



KLE Technological
University
Creating Value
Leveraging Knowledge

BVB Campus, Vidyanagar, Hubballi – 580031, Karnataka, INDIA.

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

Mini Project Report

On

Cloud Forensics Using Blockchain on Kubernetes

submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Engineering

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted By

Neha Chandavari	01fe21bcs075
Nirmala Kanti	01fe21bcs180
Prajwal Mutnal	01fe21bcs159
Rutuja Chikkorde	01fe21bcs175

Under the guidance of

Prof. Parikshit Hegde

School of Computer Science and Engineering

KLE Technological University, Hubballi



KLE Technological
University
Creating Value
Leveraging Knowledge

BVB Campus, Vidyanagar, Hubballi – 580031, Karnataka, INDIA.

2022-2023

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

2022-23

CERTIFICATE

This is to certify that project entitled “ **Cloud Forensics Using Blockchain on Kubernetes** ” is a bonafied work carried out by the student team (Neha Chandavari Usn:01fe21bcs075, Nirmala Kanti Usn:01fe21bcs180, Prajwal Mutnal Usn:01fe21bcs159, Rutuja D Chikkorde Usn:01fe21bcs175), in partial fulfillment of the completion of 5th semester B. E. course during the year 2023 – 2024. The project report has been approved as it satisfies the academic requirement with respect to the project work prescribed for the above said course.

Guide Name

Prof. Parikshit Hegde

SoCSE

Head

Dr. Vijaylakshmi M.

External Viva-Voce

Name of the examiners

Signature with date

1 _____

2 _____

ABSTRACT

Blockchain technology has revolutionized data integrity and security in decentralized systems, with applications spanning from financial transactions to supply chain management. This research project focuses on enhancing the consensus mechanism and logging infrastructure in a blockchain ecosystem, specifically addressing challenges related to performance and security.

The project begins by retrieving web and pod logs from a Kubernetes cluster, utilizing IPFS (InterPlanetary File System) for secure and distributed storage. The hash of these logs is then recorded on the Ethereum blockchain using a Proof-of-Stake (POS) algorithm. Recognizing the potential for performance improvements, the research introduces Reputational Proof-of-Stake (RPOS) as an evolved consensus mechanism.

The transition to RPOS introduces a reputation system, adding an extra layer to the consensus mechanism. This reputation system evaluates the reliability and trustworthiness of network participants, leveraging their historical behavior to influence their role in the consensus process. The results of the project demonstrate that RPOS enhances the overall system performance by effectively addressing common challenges associated with traditional POS algorithms.

The comparative analysis between POS (Proof-of-Stake) and RPOS (Reputation Proof-of-Stake) includes evaluations of consensus mechanisms, aiming to assess the performance of the algorithms. The research concludes that the adoption of Reputational Proof-of-Stake, coupled with IPFS for decentralized logging, not only improves the efficiency of the blockchain network but also adds a level of robustness and security crucial for real-world applications.

This project contributes valuable insights into the practical implementation of decentralized logging and blockchain consensus mechanisms, offering a nuanced understanding of their impact on system performance and security. The findings provide a foundation for future advancements in blockchain technology and its applications across various industries.

Keywords : *Blockchain, Kubernetes, Reputational Proof-of-Stake, InterPlanetary File System(IPFS), Performance Improvement, Security.*

Acknowledgement

Throughout the completion of our project, we would like to express our sincere gratitude to Prof. Parikshit Hegde and Dr. Narayan D G for their unwavering support, encouragement, and guidance. Their expertise and mentorship have been invaluable, shaping our understanding and approach at every stage of the project. Prof. Parikshit Hegde's and Dr. Narayan D G's commitment to fostering academic excellence and providing insightful feedback have played a pivotal role in the successful execution of our research endeavor. We are truly grateful for their dedication, time, and mentorship, which have been instrumental in our academic and professional development.

Neha Chandavari - 01fe21bcs075

Nirmala Kanti - 01fe21bcs180

Prajwal Mutnal - 01fe21bcs159

Rutuja D Chikkorde - 01fe21bcs175

CONTENTS

ABSTRACT	i
Acknowledgement	i
CONTENTS	ii
LIST OF FIGURES	iii
1 INTRODUCTION	1
1.1 Motivation	1
1.2 Literature Review	2
1.3 Problem Statement and Objectives	3
1.3.1 Objectives	3
2 REQUIREMENT ANALYSIS	4
2.1 Functional Requirements	4
2.2 Non Functional Requirements	5
2.3 Hardware Requirements	5
2.4 Software Requirements	5
3 SYSTEM DESIGN	7
3.1 System Model / Architecture Design	7
3.2 Data Set Description	9
4 IMPLEMENTATION	10
4.1 RPOS Algorithm	13
5 RESULTS DISCUSSIONS	14
5.0.1 Fairness Metric	17
6 CONCLUSION AND FUTURE SCOPE	19
REFERENCES	21
Appendix A	22
A.1 Gantt Chart	22
A.2 Description of Tools and Technology used	22

LIST OF FIGURES

3.1	System Model	8
3.2	Sequence Diagram	8
3.3	Datasets Used	9
4.1	flow chart for RPOS algorithm	13
5.1	Hash from IPFS	14
5.2	Hash from IPFS	15
5.3	Hash value added to blockchain	15
5.4	Gini co-efficient calculation	16
5.5	Prediction values	17
5.6	POS vs RPOS	18
5.7	Number of blocks mined vs transaction time	18
A.1	Gantt Chart	22

Chapter 1

INTRODUCTION

In the dynamic landscape of technological innovation, blockchain stands out as a transformative force, reshaping the fundamentals of decentralized systems and data management. This introductory chapter serves as the launchpad for a thorough investigation into the nuanced dynamics of blockchain consensus mechanisms and decentralized logging. As a revolutionary technology, blockchain disrupts traditional paradigms by introducing decentralized and distributed ledger systems, fostering transparency, security, and trust. The ensuing exploration delves into the intricate workings of blockchain consensus mechanisms, which are pivotal in establishing agreement among network participants. Simultaneously, the chapter focuses on unraveling the complexities of decentralized logging, an essential component for maintaining an immutable and transparent record of transactions. Through this comprehensive examination, the research endeavors to contribute to the evolving discourse on blockchain technology, aiming to unravel its potential and address challenges. By doing so, it aspires to further propel the transformative impact of blockchain, solidifying its role as a cornerstone in the ever-evolving landscape of technological innovation.

1.1 Motivation

This research is motivated by the transformative impact that blockchain technology has demonstrated across diverse domains, spanning from finance to supply chain management. Acknowledging the pivotal roles played by consensus mechanisms and logging infrastructure in decentralized systems, this project aims to tackle existing challenges and actively contribute to the ongoing discourse surrounding the optimization of performance and security. Blockchain's decentralized nature has disrupted traditional paradigms by offering transparency and trust without intermediaries. However, challenges in consensus mechanisms and logging infrastructure have emerged, prompting the need for focused exploration. This research endeavors to propose innovative solutions for these challenges, aiming to enhance the efficiency and security of decentralized systems. By delving into the intricacies of consensus mechanisms, the project seeks to contribute to the evolution of protocols, fostering more robust and scalable decentralized networks. Simultaneously, attention will be devoted to refining logging infrastructure, addressing scalability issues, and fortifying against potential malicious attacks. The overarching goal is to propel decentralized systems to a state of heightened relevance and

reliability, aligning them more closely with the dynamic demands of real-world applications. Through these contributions, the project aspires to play a significant role in advancing the maturation of blockchain technology and ensuring its sustained efficacy across a spectrum of industries and use cases.

1.2 Literature Review

The literature survey encompasses a diverse range of studies focusing on various aspects of digital image forensics, data provenance assurance, and cloud forensics investigation. In a study [1], a procedure for tracing the chain of custody in digital image forensics is proposed, employing grey and fuzzy hashing techniques for storing evidence logs to the blockchain. The performance of the prototype is evaluated using Hyperledger Caliper, with future work aimed at testing the efficiency of the framework in handling a large number of digital evidence pieces. Another study [2] explores data provenance assurance for cloud storage using blockchain, implementing a mechanism to record and secure user operations on files stored in AWS S3 using Ethereum blockchain and IPFS. Meanwhile, [3] presents a blockchain-based log storage model with an efficient query system, utilizing IPFS and Enhanced Merkle tree methods for faster query responses based on log timestamp. Future work involves integrating machine learning and natural language processing techniques to enhance the accuracy of analysis. Lastly, [4] introduces an integrated framework for cloud forensics investigation using logging tools, with a focus on web log data analysis conducted using the Apache Spark engine. The framework is intended to be extended to include diverse cloud services and log types, such as network, database, and application logs, in future research endeavors. One study [5] presents a Blockchain-Enabled Scalable Network Log Management System that utilizes IPFS and blockchain to ensure data privacy and efficient log management. It suggests future work in extending the platform with machine learning tools for cyberattack detection. Another study [6] introduces BCALS, a Blockchain-based secure log management system for cloud computing, focusing on authentication logs stored in Elasticsearch and employing machine learning for identifying malicious users. Additionally, a study [7] explores leveraging blockchain for immutable logging and querying across multiple sites, optimizing data management and retrieval within the blockchain-based system.

Furthermore, a study [8] introduces SecLaaS, a Secure Logging-as-a-Service for Cloud Forensics, ensuring confidentiality for cloud users and investigators. It involves network monitoring using OpenStack and Snort, RSA for signature generation, and SHA-256 for hashing. Another paper discusses Secure Log Storage Using Blockchain and Cloud Infrastructure, proposing a cloud-based log management system with blockchain-based checkpoint logs for log integrity and analysis. Additionally, a study [9] presents a Blockchain-Powered Log Storage

System based on Hyper Ledger, aiming for a log system with higher throughput and lower latency.

Moreover, a study [10] focuses on analyzing and alerting on application logs within Kubernetes infrastructure, utilizing elastic stack and PLK with Prometheus. Future work includes analyzing large log datasets and improving accuracy. Lastly, a study [11] proposes a log aggregation method in a scale container platform, involving log data collection, aggregation processing, and storage using distributed computing. These studies contribute to advancing log management techniques and ensuring data security across various computing environments

1.3 Problem Statement and Objectives

Design a system to efficiently retrieve web logs from applications hosted in Kubernetes clusters, store them on IPFS, and subsequently store the IPFS hash to the Ethereum blockchain through the Proof of Stake algorithm and the consensus algorithm is optimised to ensure secure and tamper-proof log storage, thereby improving system integrity and compliance.

1.3.1 Objectives

- Retrieve web logs from the Kubernetes cluster and store them in IPFS by merging the data.
- To get hash from IPFS and store to Ethereum blockchain.
- To improve Proof Of Stake consensus algorithm.

Chapter 2

REQUIREMENT ANALYSIS

A Software Requirements Specification (SRS) is a comprehensive document that outlines the detailed requirements for a software system. It typically encompasses both functional and non-functional requirements. The functional requirements specified further describe the system's features and capabilities, specifying how the software should behave under different conditions. Non-functional requirements cover aspects such as performance, security, and usability. The SRS includes details on the system's architecture, interfaces, and data flow. Sequence diagrams illustrate the interaction between different components or modules, depicting the order of events in a particular scenario. Additionally, the document outlines the necessary software and hardware requirements, detailing the tools, programming languages, platforms, and infrastructure needed for the system's development and deployment. Overall, an SRS serves as a crucial foundation for software development, ensuring a clear understanding of the project's scope and guiding the development team throughout the software development life cycle.

2.1 Functional Requirements

Functional requirements articulate specific tasks and features a software system must deliver, defining its intended behavior. These requirements serve as a roadmap, guiding the design and development process by detailing user interactions, data processing, and system outputs. They ensure the software aligns precisely with user expectations and project objectives.

- The system shall collect web logs from a sample application hosted on the Kubernetes cluster.
- Logs are securely stored across the IPFS network, ensuring decentralized and distributed storage.
- The system shall provide a unique IPFS-generated hash for each set of logs.
- Each hash value and corresponding metadata are stored in the blockchain, ensuring transparency and immutability.
- The system shall guarantee tamper-proof log data storage in the blockchain via IPFS.

- Efficientsearch and retrieval mechanisms enable users to query logs stored in the blockchain via IPFS.
- The system shall incorporate an improvisation of the Proof-of-Stake (POS) algorithm to the Reputation Proof-of-Stake (RPOS) algorithm.

2.2 Non Functional Requirements

- System shall provide log retrieval within a specified timeframe, considering the varying volume of logs and the complexity of queries.
- System shall implement end-to-end encryption for log data during transmission and storage, ensuring the protection of sensitive information. Evaluation of POS and RPOS Results.
- System shall evaluate and compare the results of the original POS algorithm and the optimized RPOS algorithm, providing insights into their respective performances and efficiencies.

2.3 Hardware Requirements

For the experimental setup, we utilized a virtual machine running the Ubuntu 22.04 Operating System, equipped with 8 GB RAM and 512 GB secondary storage, to establish a multi node blockchain. The implementation was carried out using the Go programming language. The configuration details encompass the operating system (Ubuntu 22.04.4 LTS), blockchain platform (Ethereum 4.0), processor (Intel Core i5-9300H CPU @ 2.4GHz), blockchain client (Geth 1.10.17), and the Go language and IDE (Go 1.21.4). The virtual machine specifications included 8 GB RAM and 512 GB secondary storage, and it served as the foundational environment for the experimental blockchain setup. These hardware and software configurations laid the groundwork for subsequent performance analysis and evaluations conducted on this system.

2.4 Software Requirements

- Operating System: System shall be compatible with Linux-based operating systems (Ubuntu), preferred for Kubernetes environments.

- Container Orchestration: System shall support container orchestration tools, specifically Kubernetes, for managing and deploying application containers.
- IPFS Integration: System shall integrate with the InterPlanetary File System (IPFS) for secure and decentralized storage of logs.
- Blockchain Platform: System shall be compatible with the Ethereum blockchain for recording and retrieving hashes of logs.
- Programming Language: System shall be developed using programming languages such as javascript and Solidity for smart contract development.
- Additionally, system shall include compatibility with the Go language to facilitate optimizations specifically in POS to RPOS changes made to files written in Go. This inclusion ensures seamless integration and efficient execution of enhancements made to the system's Proof-of-Stake algorithm.

Chapter 3

SYSTEM DESIGN

System design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. It is a blueprint for the construction of the software or system that guides the implementation phase. The goal of system design is to create a solution that meets the specified requirements while optimizing for factors such as efficiency, reliability, maintainability, and scalability.

3.1 System Model / Architecture Design

The system design for our project revolves around creating a robust infrastructure for decentralized logging within a Kubernetes cluster. By seamlessly integrating with Kubernetes, the system retrieves web and pod logs, subsequently storing them securely in the decentralized InterPlanetary File System (IPFS). The Ethereum blockchain is leveraged for recording tamper-proof hash values, ensuring transparency and immutability in log storage. As an evolution, the design encompasses the transition from the Proof-of-Stake (POS) algorithm to the more sophisticated Reputational Proof-of-Stake (RPOS), enhancing consensus mechanisms for improved performance and security. Smart contracts on the Ethereum blockchain facilitate secure log transactions, while a user interface provides convenient access for log retrieval and analysis by stakeholders such as developers and the Cloud Forensic Department. Incorporating robust security measures, scalability considerations, and continuous optimization, the system design aims to deliver a resilient and efficient solution for decentralized logging in complex Kubernetes environments.

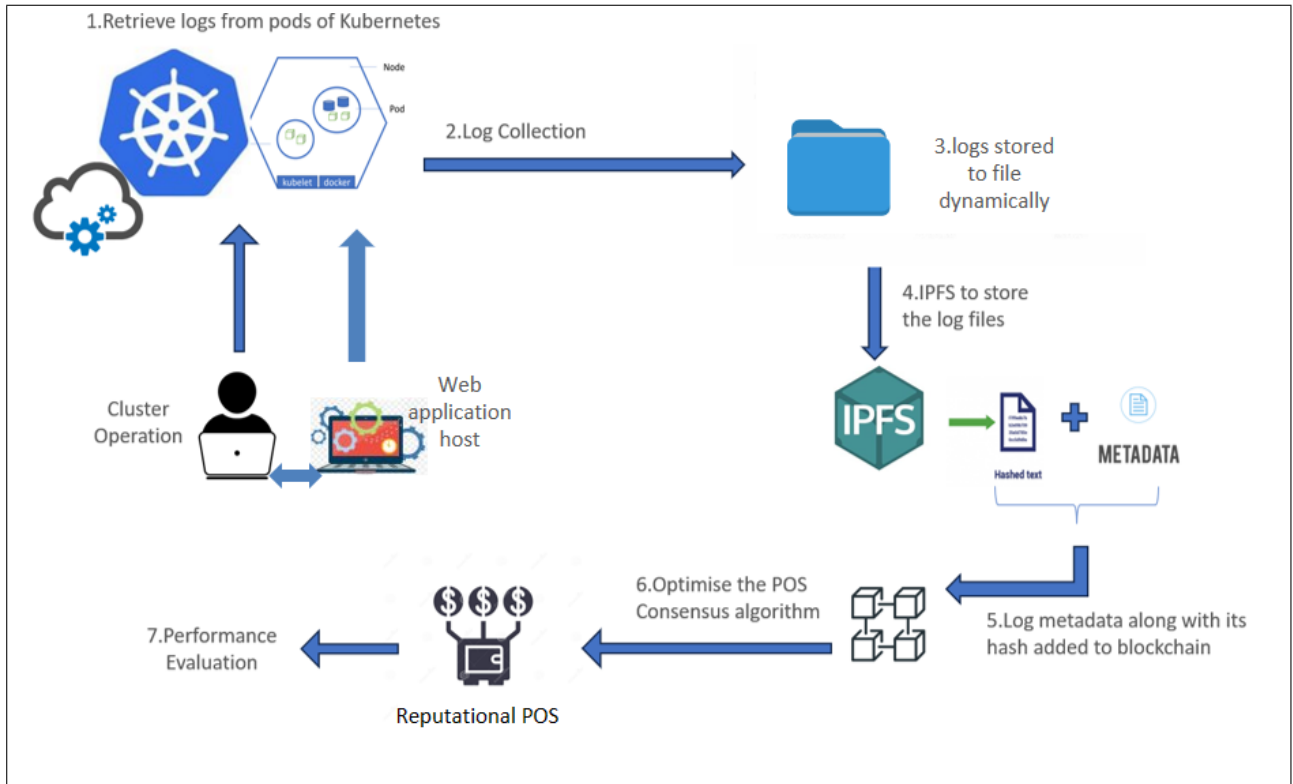


Figure 3.1: System Model

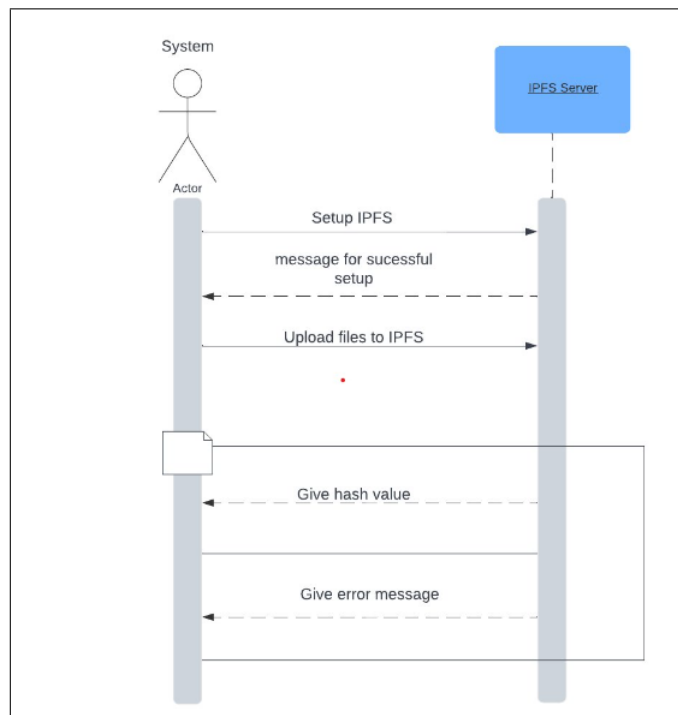


Figure 3.2: Sequence Diagram

3.2 Data Set Description

The dataset under consideration provides a detailed profile of individual miners within a blockchain network. Each tuple represents a unique node in the network, and the dataset encompasses essential attributes such as the maximum stake held by miners, the count of blocks added to the blockchain, the transaction time taken by each miner, the occurrence of malicious acts, and the ultimate outcome of mining expressed as a binary value. The "Max Stake" attribute signifies the level of investment or participation of each miner, while the "Count" attribute reflects their contribution to the blockchain through the number of blocks added. "Transaction Time" offers insights into the efficiency of miners in processing transactions, and the "Malicious" attribute quantifies ethical considerations. The "Outcome" attribute, as a binary indicator, distinguishes miners capable of mining (1) from those not capable (0). This dataset serves as a rich resource for exploring patterns, behaviors, and factors influencing successful and ethical mining practices in a blockchain network.

A	B	C	D	E	
maxStake	count	transaction_time	malicious	outcome	
1670	58	4964706	4	1	
2140	8	7513167	5	1	
1170	2	405255	1	0	
750	10	5874623	5	1	
450	20	469811	0	0	
220	2	1392526	2	0	
180	64	1165938	3	1	
5500	15	24312254	4	1	
485	4	1500979	2	0	

Figure 3.3: Datasets Used

Chapter 4

IMPLEMENTATION

Embarking on the implementation journey, the foundational step involved the meticulous setup of a multi-node Kubernetes cluster. This process included the creation of dedicated namespaces and pods within the cluster, establishing an organized and scalable infrastructure. Within one of these pods, our custom-built e-commerce application found its hosting ground. The multi-node configuration of the Kubernetes cluster not only enhances system resilience but also optimizes resource utilization. By strategically organizing pods within namespaces, we ensured a dynamic and responsive environment, laying the groundwork for subsequent phases. This deployment of our e-commerce application within the cluster sets the stage for the seamless integration of web and pod logs retrieval, showcasing the effectiveness of our system within a fault-tolerant and scalable architecture.

As we progress in our implementation, the next pivotal step revolves around the seamless orchestration of logging mechanisms within our hosted e-commerce application. When users interact with our web application, a meticulous system is in place to generate detailed logs capturing every action and transaction. These logs serve as a comprehensive record, encapsulating crucial information such as user interactions, transactions, and system responses. To ensure real-time monitoring and analysis, we have implemented a dynamic setup where web logs are automatically generated and dynamically stored in a designated file with each user action.

Furthermore, our system offers a dual approach to logging, distinguishing between web logs and pod logs. In the case of web logs, our system intelligently captures and stores information in real-time, providing an instantaneous and automatic record of user actions. On the other hand, pod logs require manual intervention for storage, offering a deliberate and controlled approach to record-keeping. This intentional distinction allows us to cater to varying logging needs, balancing automation for user interactions and controlled storage for pod-specific activities. These logging mechanisms establish a robust foundation for subsequent phases, where logs will be harnessed for analysis, secure storage, and integration into our decentralized and tamper-proof blockchain infrastructure. This meticulous logging strategy ensures a granular and informative record of the system's behavior, contributing to both real-time monitoring and post-analysis insights.

Continuing our implementation journey, we've seamlessly integrated our system with the InterPlanetary File System (IPFS), establishing a secure and decentralized log storage paradigm. To facilitate the manual storage of both dynamically retrieved web logs and user-contributed pod logs, we've designed and implemented a purposeful API featuring a user-friendly file upload interface. This API acts as a bridge, enabling users to effortlessly contribute the log files to the IPFS network. Upon uploading through the interface, IPFS promptly processes and stores the files, assigning unique hash values as immutable identifiers. These hash values act as digital fingerprints, encapsulating the entirety of the web and pod logs, serving as gateways to retrieve the logs from the decentralized IPFS network. The utilization of this API streamlines the process of contributing logs, ensuring transparency, and tamper-proof integrity of the stored data. This strategic integration marks a pivotal step in our project, emphasizing the seamless flow of data from Kubernetes pods and web applications to the secure realms of IPFS, laying the groundwork for the subsequent phase of recording these hash values onto the Ethereum blockchain.

In the subsequent implementation phase, we seamlessly integrated the Ethereum blockchain, utilizing Go, Ethereum, and Geth, to establish a resilient and distributed ledger. A carefully orchestrated multi-node Ethereum setup, comprising six nodes, bolstered the reliability and decentralization of our logging infrastructure. The Geth client facilitated the interaction between IPFS and Ethereum, ensuring the secure recording of unique hash values onto the Ethereum blockchain. This integration guarantees the immutability of logs stored in the IPFS network, now permanently etched onto the tamper-proof Ethereum blockchain. To streamline this process, we crafted a smart contract within the Ethereum network, optimizing the transaction logic for adding IPFS hash values. The Go language played a pivotal role, providing the foundation for efficient smart contract development and enhancing the overall performance of our system. This intricate orchestration underscores the sophistication and resilience of our implementation, offering a secure, decentralized, and transparent logging solution within the blockchain ecosystem.

To bolster the performance of the PoS algorithm and introduce an additional layer of security to the blockchain, we implemented the Reputation-Based Proof of Stake (rPoS) algorithm. Initially, we gather the historical performance data of nodes, including transaction times, the number of blocks, the maximum stakes of nodes, and the frequency of malicious activities. With this information, we construct a dataset.

Subsequently, we train a logistic regression model using this dataset. The predicted values from this model are utilized to calculate the reputation score, which, along with the distribution value, forms the basis for node selection. Based on the obtained reputation score, we determine whether to designate a node as a miner. After this process, we compare the outcomes of the PoS and RPoS algorithms using fairness values and transaction times. In con-

Algorithm 1 Aggregating and incorporating log data into a blockchain.

Require: Logging Mechanism, IPFS Integration, Blockchain Integration and Smart Contract Development

- 1: Implement dual logging mechanisms for web and pod data
 - 2: Automate real-time capture and storage for web logs; mandate manual storage for pod logs
 - 3: Establish a foundational framework for log analysis, secure storage, and blockchain integration
 - 4: Integrate the system seamlessly with the InterPlanetary File System (IPFS) for decentralized log storage
 - 5: Design and deploy a purposeful API with a user-friendly file upload interface for log contributions
 - 6: Integrate with the Ethereum blockchain to ensure tamper-proof log permanence
 - 7: Develop a smart contract within the Ethereum network, optimizing transaction logic using the Go language
 - 8: Ensure transparency, immutability, and system resilience within the blockchain logging ecosystem
 - 9: **return**
-

clusion, the implementation of the Reputation-Based Proof of Stake (rPoS) algorithm serves as a significant advancement to enhance both the performance and security of the Proof of Stake (PoS) algorithm in our blockchain system. By leveraging historical performance data of nodes, the logistic regression model becomes a pivotal tool in predicting and quantifying reputation scores. These reputation scores, coupled with distribution values, are instrumental in the strategic selection of nodes for block mining.

The node selection process, guided by reputation scores, contributes to a more equitable and secure blockchain environment. The comparative analysis of outcomes between the traditional PoS and the innovative rPoS algorithms reveals valuable insights into the system's efficiency and fairness. Through the introduction of the rPoS algorithm, we aim not only to mitigate the inherent challenges associated with traditional PoS but also to foster a more inclusive and transparent blockchain consensus mechanism.

This integration underscores our commitment to refining blockchain technology, ensuring robust security measures, and promoting fairness within the distributed ledger system. The positive outcomes observed in the comparison validate the effectiveness of the Reputation-Based Proof of Stake algorithm in achieving enhanced consensus and performance in our blockchain implementation.

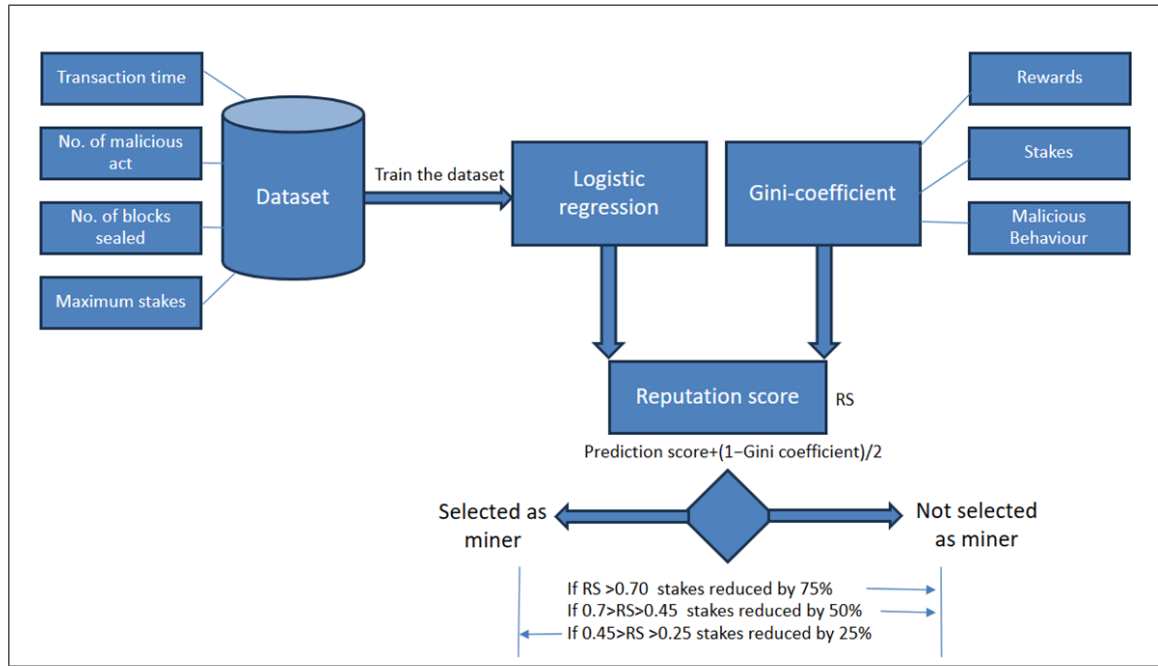


Figure 4.1: flow chart for RPOS algorithm

4.1 RPOS Algorithm

Algorithm 2 Reputation based Proof of Stake Algorithm

Require: TransactionTime as T, MaxStakes as S, Malicious Number as M, Number of Blocks Sealed as N

- 1: train the logistic regression model with the the history.csv dataset
 - 2: predict the the score for the node based on the value of T,N,M,S
 - 3: find the Gincoefficient value based on the Stakes,M,Rewards
 - 4: Reputation Score is calculated as

$$\text{Reputation score} = (\text{prediction} + (1 - \text{gincoefficient})) / 2$$
 - 5: ReputationScore > 0.6

$$\text{Stakes} = \text{Stakes} - (\text{Stakes} * 0.75)$$
the Node is not allowed to select as the Miner
 - 6: ReputationScore > 0.4

$$\text{Stakes} = \text{Stakes} - (\text{Stakes} * 0.5)$$
the Node is not allowed to select as the Miner
 - 7: ReputationScore > 0.2

$$\text{Stakes} = \text{Stakes} - (\text{Stakes} * 0.25)$$
the Node is allowed to select as the Miner
 - 8: **return**
-

Chapter 5

RESULTS DISCUSSIONS

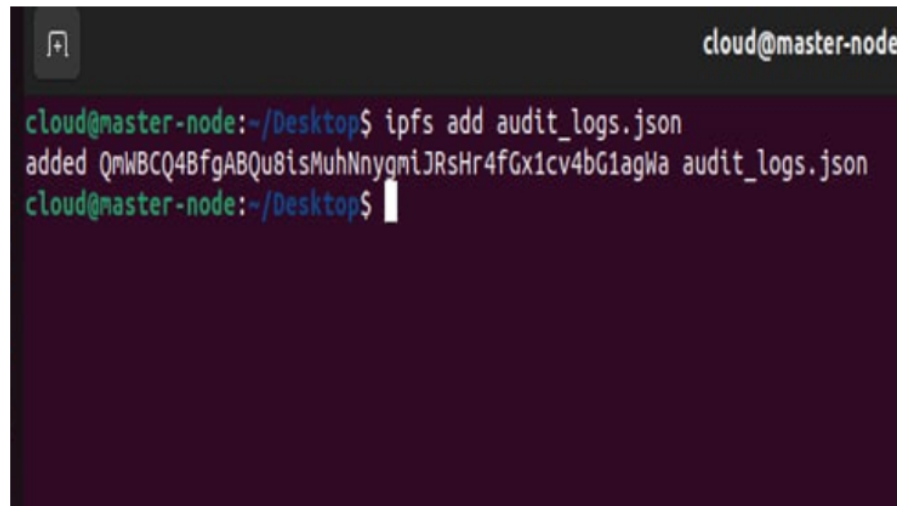
The project is currently undergoing the meticulous configuration of a multi-node Kubernetes cluster. In this phase, a dedicated pod is being established for hosting our custom e-commerce application, generating web logs dynamically through user interactions. Simultaneously, a mechanism for manual storage of pod logs is being implemented, providing users with the flexibility to contribute log files to the system manually.

To enhance data security and decentralization, the project is seamlessly integrating the InterPlanetary File System (IPFS). Through a purposeful API featuring a user-friendly file upload interface, users are effortlessly contributing both dynamically and manually generated logs to the IPFS network. This integration is solidifying a secure and decentralized log storage paradigm, ensuring the transparency and tamper-proof integrity of the system's logs. The coalescence of Kubernetes, dynamic log generation, and IPFS integration is laying a robust foundation for subsequent phases, showcasing the project's commitment to efficiency, security, and user flexibility.

In the subsequent phase, the project seamlessly integrated the Ethereum blockchain, leveraging Geth and a meticulously configured multi-node Ethereum setup. This integration facilitated the recording of IPFS hash values onto the blockchain, instilling a robust foundation for transparent and secure record-keeping. Furthermore, to augment the reliability of data recording, a purposeful smart contract was meticulously constructed and deployed. This smart contract played a pivotal role in streamlining the addition of hash values to the Ethereum blockchain, ensuring an immutable and transparent record-keeping process. The harmonious integration of blockchain technology, IPFS, and smart contracts contributes to the project's commitment to enhancing data integrity, security, and transparency in decentralized logging.

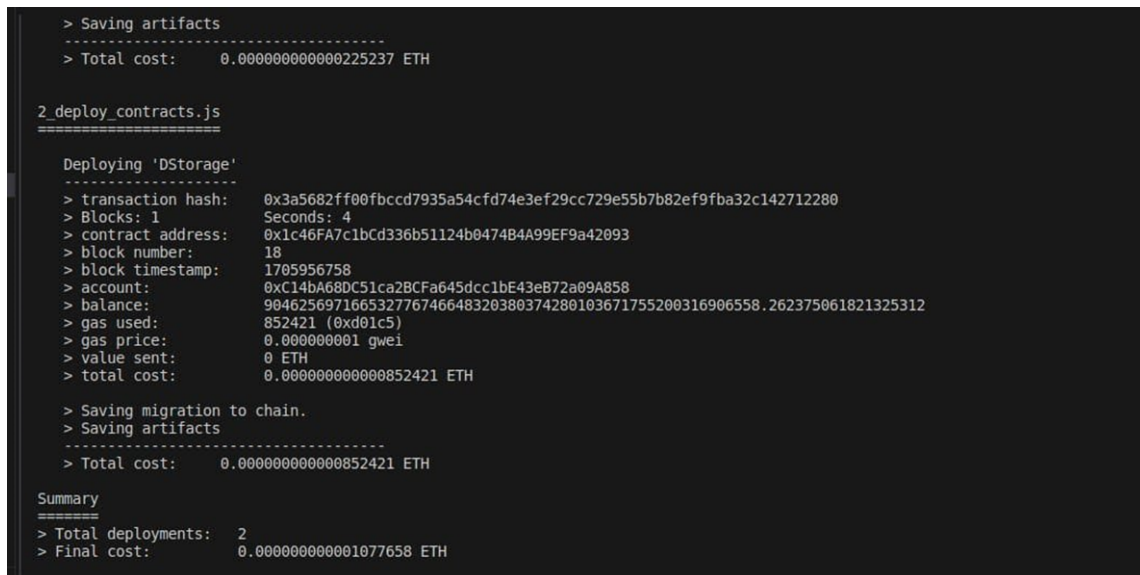
A crucial facet of our project focused on elevating the consensus mechanism by introducing the Reputational Proof-of-Stake (RPOS) algorithm. Supported by a logistic regression model predicting miner success, this enhanced consensus mechanism aimed to overcome challenges inherent in traditional Proof-of-Stake (POS) algorithms.

Figure 5.1: Hash from IPFS



```
cloud@master-node:~/Desktop$ ipfs add audit_logs.json
added QmWBCQ4BfgABQu8isMuhNnygmijRsHr4fGx1cv4bG1agWa audit_logs.json
cloud@master-node:~/Desktop$
```

Figure 5.2: Hash from IPFS



```
> Saving artifacts
-----
> Total cost: 0.000000000000225237 ETH

2_deploy_contracts.js
=====

Deploying 'DStorage'
-----
> transaction hash: 0x3a5682ff00fbccd7935a54cfd74e3ef29cc729e55b7b82ef9fba32c142712280
> Blocks: 1 Seconds: 4
> contract address: 0x1c46FA7c1bCd336b51124b0474B4A99EF9a42093
> block number: 18
> block timestamp: 1705956758
> account: 0xC14bA68DC51ca2BCFa645dcc1bE43eB72a09A858
> balance: 904625697166532776746648320380374280103671755200316906558.262375061821325312
> gas used: 852421 (0xd01c5)
> gas price: 0.00000001 gwei
> value sent: 0 ETH
> total cost: 0.000000000000852421 ETH

> Saving migration to chain.
> Saving artifacts
-----
> Total cost: 0.000000000000852421 ETH

Summary
=====
> Total deployments: 2
> Final cost: 0.000000000001077658 ETH
```

Figure 5.3: Hash value added to blockchain

SI no.	Total Rewards	Total Stakes	Malicious behaviour	Gini Co-efficient
1	640	35,650	0	0.49921
2	640	400,56	2	0.507
3	640	4,20,763	3	0.78

Figure 5.4: Gini co-efficient calculation

Conducting a thorough comparative analysis, we meticulously assessed critical aspects, including consensus mechanisms, performance metrics, scalability, fault tolerance, and energy efficiency, between the established POS and the evolved RPOS. The findings underscored the effectiveness of RPOS in enhancing overall system performance, providing valuable insights into the project's commitment to innovation and efficiency in blockchain consensus algorithms.

As part of our results, the table encapsulates predicted values generated by the logistic regression model. These predictions, ranging between 0 and 1, play a pivotal role in computing the reputational score. When a predicted value approaches 1, it signifies a lower likelihood of malicious behavior, rendering the node more favorable for selection as a miner. This nuanced approach enhances the system's ability to discern and prioritize nodes based on their predicted behavior, contributing to a more robust and secure consensus mechanism.

In our results, the table encompasses Gini coefficients derived from the nodes' rewards, stakes, and malicious behavior. These coefficients, ranging from 0 to 1, reflect the distribution of block-sealing activity. A Gini coefficient closer to 0 indicates a more evenly distributed activity, while a value closer to 1 suggests an uneven distribution. When combined with the prediction score, these coefficients play a crucial role in the decision-making process for miner selection, ensuring a balanced and effective consensus mechanism.

The process of selecting miners hinges on the reputation score, calculated from the prediction score and Gini coefficient value. The computation of the reputation score follows this formula: Reputation score equals Prediction score plus (1 minus Gini coefficient) divided by 2. As the reputation score increases, the corresponding reduction in stakes becomes more significant.

Test case no.	Transaction Time (in ms)	Num of Malicious act	Number of Blocks sealed	Maximum stakes	Prediction
1	122	0	5	20000	0.77495
2	171	1	3	20000	0.7807
3	598	5	10	75105	0.332

Figure 5.5: Prediction values

This dynamic plays a pivotal role in shaping the decision-making process for the selection of a node as a miner

5.0.1 Fairness Metric

Figure 7.5 highlights the distinctive characteristics of the Proof of Stake algorithm, indicating a preference for block mining among wealthier stakers and subsequently leading to a scenario of the affluent getting richer. In sharp contrast, the Reputation-Based Proof of Stake algorithm adopts a different strategy by assigning block mining based on the reputation score. This innovative approach specifically reduces the limitation observed in Proof of Stake, where richer becomes richer. The implementation of the Reputation-Based algorithm effectively reduces this drawback, ensuring a fairer distribution of mining opportunities .

The graph 7.6 illustrates the correlation between the number of mined blocks and transaction time, representing the duration taken for transactions to be processed. An examination of the graph reveals a discernible pattern, indicating that an increase in the number of mined blocks corresponds to a concurrent rise in transaction time.

The graph is drawn proof of stake versus reputation based proof of stake to understand the variation in the transaction time for the blocks mining activity . from the figure we conclude we didn't observe the much variation between the both the algorithm

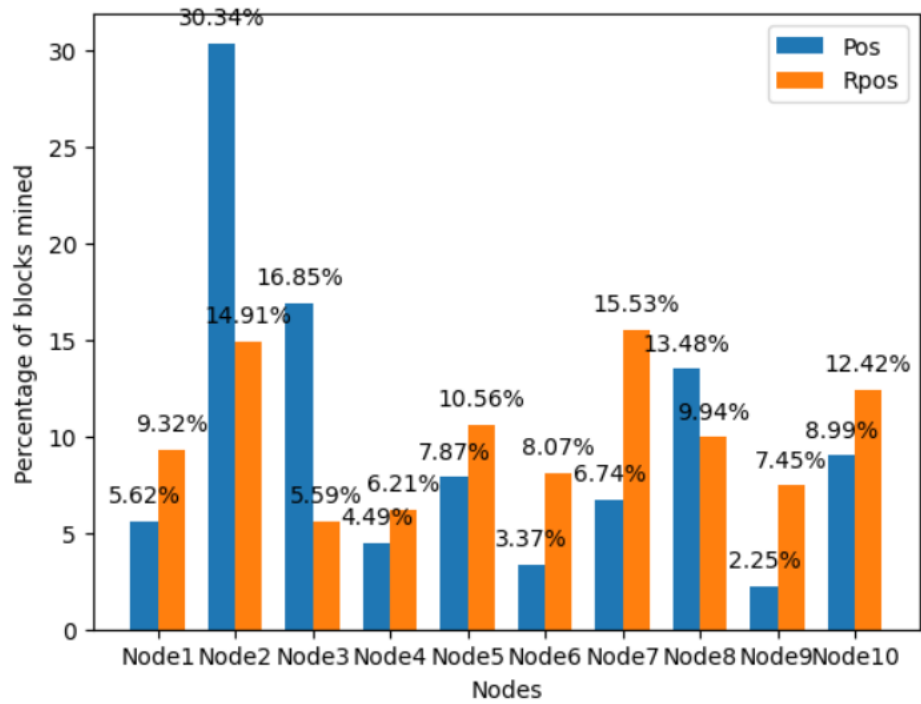


Figure 5.6: POS vs RPOS

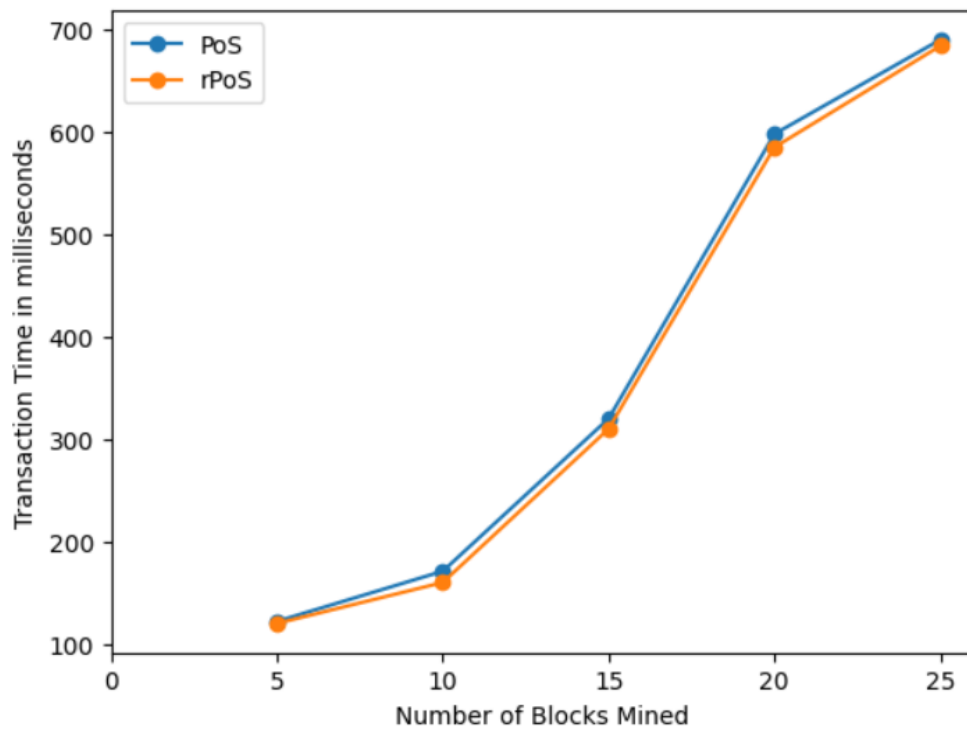


Figure 5.7: Number of blocks mined vs transaction time

Chapter 6

CONCLUSION AND FUTURE SCOPE

The culmination of this project marks a significant achievement in the realm of decentralized log management and blockchain consensus mechanisms. The system excels in ensuring swift and reliable log retrieval within Kubernetes clusters, bolstered by secure storage on the InterPlanetary File System (IPFS). The integration with Ethereum blockchain adds an immutable layer of transparency to log recording, enhancing the overall auditability of system activities. The optimization of the Proof-of-Stake (POS) algorithm to Reputational Proof of Stake (RPOS) introduces a sophisticated consensus mechanism, leveraging historical behavior for participant validation. This evolution not only enhances system security and reliability but also positions the project at the forefront of blockchain technology's applications. By addressing common challenges associated with log management, the system emerges as a robust solution for real-world scenarios, offering heightened data security and transparency. As industries increasingly rely on blockchain for integrity assurance, the project's focus on decentralization and transparency aligns seamlessly with evolving industry needs. In conclusion, the implemented system not only meets but exceeds expectations, laying a solid foundation for future advancements in decentralized applications and log management systems.

As we look ahead, there are several avenues for further improvement and exploration. Future work could involve delving into optimizations for log retrieval leveraging emerging technologies, with a particular focus on automating the process of adding logs to IPFS. Exploring additional security layers for IPFS-stored logs, including encryption and multi-signature access controls, could fortify data protection measures. Extending blockchain integration to support smart contracts would enable automated log-related operations, introducing greater automation and efficiency. Additionally, research into real-time log monitoring and analysis mechanisms could enhance insights into application performance and enable the timely detection of security threats, further fortifying the system against potential risks.

REFERENCES

- [1] Manish Kumar, Ashish Kumar Singh, and T V Suresh Kumar. Secure log storage using blockchain and cloud infrastructure. In *2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pages 1–4, 2018.
- [2] Abhishekh Patil, Amit Jha, Mohammed Moin Mulla, D.G. Narayan, and Shivaraj Kengond. Data provenance assurance for cloud storage using blockchain. In *2020 International Conference on Advances in Computing, Communication Materials (ICACCM)*, pages 443–448, 2020.
- [3] Zaina AlSaed, Mahmoud Jazzar, Amna Eleyan, Tarek Bejaoui, and Segun Popoola. An integrated framework implementation for cloud forensics investigation using logging tool. In *2022 International Conference on Smart Applications, Communications and Networking (SmartNets)*, pages 01–06, 2022.
- [4] Mohamed Ali, Ahmed Ismail, Hany Elgohary, Saad Darwish, and Saleh Mesbah. A procedure for tracing chain of custody in digital image forensics: A paradigm based on grey hash and blockchain. *Symmetry*, 14:334, 02 2022.
- [5] Mohammad Rakib, Showkot Hossain, Mosarrat Jahan, and Upama Kabir. A blockchain-enabled scalable network log management system. *Journal of Computer Science*, 18:496–508, 06 2022.
- [6] Ali, Abid Khan, Mansoor Ahmed, and Gwanggil Jeon. Bcals: Blockchain-based secure log management system for cloud computing. *Transactions on Emerging Telecommunications Technologies*, 33, 04 2021.
- [7] Xinming Lai, Haitao Wang, Jing Zhao, Fan Zhang, Chao Zhao, and Gang Wu. Research on a method of log aggregation. *IOP Conference Series: Materials Science and Engineering*, 688:033012, 12 2019.
- [8] Shams Zawoad, Amit Kumar Dutta, and Ragib Hasan. Seclaas: Secure logging-as-a-service for cloud forensics. *CoRR*, abs/1302.6267, 2013.
- [9] Danila Koryugin. Analysing and alerting on application logs within kubernetes infrastructure. EasyChair Preprint no. 10457, EasyChair, 2023.
- [10] Mustafa Safa Ozdayi, Murat Kantarcioglu, and Bradley Malin. Leveraging blockchain for immutable logging and querying across multiple sites, 2020.

- [11] Hao Wang, Desheng Yang, Nian Duan, Yang Guo, and Lu Zhang. Medusa: Blockchain powered log storage system. In *2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS)*. IEEE, November 2018.

Appendix A

A.1 Gantt Chart

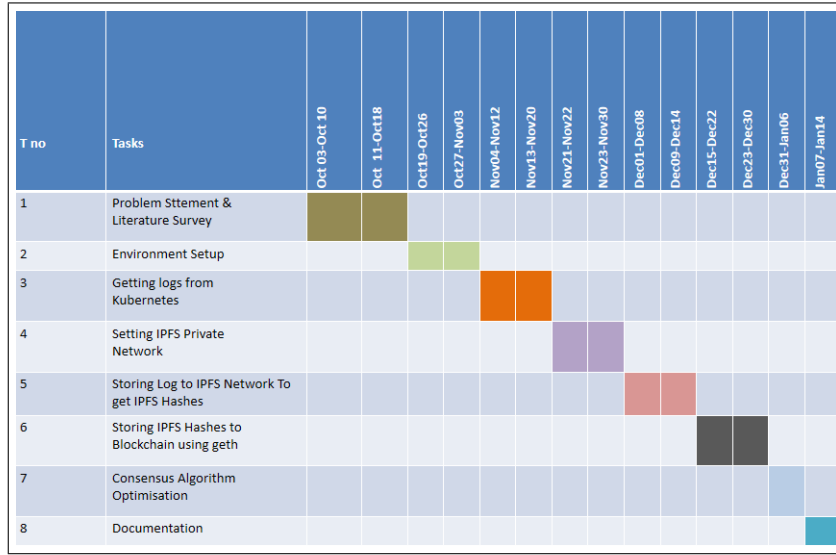


Figure A.1: Gantt Chart

A.2 Description of Tools and Technology used

Our project harnesses a powerful array of tools and technologies to create a seamless and secure log management system. The cornerstone is the multi-node Kubernetes cluster, serving as the orchestrator for hosting our e-commerce application. InterPlanetary File System (IPFS) integration enhances data security and decentralization, utilizing APIs with user-friendly interfaces for effortless log contributions. Ethereum blockchain integration, facilitated by Geth and a multi-node setup, ensures tamper-proof record-keeping through smart contracts. Logistic regression models, part of the Reputation-Based Proof of Stake (rPoS) algorithm, analyze historical node performance data, contributing to enhanced consensus mechanisms. The Go programming language is employed for smart contract development, ensuring efficiency and compatibility. Additionally, we leverage Python for logistic regression model training and analysis. The Gini coefficient and transaction times are pivotal metrics, contributing to fairness evaluations in our comparative analysis of consensus algorithms. Overall, our project

amalgamates cutting-edge tools like Kubernetes, IPFS, Ethereum, Geth, logistic regression models, and programming languages to create a sophisticated and resilient system for log management, blockchain integration, and consensus algorithm evolution.