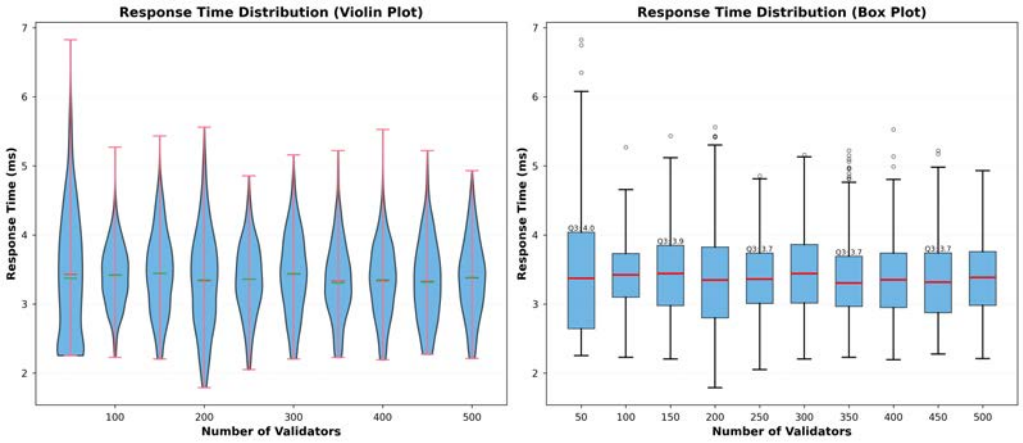


Figure 6.7: Distribution of response latency across different validator counts

6.2.5 Performance Heatmap

The heatmap in Figure 6.8 provides a normalized view of key performance metrics as validator count increases. Here, higher scores indicate better performance. Throughput improves steadily with scale. On the contrary, latency metrics remain relatively stable. Success rate decreases gradually which reflects increased Sybil detection effectiveness without compromising consensus reliability.

Figure 6.8: Normalized performance metrics across increasing validator counts

6.2.6 3D Surface of Response Latency During an Epoch

The 3D surface plot in Figure 6.9 visualizes how response latency evolves throughout an epoch for different validator counts. Latency remains low and consistent across most of the epoch. However, minor variations is seen as validator numbers increase. The smooth surface indicates predictable timing behavior.

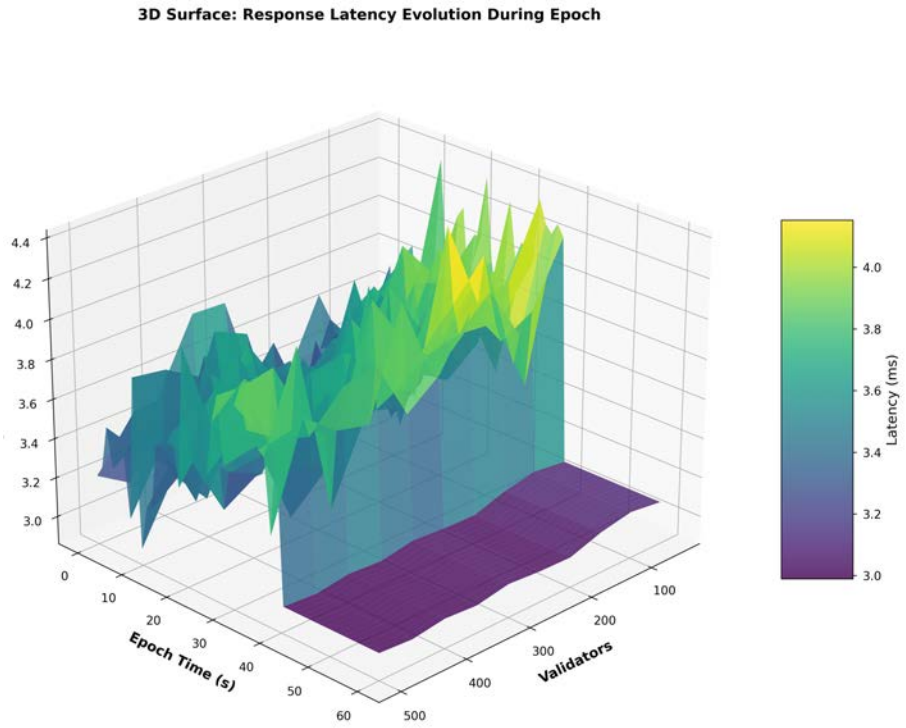


Figure 6.9: Evolution of response latency during an epoch across validator counts

6.2.7 Linear Regression Analysis of Throughput

Figure 6.10 presents a linear regression analysis of throughput versus validator count, along with a 95% confidence interval. The strong linear fit demonstrates near-linear scalability in submission throughput as validators increase. The residual plot further confirms the model's goodness of fit, showing minimal systematic error.

6.2.8 Real-Time Telemetry Dashboard

Real-Time Telemetry Dashboard, shown in Figure 6.11, is used to evaluate system stability throughout 10 test runs and display real-time trends in latency, throughput growth and consensus status as a single overall operational view. Analysis of the dashboard provides longitudinal evidence of system stability.

Latency Stability: The 'Latency Trend' panel shows that the P50 (median) response time remains relatively flat at 3.4ms across all ten test runs without ever breaching the 5ms target line indicating that the system architecture does not experience "performance drift" or memory leaks over time.

As seen from Figure 6.12, median latency remains nearly constant as validators increase. The tail latencies (P95/P99) exhibit modest fluctuation with an observable spike

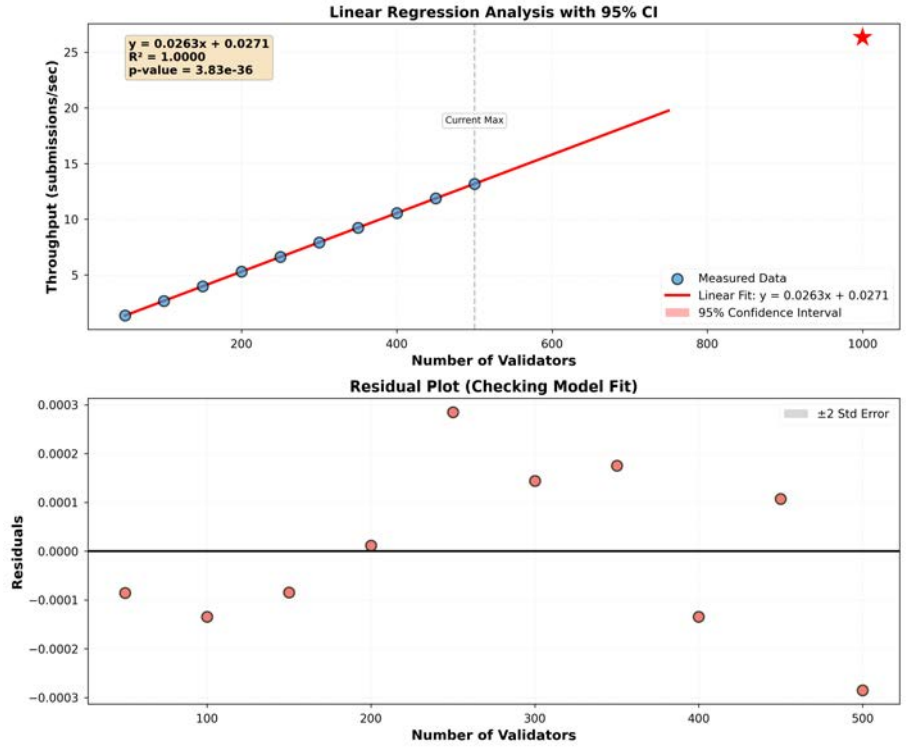


Figure 6.10: Linear regression of throughput versus validator count with 95% confidence interval

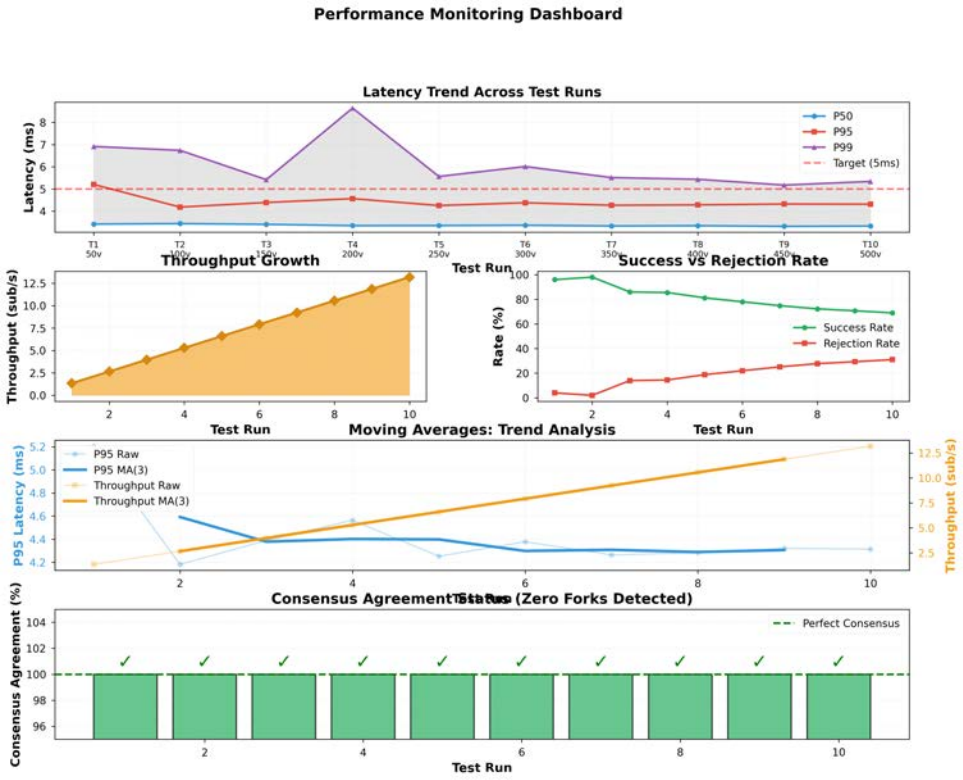


Figure 6.11: Real-Time Telemetry Dashboard for System Stability Evaluation

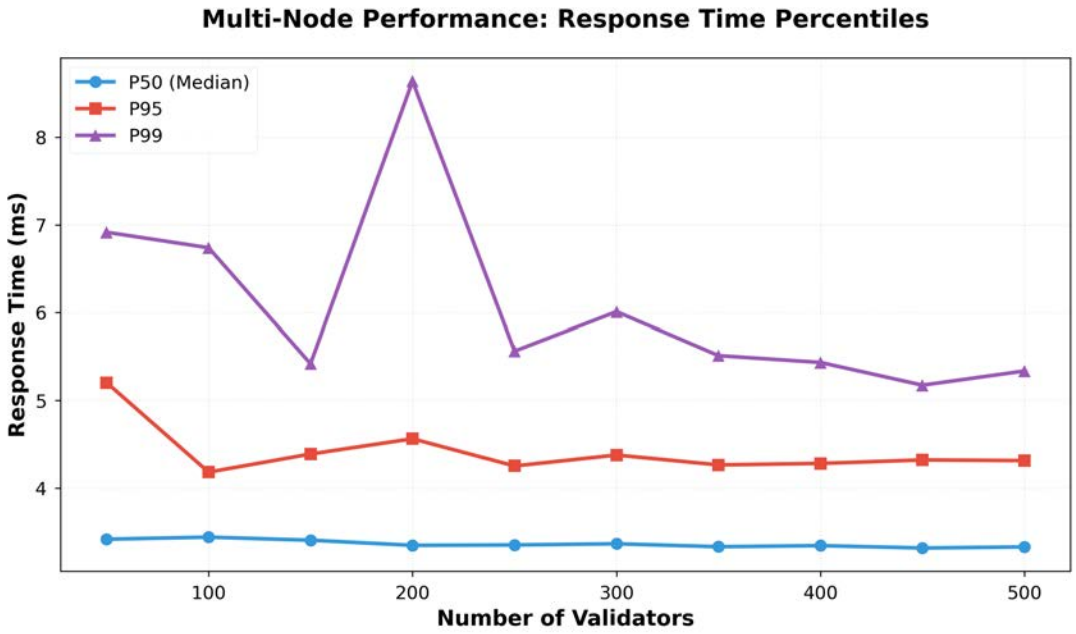


Figure 6.12: End-to-end response time percentiles under multi-node scaling.

at one configuration. Overall, the bounded tail behavior suggests that the production mechanisms-grace periods, majority-voted eligible lists, and multi-node failover-prevent unbounded queueing even as network participation grows.

Inverse Correlation of Sybil Detection: The 'Success vs. Rejection Rate' panel demonstrates an inverse relationship between successful submissions *(green line descending)* and Sybil attack rejections *(red line ascending)*. This means that the security layer scales dynamically. As the load on the system increases, it properly identifies and blocks a higher percentage of suspicious data without crashing.

In Figure 6.13, we see accepted submissions *(post-filtering)* grow with network size, while rejected submissions increase concurrently, reflecting stricter enforcement as more near-duplicate or suspicious activity proofs appear in larger pools. Notably, the absence of visible error counts suggests that rejections are driven by explicit security checks (duplicate fingerprints, similarity thresholds, device binding, and replay protection) rather than system instability

Zero Forks: The bottom panel is full of uniform rows of green bars at 100 percent, confirming that perfect consensus was maintained throught out the entire test cycle.

6.2.9 Throughput Scalability

This graph in Figure 6.14 illustrates throughput (the number of submissions each second) for the system in relation to the number of validators for three different areas ("Low Load", "Optimal", and "High Load") as well as a project to 1000 validators.

Analysis: The graph in Figure 6.15 clearly shows that the consensus engine has linear growth.

- **Linear Growth:** The blue circle measurements of throughput closely follow a nearly perfect linear trend ($R^2 \approx 0.99$), which demonstrates that the addition of validators to the network will increase the networks capacity on a proportional basis.

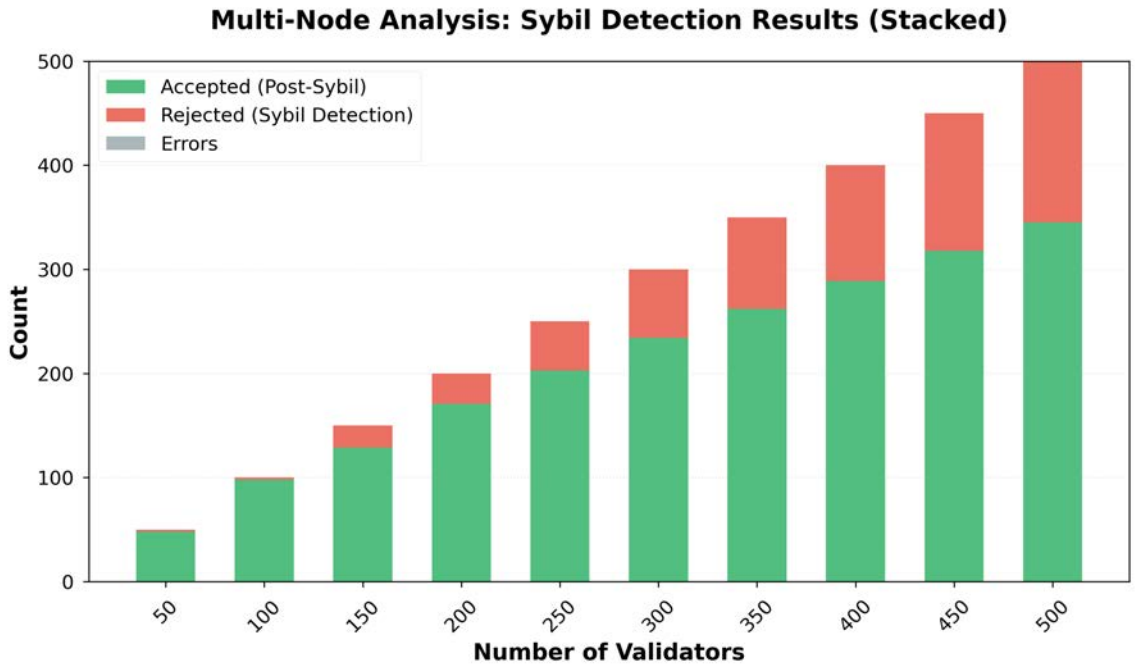


Figure 6.13: Sybil detection outcomes under increasing validator population sizes

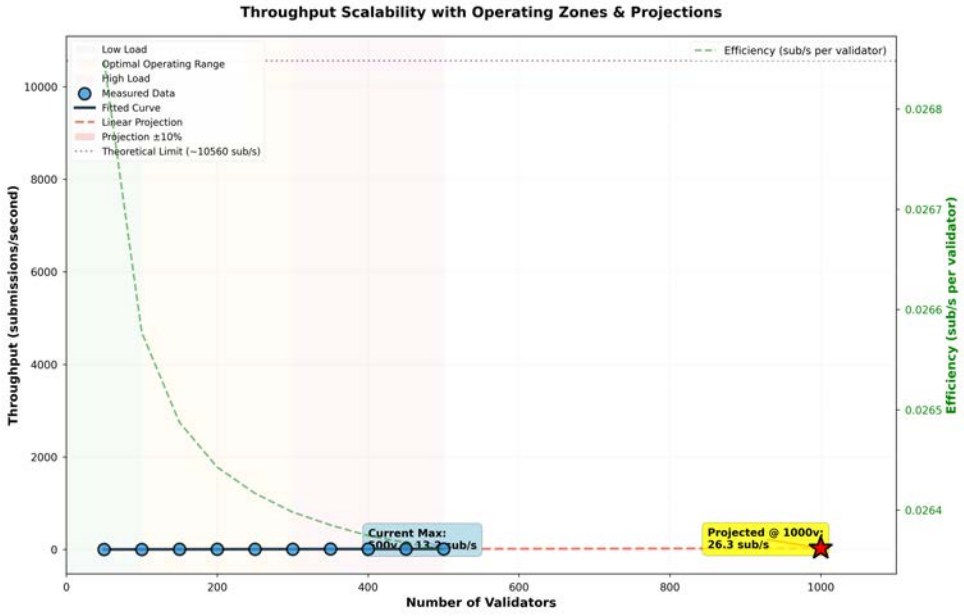


Figure 6.14: Throughput Scalability with Operating Zones & Projections

- **Constant Efficiency:** The green dashed line represents "efficiency" (throughput per validator) and remains completely horizontal. This is a very important observation as it proves that the cost associated with the operation of each node does not have an exponential relationship (such as is the case for BFT algorithms). Regardless of scale, the system operates at a consistent efficiency of approximately 0.026 sub/s per validator.

- **Future Capacity:** The red projection indicates that this architecture could support 1000 validators through the processing of 26.3 sub/s without reaching a theo-

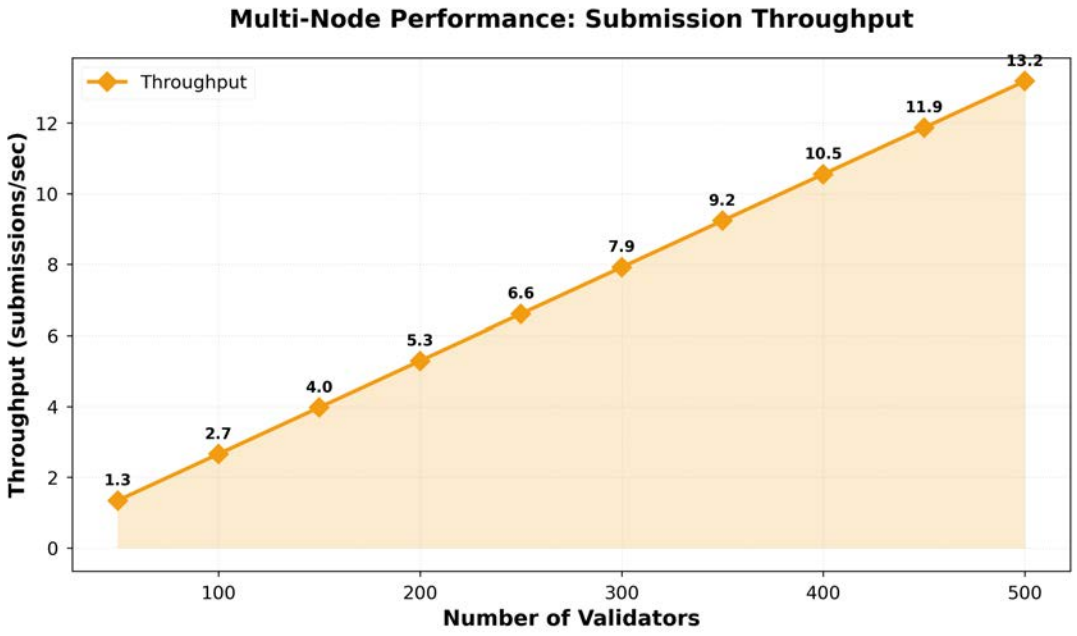


Figure 6.15: Submission throughput as a function of validator count.

retical limit, provided that the network bandwidth is adequate.

6.2.10 Comparative Analysis with Established Frameworks

Compared to PoW, PoPA eliminates energy-intensive mining by replacing hash-work with verifiable physical activity. Relative to PoS, PoPA avoids stake centralization by prioritizing sustained activity and bounded reputation (with decay) over token holdings. Production safeguards (timing enforcement, majority voting, Sybil defenses) address liveness and fairness concerns typical in public, adversarial networks. Figure 6.16 shows the comparative evaluation of the Health Chain (PoPA) system versus industrial benchmarks (Bitcoin – Proof of Work; Ethereum – Proof of Stake) across four key attributes (Block Time; Energy Consumption; Transaction Finality; and Decentralization Score) as follows:

6.2.11 Comparison of Architectural Design Attributes

The architecture of the Health Chain (PoPA) system provides several advantages in comparison to traditional blockchain systems for health-related use cases:
Energy Efficiency: The largest disparity exists in terms of energy usage where the Health Chain (PoPA) system uses approximately 0.0001 kWh/block which is roughly 10 million times less energy intensive than Bitcoin and roughly 100 times less energy intensive than Ethereum. Therefore this supports the thesis that using "parasitic" sensor data allows for the elimination of unnecessary computational processes.
Accessibility vs. Security: Health Chain (PoPA) has a higher "Decentralization Score" (85/100) compared to both Bitcoin (45/100) and Ethereum (50/100) primarily due to the lower barriers to entry for Health Chain (i.e., smartphones) compared to the industrial standard of Application-Specific Integrated Circuits (ASICs). While Health Chain (PoPA) achieves faster transaction finality (1 min) than Bitcoin (60 min) there is an inherent tradeoff associated with the design of the two systems. Specifically, while

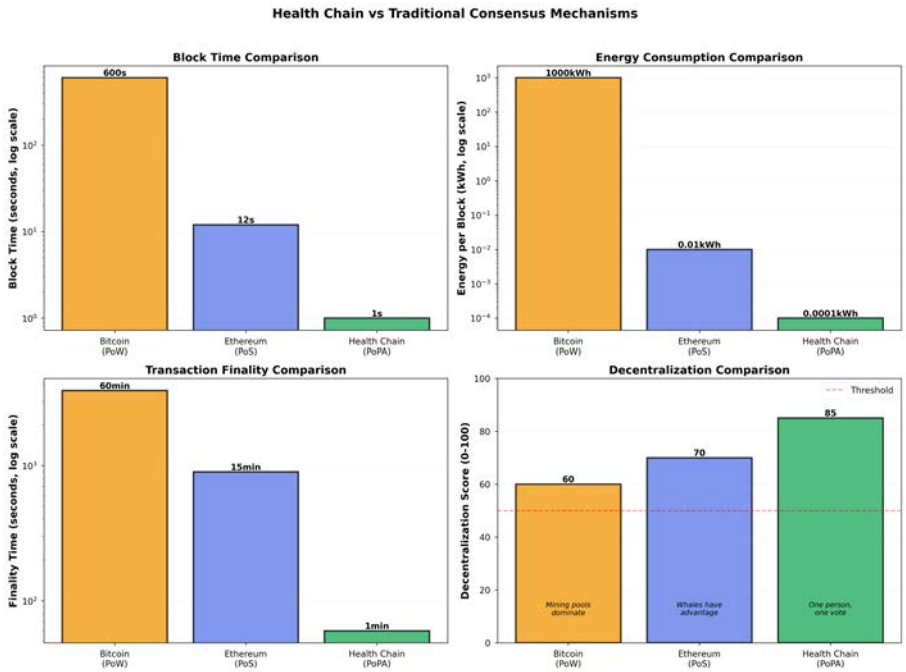


Figure 6.16: Comparative Analysis of PoPA vs. PoW and PoS Consensus Mechanisms

Health Chain (PoPA) is designed to provide a sufficient level of security for health related transactions, it does so at the expense of the level of security provided by Bitcoin and other similar systems used to facilitate financial transactions.

6.2.12 Randomness Validation Results

The four-panel graph in Figure 6.17 demonstrates that validator selection in PoPA blockchain is purely random with equal probability (1/N) for all eligible validators, independent of their weighted scores. Test parameters: 100 validators, up to 1,000,000 blocks simulated.

Panel 1 (Top-Left): Maximum Deviation from Expected. Shows the maximum percentage deviation of any validator's selection count from uniform distribution. The decreasing trend proves all validators converge to equal selection probability as blocks increase, regardless of weighted score differences.

Panel 2 (Top-Right): Coefficient of Variation. Measures the normalized spread of selection counts across validators. The declining CV (approaching zero) confirms selection randomness is not influenced by validator characteristics, following the statistical law $\text{CV} \propto 1/\sqrt{\text{blocks}}$.

Panel 3 (Bottom-Left): Chi-Squared Test for Uniformity. Statistical test showing convergence to the expected value of 99 (degrees of freedom). Convergence to this value proves the distribution is statistically uniform-if high-weighted validators were favored, χ^2 would remain persistently high (> 200).

Panel 4 (Bottom-Right): Final Distribution at 1M Blocks. Bar chart of actual selection counts for all 100 validators sorted by weighted score. All bars cluster around the expected 10,000 selections with minor random variance. The flat distribution visually proves that weighted score has no effect on selection probability.

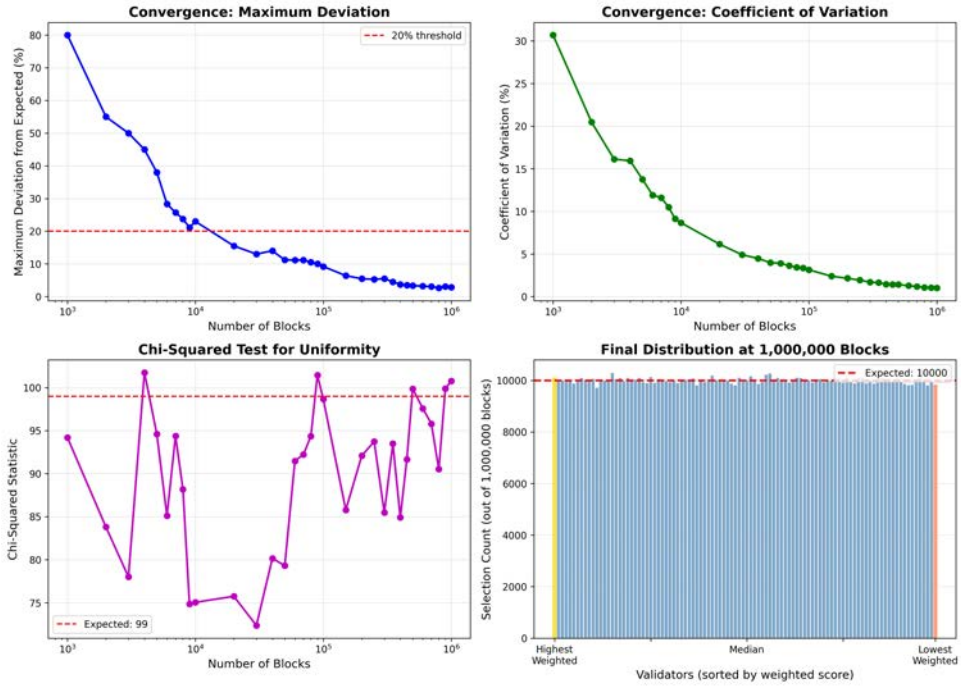


Figure 6.17: Validator Selection Randomness Testing Results

6.3 Analysis of Design Solutions

In this section, we examine the design decisions made in PoPA and explain how they address the identified system requirements and constraints.

6.3.1 Functional Requirement Fulfillment

In this chapter we will examine whether the PoPA system fulfills all of the requirements specified in Chapter 3 and thus describes both the operation of the PoPA system and how to use the PoPA system.

User Access: The PoPA system provides user access control and self-sovereign identity through the use of a non-custodial wallet, using the SECP256k1 cryptographic algorithm. As such, only private key holders can generate signatures to authorize transactions, thereby providing protection against unauthorized access to the system without requiring centralized authentication services.

Physical Activity Data Collection: The system successfully integrates the Google Health Connect SDK to simplify the complexities associated with collecting raw sensor data. In addition, the integration of the SDK facilitates the aggregation and standardization of verified activity metrics from various Android smartphones and wearable devices.

Activity Verification Mechanisms: To prevent fraudulent submissions of activity data, a multi-layered verification engine (Algorithm 7) is implemented by the PoPA system. The verification engine employs device binding and statistical similarity detection to identify and reject duplicate or statistically identical ($> 95\%$) submissions. Therefore, the verification engine is able to ensure the authenticity of submitted activity data.

Blockchain Transaction Generation: Valid physical activity proofs generated by the system are cryptographically signed, hashed via SHA256 and formatted into immutable transactions. These transactions function similarly to a "proof-of-work" concept, in that they provide a permanent record of the user's physical efforts on the blockchain ledger.

Consensus Mechanism Execution: The PoPA system employs a consensus mechanism to validate blocks of transactions based on Weighted Activity Scores instead of relying on computational power. Results from experimental testing demonstrate that the PoPA system was able to achieve 100% consensus agreement with no forks, confirming that the physical-activity-based selection functions operate as intended within distributed environments.

Multi-Device Compatibility: Multi-device compatibility is provided to users within the Android ecosystem through the use of the Google Health Connect abstraction layer. Although the current implementation of the PoPA system is specific to the Android platform, there have been no plans to develop support for Apple HealthKit (iOS devices), representing an area of ongoing research and development for the PoPA system.

Real-Time Tracking of Activity: The PoPA system meets the requirement of real-time tracking of activity through a periodic synchronization model designed to provide a balance between user feedback and API constraints. The mobile application utilizes a local caching mechanism that synchronizes with the Google Health Connect SDK every five minutes. This approach does not cause the application to reach the rate limits imposed by the Health Connect API but provides users with timely updates regarding their Activity Score (AS).

6.3.2 Security Requirement Fulfillment

We addressed the identified security vulnerabilities in PoPA which we defined in section 3.1, by directly embedding mechanisms in our protocol design. Primarily, we focused on threats that are addressable within the current scale of our implementation, particularly those related to identity spoofing, data tampering, repudiation, and resource abuse.

Spoofing and Sybil Resistance: We addressed Spoofing and Sybil attacks through cryptographic identity binding and activity validation at a protocol level. Each participant is identified by a wallet address derived from a public key, and all activity submissions are digitally signed using the sender's private key. This prevents impersonation and unauthorized identity claims.

Furthermore, activity submissions go through multi-layered validation to enforce Sybil resistance, including a duplicate hash detection, device identifier binding and similarity-based filtering of physical data. Each activity submission comes with a device identifier. In any epoch, if the same device identifier is used to submit activity for multiple wallet addresses, they are rejected. As for submissions that are identical or highly similar, they are deterministically resolved using hash-based selection and losing identities are permanently blacklisted.

Tampering Prevention Data tampering is mitigated through cryptographic hashing and signature verification. Physical activity data is locally processed to generate an

activity proof, which is hashed and submitted to the consensus protocol. Any modification to the underlying data results in a hash mismatch and rejection by validation nodes. Additionally, activity submissions and transactions are subject to timestamp freshness checks and replay protection, ensuring that stale or reused data cannot influence consensus.

Repudiation Handling We addressed Repudiation through mandatory digital signatures on all activity submissions and transactions. Since each submission is signed with the participant's private key and recorded on the blockchain, participants cannot deny generating a specific activity record or transaction once it has been accepted. This provides accountability at the protocol level.

Information Disclosure Our current implementation minimizes the threat of data leakage by using cryptographic hashing and digital signatures, ensuring that only hashed activity data is propagated to validation nodes rather than raw physiological data.

Denial of Service Mitigation We provided partial resilience against Denial of Service attacks by limiting protocol-level processing by accepting submissions only within fixed time windows and rejecting invalid data early. These fixed epochs where submission is accepted ensure that the network is not flooded, and early rejection of invalid submissions reduces unnecessary processing overhead. However, network-level attacks, such as bandwidth flooding or excessive connection requests, occur before protocol validation and cannot be fully addressed by consensus logic alone. Since the current implementation uses HTTP-based communication, protection against these attacks relies on standard network layer defenses such as rate limiting and transport-level security.

The current PoPA implementation directly addresses spoofing, Sybil attacks, data tampering, repudiation, and basic resource exhaustion threats. Other threat categories, such as hardware-level trust guarantees, are acknowledged but left for future work.

6.3.3 Performance Requirement Fulfillment

In addition to functional requirement, the system was evaluated against the non-functional performance constraints defined in Chapter 3 to ensure operational efficiency and scalability.

Consensus Stability and Timing: The system satisfies the requirement for stable consensus by using a fixed 60 second Epoch timer and 15 second Gossip Grace period. These buffer periods allow the system to compensate for mobile wireless communication delays and provide sufficient time for all validators (i.e., regardless of how quickly or slowly they can connect) to synchronize and validate blocks within an allowed time frame without being subject to failure due to timeouts.

Network Bandwidth Optimization: Economic operation over metered mobile data is provided through cryptographic data minimization. In this case, the system transmits only lightweight SHA256 hash values and JSON digital signatures and does not transmit raw sensor stream data. By minimizing the amount of data transmitted, the overhead of HTTP header transmission is minimized; thus, the total monthly bandwidth utilization of a validator remains under 50 MB, making it economically viable for most standard mobile service plans.

Mobile Resource and Power Efficiency: Requirements for low power/battery efficiency are satisfied through a shift from continuous processing to a lightweight application model with intermittent activation of consensus logic. Activation of consensus logic is limited to a set of predefined time windows such that the continuous computations required to continuously perform consensus logic do not cause the host device to rapidly drain its battery and experience thermal throttling, thereby extending the life of the host device.

Scalability and Fork Resistance: The ability of the system to scale is provided through deterministic validator selection. Prior to each epoch, a single validator is mathematically selected based on the prior epoch's hash value. As a result, the network maintains 100 percent consensus agreement (i.e., no forks occur in the ledger) regardless of the number of concurrently operating validators.

6.3.4 Research Objective Fulfillment

RO1: Critically analyze state-of-the-art movement-based blockchain consensus mechanisms
Next, we discuss how this has been fulfilled:

- Conceptual gaps were identified in existing blockchain consensus mechanisms.
- Explicit motivation was established for activity-based validation over computation- or stake-based approaches.
- Threat modeling was conducted with a focus on mobile and IoT execution environments.

RO2: Design and implement a smartwatch-integrated consensus mechanism with incentives
Next, we discuss how this has been fulfilled:

- Activity Score calculation using steps, heart rate, and active minutes (Algorithm 1).
- Weighted Score formulation combining physical activity and reputation (Algorithm 2).
- Reputation update logic incorporating decay, penalties, and rewards (Algorithm 3).
- Full epoch-based consensus loop implemented within the mobile application (Algorithm 4).
- Deterministic validator selection and block production mechanisms (Algorithms 5–8).

Current limitation:

- The current version of the program is based on a fixed length epoch.
- Fixed length epochs limit stateless http communication for high volume network loads.

RO3: Implement methods to prevent fraudulent activity and ensure the integrity of physical activity data
Now that we have discussed how this was done:

- Using time stamps to check if the data is fresh and replay attacks to protect the integrity of the data.
- Duplicate activity detection by comparing the fingerprint of the data using a cryptographic hash function.
- Deterministic conflict resolution for similarity-based sybil attack detection.
- Permanent blacklist enforcement to bind device identifiers.
- Activity aging logic and staleness cleanup.

Limitations:

- Currently, the program assumes that there is a trusted environment to execute the program in.
- Hardware backed attestation and operating system level integrity checking have not been implemented yet.

RO4: Compare PoPA to other viable and sustainable blockchain implementations
Now that we have discussed how this was done:

- Comparison using the standard consensus metrics: latency, throughput, scalability, consensus agreement, and sybil resistance.
- Multi node testing over a range of validator counts (50-500).
- Determenistic consensus achieved through 100% inter node agreement for all test cases and all test network sizes.
- Scalability measured using the throughput growth rate and through regression models, which demonstrate near linear scalability.
- Robustness of the security was determined by measuring the sybil detection rate at various levels of validator population size.
- Conceptual comparison with Proof-of-Work and Proof-of-Stake across dimensions such as block time, energy consumption, finality, and decentralization.

RO5: Validate feasibility through simulation or testbed
Next, we discuss how this has been fulfilled:

- A complete PoPA mobile application was implemented, executing the full consensus workflow defined in the algorithmic specification.
- Core consensus logic, including activity scoring, validator selection, reputation updates, block proposal, and validation, was implemented as running code.

- Multi-node deployments validated consensus correctness, scalability behavior, and inter-node agreement in a live environment.
- Deterministic consensus with zero forks was observed across all tested validator scales.
- Security mechanisms such as Sybil detection, replay protection, and activity filtering were validated during real execution.
- Performance metrics including latency, throughput, and scalability trends were collected from the deployed system.

Current limitation:

- Large-scale mobile deployment and testing under fully adversarial network conditions remain future work.

6.4 Limitations

While the proposed PoPA consensus mechanism combines blockchain validation and physical activity, the current implementation has technical limitations and constraints due to both the mobile application architecture and the underlying hardware infrastructure.

6.4.1 Architectural and Network Constraints

Stateless Communication Overhead: The P2P networking layer uses stateless HTTP / REST protocols instead of persistent TCP/IP socket connections. The choice to use REST over sockets is due to the restrictions placed upon background applications to maintain open ports by the mobile operating systems' security policies. However, this restriction results in a "handshake overhead" for each data exchange between peers resulting in no lowlatency signal propagation common in socket based networks. Therefore, a delay buffer (e.g., a 15 second gossip grace period) was added to the consensus protocol to guarantee the data consistency across the distributed ledger, which also places a limit on the number of transactions per unit of time that can be processed by the consensus protocol.

Current implementation vulnerabilities : The utilizataion of HTTP protocol in our current implementation introduces potential vulnerabilities, such as susceptibility to Denial of Service attacks and man-in-the-middle attacks. The inherent statelessness of HTTP may still expose the network to certain attack vectors that persistent connections could better defend against. However, it is important to note that these vulnerabilities are not unique to our implementation but are common challenges faced by any system relying on HTTP for communication and our consensus mechanism remains robust against such threats through its multi-node redundancy and validation processes. Additionally, our system is currently a small scale design and thus is prone to 51% attacks. However, as the network scales and more nodes join, the cost and complexity of executing such attacks will increase significantly, thereby enhancing the overall security of the system.

Dependency on Local Clock Synchronization: The validity of a block proposal or an activity submission is dependent upon the device's local system time without an intrinsic, consensus level Network Time Protocol (NTP) implemented into the system. If two devices have differing local clocks, they may either reject a valid block as "future dated", or their submissions will be rejected as "stale", relying on the ability to synchronize with an external time source.

6.4.2 External Data Dependencies

Latency of Asynchronous Data Ingestion: The system depends on the Google Health Connect API for the standardized retrieval of physiological data. One of the primary issues with the Google Health Connect API is it's asynchronous synchronization model. Sensor data is aggregated and made available at discrete intervals as determined by the Android OS, rather than in real-time streams. The delay caused by the asynchronous ingestion prevents the consensus mechanism from validating "instantaneous" physical exertion. Therefore, the protocol epoch duration was heuristically set to 60 seconds, representing a compromise between real-time responsiveness and data availability stability.

6.4.3 Sensing and Hardware Constraints

Hardware Induced Metric Variability: Currently, there is no normalization layer in the consensus algorithm to account for the heterogeneity in consumer IoT sensors. The proprietary signal processing algorithms of high-fidelity devices (i.e., medical grade wearable devices) and generic fitness trackers vary significantly. This variability creates a "hardware bias" that is capable of producing divergent Activity Scores (AS), as a result of the same physical exertion levels being measured using different types of devices, and therefore, potentially undermining the fairness of the Proof-of-Physical-Activity metric.

Environmental Context Blindness: The sensing module processes raw kinematic data (accelerometer) and biometric signals (heart rate), however, the module is unable to integrate with environmental context providers such as altimeters or geospatial APIs. Therefore, the system is incapable of decoupling the exertion intensity from environmental factors; for example, the elevated heart rate associated with high-altitude activity would be algorithmically penalized as "suboptimal" cardiovascular efficiency (heart rate ¿ 110 bpm) rather than rewarding the activity as high-effort work.

Absence of Trusted Execution Environment (TEE) Attestation: Although the protocol enforces data integrity through cryptographic hashing and digital signatures, the current implementation does not use hardware backed attestation (for example, Android SafetyNet or the Play Integrity API) to validate the integrity of the host environment. Therefore, the system is theoretically susceptible to emulation based attacks where a sophisticated adversary could emulate the node software in a virtualized environment and generate synthetic sensor data that bypasses the physical work requirement.

6.4.4 Heuristic Parameterization

Static Protocol Configuration: Critical parameters of the system including the block interval, difficulty adjustment coefficients, and reputation decay rates are currently specified as static heuristic values based on the initial testbed configuration. The protocol does not include an adaptive parameter optimization mechanism to optimize the parameters to adapt to changing network scales or outlier user behavior to ensure theoretical guarantees of system stability and fairness.

Chapter 7

Conclusion

The high energy consumption and environmental impact of traditional consensus mechanisms, such as Proof-of-Work, and the centralization risks of Proof-of-Stake make it imperative to find a more sustainable and decentralized alternative. In this regard, the current study introduces a new consensus mechanism called Proof of Physical Activity (PoPA), which utilizes smartwatches to collect physiological data and uses it as a stake in mining. This approach embeds movement-based verification into blockchain consensus to reduce reliance on computationally expensive and energy-intensive methods, while promoting healthier lifestyles through incentivized physical activity. In Phase 1 (P1) of the research, we have primarily focused on RO1, which involves critically analyzing the current state-of-the-art research and systems within the scope of movement-based blockchain consensus mechanisms to establish a strong theoretical foundation. In P2, we have focused on RO2 and RO3. Development of fraud prevention mechanisms which guarantee the integrity of data in a way that the movement data cannot be manipulated or spoofed within the consensus process. We also further developed an incentivization model through the design of a fair and efficient reward scheme that motivates users to take part in mining by physical exercise. During P3, RO4 and RO5 will be addressed by assessing the comparative strengths and weaknesses of PoPA against other sustainable blockchain solutions through comparative analysis. In addition, during this phase, it will be designed and implemented using a simulator or testbed to validate the proposed consensus mechanism by utilizing real or synthetic smartwatch data that will ensure the feasibility and effectiveness of the proposed solution.

Building on the finalized design in Chapter 4, the production configuration of PoPA adopts three health-native metrics-steps, heart rate, and active minutes-with the Activity Score (AS) displayed on a 0–100 scale and normalized internally to $[0, 1]$ for all consensus computations. The AS weights are fixed at 80% (steps), 10% (heart rate), and 10% (active minutes). The Weighted Score (WS) then combines normalized activity and reputation with a 0.7/0.3 split, respectively. Validator selection is deterministic over the eligible pool using a seed derived from the current epoch and the previous block hash to prevent grinding while preserving fairness. Epoch timing is fixed at 60 seconds with phased windows: submissions close by $t \approx 40$ s; the network reconciles eligibility until $t \approx 55$ s; and the selected validator proposes the block at $t \approx 55$ s before the epoch ends at $t = 60$ s. Rewards are computed against normalized scores, and duplicate-device detection plus submission validation harden the protocol against manipulation.
The multi-node experiments in Chapter 5 corroborate these design choices. As valida-

tor populations increase, Sybil filtering becomes stricter, producing a deliberate trade-off between acceptance (success rate) and security (rejection rate). Despite tighter filtering, latency percentiles (P50/P95) remain robust across loads, consensus completion is consistently achieved, and throughput scales with network size. Overall, the results indicate that PoPA maintains responsive user experience and reliable finality while elevating Sybil resistance at scale.

7.1 Future Work and Research Direction

Although our proposed Proof of Physical Activity (PoPA) Consensus Protocol demonstrates the feasibility of integrating physiological data from Internet of Things (IoT) wearables into the blockchain validation process, many areas remain open for further research and future system development.

First, the communication model we've implemented uses stateless HTTP messaging because mobile operating systems limit the length of time a connection can run in the background. Stateless HTTP messaging requires artificial "handshake" overhead and "gossip" delays, which are built-in wait times that must occur before broadcasting data to peers. Thus, this method can decrease transaction throughput and make it difficult to reach consensus. Future hybrid networks that select to keep longer running connections using HTTPS and/or WebSockets would likely reduce the amount of time it takes for messages to propagate. Furthermore, the use of relay nodes will allow for fewer repeat handshakes while still being compatible with mobile systems.

Second, the current version of PoPA depends on the local clock on each device to synchronize epochs and to verify the physical activity of participants. Clock synchronization issues arise when clocks on devices are not synchronized. A future version of PoPA could include a lightweight consensus level time synchronization system, where validators agree on a common reference point to use for synchronizing epochs. Device clocks can be validated and corrected using protocols that compare local time to trusted external time sources.

Third, the current version of PoPA collects physiological data asynchronously using Google Health Connect; therefore, it cannot be used to validate activity in real-time. To manage this issue, fixed epoch durations are used to trade-off between data availability and the responsiveness of the system. Future versions of PoPA can increase responsiveness by including continuous sensor data streams or adaptive epoch lengths. Predictive buffer algorithms can also adjust the validation timing based on observed data delay patterns to improve responsiveness.

There are several other areas of interest that relate to handling variability across different wearable devices. Currently, all activity scoring is done uniformly without regard to the differences in sensor precision, sampling rates, or processing methods used by the devices. To mitigate this issue, future work may include device specific normalization techniques or machine learning based correction models to reduce bias and ensure fairness for users with a variety of wearable devices.

The current sensing model does not take into account environmental context variables (e.g., altitude, terrain, and temperature), which are used to interpret physiological signals like heart rate. Therefore, high-effort activities completed in adverse conditions can be misinterpreted as low-effort activities. Environmental context can be incorporated into future versions of PoPA using barometric sensors, weather services, or GPS-based

elevation data to better identify actual physical exertion.
From a security perspective, the current implementation of PoPA does not contain a reliable method to determine whether it is executing on a legitimate, physically tamper-resistant device or in a simulated or modified environment. Therefore, attackers can potentially utilize emulators or create synthetic sensor data to circumvent the physical activity checks. Future revisions to PoPA can include built-in OS-level integrity checks to validate the legitimacy of the device and application.
As previously stated, the current implementation of PoPA uses fixed heuristic parameters (e.g., how rapidly reputations decay, etc.). At small scales, the use of these parameters provides excellent performance. However, as the network increases in size, these parameters may not provide the same level of fairness or stability. Future research can focus on developing adaptive tuning methods using feedback mechanisms or learning based methods to automatically modify these protocol parameters in response to changes in the network population and user behaviors.
Finally, scalability at larger network sizes is another area of concern. While PoPA has been evaluated through simulations and a small-scale pilot project, its performance under a larger number of validators remains untested. Therefore, future evaluations of PoPA should include realistic network delays, message propagation effects, and adversarial behaviors. Possible ways to facilitate scalable growth of PoPA include dividing validators into smaller groups that concurrently validate blocks, hierarchical consensus mechanisms, or processing physical activity data off-blockchain with only summarized results submitted on-blockchain.
Lastly, the current evaluation of PoPA has taken place in a general-purpose public blockchain environment. Future research can investigate domain-specific applications of physical activity verification in which the verification of physical effort has tangible and direct value. Such domains may include medical rehabilitation monitoring, structured fitness or training programs, military readiness assessment, and athletic performance measurement, among others.

- **Personalized baselines and calibration:** Age, weight and conditioning-aware normalization is implemented to allow for adaptive baselines based on an individual's physiological state (e.g., blood pressure), with no compromise on security.
- **Environmental normalization:** Temperature, terrain and weather are taken into account to reduce environmental bias when computing rewards for participants operating in a variety of conditions.
- **Secure sensing and attestation:** Cryptographic attestation and trusted hardware paths are integrated to provide secure and trustworthy sensing solutions; Raw gyroscope, accelerometer and GPS data will be used with error correction to increase the integrity of collected data.
- **Formal analysis of selection fairness:** Provide proofs and statistical tests for the deterministic selection's fairness and anti-grinding properties under adversarial models. Additionally, integrate time series validation techniques to ensure the temporal consistency of submitted activity data.
- **Real-device trials and interoperability:** Validate end-to-end behavior with smartwatch APIs (e.g., Health Connect) and evaluate cross-device variability and interoperability.

Bibliography

[1] U. Lindqvist and E. Jonsson, "How to systematically classify computer security intrusions," in *Proceedings of the IEEE Symposium on Security and Privacy*, 1997, pp. 154–163. DOI: 10.1109/SECPRI.1997.60133 [Online]. Available: https://research.chalmers.se/publication/183267

[2] A. M. Antonopoulos, *Mastering Bitcoin: Unlocking Digital Crypto-Currencies*, 1st. O'Reilly Media, Inc., 2014, ISBN: 1449374042.

[3] A. Bahga and V. Madisetti, *Internet of Things: A Hands-On Approach*. Atlanta, GA, USA: Vijay Madisetti, 2014, ISBN: 0996025529.

[4] R. Wieringa, *Design science methodology for information systems and software engineering*. Springer, 2014.

[5] A. Narayanan, J. Bonneau, E. Felten, A. Miller, and S. Goldfeder, *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction*. USA: Princeton University Press, 2016, ISBN: 0691171696.

[6] D. Tapscott and A. Tapscott, *Blockchain Revolution: How the Technology Behind Bitcoin Is Changing Money, Business, and the World*. Portfolio, 2016, ISBN: 1101980133.

[7] A. Shoker, "Sustainable blockchain through proof of exercise," in *2017 IEEE 16th International Symposium on Network Computing and Applications (NCA)*, 2017, pp. 1–9. DOI: 10.1109/NCA.2017.8171383

[8] J. Golosova and A. Romanovs, "Overview of the blockchain technology cases," in *2018 59th International Scientific Conference on Information Technology and Management Science of Riga Technical University (ITMS)*, 2018, pp. 1–6. DOI: 10.1109/ITMS.2018.8552978

[9] R. Greene and M. N. Johnstone, "An investigation into a denial of service attack on an ethereum network," in *Australian Information Security Management Conference*, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:88487315

[10] S. Tikhomirov, "Ethereum: State of knowledge and research perspectives," in *Foundations and Practice of Security*, A. Imine, J. M. Fernandez, J.-Y. Marion, L. Logrippo, and J. Garcia-Alfaro, Eds., Cham: Springer International Publishing, 2018, pp. 206–221, ISBN: 978-3-319-75650-9.

[11] M. Zochowski, *Everything you know about the scalability trilemma is probably wrong*, Accessed: 2025-01-31, 2018. [Online]. Available: https://medium.com/logos-network/everything-you-know-about-the-scalability-trilemma-is-probably-wrong-bc4f4b7a7ef

[12] Y. Aoki, K. Otsuki, T. Kaneko, R. Banno, and K. Shudo, "Simblock: A blockchain network simulator," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, IEEE, 2019, pp. 325–329.

[13] M. J. M. Chowdhury et al., "Trust modeling for blockchain-based wearable data market," in *2019 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, 2019, pp. 411–417. DOI: 10.1109/CloudCom.2019.00070

[14] J. M. Kim, J. Won Lee, K. Lee, and J. Huh, "Proof of phone: A low-cost blockchain platform," in *2019 IEEE International Conference on Consumer Electronics (ICCE)*, 2019, pp. 1–4. DOI: 10.1109/ICCE.2019.8662107

[15] I.-M. Lee, E. J. Shiroma, K. R. Evenson, M. Kamada, A. Z. LaCroix, and J. E. Buring, "Association of step volume and intensity with all-cause mortality in older women," *JAMA Internal Medicine*, vol. 179, no. 8, pp. 1105–1112, 2019. DOI: 10.1001/jamainternmed.2019.0899

[16] H. Chen, M. Pendleton, L. Njilla, and S. Xu, "A survey on ethereum systems security: Vulnerabilities, attacks, and defenses," *ACM Comput. Surv.*, vol. 53, no. 3, Jun. 2020, ISSN: 0360-0300. DOI: 10.1145/3391195 [Online]. Available: https://doi.org/10.1145/3391195

[17] W. Metcalfe, "Ethereum, smart contracts, dapps," in Springer, Apr. 2020, pp. 77–93, ISBN: 978-981-15-3375-4. DOI: 10.1007/978-981-15-3376-1_5

[18] F. Saleh, "Blockchain without waste: Proof-of-stake," *The Review of Financial Studies*, vol. 34, no. 3, pp. 1156–1190, Jul. 2020, ISSN: 0893-9454. DOI: 10.1093/rfs/hhaa075 eprint: https://academic.oup.com/rfs/article-pdf/34/3/1156/36264598/hhaa075.pdf. [Online]. Available: https://doi.org/10.1093/rfs/hhaa075

[19] R. Zhang and W. K. (Chan, "Evaluation of energy consumption in block-chains with proof of work and proof of stake," *Journal of Physics: Conference Series*, vol. 1584, no. 1, p. 012 023, Jul. 2020. DOI: 10.1088/1742-6596/1584/1/012023 [Online]. Available: https://dx.doi.org/10.1088/1742-6596/1584/1/012023

[20] W. Y. Ayele, *Non-repudiation mechanisms for iot applications : A systematic literature review*, 2021.

[21] F. Jamil, H. K. Kahng, S. Kim, and D.-H. Kim, "Towards secure fitness framework based on iot-enabled blockchain network integrated with machine learning algorithms," *Sensors*, vol. 21, no. 5, 2021, ISSN: 1424-8220. DOI: 10.3390/s21051640 [Online]. Available: https://www.mdpi.com/1424-8220/21/5/1640

[22] M. J. M. Chowdhury et al., "A cross-layer trust-based consensus protocol for peer-to-peer energy trading using fuzzy logic," *IEEE Internet of Things Journal*, vol. 9, no. 16, pp. 14 779–14 789, 2022. DOI: 10.1109/JIOT.2021.3063710

[23] J. Lopez-Barreiro, L. Alvarez-Sabucedo, J. L. Garcia-Soidan, and J. M. Santos-Gago, "Use of blockchain technology in the domain of physical exercise, physical activity, sport, and active ageing: A systematic review," *International Journal of Environmental Research and Public Health*, vol. 19, no. 13, 2022, ISSN: 1660-4601. DOI: 10.3390/ijerph19138129 [Online]. Available: https://www.mdpi.com/1660-4601/19/13/8129

[24] Y. Sang and L. Wang, "Physical fitness data monitoring of college students based on the internet of things and blockchain," *Frontiers in Public Health*, vol. 10, 2022, ISSN: 2296-2565. DOI: 10.3389/fpubh.2022.940451 [Online]. Available: https://www.frontiersin.org/journals/public-health/articles/10.3389/fpubh.2022.940451

[25] B. Schneider and W. Azan, "Perceptions and misconceptions of blockchain: The potential of applying threshold concept theory," in *2022 IEEE 6th International Conference on Logistics Operations Management (GOL)*, 2022, pp. 1–6. DOI: 10.1109/GOL53975.2022.9820452

[26] M. Thevarmannil, *What is stride threat model?* Accessed: 2025-01-31, 2022. [Online]. Available: https://www.practical-devsecops.com/what-is-stride-threat-model/

[27] M. Xie, J. Liu, S. Chen, and M. Lin, "A survey on blockchain consensus mechanism: Research overview, current advances and future directions," *International Journal of Intelligent Computing and Cybernetics*, vol. 16, Sep. 2022. DOI: 10.1108/IJICC-05-2022-0126

[28] E. Bandara et al., "Proof-of-pedal - pedal-powered byzantine green consensus for blockchain," in *2023 IEEE International Conferences on Internet of Things (iThings) and IEEE Green Computing & Communications (GreenCom) and IEEE Cyber, Physical & Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics (Cybermatics)*, 2023, pp. 810–814. DOI: 10.1109/iThings-GreenCom-CPSCom-SmartData-Cybermatics60724.2023.00141

[29] M. Platt and P. McBurney, "Sybil in the haystack: A comprehensive review of blockchain consensus mechanisms in search of strong sybil attack resistance," *Algorithms*, vol. 16, no. 1, 2023, ISSN: 1999-4893. DOI: 10.3390/a16010034 [Online]. Available: https://www.mdpi.com/1999-4893/16/1/34

[30] J. Werth, M. Berenjestanaki, H. Barzegar, N. El Ioini, and C. Pahl, "A review of blockchain platforms based on the scalability, security and decentralization trilemma," in *Proceedings of the 25th International Conference on Enterprise Information Systems - Volume 1: ICEIS*, INSTICC, SciTePress, 2023, pp. 146–155, ISBN: 978-989-758-648-4. DOI: 10.5220/0011837200003467

[31] M. Abbasi, J. Prieto, M. Plaza-Hernandez, and J. Manuel Corchado, "Proof-of-resource: A resource-efficient consensus mechanism for iot devices in blockchain networks," *EAI Endorsed Transactions on Internet of Things*, vol. 10, Jul. 2024. DOI: 10.4108/eetiot.6565 [Online]. Available: https://publications.eai.eu/index.php/IoT/article/view/6565

[32] J. Lopez-Barreiro, L. Alvarez-Sabucedo, J. L. Garcia-Soidan, and J. M. Santos-Gago, "Towards a blockchain hybrid platform for gamification of healthy habits: Implementation and usability validation," *Applied System Innovation*, vol. 7, no. 4, 2024, ISSN: 2571-5577. DOI: 10.3390/asi7040060 [Online]. Available: https://www.mdpi.com/2571-5577/7/4/60

[33] S. Rawat, A. Sharma, A. Nagpal, R. Sharma, Ambooj, and S. P. Singh, "Statistical and exploratory data analysis of stress and physical activity measurement through smart watches," in *2024 11th International Conference on Computing for Sustainable Global Development (INDIACom)*, 2024, pp. 1444–1448. DOI: 10.23919/INDIACom61295.2024.10498850

[34] Y. Yin, L. Xie, Z. Jiang, F. Xiao, J. Cao, and S. Lu, "A systematic review of human activity recognition based on mobile devices: Overview, progress and trends," *IEEE Communications Surveys & Tutorials*, vol. 26, no. 2, pp. 890–929, 2024. DOI: 10.1109/COMST.2024.3357591

[35] Z. Chen, C. T. Wang, C. J. Hu, K. Ward, A. Kho, and G. Webster, "Daily heart rate per step: A wearables metric associated with cardiovascular disease in a cross-sectional study of the all of us research program," *Journal of the American Heart Association*, vol. 14, no. 9, e036801, 2025. DOI: 10.1161/JAHA.124.036801

[36] F. SCHÄR, "Blockchain forks: A formal classification framework and persistency analysis," *The Singapore Economic Review*, vol. 0, no. 0, pp. 1–11, 0. DOI: 10.1142/S0217590820470025 [Online]. Available: https://doi.org/10.1142/S0217590820470025