TRUSTWORTHY AND EFFICIENT BLOCKCHAIN-BASED E-COMMERCE MODEL

by

Valli Sanghami Shankar Kumar

A Dissertation

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Doctor of Philosophy

Department of Electrical and Computer Engineering

Indianapolis, Indiana

August 2024

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. John Lee, Chair

Department of Electrical and Computer Engineering

Dr. Qin Hu

Department of Computer and Information Science

Dr. Dongsoo Stephen Kim

Department of Electrical and Computer Engineering

Dr. Brian King

Department of Electrical and Computer Engineering

Approved by:

Dr. Brian King

To my loved ones who have unwaveringly believed in me and supported me through every circumstance.

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to the following people who have played a significant role in the completion of this dissertation: First and foremost, I am deeply thankful to my research advisors, Dr. John Lee and Dr. Qin Hu. Their patience, expertise, and unwavering support have been instrumental in my growth and learning throughout this journey. As I transitioned from an Electrical background to Computer Engineering, I encountered numerous challenges, and both Dr. Lee and Dr. Hu were patient mentors who helped me navigate this transition with ease. Working with them on publishing papers was always a pleasure, as our collaborations led to the exploration of novel ideas and the integration of various technologies with blockchain. Our inspiring conversations have played a crucial role in shaping the person I am today, instilling in me a passion for creativity in all my endeavors. Without their assistance and guidance, this research journey would not have been as enriching and fulfilling as it has been. I am grateful for their mentorship and the impact they have had on my academic and personal development.

To my father, K. Shankar Kumar, whose endless support and encouragement have been a constant source of strength. More than a father, he has been my best friend, always there to lend a listening ear and offer guidance. "Though the world may stand against me, he remains steadfastly by my side, an unwavering member of my team". To my mother, Mrs. M. Uma Soundari, who has been my anchor of support and a guiding light in my life. Her wisdom has helped me navigate through life's challenges with confidence and determination. All my accomplishments are a reflection of you, Mom. Your dream to pursue a Ph.D. lives on through mine; my Ph.D. journey is a tribute to you. To Harsh Verma, whose random conversation about Bitcoin ignited my interest in blockchain technology and led me to choose this specialization. He has been not only a great friend but also a source of encouragement, insightful discussions, and moral support. He is someone who understands me better than I understand myself, constantly challenging me to step out of my comfort zone and become the best version of myself.

To my friend, Niranjan Ravi who started his journey at the same time as mine and guided me in research when needed. Your friendship has made this journey more enjoyable and meaningful. To all the remarkable individuals I had the pleasure of meeting during my time at the Center for Teaching and Learning, IUPUI: I am immensely grateful to my supervisors, Jessica and Anusha S. Rao, for their unwavering support and constant motivation throughout my journey. Their guidance has been instrumental in shaping my work. A special thank you to Marie E. Brown, the director of CTL, for her caring nature. Marie always takes the time to alleviate any stress I may have about my Ph.D. and job search, and her encouraging pep talks have been a source of comfort. I am also deeply thankful to Doug Jerolimov and Richard C. Turner, whose expertise I have come to deeply value and admire. In the Center for Teaching and Learning, I have found not only a place of work but also a second home. My colleagues have become my friends, and I am grateful for the camaraderie and support they have provided.

I extend my heartfelt gratitude to Sherrie Tucker for her invaluable assistance with timely reminders and document editing. I fondly recall the occasion when I received financial support from the department for the first time; instead of merely sending an email, Sherrie took the time to personally call me and share the news. Her understanding of the significance of this support in my life speaks volumes about her kindness and thoughtfulness. I am grateful to have met such a gem of a person during this journey. Lastly, to all those who have played a part, no matter how small, in the completion of this dissertation, I extend my heartfelt appreciation.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Amidst the rising popularity of digital marketplaces, addressing issues such as non-payment/non-delivery crimes, centralization risks, hacking threats, and the complexity of ownership transfers has become imperative. Many existing studies exploring blockchain technology in digital marketplaces and asset management merely touch upon various application scenarios without establishing a unified platform that ensures trustworthiness and efficiency across the product life cycle. In our thesis, we focus on designing a reliable and efficient e-commerce model to trade various assets. To enhance customer engagement through consensus, we utilize the XGBoost algorithm to identify loyal nodes from the platform entities pool. Alongside appointed nodes, these loyal nodes actively participate in the consensus process. The consensus algorithm guarantees that all involved nodes reach an agreement on the blockchain's current state. We introduce a novel consensus mechanism named Modified-Practical Byzantine Fault Tolerance (M-PBFT), derived from the Practical Byzantine Fault Tolerance (PBFT) protocol to minimize communication overhead and improve overall efficiency. The modifications primarily target the leader election process and the communication protocols between leader and follower nodes within the PBFT consensus framework.

In the domain of tangible assets, our primary objective is to elevate trust among various stakeholders and bolster the reputation of sellers. As a result, we aim to validate secondhand products and their descriptions provided by the sellers before the secondhand products are exchanged. This validation process also holds various entities accountable for their actions. We employ validators based on their location and qualifications to validate the products' descriptions and generate validation certificates for the products, which are then securely recorded on the blockchain. To incentivize the participation of validator nodes and uphold honest validation of product quality, we introduce an incentive mechanism leveraging Stackelberg game theory.

On the other hand, for optimizing intangible assets management, we employ Non-Fungible Tokens (NFT) technology to tokenize these assets. This approach enhances traceability of ownership, transactions, and historical data, while also automating processes like dividend distributions, royalty payments, and ownership transfers through smart contracts. Initially,

sellers mint NFTs and utilize the InterPlanetary File System (IPFS) to store the files related to NFTs, NFT metadata, or both since IPFS provides resilience and decentralized storage solutions to our network. The data stored in IPFS is encrypted for security purposes. Further, to aid sellers in pricing their NFTs efficiently, we employ the Stackelberg mechanism. Furthermore, to achieve finer access control in NFTs containing sensitive data and increase sellers' profits, we propose a Popularity-based Adaptive NFT Management Scheme (PANMS) utilizing Reinforcement Learning (RL). To facilitate prompt and effective asset sales, we design a smart contract-powered auction mechanism.

Also, to enhance data recording and event response efficiency, we introduce a weighted L-H index algorithm and transaction prioritization features in the network. The weighted L-H index algorithm determines efficient nodes to broadcast transactions. Transaction prioritization prioritizes certain transactions such as payments, verdicts during conflicts between sellers and validators, and validation reports to improve the efficiency of the platform. Simulation experiments are conducted to demonstrate the accuracy and efficiency of our proposed schemes.

1. INTRODUCTION

In the last decade, digital marketplaces have experienced substantial growth, propelled by technological advancements and increased global accessibility. This expansion has provided businesses with new opportunities to reach a broader customer base, a trend that has further accelerated in the aftermath of the COVID-19 pandemic. In the United States alone, an estimated 230.5 million people shopped online in 2021 due to the availability of a variety of products, convenience, low cost, and time effectiveness [1]. Concurrently, the secondhand market has seen a surge in popularity, driven by consumers' shift in preferences toward ethical purchasing practices. According to GlobalData, the U.S. secondhand market will reach $82 billion by 2026 [2].

Despite the expansion of digital marketplaces, consumers face numerous uncertainties due to several significant issues. These include the reliance on third-party intermediaries, which can result in data breaches containing sensitive financial and personal information, non-payment, or non-delivery crimes as well as centralization. There is also a risk to the credibility of sellers and the quality of goods, particularly in the case of secondhand products. Shopping for such secondhand goods requires considerable time, research, and courage but often they result in purchases that do not meet consumers' expectations in terms of color, texture, and quality. Thus, while technological advancements have propelled the growth of online marketplaces, they have also created opportunities for fraudsters to exploit. Customers also encounter additional challenges such as refund discrepancies between sellers and buyers, payment delays, product recalls, lack of transaction transparency, and lost transactions. Moreover, traditional digital marketplaces often impose high transaction fees.

Recently, blockchain technology has been increasingly recognized as a tool to address trust-related issues in different industries by various researchers. Blockchain is a distributed ledger that records the provenance of digital information. Each block in the blockchain contains a batch of transactions, and these blocks are linked together chronologically, forming a chain. Blockchain operates on a peer-to-peer network, where each participant (node) in the network contains a copy of the entire blockchain, thus protecting its immutability nature and maintaining the integrity and trustworthiness of the distributed ledger system.

By inherent design, blockchain technology ensures trust and provides transparency, thus making it a promising and revolutionary technology for many industries, particularly in digital marketplaces.

There are three types of blockchain:

1. **Public blockchain:** A public blockchain network is one where any user can participate and leave without any restrictions.

2. **Private blockchain:** Users can join a private blockchain network only through an invitation where their identity or other required information is authentic and verified. The owner of the network has the right to override, edit, or delete the necessary entries on the blockchain as required

3. **Consortium blockchain:** A blockchain network where the consensus process is closely controlled by a pre-selected number of users.

Smart contracts play a critical role in enhancing the usability, efficiency, security, and transparency of blockchain-based systems by enabling participants to engage in transactions securely and transparently, without relying on intermediaries. Smart contracts are digital agreements executed automatically upon meeting predefined terms and conditions. They streamline agreement execution, providing immediate certainty to all involved parties without causing delays. For example, a simple smart contract between a supplier (Jane LLC) and a retailer (V) in a pharmacy industry may have certain terms accepted by both parties. Figure 1.1 displays the working of a smart contract between the supplier and retailer while delivering pharmaceutical drugs. The smart contract may include terms like:

1. if the pharmaceutical drugs arrive on time and in the right conditions, then execute a payment from the retailer to the supplier in full amount.

2. if the goods arrive on time but 20% pharmaceutical drugs are not maintained at the right conditions (e.g., temperature), then execute a payment from the retailer to the supplier for 80% of the full amount.

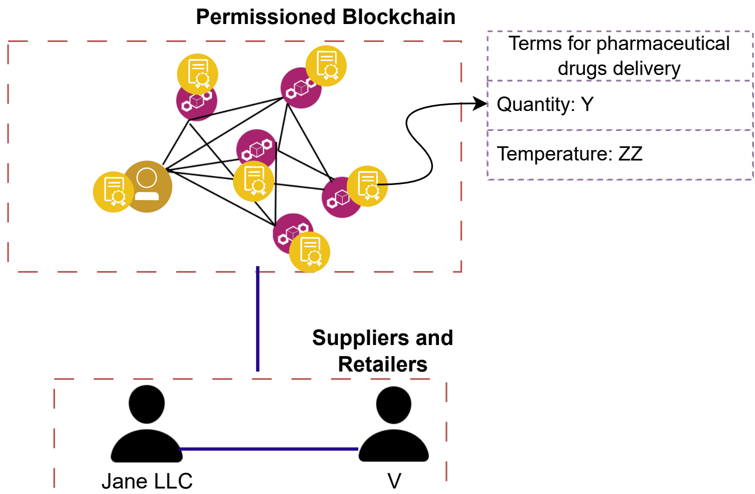


Figure 1.1. Example of execution of a smart contract between two parties.

Another technology that is making headlines as a solution to some of the critical problems related to intellectual property ownership and royalties is Non-Fungible Tokens (NFT) [3]. Unlike cryptocurrencies such as Bitcoin or Ethereum, which are fungible and can be exchanged on a one-to-one basis, NFTs represent unique digital assets that are indivisible and cannot be replicated; i.e., they are not interchangeable with other items because they have unique properties. Hence, NFTs can be considered as a signature for digital assets and are based on blockchain technology to prove authenticity, thus providing a clear and verifiable proof of ownership on the blockchain. Each NFT is minted as a unique token on the blockchain, containing metadata that provides information about the asset it represents. The immutable and transparent nature of blockchain technology ensures that the ownership history and origin of each asset are recorded and easily accessible. This helps prevent fraud and ensures that buyers can trust the authenticity of the assets they are acquiring, thus making the usage of NFTs in intangible assets (patents, novel product designs, sensitive data) more critical. They unlock new possibilities for creators, collectors, and investors, while also addressing some of the key challenges associated with the ownership and transfer of intangible assets in the digital age. NFTs also have the potential to improve finer data access control, easier collaboration, and provide dual ownership. To store NFT data efficiently and cost-effectively, the InterPlanetary File System (IPFS) is utilized. Instead of storing files on a single server, IPFS distributes files across a network of nodes, making them highly available as well as difficult for any one entity to control the distribution of the data. When an NFT is created, the associated metadata, such as the asset's description, ownership information, and image or media files, can be stored on IPFS. Figure 1.2 depicts an NFT containing the health data of a patient, with the content stored in IPFS.

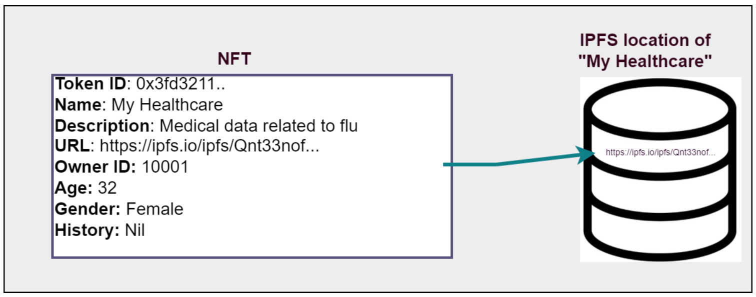


Figure 1.2. Schematic representation of an example NFT and its associated data stored in IPFS.

Blockchain technology and NFTs are the two most popular technologies gaining a lot of attention among researchers and [4]–[7] illustrate the pros and cons of blockchain technology and NFTs. There exists various research on various applications of blockchain in smart cities, such as maintenance of ID cards [8] and health records [9], verifying and sharing legal documents, automotive industry [10], interoperability of smart devices [11], and energy distribution [12]. Blockchain is also recognized as a tool to address trust-related issues in different industries by various researchers [13]–[22]. Most researches focus on leveraging blockchain technology in digital marketplaces [23]–[25], in patent management[26], [27], and in reforming the healthcare industry [28]–[31] by illustrating the current and future applications of blockchain technology within the healthcare industry, such as improving data management, secured data transfer, data storage, and managing supply chains. They also weigh up the pros and cons of deploying blockchain technology. Some studies [32]–[36] focus on leveraging blockchain technology in e-commerce but they do not detail the process of validating the quality of the products which is critical for the secondhand market. Also, transparency during validation is not enforced. On the other hand, transparency and automating certain actions are made possible in blockchain technology by leveraging smart contracts that can also lead to higher efficiency and lower administration costs.

However, many of the studies mentioned above do not establish a unified platform that ensures trustworthiness and fairness for a wide range of assets, nor do they potently enhance the network efficiency throughout the entire product life cycle. In this thesis, our central focus revolves around designing an enterprise model that encompasses trading a variety of assets through the utilization of blockchain technology. Our primary objective is to guarantee trustworthiness, accessibility, intellectual property protection, and efficiency within this model. The thesis details streamlining asset trading of both tangible (cars, electronics, clothes, etc.) and intangible assets (patents, novel product designs, sensitive health data), and incorporating techniques/algorithms to create a more efficient and frictionless experience for participants in the market, ultimately reducing costs and improving overall satisfaction. By leveraging blockchain technology, we aim to ensure the credibility of sellers, foster trust between sellers and buyers, and establish a transparent process. Here, we implement NFT technology with smart contracts to facilitate efficient ownership rights, royalties transfer, and other functionalities for intangible assets. Sellers are given the autonomy to set prices for these NFTs to provide them with competitive advantages over others. We also employ Stackelberg game theory to determine optimal pricing strategies for both sellers and buyers. To enhance finer access control, particularly for sensitive data, and boost sellers' profits, we propose the Popularity-based Adaptive NFT Management Scheme (PANMS) algorithm.

In regards to secondhand products, we introduce an additional step for product quality validation, conducted by validator nodes. These nodes authenticate the quality of products and confirm the accuracy of the original product descriptions provided by sellers. Validation occurs upon request, allowing consumers to trust the quality descriptions of purchased goods, thereby valuing the time and money invested in their purchases. Furthermore, we devise an incentive mechanism to encourage validator participation and ensure honest validation of product quality. This mechanism incorporates penalty and reputation scores to incentivize truthful validation, as it significantly impacts our model's effectiveness. Utility functions for both the platform and validators are defined within this incentive framework. Moreover, a Conflict Resolution Scheme (CRS) is designed to maintain fairness and transparency in resolving conflicts between sellers and validators regarding product quality. A jury pool comprising qualified validators is randomly selected to adjudicate any conflicts, ensuring impartial resolution. Similarly, an auction mechanism is designed for both tangible and intangible assets to assist the sellers in selling their assets effectively and promptly. Here, the highest bidder wins the product. Finally, to improve transaction efficiency, we propose two methods: efficient transaction broadcasting and transaction prioritization. To improve transaction broadcasting efficiency, we propose a weighted L-H algorithm that will determine more efficient nodes based on the nodes' degree, active time, and latency. Hence, broadcasting transactions to such nodes will improve the efficiency of transaction broadcasting. The second method is transaction prioritization which ensures the transactions are recorded not on a first-come-first-serve basis but on priorities. For example, transactions involving payments, verdicts by the jury nodes on conflicts, etc., that require immediate attention will be given the highest priority. To enforce transaction prioritization, we also introduce dynamic block creation.

In summary, the main contributions of this thesis are as follows:

1. A novel enterprise model is designed to ensure trustworthiness and efficiency in trading various assets. This model aims to facilitate seamless ownership transfer, uphold seller credibility, ensure product quality, foster trust among participants, and maintain transparency throughout the entire process.

2. Regarding the secondhand tangible products' ownership transfers, we employ validators to validate the products' quality. Also, to encourage validators' honest participation and incentivize validators, we devise an incentive mechanism using the Stackelberg game mechanism. The utilities of the platform and validators are determined efficiently to maximize payoffs.

3. To effectively transfer ownership rights of intangible assets and have better collaboration with multiple parties, NFT technology is implemented, and a Stackelberg-based pricing mechanism is implemented to determine the prices of the NFTs in an efficient manner.

4. An auction mechanism based on the first-price-sealed bid is designed for an effective and rapid means to sell assets and to improve the sellers' revenues.

5. To achieve finer access control in certain assets like critical health data, we propose an adaptive NFT management scheme based on Reinforcement Learning (RL), namely the Popularity-based Adaptive NFT Management Scheme (PANMS). Here, to deal with the uncertainty of the popularity of data fields, we model this challenge as a Combinatorial Multi-Armed Bandit (CMAB) problem and propose an online learning-based data field selection scheme to determine popular data fields and achieve higher rewards.

6. To improve transaction efficiency, the weighted L-H algorithm is designed to determine influential nodes that can broadcast transactions effectively, and transaction prioritization is introduced to prioritize certain significant transactions.

7. Experimental results showcase the effectiveness of the proposed framework.

The rest of this thesis is organized as follows. Chapter 2 investigates the most related work. Chapter 3 of this thesis introduces our system model and elucidates its significance, while Chapter 4 talks about the consensus mechanism of our model. The overall design of our proposed blockchain-based enterprise model is explained in Chapter 5, providing detailed steps on asset trading. Chapter 6 outlines the experimental evaluation setups and presents the results, whereas Chapter 7 serves as the concluding chapter of the thesis.

2. RELATED WORKS

In 2020, digital marketplaces experienced a significant surge in activity, driven by consumers' interest in the world's largest online platforms. These marketplaces boasted a wider selection, readily available inventory, and expedited delivery services, all of which became increasingly vital during the onset of the COVID-19 pandemic. Most research works investigate the significance of blockchain technology in digital marketplaces [16]–[18], [37]–[43] as blockchain technology has the potential to revolutionize marketplaces by increasing transparency, fostering trust among buyers and sellers who may not know each other directly, and creating an immutable audit trail of transactions, which is critical for accountability and dispute resolution purposes. As the technology continues to mature, its significance in marketplaces is likely to grow even further.

2.1 Intangible Assets

In recent years, the intersection of intangible assets and blockchain technology has garnered significant attention, presenting innovative solutions to various challenges in different industries. Intangible assets, such as intellectual property, sensitive data, and digital content, hold issues such as ownership rights management, privacy, and trust in transactions. Concurrently, blockchain technology has emerged as a promising tool for addressing these challenges, offering decentralized, transparent, and secure platforms for asset management and transactions. This section explores related works that delve into the integration between intangible assets and blockchain technology, aiming to unlock opportunities in the digital economy.

Some research works investigate blockchain for intangible assets like patents [26], [27], [44], [45] and sensitive health data [46]–[51]. In [44], Seyed et al. examined the requirements of presenting intellectual property assets, specifically patents, as Non-Fungible Tokens (NFTs). However, the consensus method used in the model is not addressed in detail. Also, incentive mechanisms for the validators or price determination mechanisms are not explained. Similarly, Wei et al. in [45] proposed a framework of personalized patent recommendation systems by leveraging hybrid patent analysis. Though a detailed recommendation analysis was made, [45] does not explain patent validation and price determination in detail. In [52], Omar et al. studied the idea of using NFT to track the supply chain for drugs that also enables the authentication of origin, transfer of ownership, and accessibility to drug information by different stakeholders, including end customers.

In another aspect, Xia et al. [53] proposed a system model for efficient and secured management of health records by using cloud repositories. They ensured security based on identification schemes and cryptographic methods. A setback of this model is that as the number of users increases, the latency also increases, which might cause issues in real-time operation. Also, though cryptography was used for data security, there are some problems with storing sensitive data in the cloud as explained in [54]. Although Esposito et al. [54] provided a detailed explanation of various drawbacks of using cloud storage techniques, few practical solutions exist that can overcome challenges related to cloud storage. Research works [55], [56] focus on the security aspects of using blockchain in securing sensitive health data and monetizing it. In [55], Azaria et al. designed a prototype of a decentralized record management system using blockchain which contains security measures, such as identity management, authorizations in various levels, and cryptographic encryption of sensitive patient data to prevent content tampering. Here, the data consumers act as miners for the smooth functioning of the network. In return, sellers' medical data are given as incentives to the miners/consumers. This may pose a serious threat to patients if their sensitive data are released to the hands of a malicious miner. Also, there is an absence of a finer access control mechanism for the patients to control the content of information and the miners they share it with. In [56], Li et al. proposed a novel patient-centric framework by leveraging attribute-based encryption methods for security purposes. However, attribute-based encryption has revocation problems and a lack of managing a wide range of attributes which are hard to address without high computational costs.

In most of the aforementioned systems, there is a notable absence of a comprehensive intangible assets trading system, leading to inflexibility and inefficiency in intangible asset trading. Moreover, finer access control mechanisms are lacking, further causing security issues and data breaches. Particularly in sensitive health data scenarios, hospitals typically retain ownership, thereby creating a need for innovative solutions. By incorporating price determination and auction mechanisms, as well as facilitating secure collaborations and ownerships, this thesis endeavors to establish a robust framework for intangible asset trading. Utilizing blockchain technology, dual ownership is extended to both patients and hospitals, ensuring heightened security and transparency. To streamline the trading of health data, a pricing mechanism is formulated to optimize strategies for both data owners and consumers. Additionally, to mitigate time and minting costs while enhancing access control, an adaptive NFT management scheme is proposed. This scheme aims to improve system efficiency and cost-effectiveness, ultimately improving the integrity and accessibility of intangible asset trading.

2.2 Tangible Assets

One area where blockchain holds immense promise is in asset trading, encompassing both new and secondhand tangible assets. While digital marketplaces continue to evolve, leveraging blockchain for the trading of tangible assets presents opportunities to streamline transactions, enhance trust, and optimize market efficiency. In this section, we delve into related works that explore the application of blockchain technology in the trading of tangible assets, while discussing the advancements and challenges in the marketplace ecosystem.

Research works [29], [57], [58] target blockchain for online marketplaces. In [57], Zhao focused on the transaction information that is stored in the database and tried to ensure it was not obtained by malicious actors by leveraging smart contracts and access control. However, the proposed blockchain network follows Proof of Work (PoW) consensus which would cause scalability and security issues. In [58], [59], Thomas et. al and Yi-Wei et. al implemented smart contracts in the supply chain to track shipments, delivery, and payments. Thomas et. al in [59] created a blockchain-based app to trace and maintain temperature and quality in the pharmaceutical supply chain by implementing smart contracts at various points. However, smart contract design and the role of different players in the platform are not explained in detail. Also, in [58], Yi-Wei et. al used smart contracts to perform credible transactions without trusted third parties, and the transactions on the blockchain are trackable and irreversible ensuring that there is no breach of the contract, although a complete system model design is not detailed.

Due to the significant growth of the secondhand market in recent decades and increasing affinity to sustainability [60], some research works investigate the risks of the secondhand market, the factors influencing customers, etc.,[61]–[64]. The drivers and barriers of the secondhand market, specifically the stand of the consumers regarding the uncertainties in the product quality along with the presence of counterfeit products in the market have been discussed in [61]–[63]. Lou et al. [64] focused on a survey to understand the effect of quality when purchasing second-hand luxury goods, and found that 82.1% of the consumers responded that they have trouble in confirming the products' qualities before buying the products, which affects their trust in the secondhand market. Further, [65]–[67] discusses the significance of blockchain technology in the secondhand market, particularly in ownership verification and product tracking. In [65], Kulkarni detailed the trust issues in the resale market, even when the platforms sell validated and certified products. Since the customers need to trust a centralized entity (platform) that validates the products honestly there is no transparency in the validation process. This situation can be rectified by leveraging blockchain technology, and so all information such as the history of the vehicle from the manufacturer to distributor, and retailer, with every service update will be recorded. This will instill trust in buyers and mitigate the threat of counterfeit goods.

Similarly, Herinckx et al.[66] explained how blockchain technology can prove the authenticity of luxury goods and build trust by interviewing various experts/CEOs of various brands and summarizes the significance of blockchain technology in the secondhand market. Nigam et al. [67] tried mitigating information asymmetry in secondhand marketplaces for automobiles by recording all necessary information in the blockchain. While there is a detailed analysis of the significance of blockchain technology in the secondhand marketplace, both [66], [67] do not establish any system model.

The system models [68], [69] tried to ensure the quality of the goods sold and mitigate the fraudulent sellers using smart contracts primarily. In [68], Shen et al. examined the sale of secondhand products through an online platform using smart contracts. In this model, the platform verifies the quality and decides the price of the products based on their quality. Still, there is no means to know if the platform is fair in validating products or no transparency in the validating process. Also, when a product's quality is not up to the quality criteria set by the platform, the platform returns the product, in which case the sellers need to find other means to sell their products, which might cause difficulties for the platform to maintain loyal customers (sellers using the platform). Medury et al.[69] discussed the impact of blockchain technology in determining and mitigating fraudulent sellers and misleading item listings using smart contracts, micro-escrows, and smart-lock secured Point-of-Exchange (PoE). Since the decentralized marketplace is meant to be community-owned and operated, the significant details like information about the product to be listed, and the locations of PoE are determined by the community, which might cause some serious implications if not chosen appropriately.

In most of the aforementioned system models, there is a notable absence of a comprehensive system involving the design and deployment of smart contracts for trading tangible assets, as well as a detailed analysis of conflicts and resolutions between various players within the model. Additionally, the process of validating secondhand products is often glossed over with no discussion of transparency, which may not effectively address consumer uncertainties. In contrast, our model meticulously outlines the explicit design of smart contracts related to the auction of tangible assets and conducts a thorough analysis of the behaviors exhibited by various participants within the platform. This includes the design of incentive mechanisms, penalty structures, and conflict resolution schemes for secondhand product validation. Moreover, we propose a random selection process for appointing validators for product validation based on their qualifications, thereby instilling trust within the consumer market. Furthermore, our model empowers platform participants by affording them decision-making authority. For instance, validators are entrusted with assessing product quality, with their findings recorded on the blockchain for future reference. To incentivize honest validation, we have devised an incentive mechanism that incorporates both rewards and penalties. Additionally, our platform offers flexibility to sellers by allowing the listing of all products, regardless of their quality. This approach recognizes that some buyers may prioritize cost over quality, and thus sellers have the autonomy to determine the prices of their products.

3. SYSTEM MODEL

In recent years, the popularity of online marketplaces has surged, driven by their convenience, time-saving features, and wide product selection. However, this growth has led to over-reliance on a few entities, resulting in centralization. Additionally, trusting multiple third-party intermediaries has led to significant instances of non-payment and non-delivery crimes. Sellers may also provide misleading descriptions of the product's condition, leading to misunderstandings or dissatisfaction among buyers. To address these challenges and enhance credibility, trust, transparency, and accountability, we have developed an innovative blockchain-based enterprise model for facilitating the trading of various assets. Furthermore, validators within our system will guarantee the quality of secondhand products, and by tokenizing intangible assets, such as health data, our platform will improve the efficiency of ownership transfer, royalty amount distribution, and verification of ownership.

3.1 Blockchain Network

In our model, we consider a permissioned blockchain network as it allows multiple parties (after verification of identity) to securely participate without compromising system performance and efficiency. Such a network also enhances scalability and adds an extra layer of security by allowing users to perform only authorized actions. Our system focuses on two types of assets: tangible and intangible. Tangible assets include products like watches, electronics, accessories, and clothing, both new and secondhand. Intangible assets encompass patents, novel product designs, sensitive health data, and similar items. The majority of network nodes are appointed by the platform, with the remaining nodes comprising loyal participants selected from validators, sellers, and buyers. This approach fosters customer engagement and incentivizes loyalty. To mitigate communication complexity and prevent collusion among nodes, thus safeguarding the network from corruption, we introduce a novel consensus protocol called Modified-PBFT (M-PBFT), which will be detailed in Section 4. In this protocol, one node acts as the leader for a given round, while others function as followers.

3.2 System Overview

In our system, the focus is on enhancing the product cycle of both tangible and intangible assets while bolstering the efficiency of the network. The main users in our system can be classified as sellers, validators, and buyers. In the product cycle, when sellers wish to sell intangible assets, the process begins with token creation, followed by trading and ownership transfer via auction or direct communication with the buyer. Conversely, for tangible assets, sellers initiate the trading phase directly without any tokenization process. During the trading phase, smart contracts automate various processes such as purchase, delivery, and payments for both types of assets. To optimize system efficiency, we introduce a novel transaction prioritization method and an efficient transaction broadcasting technique utilizing our weighted L-H algorithm.

Figure 3.1 illustrates the overview of the innovative blockchain-based assets enterprise model, which aims to improve the product cycle and system efficiency. Our system framework comprises two significant steps, with the first phase emphasizing the product cycle and its enhancement, while the last step prioritizes system efficiency.

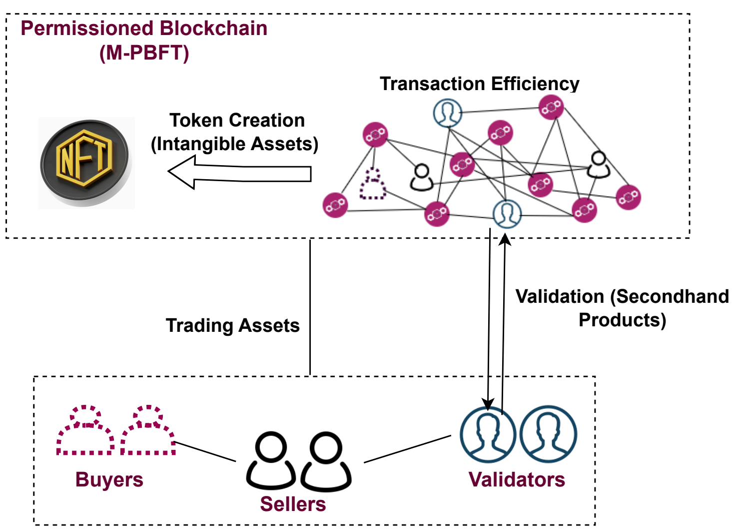


Figure 3.1. Overview of proposed Blockchain-based Assets Enterprise system.

The two main phases are explained as follows:

1. **Trading Assets**: This step focuses on trading both tangible and intangible assets. However, before the actual trading takes place, a preparatory phase is initiated to ensure that the assets are adequately prepared for the transaction.

   (a) For intangible assets, this preparatory phase includes token creation leveraging NFTs to streamline the transfer of ownership of intellectual property and sensitive data, royalties involved, and license rights, making the process more efficient and straightforward. Additionally, NFTs enhance trust and transparency in transactions. Therefore, sellers looking to sell critical data/novel designs will first generate NFTs before selling them. When it comes to tangible assets, secondhand products are validated by employing validators when requested before selling those products to gain trust among buyers.

   (b) Later, the actual trading of a product can be materialized by an auction mechanism, where the highest bidder wins the ownership. Otherwise, a seller can also decide to sell their asset to a buyer directly without the auction process after the buyer sends a private transaction exhibiting their likeness to buy the specific product. Once the buyer buys the product, the ownership of the product is also transferred. Figure 3.2 illustrates the flowchart of the product life cycle of both tangible and intangible assets.

2. **Transaction Efficiency** To improve system efficiency, we incorporate transaction prioritization and determine nodes to efficiently broadcast transactions.

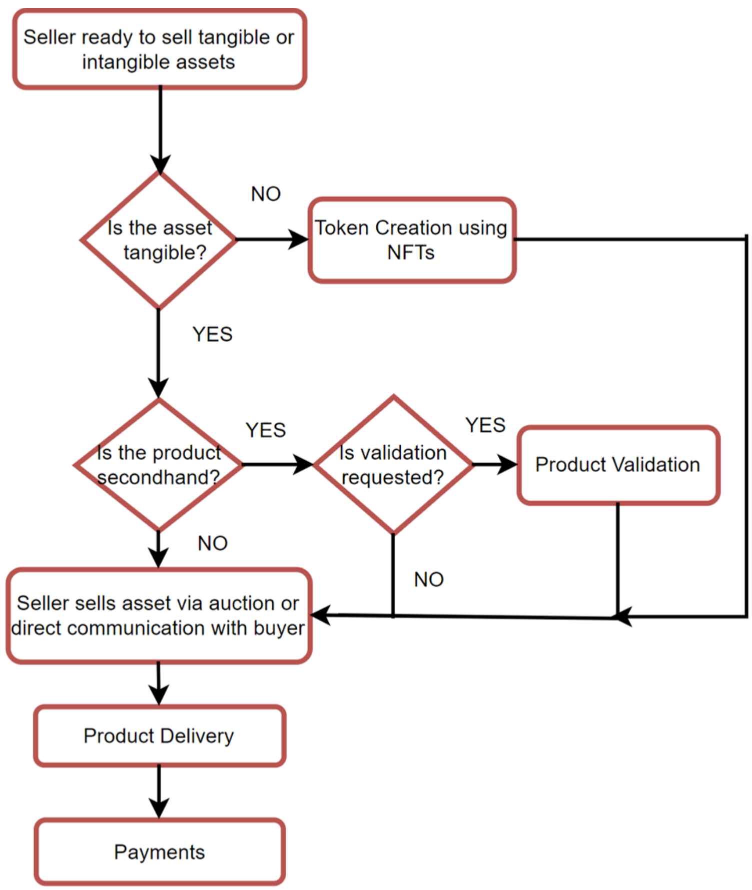


Figure 3.2. Flowchart indicating the product life cycle of tangible and intangible assets.

4. CONSENSUS FOR THE PROPOSED E-COMMERCE MODEL

In this section, we will explain in detail the consensus protocol of our proposed permissioned blockchain system for the e-commerce model. First, we describe the nodes that can participate in the consensus which includes the platform-appointed nodes and loyal nodes from the pool of users/players in the platform. The loyal players are determined by using the extreme gradient-boosted trees algorithm, i.e., XGBoost [70] based on their loyalty ranking. The XGBoost algorithm will be described in detail in Section 4.1. Later, the design of a novel consensus mechanism will be discussed elaborately in Section 4.2.

4.1 XGBoost-based Loyalty Ranking

In our consensus protocol, to incentivize customer engagement and to create a more decentralized network, loyal players are determined to be a part of blockchain nodes participating in the consensus. To that aim, we deploy the XGBoost algorithm to determine the loyal players of our system. XGBoost is a popular and efficient open-source implementation of the gradient-boosted trees algorithm which uses gradient descent to minimize the loss of an objective function against a dataset. Utilizing XGBoost algorithms for selecting loyal nodes in the platform offers the advantage of scalability and computational speed [70]. Furthermore, integrating such machine learning-based algorithms is significant since this algorithm can improve performance through dynamic adaptation, tailored optimization for specific objectives, and adept handling of complex decision-making processes, which often entail numerous criteria. To determine loyal players, the XGBoost-based Loyalty Ranking algorithm uses the following parameters:

1. Time of players' association (denoted by T),
2. Frequency of the players interacting with the platform (F), and
3. Rating given by the sellers to validators (R).

To be specific, these parameters are common for all three roles, i.e., sellers, buyers, and validators. Here, T represents the time of the players' association with the platform. Hence, a player who has associated more time with the platform (in any role capacity) will have a higher T value. F indicates the frequency of players' interaction with the platform. For example, a player who used the platform multiple times to buy, sell, or validate during their time of association will be given a higher F value. Specifically, F can be calculated by

$$F = \frac{N}{T}. \tag{4.1}$$

Here, N represents the number of times a player/user utilizes the platform for buying, selling, or validating, and T represents the player's association time with the platform. R represents the rating for the sellers by the consumers. For example, the R value for a seller becomes the maximum if the buyers are satisfied with the products' qualities, service, delivery, etc.

The general pseudo-code of our XGBoost-based Loyalty Ranking Algorithm, i.e., Algorithm 1 for a single instance is given below.

Algorithm 1 Loyalty Ranking Algorithm

Input: Data samples containing T, F, and R values
Output: Loyal players
1: Create a model for classification using the XGBoost Classifier library with players' information which consists of T, F, and R values as inputs
2: **while** accuracy $<$ Expected accuracy value **do**
3: Improve the model's performance by tuning/updating the default parameter values based on the users' desired values
4: **end while**
5: Loyal and disloyal players are predicted

In Algorithm 1, initially, we prepare our dataset containing T, F, and R values of all the players in our platform. We then proceed to create an XGBoost model for classification and split the data as train and test sets. The model will be trained to learn the relationship between T, F, and R values and ranking output (determining loyal and disloyal players). If the accuracy of this learning process is not up to the expectation, then certain default parameters such as learning rate, max_depth, eval_matric, and early_stopping_rounds are updated to improve the accuracy (Lines 1-4). When the accuracy reaches the expected mark (value may change based on the application), the model is made to predict loyal and disloyal players. Also, since this is a binary classification problem, the outputs will be either 0 indicating disloyal players, or 1 indicating the loyal ones. (Line 5).

4.2 M-PBFT

To make the entire consensus more efficient, protect the network from corruption caused by nodes colluding with each other to disrupt the consensus, and avoid communication overhead, we design our own novel consensus protocol, named Modified-PBFT (M-PBFT), which follows the basic Practical Byzantine Fault Tolerance (PBFT) protocol [71] with some modifications to the phases involved in the consensus, communication framework, and leader election process. The consensus is achieved to decide the validity of a block in the network to resist the tampering of blocks from malicious nodes.

Our proposed consensus protocol relies on the concept of views, where each view is identified by a sequentially increasing integer number. A view is a period of time that a given node can act as the leader of the PBFT network. In each view, a unique dedicated leader will be elected from the participants of the blockchain system using m-TRNG (modified true random number generator). We introduce random leader election to ensure that the nodes have a lower possibility of colluding than in a round-robin way since the sequence in which the nodes become the leader is known beforehand in the round-robin process. Once the leader is elected, the remaining members will act as followers. The leader will then lead the consensus process until the view has changed. The current leader utilizes m-TRNG based on the entropy of the network to complete the leader election process by generating true random numbers that cannot be predicted easily. This entropy will depend on the unconfirmed transaction pool in the network. Since the m-TRNG is based on the entropy of the blockchain network, the increase in incoming network traffic will raise the level of randomness, thereby causing the output to be less predictable. Here, the m-TRNG function will output two true random numbers to indicate the next leader node selected and the number of blocks that the selected leader can record.

In normal PBFT, when the leader sends a transaction proposal, the followers will send accept/reject messages regarding the proposal. Once the leader obtains $2F + 1$ messages (where F is the number of byzantine nodes) and the majority accepts the proposal, the leader will send a commit message to the followers regarding the accepted status of the proposal. When there are $2F + 1$ COMMIT messages from the followers, the proposal is added to the block. Hence, to successfully record a proposal in the block, there are two rounds of message exchanges: The first round is to ensure that there are enough follower nodes and the second round is to confirm/reject any request. However, in hybrid PBFT, there is only a single vote from the followers as we assume that there will always be a minimum quorum since most nodes are appointed by the platform. During the COMMIT phase (during which commit messages are gathered), the leader broadcasts the message after adding a new transaction to the block instead of holding a second voting process as in the PBFT [71]. This adjustment assists in reaching the consensus faster and reduces communication latency and time.

The phases and messages involved in our hybrid PBFT consensus protocol for any round are given as follows:

1. *NEW ROUND*: The leader prepares a new block proposal. The followers wait for the *PRE-PREPARE* message from the leader.

2. *PRE-PREPARE*: The leader produces a proposal containing transactions, forwards it to all followers, and changes its phase to *PRE-PREPARE* phase.

3. *PREPARE*: Upon receiving the proposal, the followers verify it, and if the block proposal is valid, the followers will broadcast a *PREPARE* message to the leader.

4. *COMMIT*: The leader broadcasts a *COMMIT* message after adding the valid block to the blockchain.

5. *VIEW CHANGE*: A follower suspecting the leader to be faulty sends a *VIEW CHANGE* message.

Figure 4.1 demonstrates various phases involved in a specific view in a normal condition. For a given view, the leader collects transactions from the transaction pool, creates a block proposal, and broadcasts it to the network. At this point, the phase of the leader changes to *PRE-PREPARE* phase. The followers receive this message of phase change and enter the *PRE-PREPARE* phase. The followers then verify the proposal and broadcast their *PREPARE* message to the leader. If there are F Byzantine nodes in the network, then the leader should obtain $2F+1$ *PREPARE* messages, for the block to be added to the network, after which the leader broadcasts a message about the successful insertion of the new block.

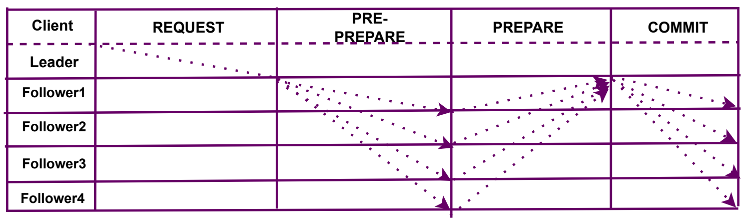


Figure 4.1. Illustration of consensus reaching process in our M-PBFT in a normal condition.

Each follower starts a timer after *PREPARE* message which is turned off after the block is added to the network. If the block is not added within a given time, then it is considered that the leader is inactive, so any follower can propose to change the leader. The proposal is accepted when that follower receives $2F+1$ acceptance messages, and a new leader is elected in a round-robin way. Similarly, if a follower suspects the leader to be faulty, then the follower can initiate a proposal to change the leader. This will ensure the leader nodes perform truthfully in the network. Also, since most of the blockchain nodes are appointed by the platform, the probability of any node acting maliciously is low. Figure 4.2 demonstrates various phases when the leader becomes inactive, thereby causing a view change.

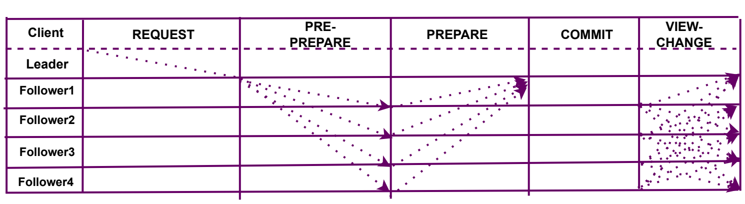


Figure 4.2. Illustration of consensus reaching process in our M-PBFT when the leader becomes inactive.

Also, in our consensus protocol, under normal conditions, the followers send their messages directly to the leader instead of broadcasting the messages to other followers in the P2P network [72]. This modification helps in reducing communication overhead in the blockchain network. Also, if the leader suddenly becomes inactive, the follower nodes can immediately start a view-change to not disrupt the consensus. In certain cases, like proposal initiation, we follow the mesh communication, i.e., original PBFT, to ensure that all nodes in the network are aware of any changes in the network.

The M-PBFT consensus for a specific view in normal conditions is shown in Algorithm 2. In the consensus, o represents the operation requested, t represents the timestamp, c represents the clients own identification, m represents the transaction request, i represents the follower nodes, p represents the leader node, v represents the number of the current domain, n is the sequence number of the request, d is the requested hash value, f is the number of byzantine nodes, σ represents the signature, σ_p is the signature of the leader, and σ_i is the signature of the follower nodes.

Algorithm 2 M-PBFT Consensus

Input: c, i, n, d, f, σ, r, p, and v
Output: Transaction recorded onto a block

```
while <REQUEST, o,t,c > σc = TRUE do
    Broadcast<<PRE-PREPARE, v, n, d > σp, m >
    if (PRE-PREPARE = TRUE) then
        Send<PREPARE, v, n, d, i > σi to p
        Send<PREPARE, v, n, d, n > σn
    end if
    if (PREPARE = TRUE) then
        Broadcast(COMMIT, v, n, p)σp
        Broadcast(COMMIT, v, n, n)σn
    end if
    Transaction is recorded
end while
```

5. STREAMLINING ASSET TRADING BY SMART CONTRACT-POWERED AUCTION

In this section, we focus on outlining the fundamental steps necessary to ensure that the products, including both tangible and intangible assets, are fully prepared for trading. Also, we delve into the auction mechanism design that plays a pivotal role in facilitating the exchange of assets within our marketplace. By leveraging blockchain technology, we aim to empower individuals and businesses, enabling them to engage in transactions with confidence within this marketplace.

5.1 NFT Tokenization and Intangible Asset Trading

Here, we discuss minting NFTs for trading intangible assets like sensitive data and intellectual property. We also explain the encryption of data incorporated into NFT to ensure security and privacy and explore auction mechanisms in detail. Additionally, we efficiently determine the base price for NFTs to facilitate trading for sellers. To achieve finer access control, we introduce a novel approach called the Popularity-based Adaptive NFT Management Scheme (PANMS), which is grounded in Reinforcement Learning (RL) [73]. The base price of these NFTs can be then incorporated during the auction of the NFTs. We propose an adaptive NFT management scheme, PANMS to ensure agile and efficient NFT management and improve user experience and asset utilization throughout the process.

5.1.1 NFT Minting and Encryption

Since intangible assets are involved with the transfer of ownership rights, royalty rights, etc, we leverage NFTs to ensure easier transfer of those rights along with creating transparency and trust throughout the entire process. NFTs are created using the ERC-721 standard and can be mapped to intangible assets. The royalty, ownership, and license rights can be updated using smart contracts. Once NFTs are minted, the information about the token is updated as an image, video, or a link that points to the location where the information is stored. Since the information is critical, the information is encrypted before storing the information using the AES encryption technique since AES is better than other techniques when brute-force attacks are applied. The AES encryption has the following basic elements: a key e (key can be represented in bits or words), the length of the key which is denoted as M_{e}, the total number of rounds which is given as M_r, and N_b, the number of columns. The three main steps of AES encryption are as follows:

- e = *KeyGen()*: Key (e) of required number of bits is generated.
- $x = \mathit{KeyExpansion}(\mathrm{e}, w)$: In *KeyExpansion* function, $N_b(M_r + 1)$ words are generated from initial key e and stored in x. For every round, a specific number of new words from a total of $N_b(M_r + 1)$ words are used in *AdRoundKey* sub-program.
- $z_n = \mathit{Cipher}(z, x)$: This function includes *SubBytes()*, *ShiftRows()*, *MixColumns()*, and *AdRoundKey()* sub-programs to encrypt data. This encrypted data can then be stored in IPFS and shared to customers. Here, z represents plaintext, i.e., the data that needs to be encrypted in words. The encrypted data or cipher text is stored in z_n.

In our system, the encryption keys are updated and changed frequently to prevent any malicious actor from obtaining any type of critical information.

Once the data is encrypted, it cannot be accessed by others. When the ownership rights are transferred; i.e., when a specific NFT is sold to a buyer, then the decryption key is sent to the buyer automatically via a smart contract. This will ensure that the buyer can now access the data. To maintain the efficiency of the system, inactive NFTs that were not bought by the buyers and left inactive for certain cycles will be burnt. Also, during this stage, the system checks the license rights of the NFTs that were already transferred to the consumers, and the rights will be revoked if they exceed the time limit provided in that NFT. Later, we also incorporate the Stackelberg game mechanism to assist the sellers in determining the prices of their NFTs as sellers will gain more control over their products as well as a competitive advantage.

Price Determination of NFTs

To determine the optimal strategies of the seller and buyer while trading NFTs, which contain sensitive data, patents, and novel designs efficiently, the Stackelberg game theory is incorporated. The prices are determined efficiently based on the extracted value or monetary value of the NFTs. In this section, we focus on determining prices of NFT containing health data.

For NFTs mapped to health data, the value would be based on the data sensitivity, i.e., the privacy of the data fields present in the NFTs for health data. Hence, the higher the sensitivity of the data stored in an NFT, the higher the price of that NFT. Similarly, an increase in the cost of an NFT has the same effect on the price of an NFT. For example, health data containing information on genetics, mental health, substance use, etc., can be sold for higher profits. Also, if there is an extra cost involved for securing sensitive information in an NFT or for storage, then the seller will fix a higher selling price to improve his/her profit. Hence, to determine the best strategies for both parties, i.e., sellers and buyers involved, we rely on the Stackelberg game theory to design an optimal pricing mechanism.

Here, the interactions between the data sellers and buyers are modeled as a two-stage Stackelberg game, where any data seller acts as the leader and the buyers act as the followers since it is the seller who decides the prices of its NFTs first, and then, based on the prices, the data consumers make decisions on whether to buy the NFTs or not.

Utility Functions

In our single-leader multiple-follower Stackelberg game mechanism, we assume that each seller will own a set of NFTs, which is represented as $\mathcal{J} = \{1, \cdots, \mathrm{j}, \cdots, J\}$. These NFTs are created for seller-generated health data, with each NFT containing specific data fields, such as token ID, name, description, URL, owner ID, age, gender, history of owners, and transfer of license rights. Let C_{j} be the cost of minting and managing NFT j, and it comprises minting cost ($C_{M\mathrm{j}}$), storage cost ($C_{S\mathrm{j}}$), and cost of implementing extra security measures ($C_{L\mathrm{j}}$) to not only protect sensitive data but also for minimization, containment, and recovery if any threat occurs, i.e., $C_{\mathrm{j}} = C_{M\mathrm{j}} + C_{S\mathrm{j}} + C_{L\mathrm{j}}$. Here, we assume that the selling price of an NFT is always greater than the cost. The set of consumers participating in our model is denoted by $\mathcal{K} = \{1, 2, 3, \cdots, k, \cdots, K\}$.

The price of NFT j is denoted as P_{j}, which also represents the strategy of the seller. P_{j} can be defined as $P_{\mathrm{j}} \in [0, P_{\mathrm{j}max}]$, and the optimal strategy of the seller is given by P_{j}^{*}. The buyer k's strategy for buying NFT j (X_{j}^{k}) is defined as the probability of the buyer buying NFT j. The optimal strategy of the buyer is denoted as X_{j}^{k*}.

Then the expected utility function of the seller, denoted as $E[U_S]$, for a particular NFT j is defined as

$$E[U_S(P_{\mathrm{j}}, X_{\mathrm{j}}^{1}, X_{\mathrm{j}}^{2}, \cdots, X_{\mathrm{j}}^{K})] = (P_{\mathrm{j}} - C_{\mathrm{j}}) \sum_{k=1}^{K} X_{\mathrm{j}}^{k}. \tag{5.1}$$

Here, we calculate the expected utility of the seller $E[U_S]$, since we need to determine the predicted utility that the seller can achieve when the buyer's strategy is uncertain. We sum the probability of all buyers on the platform buying NFT j, considering that a seller may sell the same NFT to multiple buyers. The expected utility of a buyer k for an NFT j is represented as $E[U_B^k]$, and is defined as

$$E[U_B^k(P_{\mathrm{j}}, X_{\mathrm{j}}^{k})] = \alpha^k \log(1 + X_{\mathrm{j}}^{k} Q_{\mathrm{j}}) - P_{\mathrm{j}} X_{\mathrm{j}}^{k}. \tag{5.2}$$

In the above equation, we define the utility of the buyer as the perceived utility/value derived from purchasing NFT j minus the cost [74]. Here, $\alpha^k > 0$ represents the unit price (the price of a data field in NFT j), and $\log(1 + X_{\mathrm{j}}^{k} Q_{\mathrm{j}})$ represents the satisfaction value. Since the buyer's satisfaction depends on the data quality, we introduce Q_{j} which represents the completeness of the data. Q_{j} affects the likeliness of a buyer buying the NFT j. The higher the data quality, the higher the demand for that NFT.

Game Analysis

We present the definitions of the Stackelberg equilibrium for our model and the optimal strategies of the players involved in our game design, followed by a detailed analysis of the Stackelberg game.

Definition 5.1.1 (Stackelberg Equilibrium)**.** *For our game model, $P_{\mathrm{j}}^*, X_{\mathrm{j}}^{k*}$ is the Stackelberg equilibrium if $E[U_B^k(P_{\mathrm{j}}^*, X_{\mathrm{j}}^{k*})] \geq E[U_B^k(P_{\mathrm{j}}, X_{\mathrm{j}}^{k*})]$ and $E[U_S(P_{\mathrm{j}}^*, X_{\mathrm{j}}^{k*})] \geq E[U_S(P_{\mathrm{j}}^*, X_{\mathrm{j}}^{k})]$.*

Definition 5.1.2 (Optimal Strategy)**.** *The objective of a Stackelberg game is to determine the optimal strategies of the players to maximize their utilities. In our model, we denote the optimal strategies of a seller and buyer k by P_{j}^* and X_{j}^{k*}, respectively, where*

$$X_{\mathrm{j}}^{k*} = argmax(E[U_B^k(P_{\mathrm{j}}, X_{\mathrm{j}}^k)]), \tag{5.3}$$

$$P_{\mathrm{j}}^* = argmax(E[U_S(P_{\mathrm{j}}, X_{\mathrm{j}}^1, X_{\mathrm{j}}^2, \cdots, X_{\mathrm{j}}^K)]). \tag{5.4}$$

Here, we follow backward induction to analyze this Stackelberg game. Hence, the best response function of follower k is first calculated. The best response is termed as the buyer k's likeliness to buy an NFT j. For a given price P_{j}, buyer k will determine the optimal strategy X_{j}^{k*} to maximize its own utility.

Theorem 5.1.1. *For a buyer k participating in our system, the optimal strategy, i.e., the likeliness to buy an NFT* j*, is calculated as*

$$X_{\mathrm{j}}^{k*} = \frac{\alpha^k Q_{\mathrm{j}} - P_{\mathrm{j}}}{P_{\mathrm{j}} Q_{\mathrm{j}}}. \tag{5.5}$$

Proof. In order to analyze the equilibrium strategy, we start by obtaining the first-order derivative of $E[U_B^k]$ with respect to X_{j}^k which is given as

$$\frac{\partial E[U_B^k]}{\partial X_{\mathrm{j}}^k} = \frac{\alpha^k Q_{\mathrm{j}}}{1 + X_{\mathrm{j}}^k Q_{\mathrm{j}}} - P_{\mathrm{j}}. \tag{5.6}$$

From 5.6, we obtain the second-order derivative to determine the concavity and maximum/minimum point, which is given by

$$\frac{\partial^2 E[U_B^k]}{\partial (X_{\mathrm{j}}^k)^2} = -\frac{\alpha^k Q_{\mathrm{j}}^2}{(1 + X_{\mathrm{j}}^k Q_{\mathrm{j}})^2} < 0. \tag{5.7}$$

Therefore, the utility of the buyer is strictly a concave function, and there exists an optimal X_{j}^k that maximizes $E[U_B^k]$.

Setting $\frac{\partial E[U_B^k]}{\partial X_j^k} = 0$, we obtain

$$X_j^{k*} = \frac{\alpha^k Q_j - P_j}{P_j Q_j}. \tag{5.8}$$

□

Theorem 5.1.2. *Similarly, for each seller participating in our system with* $\alpha = \sum_{k\in\mathcal{K}} \alpha^k$, *the optimal strategy for an NFT* j *is derived as*

$$P_j^* = \sqrt{\alpha C_j Q_j}. \tag{5.9}$$

Proof. To analyze the equilibrium strategy, we substitute (5.9) in (5.1) to obtain

$$E[U_S(P_j, X_j^1, X_j^2, ...X_j^K)] = (P_j - C_j)K[\frac{\alpha}{P_j} - \frac{1}{Q_j}]. \tag{5.10}$$

Then the first-order derivative of $E[U_S]$ with respect to P_j is obtained as follows:

$$\frac{\partial E[U_S]}{\partial P_j} = -\frac{K}{Q_j} + \frac{C_j K \alpha}{P_j^2}. \tag{5.11}$$

From (5.11), we obtain the second-order derivative given by

$$\frac{\partial^2 E[U_S]}{\partial P_j^2} = -\frac{C_j K \alpha}{P_j^3} < 0. \tag{5.12}$$

Therefore, the utility of the seller is also strictly a concave function; hence, there exists an optimal P_j to maximize $E[U_S]$.

Setting $\frac{\partial E[U_S]}{\partial P_j} = 0$, we obtain

$$P_j^* = \sqrt{\alpha C_j Q_j}. \tag{5.13}$$

□

Substituting P_j^* in (5.8), we get $X_\mathrm{j}^{k*} = \frac{\alpha^k Q_\mathrm{j} - (C_\mathrm{j} Q_\mathrm{j} \alpha)^{1/2}}{(C_\mathrm{j}\alpha)^{1/2} Q_\mathrm{j}^{3/2}}$. Therefore, the optimal strategies of the buyer k and the seller in our system model for NFT j are given by

$$(X_\mathrm{j}^{k*}, P_\mathrm{j}^*) = [\frac{\alpha^k Q_\mathrm{j} - (C_\mathrm{j} Q_\mathrm{j} \alpha)^{1/2}}{(C_\mathrm{j}\alpha)^{1/2} Q_\mathrm{j}^{3/2}}, (C_\mathrm{j}\alpha Q_\mathrm{j})^{1/2}]. \tag{5.14}$$

Thus, using backward induction, we obtain X_j^{k*} and P_j^*, and both values are unique and optimal strategies for the buyer k and the seller, respectively. Hence, by determining an appropriate price in the data market, the seller can maximize his/her profits. Similarly, buyer k will maximize his/her profit based on his/her inclination towards buying an NFT of good quality with an appropriate price. In other words, X_j^{k*} and P_j^* are the Stackelberg game equilibrium points.

5.1.2 Auction for Intangible Assets

Once the intangible assets are prepared for trading, the auction commences. Here, the different phases of the auction mechanism design are described in detail along with smart contract designs, reputation score design, and a decision-making algorithm to assist the companies in deciding on whether to invest in a specific NFT or not.

First, we focus on determining the top n companies to broadcast the NFT details followed by the main auction phase, i.e., the decision-making and NFT ownership transfer. Here, the value of n would vary depending on the seller's decision. The top $'n'$ credible potential buyers are determined for the seller to advertise their product details and percentage of ownership rights transfer information. This approach enhances security, particularly when dealing with intangible assets such as sensitive data or intellectual property, as it enables the seller to maintain control over who receives access to such information. Hence, determining the top $'n'$ potential buyers based on their reputation becomes critical. The seller selling an intangible asset (which is mapped to an NFT) would then share the hash of the IPFS location (that contains less sensitive information) of the NFT to the top $'n'$ potential buyers. The reputation score is evaluated using a smart contract and is based on feedback from other sellers who worked with these buyers on the platform.

In general, the reputation score is recorded in the blockchain network, by the smart contract, every time there are updates. The calculation of the reputation score of a buyer b is given as:

$$S_b(t) = S_b(1 - w)(t - 1) + f_b w(t) \tag{5.15}$$

Here, w is a dynamic variable that could represent an aggregate weight at a time t, while $f_t b$ represents the feedback of the buyer b at time t. $S_b t$ refers to the reputation score of a buyer b at time t. The feedback from other sellers depends on the recorded information available in the network, which can also prove the integrity of the buyers. Such information includes if the buyers followed the proper auction process and within the said time (time constraint during bid commit and reveal) and if the payment/royalty was properly paid in time. For example, when a buyer defaults on the royalty amount that is supposed to be paid or acts maliciously during the auction process, it shows that the buyer is not reliable. Hence, these factors are used in the feedback. If a buyer has followed these terms then the buyer will be assigned the maximum score $'1'$ by the smart contract. The reputation scores of all buyers are updated every time an auction is completed. Also, any seller can query and obtain the latest reputation score of any buyer via API. Once the top n potential buyers are chosen by the seller, a part of sensitive information is made available to those companies.

When a few potential buyers decide to participate in the auction process after viewing the information incorporated into an NFT (intangible assets), the auction process is initiated. The potential buyers sign a Non-Disclosure Agreement (NDA), and this phase is implemented using a smart contract, where a smart contract containing the terms of the NDA is sent [24] to all companies interested. Once the NDA is signed, the auction process is triggered.

Decision-making & NFT Ownership Transfer

When the potential buyers obtain a part of the sensitive information of the NFT, the potential buyers decide whether to participate in the auction process of the NFT or not. Here, we leverage an Iterative Dichotomiser (ID3)-based decision tree [23] to assist the potential buyers in determining their participation in the auction. A decision tree is a structure that contains nodes and edges and is built from a dataset. Each node is used to make a decision (known as the decision node) to represent an outcome (known as the leaf node). ID3 uses a top-down greedy approach to build a decision tree, is inexpensive to construct, and is extremely fast at classifying unknown records. The algorithm assists in the decision-making process based on different scores like practicability score, uniqueness score, future value score, and liquidity premium score of an NFT when the NFT comprises intellectual property, particularly novel designs. In contrast, for NFTs containing sensitive data, the algorithm primarily relies on uniqueness and liquidity premium scores. The uniqueness score of an NFT containing health data can be determined by its sensitivity or privacy. The higher the data sensitivity, the higher the data uniqueness. By incorporating these scores for various intangible assets within NFTs, the algorithm helps determine the viability of investment for potential buyers. A detailed explanation of the aforementioned characteristics of the NFT containing novel product design is provided here.

Definition 5.1.3. *The practicability score denoted as E_p depends on the specific technology used in the NFT-based design and is given as $E_p = \beta N$. Here, N is the number of applications that the bidder (company) can use the technology present in that NFT.*

Definition 5.1.4. *The uniqueness score denoted as E_u depends on the rarity of a specific feature used in the NFT and is given as $E_u = \frac{I}{R}$. Here, I/R is the (Total number of NFT items/Number of NFT items with that rare feature).*

Definition 5.1.5. *The future value score denoted as E_f depends on whether a technique or design is sought after in the future (based on history/data collection).*

Definition 5.1.6. *The liquidity premium score denoted as E_l focuses on NFT technology in general and is calculated based on the yield curve or realized return.*

In Algorithm 3, we discuss how the ID3-based decision tree algorithm is used by potential buyers/bidders to decide on whether to invest in a specific NFT containing novel product design or not using the aforementioned scores. In the ID3-based decision-making algorithm, the attributes are the aforementioned scores related to NFT, i.e., practicability score, uniqueness score, future value score, and liquidity premium score. Since it is a binary

Algorithm 3 ID3-based Decision-making Algorithm

Input: Decision tree (D_t) along with specific attributes, attribute list A_l=[E_p,E_u,E_f,E_l], iteration=1, and $best_{attr}$ for 1st iteration=E_p

Output: Decision related to investing in a specific NFT

1: **if** D_t is LeafNode **then**
2: Return class_label of LeafNode
3: **else**
4: Traverse D_t based on A_l starting from E_p
5: `marker:`
6: **for** iteration $\leq$ number of attributes **do**
7: Calculate $best_{attr}$ for current iteration
8: **if** $best_{attr} \in D_t$.subtree **then**
9: **go to** `marker` to traverse the specific subtree for other attributes
10: **else**
11: Return D_t.majority_class // selecting the class label that appears most frequently
12: **go to** `output`
13: **end if**
14: **end for**
15: **go to** `output`
16: **end if**
17: `output:`Decision based on class_label

classification problem, class labels are 0 and $'1'$. Based on whether the scores are above a certain threshold (decided by the potential buyers) or not, a decision is made at the end.

In Algorithm 3, if the decision tree (D_t) has just a leaf node, then the decision is made based on the class label related to that leaf node (Lines 1-3). If not, then based on a specific attribute, the D_t is traversed. If the attribute is present in a sub-tree, then it becomes a recursive problem and we start again from line 5 (Lines 5-9). If the required attribute does not match any attribute present in the D_t, then the decision is made on the majority class label of D_t. The majority class label is the class label associated with the majority of training data and is used when the attributes specified are not present in the decision tree (Lines 10-12). After all attributes in the tree are traversed, the final decision is based on the final class label (Lines 15-17).

When a few potential buyers decide to participate in the auction process, the buyers sign an NDA and once the NDA is signed, the auction process is initiated. The first-price sealed-bid auction is implemented and parameterized by a few values: the good being sold, the reserve price (R_p), and the bidding period (B_p). Here, R_p is the lowest acceptable bid, and B_p is defined as the period during which bidders may submit bids. Initially, in the auction, a bidder submits a hash of their bid. Next, during the revealing period, the bidders must present the actual bid and nonce related to the hash of the bid they submitted initially. The actual bid amount is then verified against the hash they previously provided using the nonce to ensure truthful revelation of the bids by the bidders.

When bids are revealed, the contract will keep track of the highest bidder and the highest bidders' bid amount. The current winning bid will also be reflected as the balance so that it will be available to the seller after the auction. Algorithm 4 discusses the commit and revealing-bid phases of the smart contract-based auction mechanism.

Algorithm 4 Smart Contract-based Auction

Input: end of bidding time (t_{e}), R_p, end of revealing time (t_r), and bid amount of bidder c (say B_c)
Output: Determination of the highest bidder (c_H) and highest bid (H)

```
initialize H = 0
if now < t_e then
    for c ∈ C do
        hash ← hashedBidOf(c)
    end for
end if
if (now ≥ t_e) & (now < t_r) then
    for c ∈ C do
        check(keccak256(B_c,nonce))==hashedBidOf(c)
        check if B_c ≥ R_p & if B_c ≤ balanceOf[c]
        if B_c > H then
            c_H ← c
            H ← B_c
        end if
    end for
end if
balance(c_H) − = H
balanceOf(seller) + = H
```

The smart contract-based auction Algorithm 4 involves two phases: bidding and revealing phase. Lines 1-6 focus on the bidding phase and during this phase, the bidders/companies commit/submit the hashes of their bids before the bidding time ends (current time less than t_c), and the bid is recorded by the smart contract (Lines 1-6). During the revealing phase (lines 7-16), the bidders submit their nonce related to their hashes of the bids and their actual bid amounts (B_c). The smart contract then verifies the bid amounts submitted by the bidders using their nonces submitted. If it is true and if the bid amounts are above the reserve price (R_p) set initially by the seller and the bidders' account balances contain their bid amounts, then the smart contract determines the highest bid from the bids submitted, and the winner of the auction (Lines 7-16). After the winner is determined, the bid amount submitted by the winner is transferred to the seller's wallet (Lines 17-18).

The final phase of the auction mechanism is NFT ownership transfer. Once the winner of the auction is determined, the ownership of the NFT is transferred, and the decryption key to the complete information of the NFT is given to the highest bidder using private transactions for security purposes. In this way, the buyer winning the auction can view the complete information incorporated in the NFT that the buyer won.

Bayesian Nash Equilibrium and Analysis

Here, a bidder does not know what the other bidders are going to bid and the valuations of the other bidders initially. Hence, strategically, a Bayesian game is designed. The first price sealed-bid auction follows the independent private value model, where the valuation of the NFT at auction is different for different bidders. For this design, we assume that this model focuses on multiple sellers selling their NFTs incorporating novel product designs and multiple potential buyers who are willing to bid for those NFTs. The sets of sellers and buyers in the system are denoted by $\mathcal{D} = \{1, \cdots, d, \cdots, D\}$ and $\mathcal{C} = \{1, \cdots, c, \cdots, C\}$, respectively. In the auction mechanism, the buyer is assumed to be the highest bidder. The bidders do not have a dominant strategy since they would be required to trade off between winning the bid and making a profit out of that bid.

One of the parameters involved in the utility of bidder c, (say UC_c) is the bidder's valuation of the NFT, V_c. The valuation is based on four significant characteristics of the NFT: practicability of the NFT, uniqueness of the NFT, liquidity premium (general to NFT technology), and future value held by the NFT and is presented in Definitions 5.1.3-5.1.6 in Section IV-B.

The utility function of a bidder c, denoted as UC_c can be defined as

$$UC_c = V_c - \alpha B_c - T_c. \tag{5.16}$$

Here B_c is the bid amount on the NFT by bidder c, and α denotes the spiteful behavior of the bidder. Due to the competitive nature of the auction, bidders may showcase spiteful behavior to outdo other bidders. Also, T_c represents costs incurred during the bidding process.

Then the utility function of a seller d is denoted as UD_d and can be defined as

$$UD_d = P_d + R_d - O_d - H_d, \tag{5.17}$$

where P_d represents the price of NFT that the seller d is willing to sell and R_d represents the royalty amount that the seller d might be getting from the buyer that obtains the seller's NFT. Also, O_d is the cost of the NFT which might include minting cost, transaction cost, storage cost, etc. H_d is the holding value of the NFT and is a function of price.

Theorem 5.1.3. *In a first-price auction with two risk-neutral bidders whose valuations are IID and drawn from $U(0,1)$, $(\frac{1}{2}V_1, \frac{1}{2}V_2)$ is a symmetric Bayes-Nash equilibrium strategy profile.*

Proof. Assume that bidder C_2 bids $\frac{1}{2}V_2$, and bidder C_1 bids B_1. Bidder C_1 wins when $V_2 < 2B_1$, and gains utility $V_1 - B_1$, but loses when $V_2 > 2B_1$ and then gets utility 0. The expected utility of bidder C_1 is defined as

$$E[UC_1] = \int_0^{2B_1} (V_1 - B_1)\, dV_2. \tag{5.18}$$

From (5.18), we get

$$E[UC_1] = 2V_1B_1 - 2S_1^2. \tag{5.19}$$

Bidder C_1's response to bidder C_2's strategy can be determined by

$$\frac{(2V_1B_1 - 2S_1^2)}{\partial B} = 0. \tag{5.20}$$

From (5.20), we obtain bidder C_1's strategy which is given by

$$B_1 = \frac{1}{2V_1}. \tag{5.21}$$

It can be noted that the bidders C_1 and C_2 strategies can be considered as symmetric. Hence, it is proved that the strategies of the bidders are symmetric Bayes-Nash equilibrium. □

5.1.3 Adaptive NFT Management

From the feedback received during auction, popular data fields in the market for a specific time interval t is obtained. We implement a Popularity-based Adaptive NFT Management Scheme (PANMS) to identify such popular data fields, enabling sellers to obtain greater rewards and enhance finer access control over those specific data fields before trading sensitive data, particularly health data. Through PANMS, we aim to optimize the management of NFTs, facilitating more efficient and secure transactions while prioritizing the most sought-after data fields. The popular datas are determined for every specific cycle for accuracy.

The sellers can use NFT adaptive management to create NFTs for specific information. The NFT management focuses only on critical health data or any assets that can be sold partially. For example, if a seller has an NFT that is mapped to his/her medical information, then the seller can incorporate adaptive NFT management to sell only certain data fields by creating a new NFT that maps to those specific data fields from the entire data. It is also possible to sell same data fields to multiple buyers. However, this is not possible when selling clothing, a computer, or a patent. Normally, the sellers tend to sell the aforementioned assets as a whole.

NFT management on sensitive health data focuses on determining popular data fields using the Reinforcement Learning (RL) algorithm, and NFTs are created for the popular data fields, which are sold by the data owners/sellers during every cycle t to enhance finer access control. Hence, by selling only limited data fields, the sellers can save cost and time, as they only need to mint NFTs for the selected data fields, thus, saving management costs. Also, the data fields are selected based on their popularity, and so, there is a high chance that the selected data fields for that particular cycle are high in demand, resulting in higher profits for the sellers.

In our system, the main challenge is the dynamic nature of the popularity of the data fields. Here, we define choosing popular data fields as actions, and each data field has a popularity factor value following an unknown distribution; that is, when we perform an action, we receive a reward based on the popularity factor sampled from the given distribution. Hence, the system needs to learn the popularity of the data fields and then select a set of popular ones from all data fields. The system will need to explore and exploit various options to maximize the popularity factor, and thus, profit. In our problem, the total privacy index budget (I_M) acts as a constraint, which represents the sensitivity of data. Every data field is associated with its own privacy index value. Since the seller would like to restrict the amount of sensitive data to sell in a particular cycle, the privacy index is considered a constraint. For example, a seller might not be interested in selling the most critical data that the seller owns due to various reasons, in which case the seller can set the privacy index constraint at some lower value. Now, the popular data fields are selected whose total privacy index value is within the constraint provided. This privacy index constraint is updated by the sellers for each cycle. Hence, by considering the level of privacy index constraint, popular data fields can be determined for each cycle t.

To address this challenge of the dynamic nature of popularity, determining popular fields is modeled as a Combinatorial Multi-Armed Bandit (CMAB) problem, where each data field is considered as an arm. Determining the popular data fields is considered equivalent to pulling the best set of arms which in turn contributes to the profit or reward. Here, we have used Upper Confidence Bound (UCB) [75], a widely used solution method for CMAB, to solve the exploration-exploitation dilemma.

Problem Formulation

We assume that $\mathcal{D}_{\mathcal{L}}$ is the set of data fields owned by a particular seller, where $\mathcal{D}_{\mathcal{L}} = \{D_1, D_2, D_3, \cdots, D_l, \cdots, D_L\}$. For each cycle t, a new and optimal data set (with fewer data fields than the original data) is determined based on the popularity factors of the data fields. The optimal set is denoted as $\mathcal{D}_{\mathcal{S}} = \{D_1, D_2, \cdots, D_s, \cdots, D_S\}$. The number of data fields in $\mathcal{D}_{\mathcal{S}}$ is always less than the data fields in $\mathcal{D}_{\mathcal{L}}$. The popularity factor for any data field D_l, in round t is denoted as $F_{D_l}(t)$, where

$$F_{D_l}(t) = \sum_{k \in \mathcal{K}} R_{D_l k}(t). \tag{5.22}$$

Here, $\mathcal{K}$ denotes the consumer set, and $R_{D_l k}$ represents the requests made by the customers for the data field $D_l \in \mathcal{D}_{\mathcal{L}}$. Hence, the popularity factor of any data field is equal to the total number of requests for that data field. We use $f_{D_l}(t)$ to represent the expected popularity factor of data field D_l, which is given by

$$f_{D_l}(t) = E[F_{D_l}(t)]. \tag{5.23}$$

Total reward (R) for all data fields presenting in round t is given by

$$R(t) = \sum_{D_l \in \mathcal{D}_{\mathcal{L}}} U_{D_l} F_{D_l}(t) S_{D_l}(t), \tag{5.24}$$

where U_{D_l} represents the sales margin. Here, we define a binary variable $S_{D_l} \in \{0, 1\}$ to denote whether the data field D_l has been chosen as the popular data field or not. $S_{D_l} = 1$ represents that a specific data field is selected as the popular data field, while $S_{D_l} = 0$ represents that the data field is not selected as the popular one. Also, increasing the popularity factor values of the selected data fields will increase the reward.

Based on the consumers' requests, popular data fields for every t cycle are determined from the seller's database using the Reinforcement Learning (RL) algorithm. The seller can then increase their profit by selling the NFTs containing the determined popular data fields. While choosing the popular data fields, the sellers need to ensure that the total privacy value of the selected data fields does not exceed the given constraint (I_M) set by the seller. Hence, the goal is to maximize the time-averaged total reward over time duration T, considering the total privacy index constraint. Therefore, the problem can be formulated as:

$$max \quad \frac{1}{T}\sum_{t=0}^{T-1} E[R(t)], \tag{5.25}$$

$$s.t. \quad \sum_{D_l \in D_L} I_{D_l}(t) S_{D_l}(t) \leq I_M, \tag{5.26}$$

$$S_{D_l} \in \{0, 1\}, \forall D_l \in \mathcal{D}_{\mathcal{L}}. \tag{5.27}$$

Here, (5.25) represents maximizing the time-averaged reward over time duration T, and the constraint (19) ensures that the privacy index values of the selected data fields will not cross beyond the given I_M constraint, where I_{D_l} represents the privacy index value of a specific data field D_l. (4.37) indicates that the selection variable S_{D_l} is a binary variable.

In this CMAB problem, the total profit earned by all data fields is regarded as the reward. Hence, we target maximizing the average reward while meeting the privacy index constraints of the data fields. To select appropriate data fields in each time slot, we consider the popularity factor, i.e., users' demands for each data field. We indicate the popularity level as f_{D_l} which is defined in (5.23). But in general, the popularity factor value of a data field is always uncertain. Therefore, the popularity level information must be relearned and updated to get the best outcome (reward). Here, exploring new arms and choosing the arms with maximum benefits must be in balance. Based on the learning results, we can then calculate the corresponding profit for maximum benefits.

To solve this CMAB problem, we design a Popularity-based Adaptive NFT Management Scheme (PANMS) to determine the data fields with unknown popularity levels under privacy index constraints to maximize profit.

Algorithm Design

Based on (5.24), the numerator of the objective function is written as

$$\sum_{t=0}^{T-1} E[R(t)] = \sum_{t=0}^{T-1} E[\sum_{D_l \in \mathcal{D}_\mathcal{L}} U_{D_l} F_{D_l}(t) S_{D_l}(t)]. \tag{5.28}$$

Since selection variable S_{D_l} and popularity factor F_{D_l} are independent of each other, and there exists $f_{D_l}(t) = E[F_{D_l}(t)]$ based on (5.23), we can further update the above equation as

$$\sum_{t=0}^{T-1} E[R(t)] = \sum_{t=0}^{T-1} \sum_{D_l \in \mathcal{D}_\mathcal{L}} U_{D_l} f_{D_l}(t) E[S_{D_l}(t)]. \tag{5.29}$$

It can be seen that in order to achieve the optimization goal, the data fields with the highest popularity factors are to be chosen. To that aim, we introduce variable $N_{D_l}(t)$ to represent the number of time slots/rounds in which the popularities of the data fields are learned. Also, we introduce variable $\bar{f}_{D_l}(t)$ to denote the average popularity value of a particular data field learned until time slot t. The average popularity factor value of a specific data field can be obtained according to

$$\bar{f}_{D_l}(t) = \frac{\sum_{\tau=0}^{t-1} F_{D_l}(\tau)}{N_{D_l}(t)}. \tag{5.30}$$

The values of N_{D_l} and $\bar{f}_{D_l}(t)$ need to be updated as follows in the next time slot:

$$N_{D_l}(t) = \begin{cases} N_{D_l}(t-1)+1 & D_l \in \mathcal{D}_\mathcal{S}, \\ N_{D_l}(t-1) & D_l \notin \mathcal{D}_\mathcal{S}. \end{cases} \tag{5.31}$$

$$\bar{f}_{D_l}(t) = \begin{cases} \frac{\bar{f}_{D_l}(t) N_l(t-1) + F_{D_l}(t)}{N_{D_l}(t)} & D_l \in \mathcal{D}_\mathcal{S}, \\ \bar{f}_{D_l}(t-1) & D_l \notin \mathcal{D}_\mathcal{S}. \end{cases} \tag{5.32}$$

The calculation of the estimated value of popularity level from [76] can be given as:

$$\tilde{f}_{D_l}(t) = \bar{f}_{D_l}(t) + K\sqrt{\frac{3log(t)}{2N_{D_l}(t)}}. \tag{5.33}$$

The above equations (5.31)-(5.33) form the profile of the data fields for each round that needs to be learned and updated. Initially, values of $N_{D_l}(t)$ and $\bar{f}_{D_l}(t)$ will be set to 0, and $\tilde{f}_{D_l}(t) = R_{D_l}(t)$, which indicates the presence of the data fields in the current cycle. Then for every round $t > 1$, the popularity factor values $f_{D_l}(t)$ of the data fields are determined based on the consumers' requests for each round, and accordingly, the values of $N_{D_l}(t)$, $\tilde{f}_{D_l}(t)$, and $\bar{f}_{D_l}(t)$ are updated.

Here, we use $\tilde{W}(T)$ to represent the estimated time-average of expected profit of the selected data fields after learning over time duration T, i.e.,

$$\tilde{W}(T) = \frac{1}{T} \sum_{t=0}^{T-1} E[R(t)]. \tag{5.34}$$

Then based on the learned $\tilde{f}_{D_l}(t)$ and (5.29), the above equation can be expressed as:

$$\tilde{W}(T) = \frac{1}{T} \sum_{t=0}^{T-1} \sum_{D_l \in \mathcal{D}_{\mathcal{L}}} U_{D_l} f_{D_l}(t) E[S_{D_l}(t)]. \tag{5.35}$$

Now, we rewrite the problem of determining optimal data set with uncertain data field popularity as:

$$\begin{aligned} max \qquad & \tilde{W}(T), \\ s.t. \qquad & (19), (20). \end{aligned}$$

To solve this problem, we introduce a Popularity-based Adaptive NFT Management Scheme to maximize $\tilde{W}(T)$. In this algorithm, for each time slot t, we determine the popularity level of each data field. The selection variable S_{D_l} is set by executing Algorithm 6, i.e., OptimalSetSelection strategy. We then use the determined optimal data set to obtain the maximum profit. This profit obtained will have an impact on the profile calculation during the next time slot in Algorithm 5.

In Algorithm 5, at the initial phase, when the time slot is 1, there is no presence of any historical data since the learning and updating profiles have not yet started. Hence, we initialize variables $N_{D_l}(t)$ and $\bar{f}_{D_l}(t)$ to 0, and $\tilde{f}_{D_l}(t) = \sum_{k \in \mathcal{K}} R_{D_l k}(t)$ (Lines 1-3). Then the OptimalSetSelection algorithm is called for each time slot to determine the optimal data set to select (Line 4-5). The detailed design of the aforementioned algorithm is shown in Algorithm 6 and will be introduced in the next subsection. Based on the output of Algorithm 6, the profiles of the data fields selected are learned and updated. Subsequently, we can learn and estimate the popularity level of data fields according to (5.33) for the selection of data fields in the next time slot (Lines 6-14).

Algorithm 5 Popularity-based Adaptive NFT Management Scheme (PANMS)

Input: Current time slot (t), data fields set ($\mathcal{D}_{\mathcal{L}}$), optimal data set with the highest popularity ($\mathcal{D}_{\mathcal{S}}$)

Output: Maximising rewards by selecting popular data fields within I_M

1: **for** $D_l \in \mathcal{D}_{\mathcal{L}}$ **do**

2: Initialize $N_{D_l}(0) = 0, \bar{f}_{D_l}(0) = 0,\ \tilde{f}_{D_l}(0) = R_{D_l k}(0),\ R(t) = 0$

3: **end for**

4: **for** $t = 1;\ t \leq t_{max};\ t++$ **do**

5: $\mathcal{D}_{\mathcal{S}} \leftarrow$ OptimalSetSelection$(t, \{\tilde{f}_{D_l}(t)\})$

6: Select and update the profiles of the data fields based on S_{D_l}

7: **for** $D_l \in \mathcal{D}_{\mathcal{L}}$ **do**

8: $N_{D_l}(t) = \begin{cases} N_{D_l}(t-1) + 1 & D_l \in \mathcal{D}_{\mathcal{S}}, \\ N_{D_l}(t-1) & D_l \notin \mathcal{D}_{\mathcal{S}}. \end{cases}$

9: $\bar{f}_{D_l}(t) = \begin{cases} \frac{\bar{f}_{D_l}(t) N_l(t-1) + f_{D_l}(t)}{N_{D_l}(t)} & D_l \in \mathcal{D}_{\mathcal{S}}, \\ \bar{f}_{D_l}(t-1) & D_l \notin \mathcal{D}_{\mathcal{S}}. \end{cases}$

10: **if** $N_{D_l}(t) > 0$ **then**

11: $\tilde{f}_{D_l}(t) = \bar{f}_{D_l}(t) + K\sqrt{\frac{3log(t)}{2N_{D_l}(t)}}$

12: **end if**

13: **end for**

14: Calculate R(t)

15: **end for**

Knapsack Problem

From PANMS mentioned in Algorithm 5, it can be seen that the focus of this algorithm is to determine an optimal data set $\mathcal{D}_{\mathcal{S}}$ represented as $\mathcal{D}_{\mathcal{S}} = \{D_1, D_2, \cdots, D_s, \cdots, D_S\}$. Each data field has a certain popularity factor value, represented by $\{\tilde{f}_{D_1}, \tilde{f}_{D_2}, \cdots, \tilde{f}_{D_s}, \cdots, \tilde{f}_{D_S}\}$, which in turn, affects their ability to generate profit. There is also a total privacy index constraint I_M to consider. Hence, determining an optimal set $\mathcal{D}_\mathcal{S}$ in each time slot to maximize profit R without exceeding privacy index constraint I_M constitutes a knapsack problem.

Since a data field can be either chosen to be in the optimal data set or not, this problem becomes a 0-1 knapsack problem, where 0 represents that a particular data field is not selected in the optimal set, and 1 represents its selection. We perform the 0-1 knapsack dynamic programming algorithm on the data fields to obtain the maximum profit, while making sure that the privacy index values of the individual data fields are within the mentioned constraint. During this process, we obtain and save the process variable $B(l, I)$.
Here, $I \in \{0, \cdots, I_M\}$, and for the number of data fields $l \in \{0, 1, \cdots, L\}$, the value of $B(l, I)$ can be defined as

$$B(l,I) = \begin{cases} 0, \quad l = 0, \\ B(l-1, I), \quad l > 0, \quad I_{D_l} > I, \\ max[B(l-1, I), \tilde{f}_{D_l} + B(l-1, I - I_{D_l})], o.w. \end{cases} \tag{5.36}$$

In general, for each time slot t, we determine the optimal data set with the data fields containing least privacy index values using their respective optimal 0-1 knapsack scheme to obtain the maximum profit. Hence, we first pre-determine the data fields based on their popularity factors to maximize profit, keeping the privacy constraint in mind, and then, based on the remaining total privacy index value, we select data fields with the minimum privacy index values to fill in the gaps. Based on this strategy, we can get an approximately optimal solution for maximizing profit without exceeding the privacy index value constraint given. The detailed design is given in Algorithm 6.

Algorithm 6 OptimalSetSelection($t, \{\tilde{f}_{D_l}(t)\}$)

Input: Current time slot (t), learned popularity levels ($\tilde{f}_{D_l}$)

Output: Optimal data set $\mathcal{D}_{\mathcal{S}}$

Set $\mathcal{D}_{\mathcal{S}} = \emptyset$
Initialize $B(l, I) = 0$ for all $I \in \{0, 1 \cdots, I_M\}$
for $l \leftarrow 1 \quad to \quad L$ **do**
    for $I \leftarrow 1 \quad to \quad I_M$ **do**
        if $I_{D_l} > I$ **then**
            Set $B(l, I) \leftarrow B(l1, I)$
        else
            Set $B(l, I) \leftarrow max\{B(l-1, I - I_{D_l}) + \tilde{f}_{D_l}(t), B(l1, I)\}$
        end if
    end for
    Obtain pre-deployment strategy T_{D_l} based on $B(l, I)$
    while $l \geq 1$ and $I \geq 1$ **do**
        if $II_{D_l} \geq 0$ and $B(l, I) = \tilde{f}_{D_l}(t) + B(l-1, I - I_{D_l})$ **then**
            Set $T_{D_l} \leftarrow 1$, $I \leftarrow I - I_{D_l}$, $l \leftarrow l1$
        else
            if $B(l, I) = B(l1, I)$ **then**
                Set $T_{D_l} \leftarrow 0, l \leftarrow l1$
            end if
        end if
    end while
end for
Selecting the data fields with the smallest I for the optimal data set
for every $D_l \in \mathcal{D}_{\mathcal{L}} \setminus \mathcal{D}_{\mathcal{S}}$ **do**
    if $T_{D_l} = 1$ **then**
        Set $S_{D_l} \leftarrow 1, \mathcal{D}_{\mathcal{S}} = \mathcal{D}_{\mathcal{S}} \cup D_l$
    else
        Set $S_{D_l} \leftarrow 0$
    end if
end for
return $\mathcal{D}_{\mathcal{S}}$

In Algorithm 6, we first initialize the data set $\mathcal{D}_{\mathcal{S}} = \emptyset$ (Line 1). We then proceed to initialize the value table, and then the pre-selection strategy is carried on for each data field present (Lines 3-22). We perform the 0-1 knapsack dynamic programming algorithm using a recursive approach on each data field. During this process, we obtain and save the process variable $B(l, I)$ (Lines 7-11). Based on this variable, we can get the pre-selection strategy of the data fields. This selection variable is denoted by T_{D_l} and based on the data fields selected, we calculate the remaining privacy index value available and update it dynamically (Lines 12-20). Then, we select the data fields with the minimum privacy index value for prioritized deployment (Line 22). Based on the aforementioned strategy, we can then obtain the selection strategy for the data fields S_{D_l} (Lines 23-29).

Based on this S_{D_l} value, profiles of the data fields are learned and updated with new information in Algorithm 6 to meet the end goal of maximizing profit without exceeding the constraint value.

5.2 Ensuring Quality in Tangible Asset Trading

In this section, our emphasis is on product validation, particularly for secondhand goods, where product quality plays a vital role. The secondhand market is susceptible to various challenges, including the sale of counterfeit goods by unscrupulous sellers and potential disputes arising between buyers and sellers regarding pricing or the condition of the items.

5.2.1 Secondhand Products Validation

When a buyer opts for validation, the platform will employ a validator based on his/her location and qualification to validate the product. After obtaining a validation certificate, the seller can advertise the validation certificate along with the product description and proceed with the auction mechanism. On the other hand, if the buyer or seller is not interested in validating a product, then no validator will be assigned. The product will be sold once both parties agree to the conditions.

When a validator validates a product, in return, the validator will be incentivized (with a validator fee and reputation score) for his/her work which will be explained in detail in

Section 5.2.1. Initially, the seller pays the entire validation fee. Later, whenever the product is sold at later stages, the buyer who buys this product will pay for half of the validator's reward since the validation certificate proves the legitimacy of the product which was absent before. Consequently, the seller gets back the half of validator's reward (refund fee) s/he paid initially. Figure 5.1 also includes various rewards and fees associated with each player in the system which are defined as follows:

1. **Handling Fee:** Handling fee is paid initially by the seller. It includes service charges, validation costs (half of the validator's reward for validating the product), and a penalty if the seller is untruthful about his/her product description.

2. **Refund Fee:** Reimbursement fee is the amount given back to the seller by the platform towards the end after determining the validation and penalty costs.

3. **Validation Fee:** The validators receive validation fees for validating products. If a potential buyer in a specific cycle chooses not to buy the product after requesting a product validation, then the seller will take care of the entire validator's fee temporarily. It will be given back when a buyer buys the product in successive rounds.

4. **Buyer's Fee:** A buyer has to pay a buyer's fee if the buyer chooses to obtain a validation certificate for a product (the other half of the validator's reward).

5. **Reputation Score:** A validator receives a reputation score based on the buyer's feedback.

Figure 5.2 explains the validation sequence of the product cycle in detail and the penalty incurred during this sequence. When a validator is assigned by the platform due to the validation request, the validator assignment will depend on the qualification and location of the validators in the system. The closer the proximity of the validator to the seller's place and the higher the qualification, the greater the chances of that validator being assigned for the validation assignment. The validator reward is calculated using an incentive mechanism and is split equally between the seller and the buyer. After the validation, a validation certificate is generated and recorded in the blockchain with a timestamp. Also, to save time

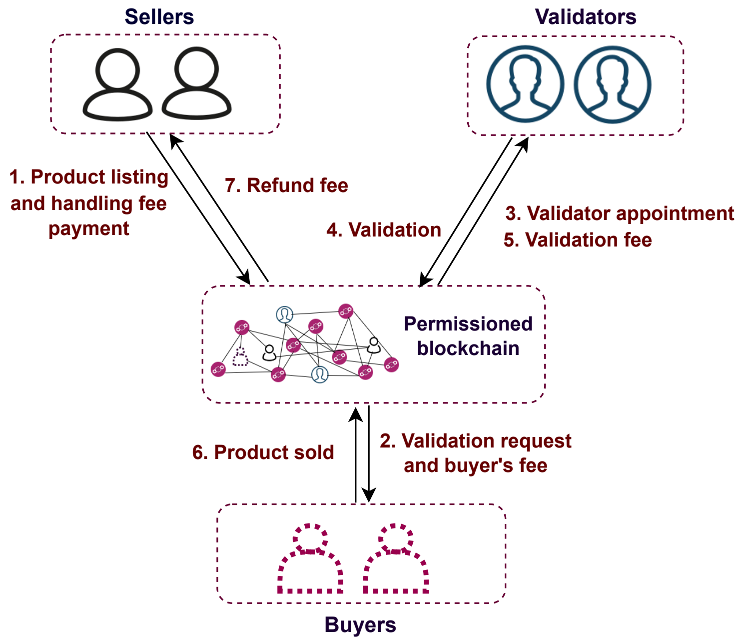


Figure 5.1. Overview of rewards and fees involved in the product validation of secondhand goods.

and validation cost, if a validation certificate is generated in a previous cycle for a product but not sold in that cycle, then the validation for the same product is not initiated again until the time given in the first validation certificate expires. If the validation determined the product to be of bad quality and the jury nodes determine that the seller was dishonest about his/her product, then a penalty will be charged to the seller. If, however, the validator has made a mistake during validation, then the validator will be given a penalty. This penalty determination will be explained below. If there is no validation certificate, then the buyer bids (in case of the auction) or sends a message to the seller regarding his/her interest in the product, and the product is sold.

Thus, the seller can initiate the auction mechanism once the validation certificate is obtained.

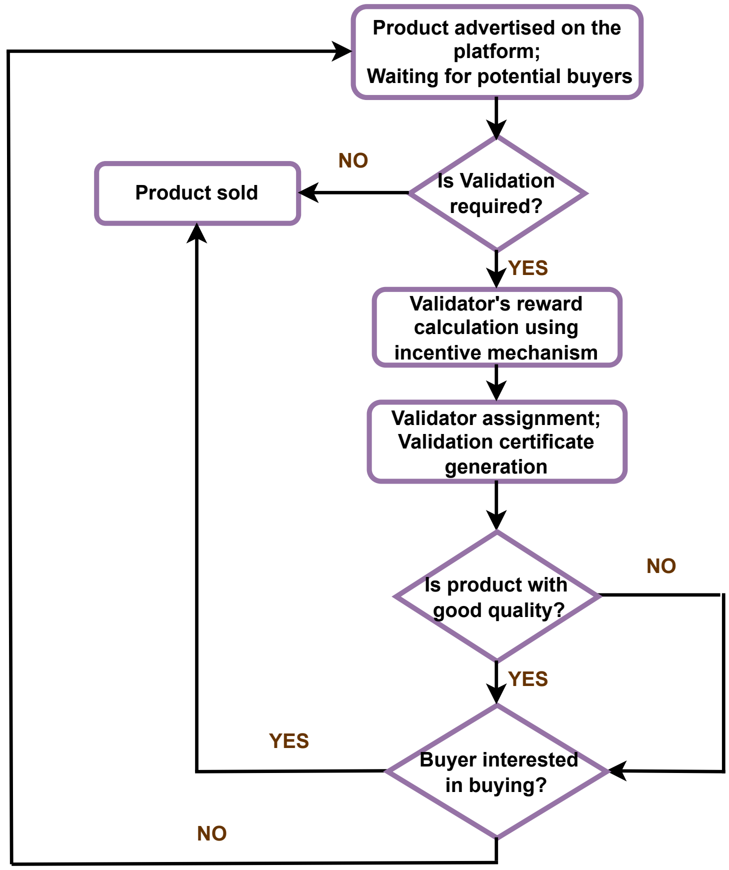


Figure 5.2. Outline of the product cycle in the validation sequence of secondhand goods.

Incentive Mechanism for Validators

To motivate the validators to participate and honestly validate the products, we design an incentive mechanism. For this design, we assume a single-leader multiple-follower Stackelberg game where the platform acts as a leader and the validators in the system act as followers. The set of validators in the system is denoted by $\mathcal{I} = \{1, \cdots, \mathrm{i}, \cdots, I\}$.

Here, we assume that the platform sets a basic wage for the validators' work based on the validators locations (proximity to the product to be validated) and qualifications. As the strategy of the platform, w_{i} denotes the wage earned by validator i per unit time from the platform, and $w_{\mathrm{i}} \in [0, w_{\mathrm{i}max}]$, where $w_{\mathrm{i}max}$ represents the maximum wage per unit time that the platform can offer to the validator i. The optimal strategy of the platform is denoted by w_{i}^{*}. The validator i's strategy denoted by $\mathrm{e_i}$ is defined as the effort exerted by validator i and is measured in terms of time taken by a validator i in validating a product. The optimal strategy of validator i is denoted as $\mathrm{e_i^*}$.

Then the utility function of the platform with respect to validator i, denoted as $U_{P\mathrm{i}}$, can be defined as

$$U_{P\mathrm{i}}(w_{\mathrm{i}}, \mathrm{e_i}) = \sum_{\mathrm{i}=1}^{I} \frac{r}{1 + \mathrm{e}xp(-\mathrm{e_i})} - w_{\mathrm{i}}\mathrm{e_i}. \tag{5.37}$$

Here, $\frac{1}{1+\mathrm{e}xp(-\mathrm{e_i})}$ is the satisfaction earned by the platform, and r is a mapping factor mapping satisfaction degree to monetary value earned by the platform. Here, we use the sigmoid function to represent satisfaction. As the platform's satisfaction with validators increases, the utility of the platform also increases. However, the relationship is not completely linear, since after a certain range, the increase in the platform's satisfaction causes a decrease in utility gained from an additional unit of effort and might even become negligible. The satisfaction ($\frac{1}{1+\mathrm{e}xp(-\mathrm{e_i})}$) is a function of effort since the platform's satisfaction depends on the validators' validations. The second term is the cost defined as total wage ($\mathrm{e_i}w_{\mathrm{i}}$). Here, for different value ranges of effort put in by a validator i, a specific basic wage factor (w_{i}) is multiplied to give the total wage. Hence, the more complex the product validation is, the higher the effort put in by a validator, and so, the higher the total wage given to the validator.

We can define the utility of validator i, i.e., $U_{V\mathrm{i}}$, as

$$U_{V\mathrm{i}}(w_{\mathrm{i}}, \mathrm{e_i}) = w_{\mathrm{i}}\mathrm{e_i} - L_{\mathrm{i}}. \tag{5.38}$$

The first term is the total revenue of the validator i. L_{i} is labor consumption for the validator while validating a product. Inspired by the concept in [21], L_{i} is directly proportional to the effort of validator i, and is defined as follows:

$$L_{\mathrm{i}} = l\mathrm{e_i^2}. \tag{5.39}$$

Here, $l > 0$ is a constant. This input factor maps the quality time to labor consumption.

We also include a budget constraint (B) for the platform where $\sum_{\mathrm{i}=1}^{I} w_{\mathrm{i}}\mathrm{e_i} \leq B$.

Game Analysis

We present the definitions of the Stackelberg equilibrium for our model and the optimal strategies of the players involved in our game design, followed by a detailed analysis of the Stackelberg game.

Definition 5.2.1 (Stackelberg Equilibrium)**.** *For our game model, the point ($w_\mathrm{i}^*, \mathrm{e}_\mathrm{i}^*$) is the Stackelberg equilibrium if $U_{V\mathrm{i}}(w_\mathrm{i}^*, \mathrm{e}_\mathrm{i}^*) \geq U_{V\mathrm{i}}(w_\mathrm{i}, \mathrm{e}_\mathrm{i}^*)$ and $U_{P\mathrm{i}}(w_\mathrm{i}^*, \mathrm{e}_\mathrm{i}^*) \geq U_{P\mathrm{i}}(w^*, \mathrm{e}_\mathrm{i})$.*

Definition 5.2.2 (Optimal Strategy)**.** *The objective of a Stackelberg game is to determine the optimal strategies of the players to maximize their utilities. In our model, we denote the optimal strategies of the platform and validator* i *by w_i^* and e_i^*, respectively, where*

$$\mathrm{e}_\mathrm{i}^* = argmax(U_{V\mathrm{i}}(w_\mathrm{i}, \mathrm{e}_\mathrm{i})), \tag{5.40}$$

$$w_\mathrm{i}^* = argmax(U_{P\mathrm{i}}(w_\mathrm{i}, \mathrm{e}_\mathrm{i})). \tag{5.41}$$

Here, we follow backward induction to analyze this Stackelberg game. Hence, the best response function of a validator i, i.e., e_i^*, is first calculated before calculating w_i^*. In this typical inequality-constrained optimization problem, the classic KKT conditions method [77] is used to derive the optimized strategy.

Optimal strategy of validator i

In order to analyze the equilibrium strategy, we start by obtaining the first-order derivative of $U_{V\mathrm{i}}$ with respect to e_i

$$\frac{\partial U_{V\mathrm{i}}}{\partial \mathrm{e}_\mathrm{i}} = w_\mathrm{i} - 2l\mathrm{e}_\mathrm{i}. \tag{5.42}$$

From (5.42), we obtain the second-order derivative to determine the concavity and maximum/minimum point, which is given by

$$\frac{\partial^2 U_{V\mathrm{i}}}{\partial \mathrm{e}_\mathrm{i}^2} = -2l < 0. \tag{5.43}$$

Thus, the utility of the validator is strictly a concave function, and there exists an optimal e_{i} that maximizes $U_{V\mathrm{i}}$. Setting $\frac{\partial U_{V\mathrm{i}}}{\partial \mathrm{e}_{\mathrm{i}}} = 0$, we obtain

$$\mathrm{e}_{\mathrm{i}}^{*} = \frac{w_{\mathrm{i}}}{2l}. \tag{5.44}$$

Optimal strategy of the platform

We substitute (5.44) in (5.37) to get

$$U_{P\mathrm{i}} = \sum_{\mathrm{i}=1}^{I} \frac{r}{1 + exp(-\frac{w_{\mathrm{i}}}{2l})} - \frac{w_{\mathrm{i}}^{2}}{2l}. \tag{5.45}$$

The KKT conditions method is often used to achieve a global minimum. Hence, to maximize the utility of the platform with the KKT conditions, we have the problem restated,

$$\text{Minimize:} - U_{P\mathrm{i}} = -(\sum_{\mathrm{i}=1}^{I} \frac{r}{1 + exp(-\frac{w_{\mathrm{i}}}{2l})} - \frac{w_{\mathrm{i}}^{2}}{2l}). \tag{5.46}$$

In order to utilize the KKT conditions, we first construct the Lagrange function L.

$$L(w_1, w_2, \cdots, \lambda) = -U_{P\mathrm{i}} + \sum_{\mathrm{i}=1}^{I} \lambda(w_{\mathrm{i}}\mathrm{e}_{\mathrm{i}} - B). \tag{5.47}$$

Based on KKT condition, optimality is defined as,

$$\frac{\partial U_L}{\partial w_{\mathrm{i}}} = 0. \tag{5.48}$$

The complementary slackness of KKT method is

$$\lambda(\sum_{\mathrm{i}=1}^{I} w_{\mathrm{i}}\mathrm{e}_{\mathrm{i}} - B) = 0. \tag{5.49}$$

Also, we can have two cases for discussion based on the budget constraint (B) and $\sum_{\mathrm{i}=1}^{I} w_{\mathrm{i}}\mathrm{e}_{\mathrm{i}}$ value. First, when the platform is idle, i.e., where the platform does not spend all the budget ($\sum_{\mathrm{i}=1}^{I} w_{\mathrm{i}}\mathrm{e}_{\mathrm{i}} < B$). Second, when the platform is busy ($\sum_{\mathrm{i}=1}^{I} w_{\mathrm{i}}\mathrm{e}_{\mathrm{i}} = B$), the platform may hire more validators than that in the ideal case, thereby using all the budget allocated. Here, we explain the effect of budget constraints on the utility of the platform for both cases.

Case 1: $\sum_{\mathrm{i}=1}^{I} w_{\mathrm{i}}\mathrm{e}_{\mathrm{i}} < B$. Here, the budget is not a tight restraint. Based on (5.49) we have $\lambda = 0$, and incorporating this value in (5.47), we get

$$\frac{\partial U_L}{\partial w_{\mathrm{i}}} = \sum_{\mathrm{i}=1}^{I} (-\frac{r\frac{w_{\mathrm{i}}}{2l}}{1 + exp(-w_{\mathrm{i}}/2l)^2} + \frac{2w_{\mathrm{i}}}{l}) = 0. \tag{5.50}$$

Hence, the optimal strategy of the platform (w_i^*) is determined as

$$w_i^* = l * log(\frac{4}{r}). \tag{5.51}$$

Thus, w_i^* and e_i^* are determined for case 1.

Case 2: $\sum_{i=1}^{I} w_i e_i = B$. Here, the validation requests increase and the platform gets busier, and the budget is tightly used by the platform. Then according to the non-negativity constraint and the budget constraint, we have $\lambda > 0$. Incorporating this value in (5.47), we can solve every validators optimal wage and effort in a form that consists of a global optimizing determinant factor λ:

$$\begin{aligned} w_i^* &= w_i^*(\lambda), \\ e_i^* &= e_i^*(\lambda). \end{aligned} \tag{5.52}$$

Thus, (5.46) becomes,

$$\text{Minimize:} - U_{Pi} = -(\sum_{i=1}^{I} \frac{r}{1 + exp(-\frac{w_i^*(\lambda)}{2l})} - \frac{w_i^*(\lambda)^2}{2l}). \tag{5.53}$$

Substituting optimal λ in (5.52), we get optimal strategies.

Conflict Resolution between Validators and Sellers

To avoid any unnecessary disagreements between sellers and validators who validate the product, a conflict resolution scheme is designed. Whenever a product is deemed as a bad quality by a validator then one of the three scenarios may occur:

1. When the seller's initial description was honest, announcing that the product has few unlikable qualities. In this scenario, no action is taken and the process of selling the product to the potential buyers starts.

2. The seller was dishonest in his/her initial product description, but accepts the validation, then the product is removed from the listing. If needed, the seller can relist the product with an honest product description.

3. When the validation does not match the seller's initial description of the product and the seller does not agree with the validation. During this scenario, the platform will appoint three jury nodes, chosen randomly among the pool of validators who have expertise/qualification in the field of the said product. After the appointment of the jury nodes, the seller and the validator will be allowed to upload the images or videos regarding the product's working condition and quality and then the jury will decide the dishonest party based on the majority of the jury nodes' decisions. Once this information is recorded, the dishonest party will be given a penalty. If the seller is at fault, then a double penalty is given and the amount is deducted from the refund fee set to be paid by the platform to the seller at the later stage. If the validator is at fault, the validation fee for that product is reverted.

We introduce the algorithm named Conflict Resolution Scheme (CRS) to incorporate all the possible scenarios described above in our system, which is given as in Algorithm 7.

Algorithm 7 Conflict Resolution Scheme (CRS)

Input: Validation for a product, seller's product description
Output: Updated validation fee (D), penalty (P), and product status

```
Product deemed bad quality during the validation
if seller's description = validation then
    No penalty (P = 0)
    Determine buyer's interest to buy the product
else
    Record seller's response to the product's validation
    if Seller's response = Accept then
        Product removed from the listing and penalty amount is deducted
    else
        Jury of three random validators is appointed by the platform
        Seller and validator submit all evidence
        Verdict of the majority jury nodes is recorded
        if Seller is dishonest then
            Double penalty is deducted from the refund fee and the product is removed from the listing
        else
            Validation fee (D) assigned for the validator is reverted
        end if
    end if
end if
```

5.2.2 Auction for Tangible Assets

This section explains the intricacies of the auction process employed once the tangible assets are prepared to be traded. For a tangible asset, the seller will broadcast their product details to the top $'n'$ potential buyers, after which the interested ones will participate in the auction. The process of determining the top $'n'$ potential buyers using reputation score and the smart contract-based auction is detailed in Section 5.1.2 and Algorithm 4 respectively since the process (i.e., decision-making and auction process for ownership transfer) remains largely similar for both tangible and intangible assets. The key distinction lies in the involvement of Non-Disclosure Agreements (NDAs) exclusively in intangible asset trading, while tangible assets do not require such agreements. Once the winner of the auction is determined, the ownership of the product is transferred and the product is later delivered. The delivery process can be tracked using a smart contract to avoid any fraudulent activities.

5.3 Transaction Efficiency

In this section, we focus on improving the transaction efficiency of the network by proposing a weighted L-H algorithm to determine efficient nodes that can broadcast transactions effectively and introducing transaction prioritization.

5.3.1 Weighted L-H Algorithm

Here, to identify influential nodes based on node parameters, we use a weighted L-H index algorithm. In our improved L-H index algorithm, we consider node parameters such as latency and activity time in addition to node degree proposed by the basic L-H index algorithm to determine influential miners [78]. By doing so, nodes with better performance characteristics in addition to the nodes with better connections are chosen as important miners. In our approach, we use parameters such as node degree, latency, and active time of a node. Here, we regard the impacts of latency and active time on node degree d. Active time and latency will assist in determining the reachable influential nodes and ensuring the quick transmission of data, respectively. Our weighted L-H index algorithm uses the basic

L-H index algorithm with latency and active time added as weights to it. Hence, we will be using a weighted degree to calculate the L-H index value of the nodes. Thus, node i's weighted degree ($d_{w\mathrm{i}}$) is calculated as, $d_{w\mathrm{i}}$ =$(\alpha B + \beta C)d_{\mathrm{i}}$. Here, B is assumed as the latency index and C is the active time index, and d_{i} denotes the degree of node i. The values of parameters α and β range from 0 to 1, and $\alpha + \beta = 1$. These parameters indicate the influence of latency and active time in the network. To illustrate, if $\alpha = 0.75$ and $\beta = 0.25$, then the latency has more influence in the network. C is determined as follows:

$$C = \frac{\text{active time of a node}}{\text{total time}}. \tag{5.54}$$

Here, the *active time* of a node represents the duration in which the node has been active in the network. *Total time* indicates a specified time frame in which the node was observed. Similarly, B is calculated as,

$$B = 1 - \frac{\text{latency}\ \ \text{minimum latency in the network}}{\text{maximum latency}\ \ \text{minimum latency}}. \tag{5.55}$$

Latency represents the observed node's latency. The denominator represents the range of the observed latency values in the network. Thus, the lower the latency of a node, the higher B. The L-H index value of a node i is $d_{w\mathrm{i}} + d^{\mathrm{i}}_{wn}$. Here d^{i}_{wn} represents the total summation of the weighted degree of all neighbors of the node i and is given by the formula, $d^{\mathrm{i}}_{wn} = \sum_{\mathrm{j}=1}^{n} d_{w\mathrm{j}}$, where $d_{w\mathrm{j}}$ is the weighted degree of any neighbor j and n is the total number of neighbors of node i. Our weighted L-H index algorithm uses the above calculation to determine the weighted degree of the nodes in the network. Then, the weighted degree of a node and its neighbors are used in determining the L-H index of the node. The higher the node's L-H index value, the higher its ranking. Based on this ranking, the nodes can be determined as influential nodes or not. The detailed process of our proposed weighted L-H index algorithm is presented in Algorithm 8.

In Algorithm 8, variables such as latency, active time, and degree are initialized. Neighboring nodes of all nodes in the platform are determined (Lines 1-3). The weighted degree of any node is calculated using latency, active time, and degree (Lines 4-6). The weighted

Algorithm 8 Weighted L-H index algorithm

Input: Degree (d), latency (B), active time (C), number of nodes in the network (m), parameters α and β

Output: Ranking of nodes

1: Initialize variables α, β
2: Initialize and calculate arrays B, C
3: Determine the neighbors of all the nodes
4: **for** i = 1 to m **do**
5: $\quad d_{w\mathrm{i}} = (\alpha B + \beta C) d_{\mathrm{i}}$
6: **end for**
7: **for** i = 1 to m **do**
8: $\quad$ Weighted L-H index value $= d_{w\mathrm{i}} + d_{wn}^{\mathrm{i}}$
9: **end for**
10: Sort nodes' weighted L-H index values in descending order

L-H index for each node i is calculated using the summation of the weighted degree of any neighbor i and the total summation of the weighted degree of all neighbors of the node i. All these nodes are sorted to determine the most efficient nodes in the platform (Lines 7-10).

5.3.2 Transaction Prioritization

Transaction prioritization mechanism focuses on the general welfare without giving precedence to fees. In our network, if there is a priority transaction like payments, validation of secondhand products, the judgment of the conflict resolution, etc. then that transaction is immediately recorded without delay, while a normal transaction in the blockchain network, generally, undergoes a wait time. Based on this priority, an efficient leader will record transactions in the blockchain, and the follower nodes will verify the leader's work.

The leader records transactions present in the mempool onto the main chain by following certain rules. That is, while creating a block, priority transactions are given more significance. Hence, when there is even one priority transaction appearing, the leader needs to immediately create a block and record this transaction along with other normal transactions in the mempool, which is then broadcast to the blockchain network. This rule will ensure that priority transactions do not encounter any delay. Sometimes, however, when there are a limited number of transactions entering the network, there might be some unused spaces in the blocks created. On the other side, normal transactions might undergo some delay in this process. To achieve this goal, we design a dynamic block creation policy for the leader to properly create blocks and record transactions. Detailed steps involved in the block creation process are shown in Algorithm 9.

Algorithm 9 Dynamic Block Creation

Input: Maximum number of transactions can be recorded in a block (m), the number of normal transactions (n_t), the number of priority transactions (p), the current waiting time (T_C), and the maximum waiting time (w)
Output: A new block
1: Initialize $T_C = 0$
2: **while** true **do**
3:     **if** $p \geq 1$ **then**
4:         Block is created with timestamp
5:         Leader waits for the followers' feedback
6:     **else**
7:         **if** $n_t \geq m$ or $T_C \geq w$ **then**
8:             Block is created with timestamp
9:             Leader waits for the followers' feedback
10:         **end if**
11:     **end if**
12: **end while**

At any moment, the presence of even one priority transaction in the pool will lead to the creation of block immediately (Lines 3-4). After finishing creating a block, this block is broadcast to the network to let the followers review the work of the leader, and the leader has to wait (Line 5). Here, all priority transactions present in the mempool will be recorded in the block followed by normal transactions. If there is no space to record all normal transactions, then the leader will leave some waiting transactions for the next block creation. If all transactions in the mempool are normal transactions, then the transactions are recorded based on their arrival time, i.e., a normal transaction n_{t1} is recorded before a normal transaction n_{t2} if n_{t1} has entered the network first. Block creation time for normal transactions depends on whether there are enough normal transactions in the mempool to entirely occupy a block. If the current number of normal transactions in the pool is equal to or larger than m, or if the time elapsed is equal to or larger than w, then a block is created (Lines 6-8). This will ensure a trade-off between the block space utilization and system efficiency on packaging priority transactions. After this block creation, the leader needs to wait for the feedback of that block from the followers (Line 9) in the network before it can create a new block. By integrating dynamic block creation into our platform, the network can achieve enhanced efficiency in processing critical transactions by mitigating any latencies.

6. EXPERIMENTAL RESULTS

In this section, we perform various simulation experiments to analyze the performance of our blockchain-based trading platform for both tangible and intangible assets.

6.1 Evaluation of Loyalty Ranking Algorithm Involved in Consensus

In this section, we perform simulation experiments to determine the performance of the loyalty ranking algorithm, which is based on our XGBoost. Loyalty players are determined who will be involved in the consensus.

To determine loyal players to be a part of the blockchain network, 1000 data samples from players participating in our platform were obtained and loaded as an input dataset to our XGBoost classifier algorithm. The dataset was divided into training data and test data, with a test size ratio set as 0.33. Hence, 330 data samples are treated as test data. The test ratio (test data samples are always less than train samples) can be varied and the model needs to be tuned accordingly to get the desired accuracy. Data were then pre-processed to clean and organize. Figure 6.1 represents the values of binary_logloss function, and it can be seen that the values converge around 0.07, thus making the algorithm more accurate, since the metric is a loss function, the lesser the value, the greater the accuracy.

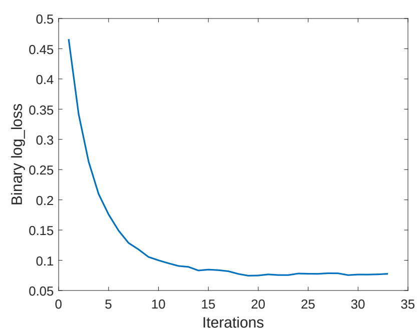


Figure 6.1. Evolution of Binary_logloss function.

Parameters should be tuned to obtain good accuracy. The loss metric was chosen as logloss since this function can best determine how good our model is in predictions. The

Table 6.1. Parameters tuning

Parameters	Values
Learning rate	0.04
Max_depth	10
Early_stopping_rounds	15
Eval_metric	Binary_logloss

Table 6.2. Confusion matrix for XGBoost

Possible Outcomes	Values
True Positive	211
True Negative	116
False Positive	2
False Negative	1

other parameters and their values are shown in Table 6.1. The accuracy obtained by the algorithm is 99%. Accuracy is calculated according to

$$\text{Accuracy} = \frac{\text{Number of loyal/disloyal players correctly determined}}{\text{Total number of predictions}}. \tag{6.1}$$

Table 6.2 describes the performance of the algorithm on test data using a confusion matrix of XGBoost Algorithm.

To illustrate the effect of the total score on the predicted output, we determine the total score (TS) based on T, F, and R values, with all the afore-mentioned values having an equal weightage. Hence, total score can be given as, $TS = x_1T + x_2F + x_3R$, where x_1, x_2, and x_3 are weights. Here, T, F, and R are assumed to have the same weightage. Hence, the values of x_1, x_2, and x_3 are set as 0.33. Also, the maximum values of T and F are set as 60 (months) and 10. Similarly, the maximum value of R is assumed as 1. Also, we assume a target score to determine loyal and disloyal players. The loyal players are players whose TS value is above the target score and disloyal players are players whose TS is less than the target point. Figure 6.2 explains the impact of the total score on the predicted output. For this scenario, we focused on 10 out of 1000 data samples and obtained their T, F, and R values based on which their TS value was calculated. Based on the TS value calculation, we assumed the target score to be 15. Also, the maximum TS that can be obtained is around 25. Hence, any player having a total score above 15 is considered a loyal player; i.e., the output predicted by the XGBoost-based algorithm will be 1, once the algorithm is passed the input values and the classifier model is properly tuned and trained. Similarly, players scoring less than 15 will be predicted by the algorithm to be disloyal players.

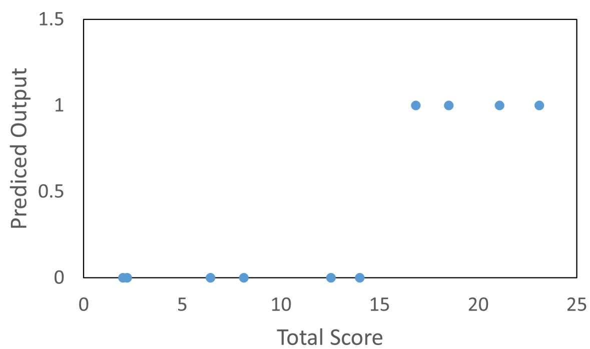


Figure 6.2. Predicted output based on the total score.

6.2 Performance Analysis of Intangible Asset Trading

Here, we focus on analyzing the efficacy of intangible asset trading, specifically, evaluating the pricing mechanism for NFTs and assessing the effectiveness of the ID3-based decision tree algorithm (Algorithm 3). Additionally, we conduct a comparative analysis between our adaptive NFT management framework and established benchmark algorithms. Moreover, we illustrate how various parameters influence bidder's utility within the auction mechanism design.

6.2.1 Price Determination of NFT containing health data

For evaluating the performance of our pricing mechanism, we analyze the influence of various parameters on utilities and optimal strategies of buyer k and a seller. Here, we denote $\alpha = \sum_{k \in \mathcal{K}} \alpha^k$, where α indicates the summation of the unit prices for data purchased by all buyers and α^k is the price of one data field obtained by buyer k. In Figure 6.3, we determine the impact of scalar parameter α and quality of NFT j (Q_{j}), on the optimal strategies of the seller and buyer k. The quality of NFT j is assumed to be ranging from 0-100. It can be noted from Figure 6.3 that for the increase in Q_j, the optimal strategy of the seller (P_j^*) increases gradually but on the other hand, the optimal strategy of the buyer k (X_j^{k*}) decreases. This effect of Q_j on P_j^* and X_j^{k*} is due to the fact that when Q_j, i.e., the quality of an NFT is improved, the cost of that NFT may also increase causing the seller to set a higher P_j (5.14). Meanwhile, if the seller sets a higher price, the probability of the buyer buying the NFT j will drop down, after a certain threshold. Since buying the NFT for a higher price (beyond the product's worth) will result in less profit for the buyer. Also, since α and α^k are directly proportional to the optimal strategies (5.14), increasing α and α^k values increases P_j^* and X_j^{k*}.

Similarly, Figure 6.4 illustrates the significance of Q_j and α on the expected utilities of the seller and the buyer. When quality increases, the price of the NFT also increases from (5.10), resulting in a slight increase in the seller's expected utility compared to the buyer's expected utility. On the other hand, the increase in quality increases the likeliness of a buyer to buy that NFT, assuming that the effect of price on the buyer's decision is negligible. Also, from (5.2) and (5.10), it can be seen that α^k and α are directly proportional to the utilities of the buyer and the seller.

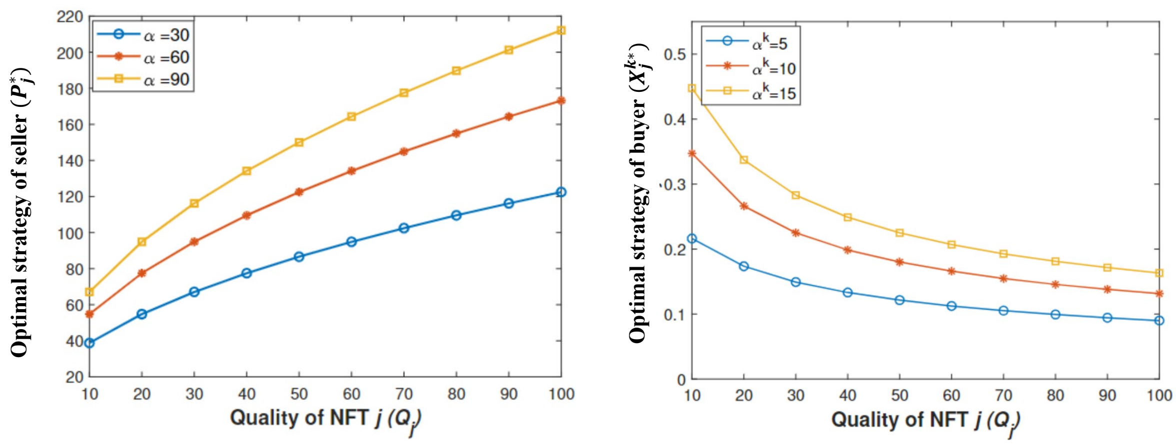


Figure 6.3. Influence of unit price (α) and quality of NFT j on strategies of the seller and buyer k.

Figure 6.5 illustrates the effect of price and buyer k's likeliness to buy an NFT j on the seller's utility (U_S), and Figure 6.5(a) depicts the three dimensional plot for the same. Figure

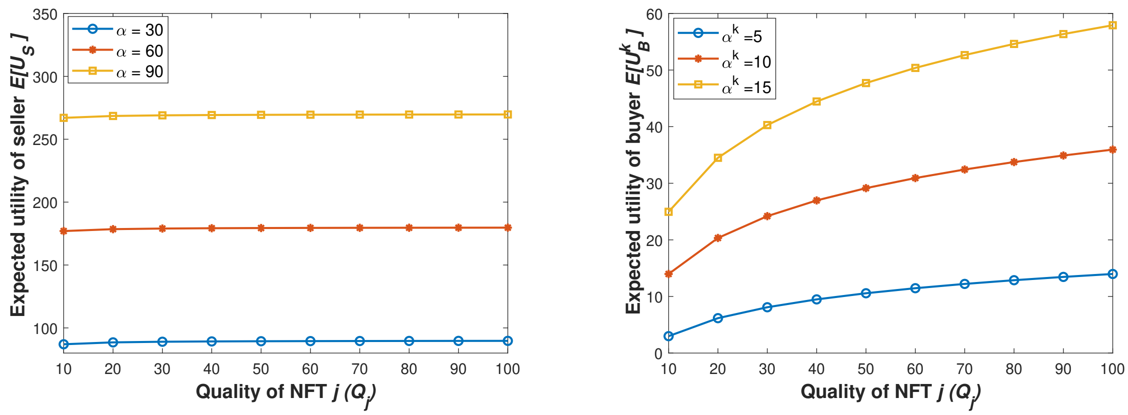
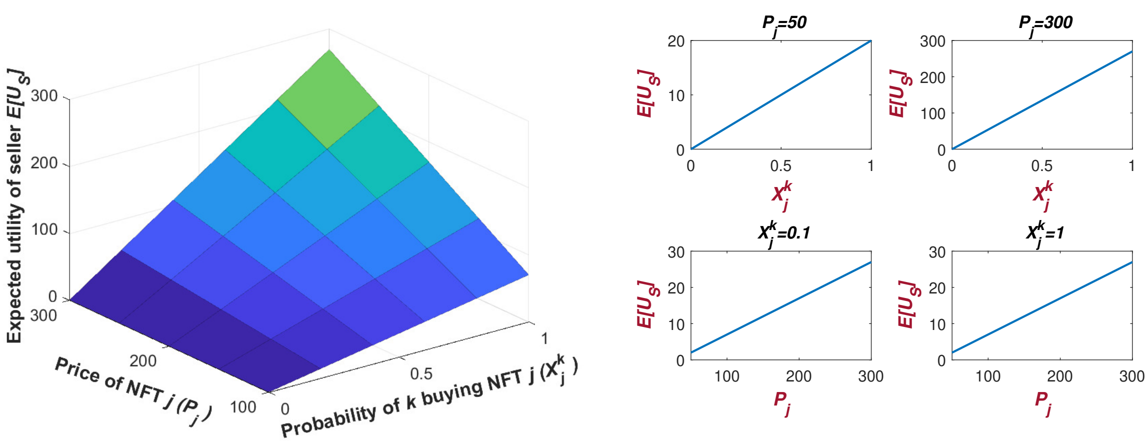


Figure 6.4. Influence of unit price (α) and quality of NFT j on the expected utilities of the seller and the buyer.

6.5(b) shows the individual relationships of price and demand level on the seller's utility. By increasing both P_j, i.e., price of an NFT j, and X_j^k (buyer k's likeliness to buy the NFT), the seller's utility increases. This is due to the fact that as the NFT's selling price increases, the profit of the seller also increases. Similarly, when a buyer's inclination to buy the NFT increases, the seller is expected to achieve higher profits, owing to the possibility of more customers buying the NFT.

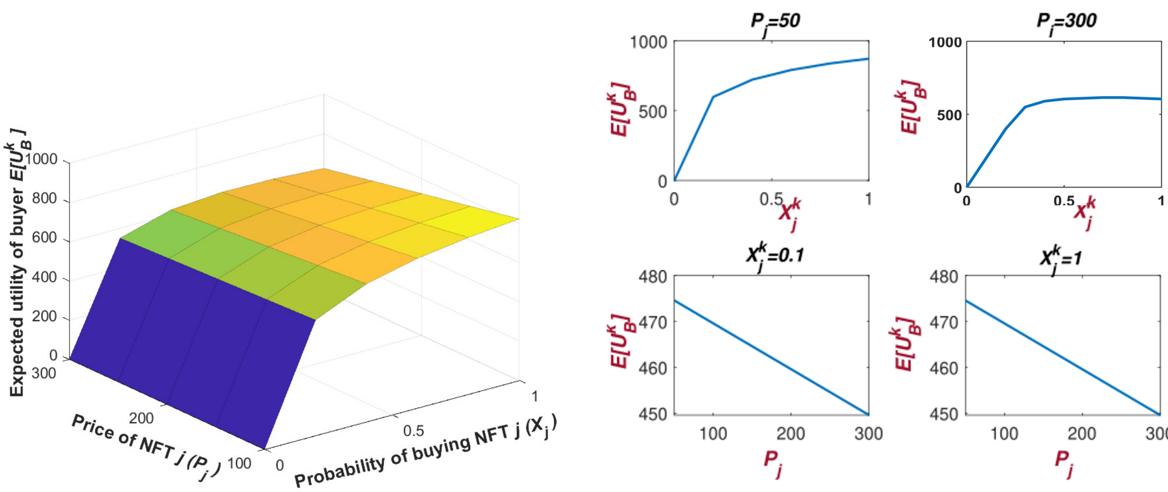


(a) Three dimensional plot of the seller's utility

(b) Utility of seller vs. P_j and X_j^k

Figure 6.5. Effect of price and buyer k's likeliness to buy an NFT on seller's utility.

Figure 6.6 represents the effect of price and the buyer k's likeliness to buy NFT j on the buyer's utility (U_B^k). It can be seen that as the price of the product (NFT) increases from 50 to 300, the buyer's utility decreases. This effect is due to the fact that when the buyer pays more for the NFT, the buyer's profit decreases, assuming quality remains the same. Also, when the buyer's likeliness to buy NFT j increases from 0.1 to 1, then his/her utility will also more likely increase, assuming all other parameters (price, quality) are constant.



(a) Three dimensional plot of buyer k's utility

(b) Utility of buyer k vs. P_{j} and X_{j}^k

Figure 6.6. Effect of price and buyer k's likeliness to buy an NFT on buyer k's utility.

6.2.2 Adaptive NFT Management

To evaluate our adaptive NFT management design, we simulate the output of our algorithm and investigate the algorithm's performance against a benchmark algorithm. Also, we analyze the impact of various critical factors on the selection of the data fields. We considered 10 data fields, and the popularity factor for these data fields was set to follow Poisson distributions. Here, 30 samples were generated for 10 data fields with mean popularity values ranging from $50 - 150$. Thus, the popularity of the data fields is learned by our PANMS algorithm, and the arms are chosen based on the learning, for every iteration (t). First, we obtained a bar graph determining the popularity of the data fields to understand the learning aspect of our algorithm over time. Figure 6.7 depicts the learning behavior of our algorithm, based on the popularity factors of the data fields over time t. Here, we assumed that the data field D5 has the highest popularity factor, followed by the data fields, D3 and D7. It can be noted that at $t = 10$, all data fields are equally explored. In the graphs for $t = 20$ and $t = 30$, the algorithm has learned that the data fields D5, followed by D3 and D7, have higher popularity factors. Hence, they are exploited more compared to other data fields.

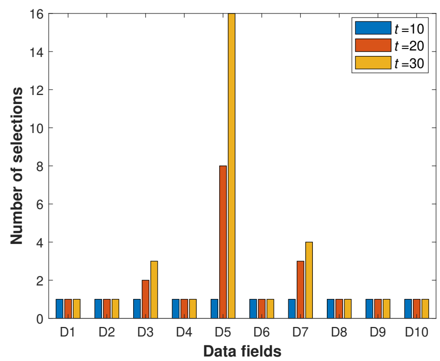


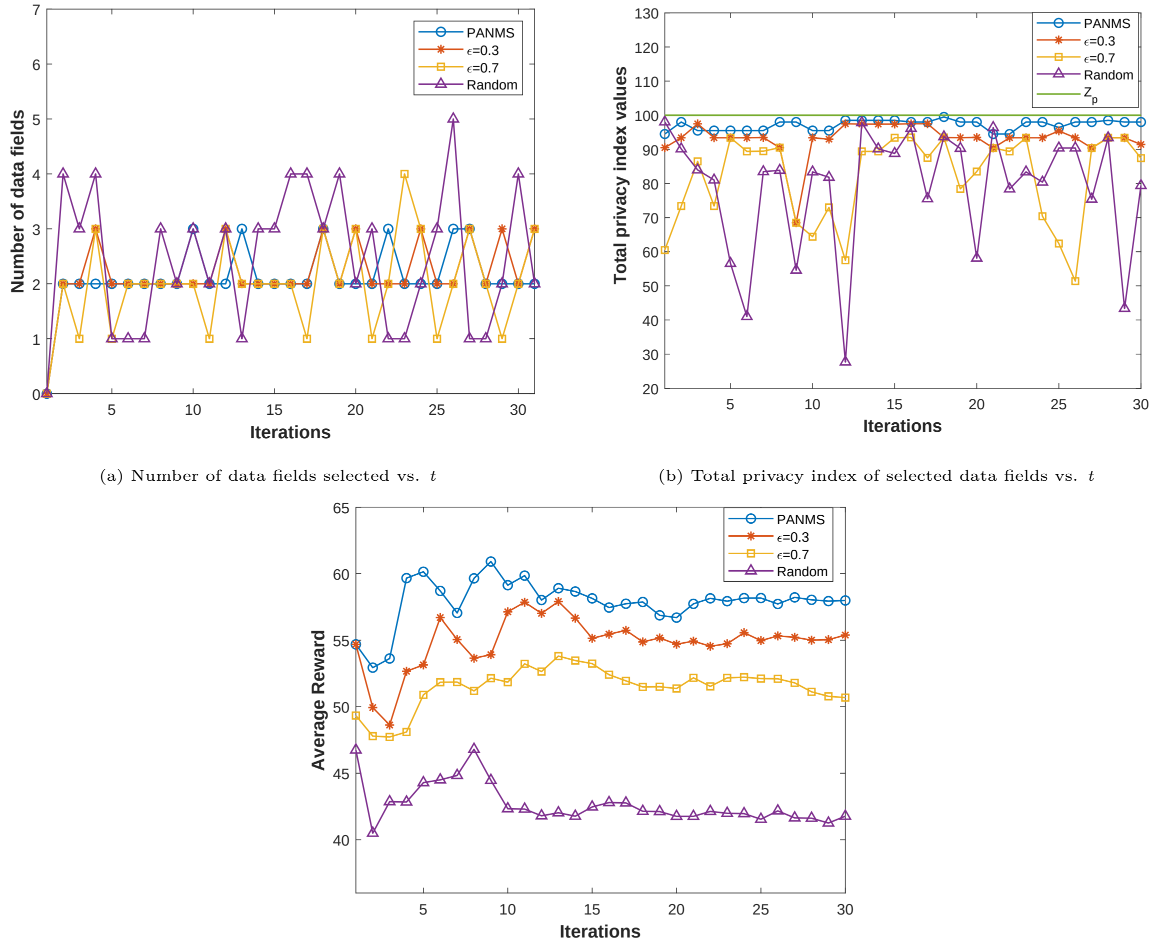
Figure 6.7. Representation of the number of times each data field is selected over time t.

To demonstrate the effectiveness of our algorithm, we compare our PANMS algorithm with two benchmark schemes: epsilon greedy and random. Our PANMS explores all data fields equally at the start but then proceeds to learn the popularity levels of the data fields and exploits such optimal data fields while considering any new information available. On the contrary, the epsilon greedy algorithm chooses between exploration (random arm) and exploitation (arm with maximum reward) based on the ϵ value. Here, we use two ϵ values: $\epsilon = 0.3$ and $\epsilon = 0.7$, where ϵ indicates the probability of exploration. For example, for $\epsilon = 0.3$, arms are randomly chosen 30% of the time, and the remaining 70% of the time exploits the best arm without taking any new information into account. Also, the random algorithm randomly selects an arm during each iteration without undergoing any learning process. The popularity values are changed randomly for every iteration. The privacy index constraint Z_p was set to 100 for all iterations.

Figure 6.8(a) shows the number of popular data fields that each algorithm chooses for every iteration based on the popularity of the data fields ensuring the total privacy index values of these chosen fields are within the constraint. Since PANMS undergoes learning, it always selects the combination of data fields with higher popularity values, thereby achieving the best reward. Hence, for most iterations, the number of data fields selected by our algorithm will not deviate much. Similarly, for a given ϵ value, the epsilon-greedy algorithm determines the best set of data fields and the number in each iteration remains mostly the same. However, as the ϵ value becomes higher, more randomness is involved in selecting the data fields, causing deviation in the number of data fields selected. On the other hand, the random algorithm selects data fields randomly causing the number of data fields selected to be completely random. Figure 6.8(b) shows the total privacy index values for the selected data fields for each iteration. All three algorithms considered in our experiment select data fields while considering the given Z_p constraint value. PANMS and epsilon-greedy select more popular data fields while satisfying the privacy index constraint to achieve the maximum reward. Epsilon greedy algorithm with higher ϵ value behaves similarly to the random algorithm, causing the total privacy values to vary more. The random algorithm selects data fields completely in a random manner, which results in low total privacy index values in most cases (e.g., $t = 11$) and fewer rewards.

Also, Figure 6.8(c) shows that the average reward of our algorithm is higher than random and epsilon-greedy algorithms. Since our algorithm learns the popularity of each data field, higher rewards are obtained in PANMS as can be seen in (5.28). Meanwhile, the random algorithm just randomly selects data fields in every iteration causing inconsistent performance. Also, the reward of the epsilon-greedy algorithm is less than PANMS since the epsilon-greedy algorithm chooses to explore (selecting random arm) or exploit (selecting best arm) based on the ϵ value. A higher ϵ value will cause a higher exploration, thereby losing maximum rewards. However, a lower ϵ value will lead to sub-optimal behavior. Hence, getting the right ϵ value is significant yet difficult. On the other hand, PANMS performs consistently better in most cases due to its learning of popularity factor.

In Figure 6.9, the effect of various parameters on the cumulative reward obtained by a seller is studied in detail. Figure 6.9(a) demonstrates the effect of selling margin (U_{D_l}) on the cumulative reward obtained. Since, the sales margin has a direct effect on reward from (5.29), with an increase in U_{D_l}, cumulative reward also increases. Figures 6.9(b)-6.9(d)



(a) Number of data fields selected vs. t

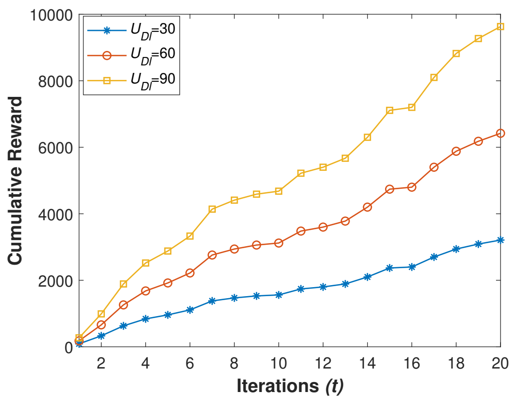
(b) Total privacy index of selected data fields vs. t

(c) Average rewards vs. t

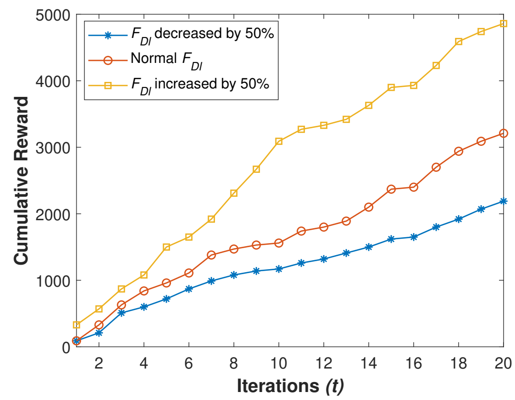
Figure 6.8. Comparison of Random, PANMS, epsilon-greedy algorithms for various critical parameters.

show the effect of the popularity factor, the number of customers, and the number of data fields on the cumulative reward obtained by the seller. In all the afore-mentioned cases, the cumulative reward obtained by the seller increases when the number of customers, popularity factor (affected by demand), and the number of data fields sold by the seller increase.

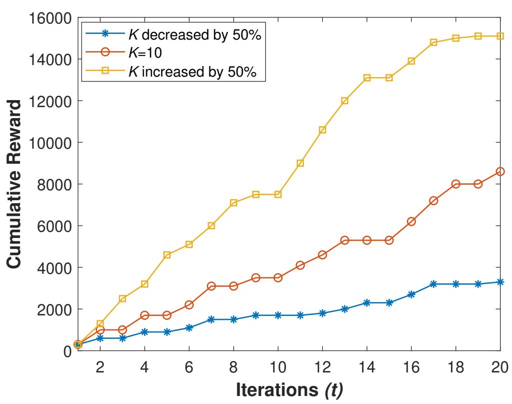
When the number of customers is increased in our system, there are chances that the new customers might also buy the data fields offered by the seller, in addition to the old customers, causing the seller's reward to increase. Similarly, even when the number of customers remain same, but if the popularity level of some data fields increases, then the number of customers buying those data fields will increase from the original, owing to the new interest in the data fields, thus improving the seller's reward. Also, when the seller opts to bring in new NFTs to sell, there will be an additional demand for the new NFTs apart from the demand for the old products/NFTs, thereby increasing the seller's cumulative reward.



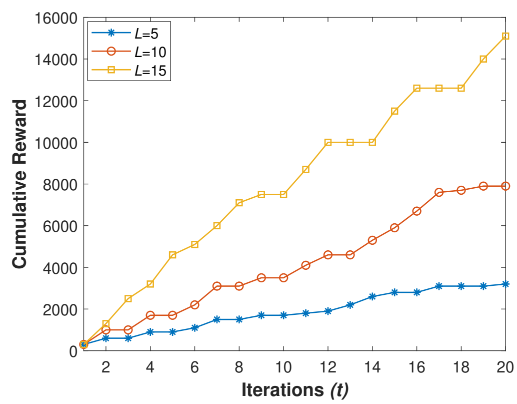
(a) Effect of sales margin on cumulative reward



(b) Effect of popularity level on cumulative reward



(c) Effect of number of consumers on cumulative reward



(d) Effect of number of data fields on cumulative reward

Figure 6.9. Influence of various parameters on the cumulative reward obtained by the seller.

6.2.3 Analysis of Auction Mechanism for Intangible Assets Trading

In this section, we focus on analyzing the ID3-based decision-making algorithm's effectiveness in aiding buyers in determining their participation in auctions. Additionally, we discuss the effectiveness of the smart contract-based auction mechanism (Algorithm 4).

Performance Analysis of ID3-based Decision-making Algorithm

To assist companies in deciding if they need to invest in a specific NFT or not, we incorporate the ID3-based decision-making algorithm. The classifier used here was binary, i.e., 0 and 1 classifier. The dataset containing 1000 samples was divided into training data and test data, with a test size ratio set as 0.33, the most used value for the test size. Hence, 330 data samples are treated as test data. The hyperparameters like max_depth, random state, and max-leaf nodes should be tuned to obtain a better model performance. The loss function chosen is log loss since this function is commonly used in classification problems to determine the performance of the model.

To determine whether to invest in a specific NFT or not, we determine the valuation of the NFT based on the scores defined in Definitions 5.1.3-5.1.6. Here, the value of each score for the dataset is randomly generated and ranges between 0-10. Once all four scores are obtained, a decision tree is built with these scores as attributes and binary 0 and 1 as output labels. Here, all aforementioned four scores are given equal weights as all scores are significant in determining the NFT's valuation. Based on the trained data, if the score value of each attribute is greater than 50% then the decision is made by a company to invest in the specific NFT. If even one attribute score is less than 50%, then the algorithm outputs the value 0, i.e., not to invest in the NFT. This decision rule ensures that only well-rounded technologies with strengths in all evaluated areas are selected. This helps in reducing the risk of poor investments. Also, setting a threshold at 50% balances risks and rewards.

The accuracy obtained by the algorithm is 89%. To represent the predictive performance of our decision tree classifier model on the dataset, a confusion matrix is obtained and is shown in Table 6.3. It can be seen that the true positive and true negative values of the confusion matrix are 150 and 116. True positive is where our algorithm correctly predicts the positive class. Similarly, a true negative is where our algorithm correctly predicts the negative class.

Table 6.3. Confusion matrix of ID3-based decision matrix

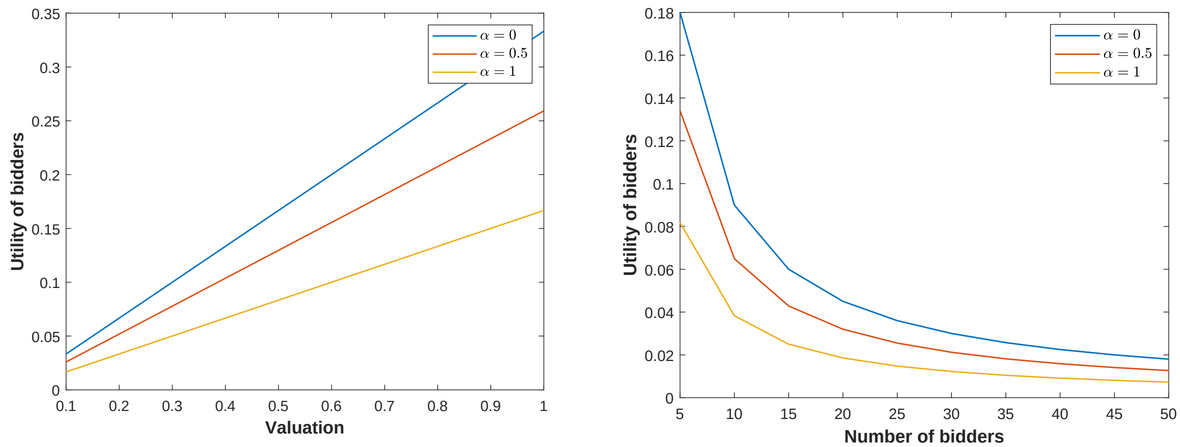
ClassLabel	0	1
0	150	10
1	24	116

First Price Sealed-bid Auction Mechanism Analysis

Here, the factors impacting the utility of bidders are analyzed, and our auction mechanism is compared against other benchmark algorithms.

Figure 6.10 analyzes the impact of valuation, number of bidders, and bidders' spitefulness on the utility of the bidders. Here, we assume that the bidders' valuations of a specific NFT are assumed to be in a uniform distribution between 0 and 1. With an increase in the average valuation by bidders on the NFT, the utility of bidders in an auction also increases as shown in equation (5.16), assuming that the average bid amount by bidders is constant. Also, spitefulness can be defined as the bidders' intention to decrease the utility of other bidders even by harming their utility. From Figure 6.10a, it can be seen that as the spitefulness value of a bidder increases, the average bid amount is driven up by this competitiveness of the bidder without the bidder even wanting to win, thus causing a decrease in the utility of other bidders. This is because each bidder tries to obtain the NFT for themselves and lower the utility of other bidders, and so bids more. In Figure 6.10b, as the number of bidders increases, the utility of the bidders decreases due to the increase in competition which causes the bidders to increase their bid amounts, thereby decreasing their utilities.

In Figure 6.11, we compare our first price sealed-bid auction (FPSB) with other benchmark auction mechanisms: all-pay auction and senior auction. From Figure 6.11a, it can be seen that the bidders in the senior auction as well as in the all-pay auction bid lower than in our FPSB. This is because the risk of the bidders in the senior auction and all-pay auction is higher than in the FPSB auction. Figure 6.11b illustrates that as the number of bidders increases, in the first-price auction, the bid amount slightly increases and becomes almost constant. This trend shows that the number of bidders increases initially when competition



(a) Effect of valuation on the utility of bidders.

(b) Effect of the number of bidders on the utility of bidders.

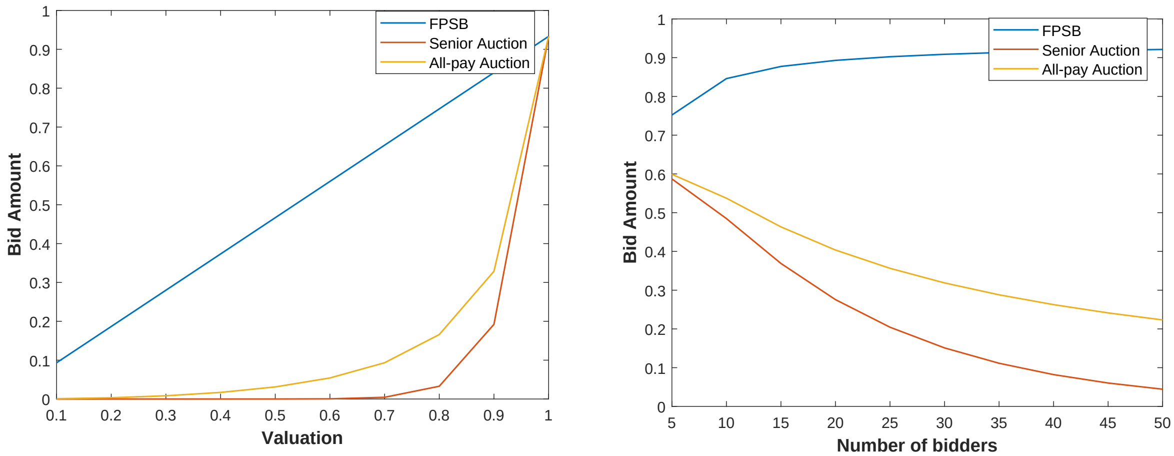
Figure 6.10. Impact of valuation and number of bidders on the utility of bidders in first-price sealed bid auction.

increases. Later, the bid amount does not increase as the bidders also try to gain some profits which is not possible if they keep increasing bid amounts. Similarly, the bid amount decreases in other auctions as the competition grows due to the penalty. Hence, the bidders in other auctions try to decrease their penalty by bidding less as there is an increase in the competitors (which might lessen the probability of winning for each bidder).

Smart Contract Implementation

Here, the smart contract related to the first-price sealed bid auction is designed and deployed in Solidity. In Remix IDE, we design a SealedBidAuction.sol file to implement a first-price sealed bid auction, and the contract consists of two main functions: submitSealedBid and revealBid. In the submitSealedBid function, the bidders (different accounts) submit their sealed bid amount. Similarly, in the revealBid function, the contract obtains the bidders' actual bid amount and compares if the actual bid amounts are the same as the bidders' sealed bids. Later, the highest bid amount and bidder (account) are determined.

Figure 6.12a shows the display screen once our contract is compiled and deployed successfully. Here, the inputs are submitSealedBid and revealBid. The highest bidder and highest bid are calculated after the auction is ended. Figure 6.12b displays the submitting bid phase



(a) Effect of valuation on bid amount.

(b) Effect of number of bidders on bid amount.

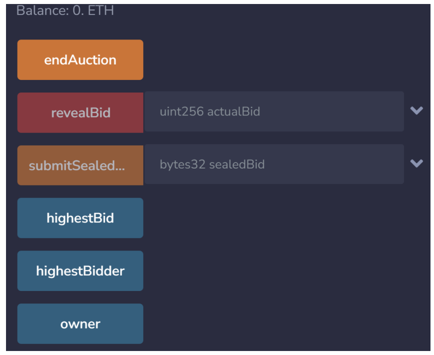
Figure 6.11. Comparisons of other auction mechanisms with our first price sealed-bid auction.

where the bidders submit their bids as a hash to ensure the other bidders do not have prior knowledge. Figure 6.12c shows the specific bidder submitting their real bid amount during the revealBid phase. If the actual bid amount submitted is the same as the hash provided in the submitSealedBid function, then the transaction succeeds. Otherwise, there will be an error notification. Finally, once all the bidders (accounts) complete the submitSealedBid and revealBid functions, the highest bid and the highest bidder are determined and are shown in Figure 6.12d. These outputs are then displayed under the highestBid and highestBidder tabs.

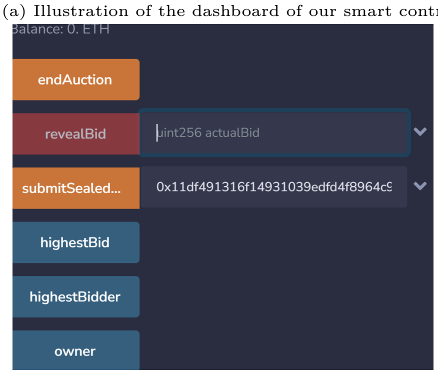
6.3 Performance Analysis of Tangible Asset Trading

For evaluating the performance of incentive mechanism design for validators involved in the validation of secondhand goods, we analyze the influence of various parameters on the optimal strategies of validator i and the platform.

Figure 6.13 focuses on the effect of wage on the optimal strategy of a validator i. From the figure, it can be seen that increasing wages motivate the validator to put in a higher effort during validation. l (input factor mapping quality time involved in validation to labor



(a) Illustration of the dashboard of our smart contract.



(b) Demonstrating the bid submitting phase of bidders in hash.

(c) Displaying the bid revealing phase.

(d) Depicting the final output of highest bid and winner.

Figure 6.12. Screenshots illustrating the smart contract implementation of first-price sealed bid auction.

consumption) value affects the optimal strategy of the validator directly as can be seen in (5.45).

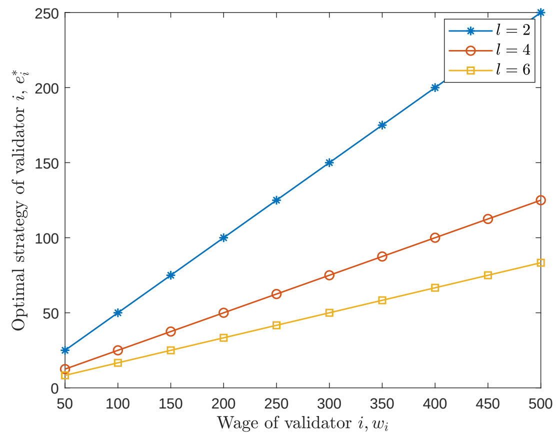


Figure 6.13. Influence of wage on the optimal strategy of validator i.

Figure 6.14 explains the effect of r (mapping factor) and l on the optimal strategy of the platform for two cases: case 1 ($we < B$) and case 2 ($we = B$). From Figure 6.14, it can be seen that as r value increases, the optimal strategy of the platform for case 1 decreases. Also, increasing the l value increases the optimal strategy. The same effect is also determined in (5.53). The effect of the same parameters for case 2 ($we = B$) is seen in Figure 6.15. For the same values of l and r, case 1 shows better utility compared to case 2, since for a specific value of B, we of case 1 will be less than we of case 2. Hence, as the cost of the platform (we) decreases, the utility of the platform increases.

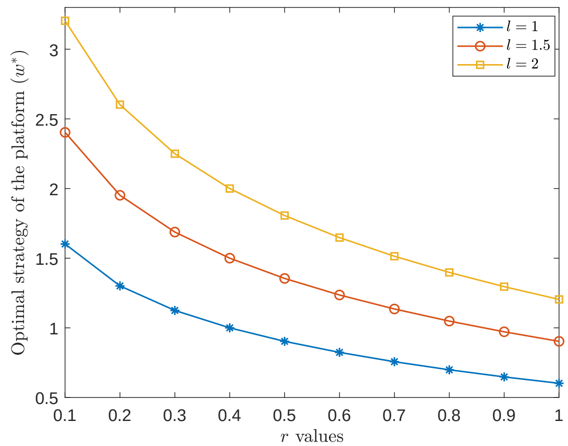


Figure 6.14. Influence of input factor and mapping factor on the optimal strategy of platform for $w * \mathrm{e} < B$.

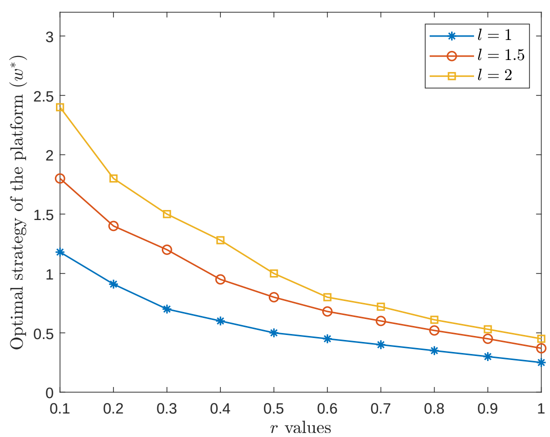


Figure 6.15. Influence of input factor and mapping factor on the optimal strategy of platform for $w * \mathrm{e} = B$.

Figure 6.16 focuses on the effect of the budget (for both cases) on the utility of the platform. It can be seen that for a specific budget allocated to the platform (B), the utility of the platform is greater in case 1 compared to case 2. Hence, increasing the cost of the platform will have an adverse effect on the utility of the platform as can be seen in (5.37). Also, when there is a budget constraint and the platform is busy with incoming validation requests (case 2: $we = B$), the platform might not be able to afford to hire more than a specific number of validators. This might cause the effort put in by each validator already present in the platform to increase along with the wage given (to ensure the validator's satisfaction), affecting the utility of the platform slightly in an adverse way. Here, we assume that during the initial development phase of the platform, the platform motivates nodes/users with extra monetary benefits to join in validation.

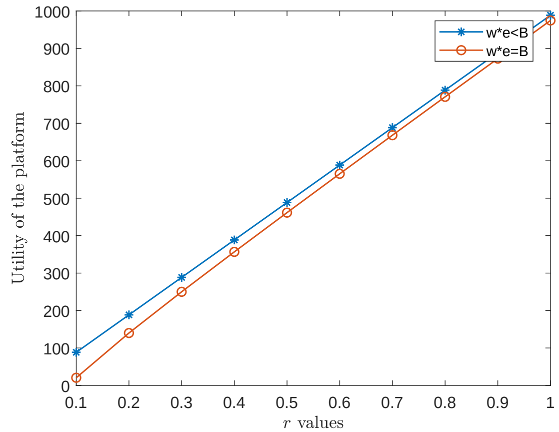


Figure 6.16. The effect of cost ($w * \mathrm{e}$) on the utility of the platform.

Table 6.4 details the results of the Conflict Resolution Scheme (CRS). Various possible scenarios are discussed focusing on the validation fee (D), penalty (P) updates, and product status. Assuming that the validator will get \$100 if the validation is successful and the penalty amount is \$100 for dishonesty, we can proceed with four different scenarios as follows:

Table 6.4. Results of CRS

Scenario 1	Scenario 2	Scenario 3	Scenario 4
D=100;P=0	D=100;P=100	D=100;P=200	D=0;P=0
Wait	Product removed	Product removed	Wait

1. Scenario 1: When the seller's description is the same as the validation, then there is no penalty ($P = 0$) to the seller. Also, the validator gets his/her validation fee ($D = \$100$), and the product status goes to wait until a buyer buys the product.

2. Scenario 2: When the seller's description is different from the validation and the seller accepts his/her mistake, then there will be a penalty amount (P = \$100). D will be \$100, and the product is removed from the listing.

3. Scenario 3: When a conflict arises between the seller and the validator's validation, and if the jury decides the seller to be at fault, then D becomes \$100, and the seller gets double penalty \$200 which also includes the fee to the jury nodes. The product is removed from the listing.

4. Scenario 4: Similar to Scenario 3, if the jury finds the validator to be at fault instead of the seller, then the validation fee D will be 0. There will be no penalty for the seller ($P = 0$), and the product gets into wait status until a buyer buys the product.

6.4 Performance Analysis of the Weighted L-H Algorithm

To rank the influential nodes based on the node's L-H index value using our weighted L-H index algorithm, we simulated a part of the blockchain network with 26 nodes. Among them, nodes 21-26 are assumed to be nodes that form a pool. The ranking is obtained for different scenarios by using our weighted L-H index algorithm. In the ranking list, the pools are not included as this method focuses only on single nodes (miners). Nodes with minimum latency, maximum active time, and a higher degree are given higher ranks. The simulated network can be seen in Figure 6.17.

Figure 6.18 indicates the change in L-H index values of the nodes based on active time and latency. Case 1 illustrates the L-H index value when no parameters are added as weights (i.e., prior L-H indexing algorithm [78]). Case 2 represents the L-H index with only active time. Case 3 indicates the L-H index value with only the latency parameter, whereas case 4 indicates the L-H index value with both active time and latency. It can be seen from Figure 6.18 that node 14 has a higher degree value in case 4 (when our weighted L-H index algorithm is applied). This is because node 14 and its neighbors have a minimum latency value and maximum active time than other nodes. Similarly, rankings of nodes 1, 8, 11, 13, and 17 are also changed.

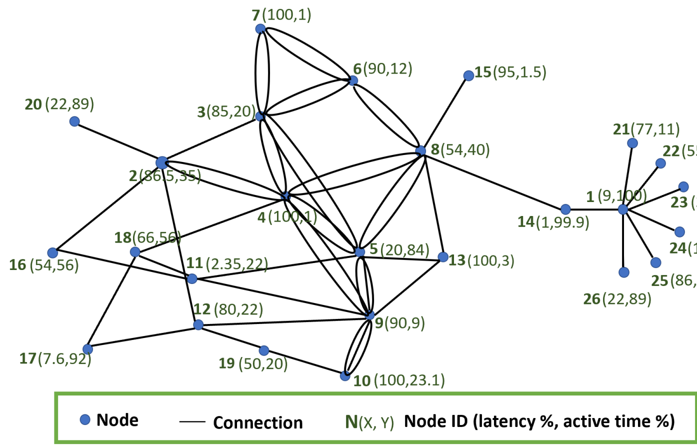


Figure 6.17. Simulated blockchain network.

Hence, our weighted L-H index algorithm is more effective in ranking influential nodes as it takes all node parameters such as degree, latency, active time of a node and its neighbors into consideration while ranking. Also, our weighted L-H algorithm achieves higher possibilities for efficient nodes with all desirable parameters to be chosen as influential nodes, thereby helping in efficient and prompt transaction publishing in the blockchain network.

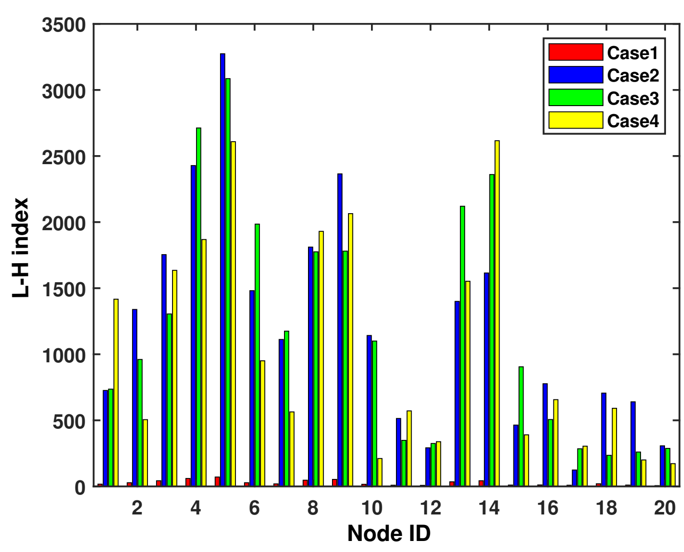


Figure 6.18. L-H index values for all scenarios.

7. CONCLUSION

In this thesis, we present a novel and reliable blockchain-based e-commerce model aimed at enhancing trust, transparency, and credibility during asset trading. Leveraging blockchain technology ensures integrity among participants and fosters trust in sellers. Our persuasive simulation results demonstrate the effectiveness of our XGBoost algorithm in accurately identifying loyal participants within the network. We incorporate Non-Fungible Token (NFT) technology to tokenize intangible assets, enabling dual ownership, refined access control, and improved efficiency in sharing NFTs among multiple parties. Additionally, we develop a Stackelberg game theory approach to determine optimal strategies for both sellers and buyers, ensuring the maximization of benefits for all parties involved. Further, we modeled the determination of popular data fields as a constrained Combinatorial Multi-Armed Bandit (CMAB) problem with unknown popularity levels, which was then resolved by our algorithm Popularity-based Adaptive NFT Management Scheme (PANMS). The simulation results show that our algorithm works better than the benchmark algorithms. To transfer ownership rights of both tangible and intangible assets, sellers have the option to efficiently and promptly sell their assets through an auction employing a first-price sealed-bid mechanism when there is significant interest from multiple potential buyers. Alternatively, the sellers may choose to wait for direct communication from potential buyers to sell their assets if there are only a few interested parties.

By employing validators for secondhand tangible assets' ownership transfer sequence, the quality of products is ensured. We designed an incentive mechanism using the Stackelberg game mechanism to motivate the validators to honestly validate products and determine the most efficient strategy for the platform. We also proposed a Conflict Resolution Scheme (CRS) algorithm to ensure fairness in resolving conflicts between sellers and validators regarding the description of the product's quality. Our simulation results show various parameters affecting the players' strategies and ensure that the strategies of validators and buyers are the optimal strategies. To enhance network efficiency, we implement a weighted L-H algorithm for identifying influential nodes and prioritizing transactions, along with a dynamic block creation policy. Our simulation results highlight the efficiency of our algorithm in accurately ranking nodes based on parameters such as node degree, latency, and active time, compared to alternative methods.

REFERENCES

[1] Statisca. "Key figures of e-commerce." (), [Online]. Available: https://www.statista.com/statistics/183755/number-of-us-internet-shoppers-since-2009/.

[2] T. Walk-Morris, *Secondhand US market to reach* 82*B by* 2026*:thredup*, RetailDive, 2022. [Online]. Available: https://www.retaildive.com/news/secondhand-market-reach-82b-2026/-thredup/.

[3] I. E. Okonkwo, "Nft, copyright and intellectual property commercialization," *International Journal of Law and Information Technology*, vol. 29, no. 4, pp. 296–304, 2021.

[4] A. A. Monrat, O. Schelén, and K. Andersson, "A survey of blockchain from the perspectives of applications, challenges, and opportunities," *IEEE Access*, vol. 7, pp. 117 134–117 151, 2019.

[5] Z. Zheng, S. Xie, H.-N. Dai, X. Chen, and H. Wang, "Blockchain challenges and opportunities: A survey," *International journal of web and grid services*, vol. 14, no. 4, pp. 352–375, 2018.

[6] Q. Wang, R. Li, Q. Wang, and S. Chen, "Non-fungible token (nft): Overview, evaluation, opportunities and challenges," *arXiv preprint arXiv:2105.07447*, 2021.

[7] L. Ante, "The non-fungible token (nft) market and its relationship with bitcoin and ethereum," *FinTech*, vol. 1, no. 3, pp. 216–224, 2022.

[8] K. O. Asamoah, H. Xia, S. Amofa, *et al.*, "Zero-chain: A blockchain-based identity for digital city operating system," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 10 336–10 346, 2020. DOI: 10.1109/JIOT.2020.2986367.

[9] L. A. Linn and M. B. Koo, "Blockchain for health data and its potential use in health IT and health care related research," in *ONC/NIST Use of Blockchain for Healthcare and Research Workshop. Gaithersburg, Maryland, United States: ONC/NIST*, 2016.

[10] P. K. Sharma, N. Kumar, and J. H. Park, "Blockchain-based distributed framework for automotive industry in a smart city," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 7, pp. 4197–4205, 2019. DOI: 10.1109/TII.2018.2887101.

[11] W. Viriyasitavat, L. Da Xu, Z. Bi, and A. Sapsomboon, "New blockchain-based architecture for service interoperations in internet of things," *IEEE Transactions on Computational Social Systems*, vol. 6, no. 4, pp. 739–748, 2019.

[12] J. Guo, X. Ding, and W. Wu, "A blockchain-enabled ecosystem for distributed electricity trading in smart city," *IEEE Internet of Things Journal*, 2020. DOI: 10.1109/JIOT.2020.3015980.

[13] Y. Tan, "Implications of blockchain-powered marketplace of preowned virtual goods," *Production and Operations Management*, 2022.

[14] L. Medury and S. Ghosh, "Design and analysis of blockchain-based resale marketplace," in *IOT with Smart Systems*, Springer, 2022, pp. 481–490.

[15] Y. Yu, C. Yao, Y. Zhang, and R. Jiang, "Second-hand car trading framework based on blockchain in cloud service environment," in *2021 2nd Asia Conference on Computers and Communications (ACCC)*, 2021, pp. 115–121. DOI: 10.1109/ACCC54619.2021.00026.

[16] H. Treiblmaier and C. Sillaber, "The impact of blockchain on e-commerce: A framework for salient research topics," *Electronic Commerce Research and Applications*, vol. 48, p. 101 054, 2021.

[17] H. P. Bulsara and P. S. Vaghela, "Blockchain technology for e-commerce industry," *International Journal of Advanced Science and Technology*, vol. 29, no. 5, pp. 3793–3798, 2020.

[18] X. Zhu and D. Wang, "Research on blockchain application for e-commerce, finance and energy," in *IOP Conference Series: Earth and Environmental Science*, IOP Publishing, vol. 252, 2019, p. 042 126.

[19] A. Kulkarni. "How resale items can get a second life with blockchain verification." (2018), [Online]. Available: https://medium.com/@ajitkulkarni/how-resale-items-can-get-a-second-life-with-blockchain-verification-63748302bf68.

[20] J. Herinckx and R. Ghislain, "The use of blockchain to fight counterfeiting in the second-hand luxury fashion market," 2022. [Online]. Available: http://hdl.handle.net/2078.1/thesis:35381.

[21] A. Nigam, S. Sangal, A. Behl, *et al.*, "Blockchain as a resource for building trust in pre-owned goods marketing: A case of automobile industry in an emerging economy," *Journal of Strategic Marketing*, pp. 1–19, 2022.

[22] B. Shen, X. Xu, and Q. Yuan, "Selling secondhand products through an online platform with blockchain," *Transportation Research Part E: Logistics and Transportation Review*, vol. 142, p. 102 066, 2020.

[23] Y.-W. Chang, K.-P. Lin, and C.-Y. Shen, "Blockchain technology for e-marketplace," in *2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, IEEE, 2019, pp. 429–430.

[24] H. Subramanian, "Decentralized blockchain-based electronic marketplaces," *Communications of the ACM*, vol. 61, no. 1, pp. 78–84, 2017.

[25] O. R. Kabi and V. N. Franqueira, "Blockchain-based distributed marketplace," in *International Conference on Business Information Systems*, Springer, 2018, pp. 197–210.

[26] N. M. Denter, F. Seeger, and M. G. Moehrle, "How can blockchain technology support patent management? a systematic literature review," *International Journal of Information Management*, p. 102 506, 2022.

[27] S. M. H. Bamakan, A. B. Bondarti, P. B. Bondarti, and Q. Qu, "Blockchain technology forecasting by patent analytics and text mining," *Blockchain: Research and Applications*, vol. 2, no. 2, p. 100 019, 2021.

[28] D. V. Dimitrov, "Blockchain applications for healthcare data management," *Healthcare Informatics Research*, vol. 25, no. 1, 2019.

[29] S. Angraal, H. M. Krumholz, and W. L. Schulz, "Blockchain technology: Applications in health care," *Circulation: Cardiovascular Quality and Outcomes*, vol. 10, no. 9, 2017.

[30] A. A. Siyal, A. Z. Junejo, M. Zawish, K. Ahmed, A. Khalil, and G. Soursou, "Applications of blockchain technology in medicine and healthcare: Challenges and future perspectives," *Cryptography*, vol. 3, no. 1, 2019.

[31] I. Radanovi and R. Liki, "Opportunities for use of blockchain technology in medicine," *Applied Health Economics and Health Policy*, vol. 16, no. 5, 2018.

[32] A. Kulkarni, *How resale items can get a second life with blockchain verification, (2018)*. [Online]. Available: https://medium.com/@ajitkulkarni/how-resale-items-can-get-a-second/-life-with%20-blockchain-verification-%2063748302bf68.

[33] J. Herinckx and R. Ghislain, “The use of blockchain to fight counterfeiting in the second-hand luxury fashion market,” 2022. [Online]. Available: http://hdl.handle.net/2078.1/thesis:35381.

[34] A. Nigam, S. Sangal, A. Behl, *et al.*, “Blockchain as a resource for building trust in pre-owned goods marketing: A case of automobile industry in an emerging economy,” *Journal of Strategic Marketing*, pp. 1–19, 2022.

[35] B. Shen, X. Xu, and Q. Yuan, “Selling secondhand products through an online platform with blockchain,” *Transportation Research Part E: Logistics and Transportation Review*, vol. 142, p. 102 066, 2020.

[36] L. Medury and S. Ghosh, “Design and analysis of blockchain-based resale marketplace,” in *IoT with Smart Systems*, Springer, 2022, pp. 481–490.

[37] H. Subramanian, “Decentralized blockchain-based electronic marketplaces,” *Communications of the ACM*, vol. 61, no. 1, pp. 78–84, 2017.

[38] Y.-W. Chang, K.-P. Lin, and C.-Y. Shen, “Blockchain technology for e-marketplace,” in *2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, IEEE, 2019, pp. 429–430.

[39] O. R. Kabi and V. N. Franqueira, “Blockchain-based distributed marketplace,” in *International Conference on Business Information Systems*, Springer, 2018, pp. 197–210.

[40] S. Lawrenz, P. Sharma, and A. Rausch, “Blockchain technology as an approach for data marketplaces,” in *Proceedings of the 2019 International Conference on Blockchain Technology*, 2019, pp. 55–59.

[41] S. Lawrenz, P. Sharma, and A. Rausch, “Blockchain technology as an approach for data marketplaces,” in *Proceedings of the 2019 International Conference on Blockchain Technology*, 2019, pp. 55–59.

[42] V. P. Ranganthan, R. Dantu, A. Paul, P. Mears, and K. Morozov, “A decentralized marketplace application on the ethereum blockchain,” in *2018 IEEE 4th International Conference on Collaboration and Internet Computing (CIC)*, IEEE, 2018, pp. 90–97.

[43] M. Klems, J. Eberhardt, S. Tai, S. Härtlein, S. Buchholz, and A. Tidjani, "Trustless intermediation in blockchain-based decentralized service marketplaces," in *International Conference on Service-Oriented Computing*, Springer, 2017, pp. 731–739.

[44] S. M. H. Bamakan, N. Nezhadsistani, O. Bodaghi, and Q. Qu, "Patents and intellectual property assets as non-fungible tokens; key technologies and challenges," *Scientific Reports*, vol. 12, no. 1, pp. 1–13, 2022.

[45] W. Du, Y. Wang, W. Xu, and J. Ma, "A personalized recommendation system for high-quality patent trading by leveraging hybrid patent analysis," *Scientometrics*, vol. 126, no. 12, pp. 9369–9391, 2021.

[46] S. Jiang, J. Cao, H. Wu, Y. Yang, M. Ma, and J. He, "Blochie: A blockchain-based platform for healthcare information exchange," in *IEEE International Conference on Smart Computing*, (2018).

[47] R. Kumar, N. Marchang, and R. Tripathi, "Distributed off-chain storage of patient diagnostic reports in healthcare system using IPFS and blockchain," in *International Conference on COMmunication Systems & NETworkS (COMSNETS)*, (2020).

[48] Q. Xia, E. B. Sifah, K. O. Asamoah, J. Gao, X. Du, and M. Guizani, "Medshare: Trust-less medical data sharing among cloud service providers via blockchain," *IEEE Access*, vol. 5, 2017.

[49] C. Esposito, A. De Santis, G. Tortora, H. Chang, and K. K. R. Choo, "Blockchain: A panacea for healthcare cloud-based data security and privacy?" *IEEE Cloud Computing*, vol. 5, no. 1, 2018. DOI: 10.1109/MCC.2018.011791712.

[50] A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman, "Medrec: Using blockchain for medical data access and permission management," in *2nd International Conference on Open and Big Data (OBD)*, (2016). DOI: 10.1109/OBD.2016.11.

[51] M. Li, S. Yu, Y. Zheng, K. Ren, and W. Lou, "Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 1, 2013. DOI: 10.1109/TPDS.2012.97.

[52] A. S. Omar and O. Basir, "Secure anti-counterfeiting pharmaceuticals supply chain system using composable non-fungible tokens," in *Blockchain for Cybersecurity and Privacy*, (2020).

[53] Q. Xia, E. B. Sifah, K. O. Asamoah, J. Gao, X. Du, and M. Guizani, "Medshare: Trust-less medical data sharing among cloud service providers via blockchain," *IEEE Access*, vol. 5, 2017.

[54] C. Esposito, A. De Santis, G. Tortora, H. Chang, and K. K. R. Choo, "Blockchain: A panacea for healthcare cloud-based data security and privacy?" *IEEE Cloud Computing*, vol. 5, no. 1, 2018. DOI: 10.1109/MCC.2018.011791712.

[55] A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman, "Medrec: Using blockchain for medical data access and permission management," in *2nd International Conference on Open and Big Data (OBD)*, (2016). DOI: 10.1109/OBD.2016.11.

[56] M. Li, S. Yu, Y. Zheng, K. Ren, and W. Lou, "Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 1, 2013. DOI: 10.1109/TPDS.2012.97.

[57] Z. Liu and Z. Li, "A blockchain-based framework of cross-border e-commerce supply chain," *International Journal of Information Management*, vol. 52, p. 102 059, 2020.

[58] Y.-W. Chang, K.-P. Lin, and C.-Y. Shen, *Blockchain technology for e-marketplace*, 2019. DOI: 10.1109/PERCOMW.2019.8730733.

[59] T. Bocek, B. B. Rodrigues, T. Strasser, and B. Stiller, "Blockchains everywhere-a use-case of blockchains in the pharma supply-chain," in *2017 IFIP/IEEE symposium on integrated network and service management (IM)*, IEEE, 2017, pp. 772–777.

[60] G. Khusainova, "The secondhand market is growing rapidly, can challengers like vinokilo thrive and scale?.," 2021. [Online]. Available: https://www.forbes.com/sites/gulnazkhusainova/2021/01/28/the-secondhand-market-is-growing-rapidly-can-challengers-like-vinokilo-thrive-and-scale/?sh=6efd86a7ccb6.

[61] E. Hur, "Rebirth fashion: Secondhand clothing consumption values and perceived risks," *Journal of Cleaner Production*, vol. 273, p. 122 951, 2020.

[62] C. Lopez and F. Ouattara, *Customer engagement in second-hand fashion marketplaces after the pandemic*, 2021.

[63] M. D. Alam, *Factors that influence the decision when buying second-hand products.* 2015.

[64] X. Lou, T. Chi, J. Janke, and G. Desch, "How do perceived value and risk affect purchase intention toward second-hand luxury goods? an empirical study of us consumers," *Sustainability*, vol. 14, no. 18, p. 11 730, 2022.

[65] A. Kulkarni. "How resale items can get a second life with blockchain verification." (2018), [Online]. Available: https://medium.com/@ajitkulkarni/how-resale-items-can-get-a-second-life-with-blockchain-verification-63748302bf68.

[66] J. Herinckx and R. Ghislain, "The use of blockchain to fight counterfeiting in the second-hand luxury fashion market," 2022. [Online]. Available: http://hdl.handle.net/2078.1/thesis:35381.

[67] A. Nigam, S. Sangal, A. Behl, *et al.*, "Blockchain as a resource for building trust in pre-owned goods marketing: A case of automobile industry in an emerging economy," *Journal of Strategic Marketing*, pp. 1–19, 2022.

[68] B. Shen, X. Xu, and Q. Yuan, "Selling secondhand products through an online platform with blockchain," *Transportation Research Part E: Logistics and Transportation Review*, vol. 142, p. 102 066, 2020.

[69] L. Medury and S. Ghosh, "Design and analysis of blockchain-based resale marketplace," in *IOT with Smart Systems*, Springer, 2022, pp. 481–490.

[70] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785–794.

[71] W. Li, C. Feng, L. Zhang, H. Xu, B. Cao, and M. A. Imran, "A scalable multi-layer pbft consensus for blockchain," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 5, pp. 1146–1160, 2020.

[72] M. Yin, D. Malkhi, M. K. Reiter, G. G. Gueta, and I. Abraham, "Hotstuff: BFT consensus with linearity and responsiveness," in *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing*, 2019, pp. 347–356.

[73] R. S. Sutton, A. G. Barto, *et al.*, "Reinforcement learning," *Journal of Cognitive Neuroscience*, vol. 11, no. 1, pp. 126–134, 1999.

[74] C. Xu, K. Zhu, C. Yi, and R. Wang, "Data pricing for blockchain-based car sharing: A Stackelberg game approach," in *GLOBECOM 2020 - IEEE Global Communications Conference*, (2020). DOI: 10.1109/GLOBECOM42002.2020.9322221.

[75] C. Wang, Q. Hu, D. Yu, and X. Cheng, "Proactive deployment of chain-based VNF backup at the edge using online bandit learning," in *IEEE 41st International Conference on Distributed Computing Systems (ICDCS)*, (2021).

[76] C. Wang, Q. Hu, D. Yu, and X. Cheng, "Proactive deployment of chain-based VNF backup at the edge using online bandit learning," in *IEEE 41st International Conference on Distributed Computing Systems (ICDCS)*, (2021).

[77] C. Liu, R. Du, S. Wang, and R. Bie, "Cooperative stackelberg game based optimal allocation and pricing mechanism in crowdsensing," *International Journal of Sensor Networks*, vol. 28, no. 1, pp. 57–68, 2018.

[78] Q. Liu, Y.-X. Zhu, Y. Jia, *et al.*, "Leveraging local h-index to identify and rank influential spreaders in networks," *Physica A: Statistical Mechanics and its Applications*, vol. 512, pp. 379–391, 2018.