COMBINATORIAL SOLUTIONS FOR CONSENSUS ALGORITHMS AND BLOCKCHAIN SHARDING

MUTABAKAT ALGORİTMALARI VE BLOKZİNCİRİ PARÇALANMASI İÇİN KOMBİNATORYAL ÇÖZÜMLER

MARWAN JAMEEL

ASSOC. Prof. Dr. İSMET YURDUŞEN

Supervisor

ASSOC. Prof. Dr. OĞUZ YAYLA

Co-Supervisor

Submitted to

Graduate School of Science and Engineering of Hacettepe University

as a Partial Fulfillment to the Requirements

for the Award of the degree of Doctor of Philosophy

in Mathematics.

2021

To my dear teacher Oğuz Yayla

ABSTRACT

COMBINATORIAL SOLUTIONS FOR CONSENSUS ALGORITHMS AND BLOCKCHAIN SHARDING

Marwan JAMEEL

Doctor of Philosophy, Department of Mathematics

Supervisor: Assoc. Prof. Dr. İsmet YURDUŞEN

Co-Supervisor: Assoc. Prof. Dr. Oğuz YAYLA

December 2021, 75 pages

The scalability problem in blockchain technology seems to be the essential issue to be solved. It is known that the choice of a compromised algorithm is critical for the practical solution of this important problem. Usually, Byzantine Fault Tolerance (BFT) methods based on the public blockchain networks have been most widely applied to solve scalability.

In this thesis, we formulate two possible cases to scale the blockchain. Instead of the frequently used proof-of-work or stake methods to form the consensus committee, allowing BFT-based methods, we propose a new model. This new model calculates the reputation value for the nodes that want to join the leader (trust) committee using particle swarm optimization (PSO). It is a computational method for optimizing a problem by improving a candidate solution against a specified quality metric. It solves the problem by populating the search space, so-called particles and moving these particles around according to a simple mathematical formula over the particle's position and velocity. To discard the misbehaving nodes from the trust leader committee, new nodes with high reputation values are selected. Since this study focuses on creating the consensus committee, a simulation test the proposed model more effectively. The results show that the proposed model successfully selects the nodes with high confidence to the consensus committee instead of the malicious nodes. To

select an updated trustworthy committee and then allow all network users to join at any time to protect the blockchain network's security is in general insufficient. However, suspicious nodes must be avoided at all costs. We utilize a straightforward strategy inspired by bio-dynamic systems to deflect the trust committee's focus from the assaulting nodes. Removing poorly-tailored nodes increases the selection of honest nodes or participants. We propose an unsupervised machine learning to solve the current challenge by applying a Grey Wolf Optimization (GWO) technique.

In addition, blockchain studies have recently been splitting the blockchain to address the scalability problem focused on sharding. Sharding is a helpful technique for exploring fundamental computational challenges in blockchain technology, such as consensus, Byzantine fault tolerance, and self-stabilization. The sharding method creates a small, segmented blockchain network. Rather than creating a more extensive network, networks with fewer nodes are established. Additionally, successful sharding can be applied to various areas, resulting in significantly speedier processes. Our solution will give a safe and dependable use of blockchain components by analyzing the system and fitting the shard size using the Topological Data Analysis (TDA) with the help of an unsupervised machine learning technique. In order to achieve our goal, the Linear Programming Problem (LPP) is constructed and solved using the Dual-Simplex approach to determine the best shard size. Additionally, we segmented the blockchain network using our system. The test results show that reputation values boosted the parties' reliability. Then the likelihood of any piece collapsing and harming the entire blockchain decreases.

Keywords: Blockchain, Scalability, Consensus Protocol, Sharding, Byzantine Fault Tolerance (BFT), Machine Learning, Particle Swarm Optimization (PSO), Grey Wolf Optimizer (GWO), Topological Data Analysis (TDA), Betti number, Simplicial Complex, Linear Programming Problem (LPP), Dual-simplex optimization method.

ÖZET

MUTABAKAT ALGORİTMALARI VE BLOKZİNCİRİ PARÇALANMASI İÇİN KOMBİNATORYAL ÇÖZÜMLER

Marwan JAMEEL

Doktora, Matematik Bölümü

Tez Danışmanı: Doç. Dr. İsmet YURDUŞEN

Eş Danışmanı: Doç. Dr. Oğuz YAYLA

Aralık 2021, 75 sayfa

Blokzinciri teknolojisindeki en büyük zorluklardan biri ölçeklenebilirlik sorunudur. Ölçeklenebilirlik probleminin pratik çözümü için kritik öneme sahiptir. Ölçeklenebilirliği artırmak için, blokzinciri uygulamalarına katılan uzmanlar, teknolojinin sınırlamalarını belirtebilir. Byzantine Fault Tolerance (BFT) tabanlı yöntemler en yaygın olarak genel blokzincir ağlarına dayalı olarak uygulanmıştır. Ölçeklendirme için iki olası durum çalışılır.

Konsensüs komitesini oluşturmak için sıklıkla kullanılan Proof of (Work, stake,...) yöntemleri yerine, BFT tabanlı yöntemlerin kullanımına izin vererek yeni bir model değerlendirmesi önerilmiştir. Bu model, particle swarm optimizasyonunu (PSO) kullanarak lider komiteye katılmak isteyen düğümler için itibar değerini hesaplar. Bu, belirli bir kalite metriğine göre bir aday çözümü geliştirerek bir sorunu optimize etmeye yönelik bir hesaplama yöntemidir. Arama uzayını parçacık denilen olası çözümlerle doldurarak ve bu parçacıkları parçacığın konumu ve hızı üzerinde basit bir matematiksel formüle göre hareket ettirerek çözer. Güven Lideri Komitesindeki düğümlerin zararlı olma olasılığını azaltmak için, komite için yüksek itibar değerlerine sahip uzlaşma düğümleri seçilir. Bu çalışma fikir birliği komitesi oluşturmaya odaklandığından, önerilen modeli daha etkin bir şekilde test etmek için python'da simülasyon kullanılmıştır. Test sonuçları, önerilen modelin, kötü niyetli düğümün varlığında fikir birliği komitesnin yüksek güven ile düğümleri başarıyla seçtiğini göstermektedir.

Güncellenmiş güvenilir bir komite seçip, komiteyi güncelleme ve ardından blokzinciri ağının güvenliğini korumak için tüm ağ kullanıcılarının herhangi bir zamandaki katılımına izin vermek hedeflerini karşılamak genel olarak yetersizdir. Şüpheli düğümlerden ne pahasına olursa olsun kaçınılmalıdır. Liderlik komitesinin odağını saldıran düğümlerden saptırmak için biyo-dinamik sistemlerden ilham alan basit bir strateji kullanıyoruz. Yıkıcı düğümleri tespit etmek için elde edilen yöntemden kötü uyarlanmış düğümleri kaldırmak, dürüst düğümlerin veya katılımcıların seçimini de artırır. Grey Wolf Optimizasyonu (GWO) tekniği uygulayarak mevcut sorunu çözmek için denetimsiz bir makine öğrenimi öneriyoruz.
Ayrıca, blokzinciri çalışmaları son zamanlarda parçalamaya odaklanan ölçeklenebilirlik sorununu ele almak için blokzincirini bölüyor.
Sharding, blokzinciri teknolojisindeki fikir birliği, Bizans hata toleransı ve kendi kendini dengeleme gibi temel hesaplama zorluklarını araştırmak için yararlı bir tekniktir. Parçalama yöntemi, küçük, bölümlere ayrılmış bir blokzinciri ağı oluşturur. Daha kapsamlı bir ağ oluşturmak yerine, daha az düğümlü ağlar kurulur. Ek olarak, başarılı bir parçalama çeşitli alanlara uygulanabilir ve bu da önemli ölçüde daha hızlı işlemlerle sonuçlanır. Ölçeklenebilirlik çözümümüz, denetimsiz bir makine öğrenimi tekniği yardımıyla Topological Data Analysis (TDA) kullanarak sistemi analiz ederek ve parça boyutunu bu süreç için uygun hale getirerek blokzinciri bileşenlerinin güvenli ve güvenilir kullanımına katkıda bulunacaktır. Doğrusal Programlama Problemi (LPP), en iyi parça boyutunu belirlemek için Dual-Simplex yaklaşımı kullanılarak oluşturulur ve çözülür. Ek olarak, sistemimizi kullanarak blokzinciri ağını bölümlere ayırdık. Test bulguları, itibar değerlerinin eklenmesinin parçaların güvenilirliğini artırdığını göstermektedir. Daha sonra herhangi bir parçanın çökme ve tüm blokzincirine zarar verme olasılığı azalır.

Anahtar Kelimeler: Blokzinciri, Ölçeklenebilirlik, Mutabakat Protokolü, Parçalama, Byzantine Fault Tolerance (BFT), Makine Öğrenimi, Particle Swarm Optimization (PSO), Grey Wolf Optimizer (GWO), Topological Data Analysis (TDA), Betti number, Simplicial Complex, Linear Programming Problem (LPP), Dual-simplex optimization method.

ACKNOWLEDGEMENT

First of all, I would like to express my sincere warmest thanks to my supervisors Assoc. Prof. Dr. İsmet Yurduşen and Assoc. Prof. Dr. Oğuz Yayla for their invaluable suggestions, motivation and patience. My particular gratitude goes to Dr. Yayla for his continuous support not only in this thesis but also in my life. His guidance helped me in all the time of research and writing of this thesis.

Besides my supervisor, I would like to thank the rest of my thesis committee members: Prof. Dr. Seçkin Kürkçüoüğlu , Assoc. Prof. Dr. Emre Taşcı, Assoc. Prof. Dr. Zülfükar Saygı, Assist. Prof. Dr. Turgut Hanoymak, and Assoc. Prof. Dr. Mesut Şahin, for their encouragement and insightful comments.

I would like to my special thank to my dear teachers in the Department of Mathematics and the TÖMER at Hacettepe University for their support.

I would like to express my thanks to the YTP program for allowing me to study in my second home Turkey and for supporting me during this study.

I must express my deepest gratitude to my parents for supporting me throughout my life.

Finally, thanks to Allah, for letting me through all the difficulties.

Marwan Jameel

December 2021, Ankara

Contents

List of Figures

List of Tables

NOTATIONS AND ACRONYMS

Notations

$\mathbb{Z}$	Integers
$\mathbb{R}$	Real numbers
$\mathbb{R}^d$	Euclidean space dimension d
β_i	Betti number of dimension i
SC	Simplicial Complex
k-SC	k-dimensional Simplicial Complex
α	Fittest Grey wolf in pack
β	Second fittest Grey wolf in pack
γ	Third fittest Grey wolf in pack
δ	Rest of Grey wolf in pack

Acronyms

BC	Blockchain
ID	Identity
BFT	Byzantine Fault Tolerance
PBFT	Practical Byzantine Fault Tolerance
PoW	Proof of Work
PoS	Proof of Stake
VRF	Verifiable Random Functions
TC	Trust Committee
TDA	Topological data analysis
PH	Persistent homology
SC	Simplicial Complex
PSO	Particle Swarm Optimizatio
GWO	Grey Wolf Optimizer
SGWO	Security Grey Wolf Optimizer
LPP	Linear Programming Problem
DSM	Dual-simplex method
PC	Personal Computer
AI	Artificial Intelligence
DAG	Directed Acyclic Graph

1 INTRODUCTION

This thesis investigates the scaling problem of blockchain data structure by focusing mainly on consensus and sharding methods. We have done this through our attempt to solve a problem in the blockchain literature. Then we add our contributions for optimum consensus and sharding applications throughout the thesis structure.

One of the recent studies in this area is done by Buğday et. al. [8]. It is mentioned that blockchain technology began to be paid more attention in recent years because of the increasing interest in cryptocurrencies and their working principle. Blockchain can be described as a data structure that contains unchangeable public records, and its validation requires a consensus throughout the nodes in the blockchain network itself [9]. In other words, rather than being validated by an external agent, the validity of the blocks requires a consensus of the nodes in the blockchain network itself. However, this kind of validation brings some problems with it. For example, the high number of transactions in limited block size can create a problem about scalability because it might cause complexity in the validation algorithm.

Until now, Proof-of-Work (PoW) method [10] have been used and mentioned as the most favorable consensus algorithm to tackle with the scalability problem [8, 11] and the PoW consensus algorithm that is used in the blockchain data is called as Hashcash [12]. PoW is also known as one of the most robust method against network hijacking (Sybil) [13] with multiple identities [14, 15]. It is used in different applications such as Bitcoin [9] and Ethereum [16]). However, these applications require a very high energy consumption or long settlement times. Thus, up to now, there is no single or exact solution to the blockchain consensus problem and more work needs to be done to find the optimum consensus and scalability solution.

Byzantine Fault Tolerant (BFT) based methods are also widely used alternative to PoW to solve the scalability problem in addition to prevent the system's impairment against malicious nodes and establish its security [17]. In comparison to PoW methods, arriving at a consensus in BFT-based methods is less time consuming and cost effective. In this method, reaching a consensus requires a selection of subset of the nodes and creating a consensus committee to prevent the performance problem of the blockchain [15]. Creating a consensus committee is especially important because beside the performance problem, the reliability of all the nodes in the public blockchain is usually uncertain. Thus, in the BFT-based methods, the first step is creating a consensus committee [18].

Since BFT-based approaches use a subset of nodes in the network, the criteria for subset selection emerges as an important problem to deal with [8], [19], [20]. In [20] we used a Particle Swarm Optimization (PSO) based on blockchain member selection to overcome this problem. The detailed analysis of this study will be discussed in Chapter 3. On the other hand, the similar problem was also discussed by the authors Buğday et. al [8] where they created a consensus committee by choosing the nodes with high reputation based on their feature values in the system using blockchain simulation environment. They did this through nine different testing and arrived at an agreement about the criteria to choose the reliable nodes for the consensus committee which could measure the behavior change effectively.
Another widely used method to deal with the scaling problem is sharding of blockchain [8]. This method uses dividing the whole chain into different parts in order to facilitate the management of the parts and the response to requests easier [21, 22]. Blockchain sharding not only achieves improvement in the performance; it also reduces the workload on the database. By sharding we expect the following benefits:

1. Each part is running its own blockchain shard so that a rise in the number of transactions is ensured.

2. Nodes communicate with only their own network nodes so that the network efficiency increases.

3. Using and downloading only the necessary section of the blocks will result storage utilization.

When the sharding is used, since the nodes in the network are fragmented, the shards are to be arranged large enough for increasing the reliability and more attention should be paid to the selection of the nodes. To deal with this problem, researchers [23, 24] and Algorand [25] used orientation estimation resistant randomness (Verifiable Random Functions (VRF) [26]) technique in the assignment process. Unlike other methods, the assignment of nodes to parts is performed initially using the confidence value obtained in this study [27], after which the appropriate number of shards are determined. In this way the number of faulty nodes in the parts is expected to be kept in the minimum.
However, to date, no study suggests an optimal number of shards. Thus, in this study, we aim to find the optimum number of partitions. Focusing on the scaling problem in the blockchain, we realized that the main cause of the problem is associated with complex and time consum-

ing character of the consensus algorithm. After literature review, we have noticed that two important methods, namely the BFT-based consensus algorithms and the sharding method are used in an efficient way to solve this problem. However, in the process of adapting these methods some new problems start to arise such as the selection of different consensus committees and the assignment of nodes to shards. For example, an increase in the number of joined participants and the possible changes in the capabilities of participants in the future give birth to the following natural question; what would be the appropriate number of shards that must be chosen. In this study, we propose solutions to these problems. To summarize, we have the following contributions:

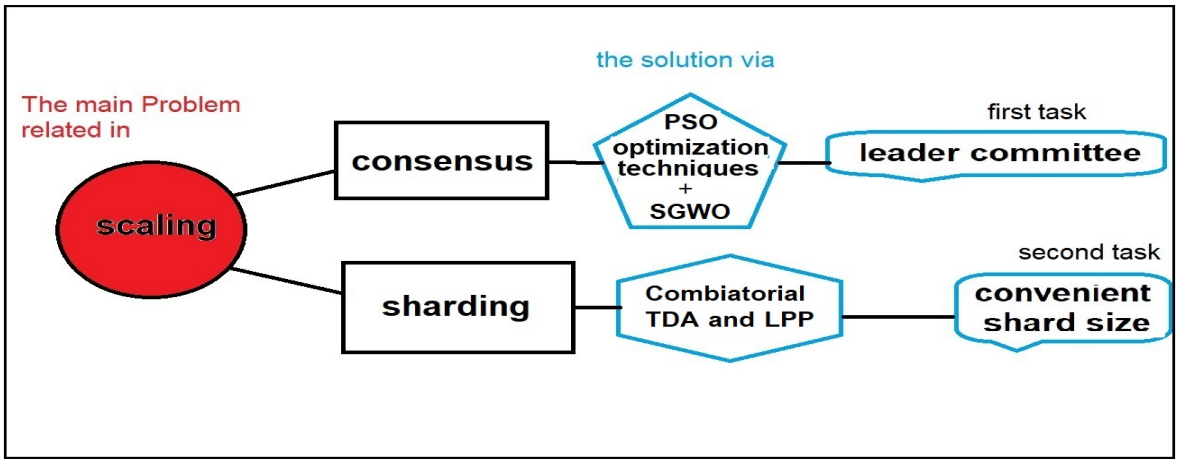

Figure 1.1: A diagram of the applied division of research as main ideas methodology.

- A general approach to addressing blockchain challenges via mathematical algorithms.
- We represent nodes in the form of vectors, indicating the quality and strength of the node's blockchain connection.
- Using particle swarm optimization, one of the essential swarm intelligence machine learning processes, in consensus trust committee selection. Encourage nodes to act honestly by assigning trustful values to them in the blockchain network.
- We are using confidence value in assigning nodes to parts.
- We were able to reduce the chance of failure in Trust Committee (TC) by enhancing the security efficiency of the selection committee by applying an GWO algorithm.

- We are implementing topological data analysis to enable combinatorial building for participants in the blockchain.
- Finding the ideal number of shards and the number of nodes in those shards

The organization of this thesis is as follows: Chapter 2 discusses the preliminaries and related previous works. Chapter 3 deals with the suggested model design and algorithm for selecting a consensus trust committee, as well as a brief introduction to the particle swarm optimization (PSO) algorithm. Additionally, it provides information on the experimental work and our consensus committee selection results. Chapter 3 continues by delving into the specifics of the associated combination of the GWO algorithm to the PSO algorithm in order to enhance the security aspect by excluding some attackers. Then, Chapter 4 focuses on implementation of the topological data analysis (TDA) to the sharding scalability problem and the remainder of it describes the possible numerical simulations. Finally, Chapter 5 gives the conclusions and also some discussions for the possible future work.

2 PRELIMINARIES AND RELATED PREVIOUS WORKS

Blockchain is a specific type of basic data; that is, it is a data structure obtained by linking blocks containing transaction information in the form of a table to facilitate searching and filtering for specific information, *e.g.* block creation time and cryptographic hash information of the previous block. A block is divided into two main areas. These are referred as the block

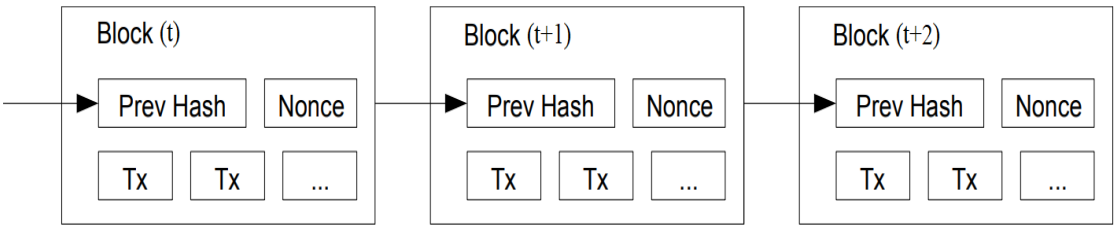

Figure 2.1: Chain of blocks via hashes (Satoshi Nakamoto)

title and the block detail, see Figure 2.1. The information of the former is used for easier verification of the block. The block title is considered metadata that defines the block. It contains basic information such as block hash, previous block hash, Merkle root value and creation time. The definitions of each item are given below:

Definition 2.1. *Peer-to-Peer network is a distributed application architecture that connects nodes, often peers, in an efficient manner to accomplish a common purpose. Peers share resources with or without the assistance of a central authority for administration. In a blockchain system, the nodes independently store the data.*

Definition 2.2. *Block hash is the result of a cryptographic hash algorithm applied to the block.*

Definition 2.3. *Previous block hash is the hash of the previous block. Thanks to it, the consistency of the blockchain is guaranteed to be immutable.*

As shown in Figure 2.1, the previous block hash value goes up to the first block. In this way, changing of the blocks is prevented. In order to change the records in the blockchain, it is necessary to update the previous block's digest information along with the published blocks and quite a lot of processing power is required to do this. Therefore, records in the

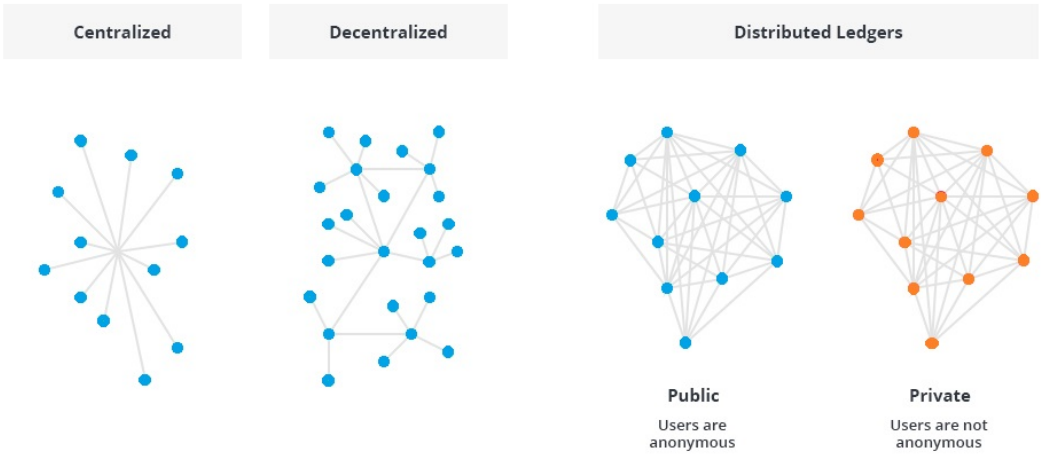

Figure 2.2: Centralized, decentralized, and distributed nodes [1]

blockchain cannot be changed unless the majority of the processing power is concentrated in one place. Transaction information within the block is not kept in the block header. But without needing the whole block, there is a need to verify whether the desired data is in the block.

Definition 2.4. *A Merkle tree is created and the root value is stored in the block header, allowing the desired motion to be reached [28, 29, 30].*

Definition 2.5. *Creation time indicates the block's creation time.*

For a block to be added to the blockchain, the nodes in the network must reach consensus on the block. Depending on the consensus algorithm used, different blocks can be broadcast simultaneously in some cases, for example, when using PoW. Subsequent blocks determine which block and chain are valid. The longest chain is accepted as correct and new blocks are added to this chain. Such a case of different chain formation is called bifurcation. Forking, which is a creation of new chain, can occur in the blockchain for different reasons.

There is no need for a reliable central authority for the verification of transactions in the blockchain network. The nodes in the network decide the correctness of the transactions. While transactions are carried out by connecting to a central place in the usual client-server architecture, in the blockchain, every client suitable for a distributed architecture can directly communicate and transact with each other. Figure 2.2 shows example network structures. In classical databases, every data can be kept on centralized or decentralized servers, while in

blockchain, a copy of all information is found at all nodes in the network. Even if the data in one node is corrupted, it can recover information from other nodes.
There are basically two different types of blockchain. These are public blockchains and private blockchains.

Definition 2.6. *Public Blockchain: In public blockchain networks, any user who wants to join the network can join the network without permission. As long as the nodes in the network comply with the rules without any restrictions, they can perform all operations in the network such as accessing historical records, creating a new record, block validation. Cryptocurrencies such as Bitcoin and Ethereum are examples of public blockchains.*

Definition 2.7. *Private Blockchain: In private blockchain networks, joining the network is permission dependent. Private blockchains are usually blockchains created by interrelated organizations. The security level of the blockchain and the rules for joining the blockchain network are determined by the institution that establishes the blockchain. There is a risk of centralization as it is open to more limited users.*

2.1 Blockchain Technology Operations

Proof is some (computational) effort that must be performed; this is referred to as proof-of-{work,stake,activity,storage,...}. As illustrated in Figure 2.1, blocks are connected together. As a result, it is not possible to modify an individual block in the chain without altering the subsequent blocks. If an attacker wishes to modify a block, re-computation of all subsequent blocks is also required. That would be exceedingly tough to accomplish, much more so given the constant addition of new blocks at the chain's conclusion. Only when more than 50% of processing power is owned by honest nodes can a blockchain be called secure. Otherwise, the majority is capable of recreating the entire chain.
After an adequate nonce is discovered mined, it is distributed to the whole network. After receiving a new block, each node verifies the new block. The block is added to the chain if it is accepted. When a chain has numerous branches, the longest one is favored. The blockchain network works according to the stages outlined by Nakamoto [9], which are as follows:
step 1. All nodes will get new transactions.
step 2. New transactions are collected in a block for every node.
step 3. Each node attempts to locate its own block with challenging proofs.

step 4. If the node discovers evidence of work, i.e. Nonce, the block is sent to all nodes.

step 5. Nodes only accept the block if all of its transactions are legitimate and have not been spent.

step 6. Nodes indicate their acceptance of the block by utilizing the hash of the accepted block like the previous hash to create a new block in the chain.

Nodes can go out or join the network free of charge. When a node joins the chain, it takes from all the known nodes the longest chain to build on.

2.1.1 The need for scaling

The Bitcoin network uses a lot of processing power and can handle up to 7 transactions per second [31]. Other blockchain consensus protocols (e.g. Ethereum, Ripple, Tendermint) that are actively used have essential limitations [32]. Regrettably, the capacity of Bitcoin's transaction network does not scale effectively. According to MasterCard and Visa, centralized fiat payment processing systems can process anywhere from 1200 to 56000 transactions per second. Actually the demand which is generated by practical applications is three to four orders of magnitude more [33]. Although the modifications of the existing protocols to increase their scalability are a source of contention in the Bitcoin community [31, 34, 35], the recent analysis demonstrates that those ideas have scalability limitations [36]. A proposed solution to this is to implement a sharding protocol among the identities, a fraction f of which are byzantine. Our objective here is to distribute all identities evenly among multiple committees, with the restriction that the majority of each committee is likely to be honest. If one assumes a shared random coin to appropriately partition the data, then a protocol becomes simple [31, 35].

Blockchains have proven to be an appropriate manner of logging all types of information with several benefits such as transparency, trust, security, or cost, depending on how the design parameters are configured. Table 2.1 provides an overview of areas where the use of blockchain technology has a prospective impact. We noted that while these prospects face various obstacles, significant research is going to be undertaken in each category to overcome those obstacles in the near future.

Sector	Blockchain technology's potential in related	Example
Education	Create a trustworthy database of digital credentials that includes academic certifications, degrees, transcripts, and assessments, as well as tight policies for issuing and revoking credentials	Blockcerts [37]
Health care	Store electronic medical records with biometrics and/or multi-signature access and update rights to simplify the process of recording, distributing, and updating patient data in a safe manner.	HealthWizz [38]
Government	Serve as an intelligent database for personal identifying information, criminal records, and biometric-based e-citizenship.	ID2020 [39]

Table 2.1: Possible blockchain applications

2.2 Blockchain Consensus Protocols

The Bitcoin cryptocurrency was introduced more than a decade ago, the centralized system was replaced by a consensus process based on proof of work [9]. The initial bitcoin technology of blockchain, which is built on a decentralized system, gave a fresh perspective to Chaum's proposal. The blockchain technology behind cryptocurrencies has increased its popularity within the last years [9]. Initially, implemented as employing a public network which allows open participation and data access. Blockchain had some problems, such as scalability, energy-consumption, and security and privacy threats. The public network addressing is used for solving the problem of security and privacy [11]. However, the amount of energy consumption in blockchain consensus protocol is still a challenging problem and Chapter 3 involves our contribution in this direction [20]. On the other hand, blockchain system with voting based consensus protocol reduces the energy consumption. But, this leads to less scalable and decentralized architecture. A better solution might be semi proof-of-work based consensus protocol with less energy consumption and computation power [40].

The blockchain consensus protocol has so many concrete objectives, see Figure 2.3, reaching an agreement, cooperating, providing equal rights to all nodes, and requiring all nodes to participate in the consensus process. As a result, a consensus protocol aims to find a common agreement, which is a win for the entire network [11]. Thus, the blockchain nodes need to

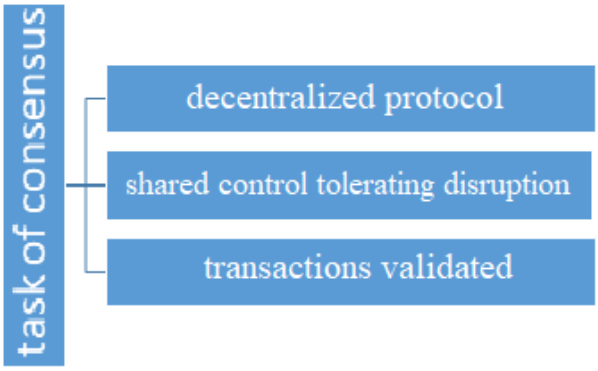

Figure 2.3: Task of consensus protocol in general

reach to a consensus on the validity of the block within the network. Thus, the system does not require clearance from a third party. On the other side, due to the high volume of transactions, the scalability issue arises due to the block size limitation. The proof of work (PoW)-based solution is primarily used to protect against Sybil attacks. There are lots of alternative solutions to the blockchain consensus method. But designing safe and practical consensus algorithms for the blockchain is still a general problem.

The Byzantine Fault Tolerant (BFT) is another popular method for achieving consensus when dealing with scalability issues. The BFT-based technique also secures the blocks' integrity and validity against other parties in the system, according to [41]. In comparison to other consensus methods, BFT-based methods require less time and are less expensive than PoW approaches. In BFT-based approaches, the consensus committee must be chosen first, before a block may be reached. Making this committee in the situation of a big network where the reliability of nodes is unknown is another issue. That all of the parties in the blockchain network can't be selected as a member into the consensus committee, which slows down the network so much. In Chapter 3, we point an alternative solution to this problem based on particle swarm optimization.

There are many different systems for proving that something nontrivial confirms trust. For example computational power, storage space, etc. The most popular and widely used proof systems are discussed in the following paragraphs:

2.2.1 Proof of Work (PoW)

The most well-known consensus algorithm is the PoW algorithm. In this algorithm, a new cryptographic hash value is created by adding a variable to the block. If the resulting cryptographic hash value meets the difficulty level, then the PoW solution has been found. Nodes in the network check this information and add it to their blockchain if it is correct. The higher the difficulty, the more operations are required to find the variable. PoW is a method with very high operation cost, energy requirement and some of them suffer from the long settlement time. Therefore, less costly yet secure and trustworthy consensus approaches are being investigated [14].

2.2.2 Proof of Stake (PoS)

In this method, which was developed as a PoW alternative, it is necessary to have shares (cryptocurrency) in order to verify the block. The basic principle of this method is any people having a large stake are naturally reliable and hence they work to ensure that the system does not break down and works according to the rules. Mining power is proportional to the amount of shares. Proof of stake was first used in [42]. Later, authors in [43, 44, 25] studied the PoS method. In the selection of the node to issue a block, only looking at the amount of shares will turn the system into a monopoly of the person with the most shares, so it requires looking at features such as the age of money, the lowest hash value along with the amount of shares [43]. This consensus method is vulnerable to nothing at stake [45]. In order to prevent this attack, subsequent studies have developed by different methods such as penalizing the node's share.

2.2.3 Byzantine Consensus

Another consensus alternative method is BFT based methods. In these methods, the block to be published is decided by a group rather than an individual. It is assumed that there are nodes that behave incorrectly within the decision-making group, and the proposed method is ensured to work under these conditions. If the number of bad nodes in the group broadcasting the block is more than 1/3 of the number of nodes in the group, the methods will not work correctly. So that the leader node makes a proposal and the other nodes vote on this proposal and it is decided whether the proposal will be accepted or not. The Practical Byzantine consensus (PBFT) method, considering the updated version of BFT, was the first proposed

method for the Byzantine consensus [46]. The performance of the system decreases as the number of nodes increases. Later, methods such as HoneyBadger [47], BFT-Smart [48] have been developed to increase system performance by reducing communication costs.

2.2.4 Blockchain Consensus Protocols Related Significant Works

- Despite its exceptional security and extensive use, PoW is quite expensive in terms of energy and processing time, as indicated in Subsection 2.2.1 and the study with reference [49].
- In order to reduce these costs, PoS has been used instead of PoW in [34, 42]. However, even when nothing is at risk, using PoS alone is insufficient for security [45] and highlighted in Subsection 2.2.2.
- Generally, in the literature, the consensus committee uses a subset of the network rather than all nodes in the system. Numerous studies are [21, 25, 33, 50, 51, 52, 53, 54, 55] as examples of research that validate blocks using methods other than PoW and PoS.
- Elastico [33] proposes the integration of PBFT into an open blockchain network. Nodes wishing to join the committee via this method must present a proof-of-work solution compatible with their credentials. Periods of block verification are used, and the committee is renewed at the conclusion of each time.
- In some studies [25, 51, 52] nodes that want to join the subset prove their knowledge by solving PoW.
- ByzCoin [50] requires nodes interested in joining the committee to solve PoW for a specified period of time. After this period, nodes that solve the PoW are included by voting abilities based on their solutions when the committee is refreshed. Block broadcasting uses PBFT instead of PoW. The PoW is used for node selection to the committee.
- Solidus [52] chooses a committee similar to ByzCoin.Unlike previous work, it incorporates ways to ensure the proper working of committee nodes.

- In research conducted for private blockchain networks, the compromise is chosen statistically because the system's user information is known [53]. Throughout [53], it is assumed that the network is managed centrally by a central bank.

- Numerous studies have been conducted to determine a node's reputation on generic peer-to-peer networks [56, 57, 58]; however, these studies are unrelated to the blockchain area. To select the conciliation committee, Buğday [8] proposes a new model for consensus group formation that enables the use of BFT-based methods rather than PoW on the public blockchain network. The proposed model uses the adaptive hedge approach, a decision-theoretic tool for online learning.

Universal reputation module for distributed consensus protocols (GURU) is the subject of a detailed study which presents a research as an alternative to an entire blockchain process [11]. As a result, more precise findings are achieved when computing the confidence value. Additionally, the feature vector employed in this approach is easily extensible with additional characteristics. While GURU's model lacks a confidence calculation approach, our model employs an unsupervised PSO learning scheme to determine the reputation value of active nodes in order to construct the consensus trust committee [20]. While the learning model is the selection of nodes with behaviour monitors, it is unique in the literature in terms of the transition from honest to malicious and malicious to honest, as opposed to GURU's cumulative approach. The swarm intelligence method is used to determine which nodes are trustworthy and malicious. It differs from [8] and GURU in that it looks for sufficient nodes to add to the committee. It is more resilient, secure, and equitable. Additionally, nodes are incentivized to operate honestly by assigning each node a trust value based on our proposed paradigm. Nodes can operate more prudently while trading with one another by considering the trust value, or smart contracts can impose limits based on the trust value.

2.3 Sharding scalability

The process of dividing a whole into different parts, making the parts easier to manage and faster processing is called sharding. Sharding method has been used for a long time in database management systems to reduce the load on the database and for performance improvements. Also known as horizontal scaling. An example would be dividing the customer table by geographic regions for fragmentation. By creating different databases for

each region, it is ensured that it works without being affected by other regions.
In the blockchain world, a node will only be responsible for keeping the information in the piece it is in. Instead of knowing the whole blockchain, the node will only have knowledge of the parts it is interested in. This will provide scaling in terms of storage and processing load. While the blockchain is being disassembled, the process is favored with dynamic organized, special attention should be paid not to lose its decentralized and secure features, which are the basic principles of blockchain.
Sharding should not be limited to just splitting transactions. If we consider the division of data storage and network communication together with transactions, the blockchain becomes competitive with commercial products. The transactions are divided into different parts and they are processed simultaneously. In this way, the number of transactions processed per unit time increases. Nodes in the main blockchain network are divided into subnets, allowing transactions to be processed in parallel. Each part does its own block validation and publishing. Intra-shard communication is not different from the general blockchain operation, but different methods need to be created for inter-shard communication.
By having every node stop processing every transaction, it theoretically increases the risk that invalid transactions could be logged by a malicious node cluster and remain unchecked. As the number of nodes in the network increases, its ability to divide the network into pieces that can share the load of processing transactions increases, and at the same time, each piece remains large enough for reliable negotiation. Thus each blockchain provides a consensus and publishes blocks. In this way, the processing and storage load is divided. However, the communication of blockchains will be more difficult due to the different protocols. For this reason, it is necessary to establish a blockchain above the blockchain, which will be more difficult than disassembling the blockchain, since it is necessary to gather blockchains with different cultures around the same table and persuade them to work together. Studies on databases can also be a solution to the scalability problem e.g. [59]. There are some problems that need to be solved in order to use sharding on the blockchain.

1. In the fragmentation structure, the transactions are tried to be processed in parallel on different committees. If a transaction is dependent on another transaction in other words, if another transaction must occur for a transaction to be considered true, these transactions should not be run in parallel.

2. Ensuring communication between parts, managing transfers between different parts, a

design made without considering the disruption of a part can affect the whole blockchain.

3. According to the above two points and before starting the sharding procedure, we ask what is the ideal number of shards with the possibility of having common contracts between the shards? The reason is there are nodes close to more than one shard as a working principle and the presence of transactions that require contact with more than one part.

2.4 Related Work about Sharding

Studies that also use the partitioning method to increase the scalability of the blockchain are explained in this paragraph. The OmniLedger [21] study is based on ByzCoin. In OmniLedger [21], nodes in the network are divided into two categories: validators and monitors. OmniLedger offers a scalable, secure distributed ledger structure using blockchain sharding and randomly assigning nodes to shards. It can be viewed as a fork of ByzCoin with the blockchain sharded.

The RandHound [60] method is used to provide distributed randomness against orientations. In order for the processes to be evaluated in parallel, there should be no connection between each other. As a solution to this situation, directional acyclic graph method is used.

Elastico [33] divides the network into small committees and each piece is handled with different transactions. Each committee is made up of a small number of nodes in parallel that can effectively operate the classical Byzantine consensus. Elastico has some assumptions:

1. Honest nodes are directly connected to each other.

2. Communication channels between honest nodes are synchronized.

2.5 More Concepts on Blockchain

Using simply Bitcoin, Ethereum and Ripple, we would not offer a complete picture of the existing blockchain world. Many derivations from the blockchain protocol have evolved with slight or mere basic alterations. Therefore, in addition to these three, we also need to analyze the design decisions of the specified important concepts. We compare them to blockchain and discuss uniqueness of their usual blockchains.

2.5.1 Interfaces and Access

To fit into the existing world of a more centralized internet dominated by numerous companies with established communication protocols for their users, we must devise a method for evaluating various blockchain interface options. As a result, we regard oracles as a mean of communicating data, while algorithms for privacy and intransparency serve as means of communicating identity and access possibilities.

2.5.2 Network Structure

Defining the rules governing how nodes communicate with one another and whether they are equipotent peers has a significant impact on the network's structure – and thus on the allocation of roles and power. For this, we see its possible to contrast two approaches: a multi-tiered peer-to-peer network and a private centralized network or whatever the designer deems appropriate alternative.

2.5.3 Distributed Consensus

Data integrity, specifically the quality and consistency of data, is an important consideration when designing a database. A functioning consensus mechanism is critical for the blockchain to function as a distributed database. That is why we examine the ramifications of various variants.

2.5.4 Cryptography

Modern cryptographic methods are the core of communication channel that is open to other participants. As a result, we use major cryptographic components, namely hashing algorithms and digital signature systems.

2.6 Machine Learning

The primary objective of machine learning is to train data using a specific algorithm. Machine learning's objective is to extract information from data. Additionally referred to as predictive analytic or mathematical learning, it is a topic of study that combines statistics, artificial intelligence, and computer science [61]. Machine learning approaches have grown

in popularity in recent years. Any component of a complex website, such as Facebook, Amazon, or Netflix, is quite likely to incorporate many machine learning models [62].

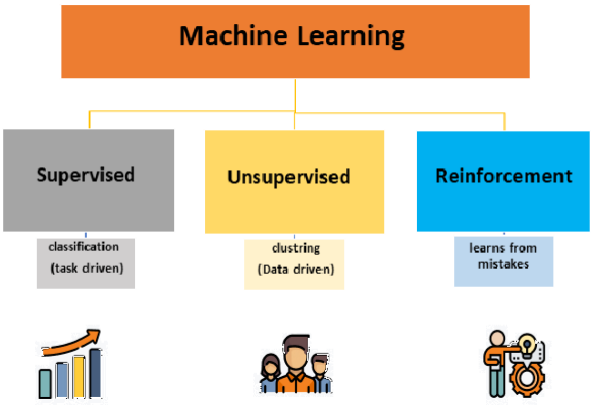

Figure 2.4: Type of Machine Learning (modified version from [2])

Consider a spam filter, which is responsible for forwarding relevant incoming email messages to a spam folder. You could develop a word blacklist that would automatically label an email as spam. This is an illustration of how a rule structure created by an expert can be utilized to develop an intelligent application. Manually constructing decision rules is a suitable technique for certain situations, such as those where humans have a firm grasp on the method to model [63]. The most efficient machine learning algorithms automate decision-making processes by generalising from existing examples. In this case, in supervised learning, the user gives the algorithm pairs of inputs and desired outputs, and the system figures out how to get the desired outcome given an input. The algorithm can generate output for an input it has never seen before without the participation of a human [64].

2.7 Combinatorics

Combinatorics is a branch of mathematics concerned with problems involving the selection, arrangement, and operation of finite or discrete systems. The closely related field of combinatorial geometry is also included [65, 66]. It is also dealt with in Chapter 4 when sharding works through simplicial complex in topological data analysis. Calculating the number of potential configurations (e.g. graphs, designs, arrays) of a given type is a fundamental problem in combinatorics as we interest in trust committee selection in Chapter 3. Even when the configuration rules are very basic, enumeration can occasionally provide serious obstacles. In mathematics, it is settled for an approximation or at the very least a reasonable lower and

Figure 2.5: Approaches and subfields of combinatorics

upper bound [65].

Finally, there are optimization issues to consider combined to combinatorial that different choice to x make the function f is optimal design. For instance, the economic function f assigns the numerical value $f(x)$ to any configuration x that meets certain stated criteria. The objective in this example is to pick a configuration x_0 that minimizes or equals $f(x)$-that is, for any number $\epsilon > 0$, $f(x_0) \geqslant f(x)+\epsilon$, for all configurations x with the required properties.

2.8 Optimization

Mathematical optimization is the process of picking the optimal element from a set of available alternatives depending on some criterion [67]. Numerous optimization problems arise in all quantitative areas, ranging from computer science and engineering to operations research and economics [68]. Optimization theory and techniques are applied to a variety of formulations is a significant field of applied mathematics. In its simplest form, optimization

is maximizing or minimizing an objective real function through the systematic selection of acceptable input values and calculation of the function's value. In a broad sense, optimization is the process of discovering the "best available" values for an objective function given a specific domain (or input), which may contain a range of different objective functions and domains.

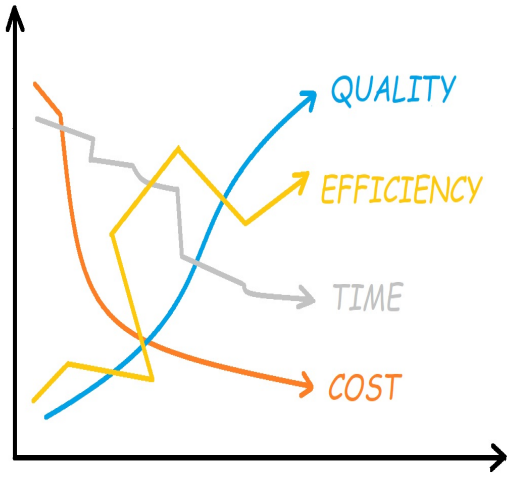

Figure 2.6: The goal of optimization

2.9 Metaheuristic optimization

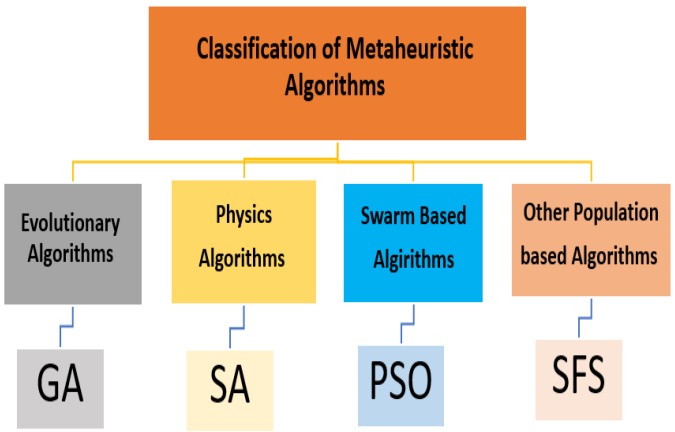

Figure 2.7: Classification of Metaheufistic Optimization Algorithms

A metaheuristic is a higher-level technique or heuristic used in mathematical optimization and computer science to discover, generate, or select a heuristic (partial search algorithm) capable of providing a sufficiently good solution to an optimization problem, mainly when the

information is incomplete or imperfect, or the computation capacity is limited [69]. Meta-heuristic approaches select a subset of answers that would otherwise be too numerous or complex to enumerate or examine thoroughly. Meta heuristics contain little assumptions about the optimization problem at hand, which allows them to be applied to a broad variety of tasks [70].

3 SWARM INTELLIGENCE MACHINE LEARNING BASED CONSENSUS COMMITTEE

The issue of consensus verification in blockchain technology is highlighted in this chapter. Consensus protocols are the basic processes in this technology, and most of the scalable operations in the blockchain are done through them. The contribution is to train intelligent algorithms to purify and sort qualified members and include them in the trust committee. The primary purpose is to delegate a group of nodes to all members with complete reliability. The idea has been fed by using the Particle Swarm Optimization (PSO) algorithm, the system in Figure 3.1-(a), which we will deal with in the first item. The committee's fear of

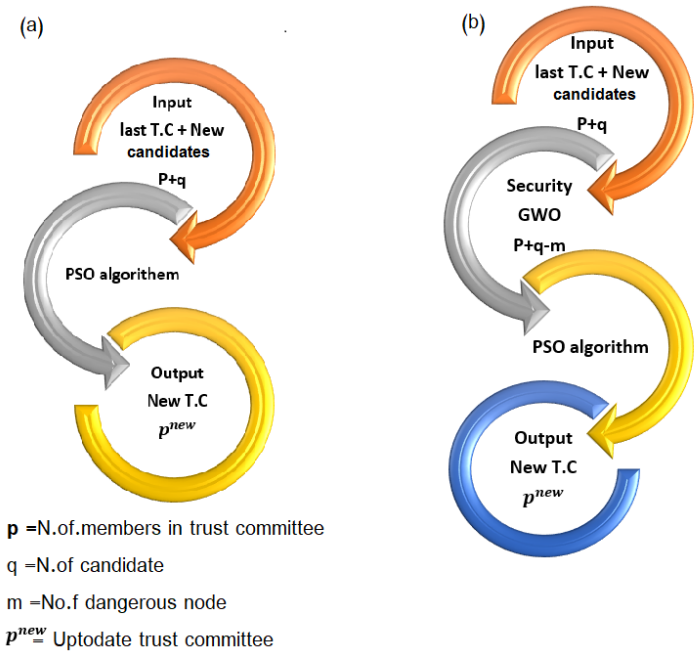

Figure 3.1: Diagram of consensus committee update

being attacked by nodes is still remained. Thanks to the PSO algorithm, the committee is formed, updating itself from time to time with the desire of a new contract for nomination and competition. The committee feels the need of taking a precautionary measure before the PSO begins its operation. We employed the Grey Wolf Optimization (GWO) algorithm to protect competitors for the new leadership committee from attackers who distort the work of the PSO algorithm, given in Subsection 3.2. Searching for misbehaving nodes and keeping them away from competition enhances the security of the inputs to the PSO algorithm by combining them with the intelligent GWO algorithm, as seen in Figure 3.1-(b).

3.1 PSO Based Trust Committee Selection in Blockchain

This section aims to approximate the selection issue of trusted nodes from the total participants as an optimization problem. Due to the presence of several optimization algorithms as ways to solve the problem, the particle swarm optimization method inspired by nature was chosen. A detailed explanation of the model and the results of experimental studies are given in this section. The results were published in a conference paper [20].

3.1.1 Introduction

Reaching consensus is one of the challenging problems in distributed computing. No solution has been given since 1985; on the other hand, in the past three decades, new protocols were designed to resolve consensus under various assumptions [49]. A blockchain is an immutable chain of blocks whose accuracy is guaranteed by the network's participants [9]. Today, by deploying new types of blockchain, many consensus mechanisms were proposed to agree on the order of transactions, which is often defined as a distributed ledger. However, not much work has been dedicated to exploring its theoretical ramifications. Hence, the known proposals are usually not clear, and they have implementation bugs or some design issues [49]. A vital scalability issue for implementing a consensus protocol is selecting nodes that accept the transaction. If many nodes are picked, this provides an appropriate setting and removes the necessity of a small trusted party. On the other hand, a small set of nodes makes the system faster [55]. Selecting a committee with a fixed number of nodes would be a solution. Nevertheless, there must be a method for the selection of this committee. Generally, Byzantine Fault Tolerance (BFT) based solutions use a subset of the nodes in the network. Therefore, the selection of a subset becomes an essential issue in the implementation of blockchain [8]. Since implementing blockchain systems with less decentralized computation power is a challenging task, one of the essential solutions in scaling the blockchain is a selection of a trust committee (TC). We intend to develop and apply a new method for selecting committee members depending on their reputation via particle swarm optimization (PSO) based update of the behaviour of nodes. Indeed, this is the same concept as accumulating participants in the blockchain throughout the previous rounds of the consensus procedure.

3.1.2 Particle Swarm Optimization Algorithm

Swarm intelligence (SI) is the collective behaviour of self-organizing decentralized systems, such as blockchains, whether artificial or natural and is widely employed in machine learning. The particle swarm method is one of the most important solutions for swarm intelligence [55]. The Particle Swarm Optimization (PSO) algorithm collectively optimizes a solution to a problem by iteratively attempting to enhance a candidate solution according to a specified quality criterion. In the first stages of the search space, the population of points as a set of moving particles is random. The PSO approach seeks the best or optimal position by adaptively updating their position until they achieve a generally stable position or the system is configured to cease after a predetermined number of iterations. Local and global bests will be calculated at each iteration based on the reputation and objectives of the current particles. Each particle is represented mathematically as a point in a k-dimensional space, with its status determined by its position and velocity [71]. Each particle possesses the characteristics described below.

1. It possesses both a position and a velocity.
2. It is aware of its own position and the value of the objective function associated with that position.
3. It retains its best previously discovered position.
4. It is aware of its neighbours' best past position and objective function values.

We note that the items 3 and 4 can be combined. At each iteration, the behaviour of a given particle is either of the following:

- to proceed in its manner,
- to return its former optimal location,
- to move to the best neighbour's former position or the best neighbour's current position (variant).

In our system, each particle i has j characteristic properties to be updated. This behaviour is formalized by the following equations, as given in [72]:

$$\begin{aligned} {v_{ij}}^t &= c_1 v_{ij}^{t-1} + c_2 r_1 (p_{ij}^{t-1} - x_{ij}^{t-1}) + c_3 r_2 (G_j^{t-1} - x_{ij}^{t-1}) \\ {x_{ij}}^t &= x_{ij}^{t-1} + v_{ij}^t \end{aligned} \tag{3.1}$$

for the j-th property of the i-th particle, where

x_{ij}^t: denotes the particle's position at time t,

v_{ij}^t: denotes the particle's velocity (behavior) at time t,

p_{ij}^t: denotes the node's best position till time t,

G_j^t: denotes the swarm (blockchain network members) best position at time t.

r_1, r_2: Two random real values selected from U(0, 1).

c_1, c_2, c_3: The parameters of velocity, cognitive and social updates, respectively.

In vector form, these can be seen in Figure 3.2 move-in one step and one particle into the swarm, while each individual is in the swarm, and for the number of repetitions specified in Figure 3.3. where the vector magnitude represents the weight value of that specific vector.

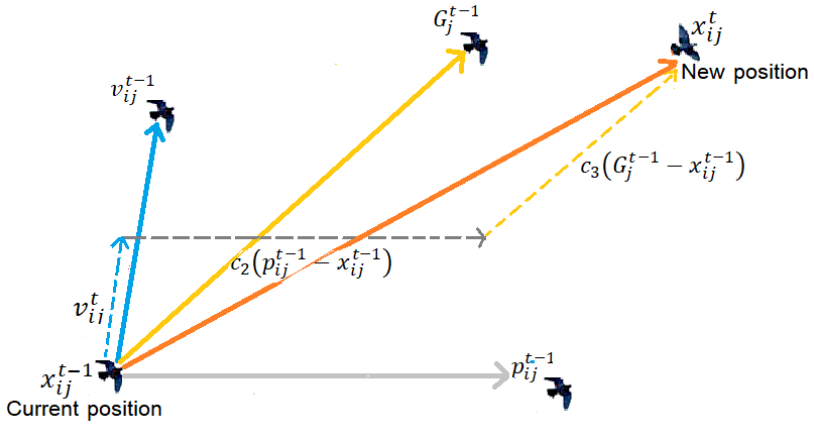

Figure 3.2: Illustration of the update of velocity and position in PSO

In summary, the following is a **brief description of the PSO process** in its entirety:

1. Create a population in hyperspace and populate it with random people.

2. Determine the fitness of each particle using the initial values.

3. Adjust velocities according to Equation (3.1) following prior best and global (or neighbourhood) best outcomes from previous trials.

4. Make a decision to terminate based on certain criteria.

5. Proceed to step 3.

3.1.3 Designing reinforcement reputation model

Initially, internal blockchain network for containers was created as simulation illustrated in Figure 3.5 and data scheme representation for blockchain in Figure 3.6 with adaptation of PSO to blockchain basics listed in the Table 3.1. Our goal is an easily deployable prototype

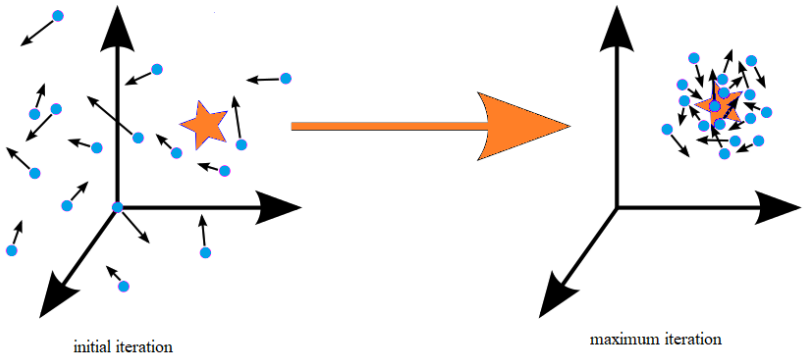

Figure 3.3: PSO dynamics (modified as in [3])

to make a decision to choose the trust committee from community or network participants, we attempt to classify and recognize some active nodes. This is accomplished by employing the PSO algorithm to capture the positive behaviour monitoring system that the nodes have. To adapt the PSO algorithm to blockchain technology, each component has its corresponding terminology as in Table 3.1.

Table 3.1: Remake of the PSO to Blockchain Technology concepts

Variable	PSO	Blockchain
x_{ij}^t	Position of particle i at time t	Node behaviour at t-th iteration
v_{ij}^t	Velocity at time t	Behavior changes in $[t-1, t]$
p_{ij}^t	Best position for particle i	The best score of node i for any time
G_j^t	Best position in the swarm	Best node in whole blockchain

Our approach starts by producing a swarm of particles, each one represents a node in the blockchain network which is a possible solution to the problem. PSO then assists in the selecting the most effective and successful nodes by monitoring its behavior and motions. Subsequently, we easily select the participants (examined nodes) who has earned enough reputation to be considered as a member in trust committee (TC). The PSO approach's objective function and technique are shown in Figure 3.4. The algorithm with the flowchart will be explained in detail in the following section.

Steps in our methodology

Step 1: Simulation

Blockchain network for containers was created using Monte Carlo simulation. A number n

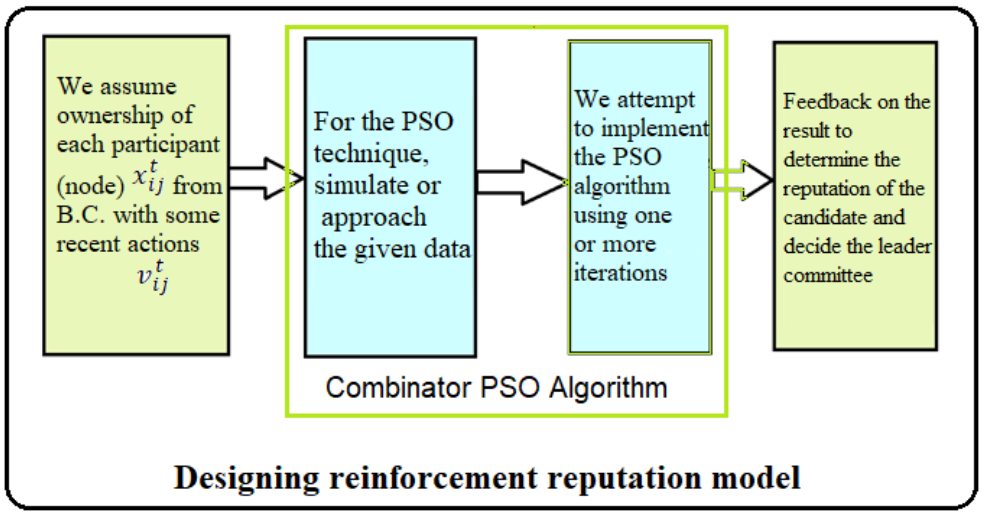

Figure 3.4: The diagram of the new model

of particles (nodes) in the network are randomly sampled, as many as the network's clients, shown in Figure 3.5.

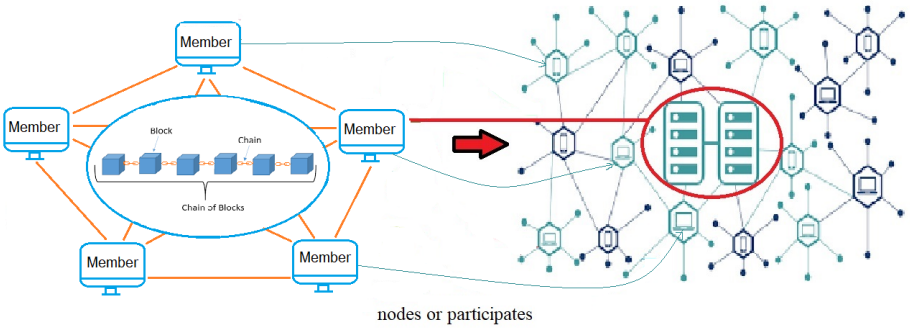

Figure 3.5: Overview of blockchain network simulation

Step 2: Insert Data

We have n nodes in search space and 5 properties per node. But the number of characteristic properties can be incremented in demand. We describe each node i as a vector x_i based on its characteristics $\{x_{i1}, x_{i2}, x_{i3}, x_{i4}, x_{i5}\}$, and each component corresponds to an attribute as follows:

$$x_i = \begin{bmatrix} x_{i1} \\ x_{i2} \\ x_{i3} \\ x_{i4} \\ x_{i5} \end{bmatrix} = \begin{bmatrix} belong\ time \\ response\ time \\ rate\ of\ success \\ amount\ of\ stake \\ type\ (new\ or\ old) \end{bmatrix} = \begin{bmatrix} rand[0,1] \\ rand[0,1] \\ rand[0,1] \\ rand[0,1] \\ \{0,1\} \end{bmatrix}$$

Belong time of a node is weighted amount of time it exists in the blockchain network. Response time of a node is weighted amount of average time it reacts for the approval of the transactions. Rate of success measures the ratio of node's successful approval of transactions. Amount of stake is weighted amount of coins/tokens the node has. Finally, type of a node is 1 if it is a new candidate for TC; 0 otherwise. Nodes participate in consensus, then they have belonged time as a priority in joining for the rest of the contract.

At each round, nodes decide the status of current transactions: confirm transactions and keep copies of confirmations, and participate in the creation of new blocks in the chain, for which they are rewarded with a stake increase. Their responses to given duties, updates their rely time to the transaction, the rate of success and others.

We illustrate the initial state of nodes in Figure 3.6. The more features $x_{i6}, \ldots, x_{ik}$ we add, the more accurate outcome we get. Initially, we choose the values of each property at random from the range $[0, 1]$. At each update time, we normalize the value to a range between 0 and 1 among all nodes of the blockchain. We note that type of a node is assigned whether it is committed member or not; hence it is only updated when its membership is changed. Also we can utilize fuzzification to incorporate data about recent actions via the velocity vector. For its j-th feature, each node i has a specific state x_{ij}^t at time t. The j-th feature of node i fluctuate with time depending on the type and amount of its activity; hence, the difference from the scenario that they had is measured and stored in a vector v_i consisting of components v_{ij} for $j = 1, 2, \ldots, 5$:

$$v_i = \begin{bmatrix} v_{i1} \\ v_{i2} \\ v_{i3} \\ v_{i4} \\ v_{i5} \end{bmatrix}$$

Step 3: Implementation

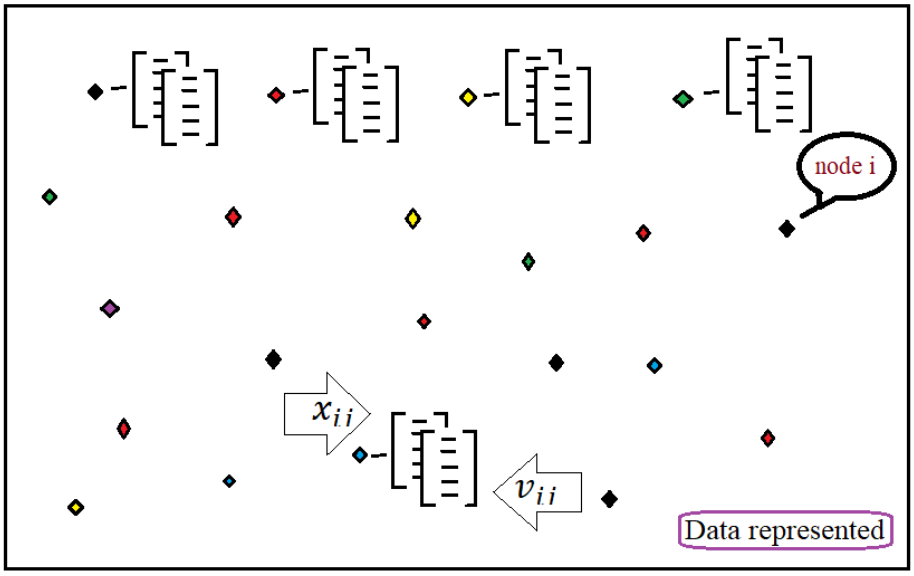

Figure 3.6: Simulation of nodes

Initialization:

A- Create a random variation vector v_i and a random feature vector x_i for each node i.

B- To compute reputation, use the fitness function $f(x)$ as in the PSO approach:

$$f(x) = \|x\|_2 = (\sum_{i=1}^{k} x^2)^{\frac{1}{2}}$$

And calculate the probability $p(x)$ of fitness function:

$$p(x) = \frac{f(x)}{\sum_{i=1}^{n} f_i(x)}$$

C- An initial TC is assigned by ordering the probability values of each node.

Epoch:

A- Blockchain runs finite number s of iterations (blocks) $t = rs+1, rs+2, \ldots, rs+s$ that is sufficient to recognize nodes in TC and new candidates for TC at each round $r = 0, 1, 2, 3 \ldots$. This is also called epoch in PSO literature. The velocity v_{ij}^t and the position values x_{ij}^t of each node i for the j-th feature is updated at each $t = 1, 2, 3, \ldots$ as given in (3.1), where $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, 5$. This is a competition among the new candidates for TC and existing nodes in TC.

B- Compute reputation, use the fitness function $f(x)$ and probability function $p(x)$ as in the PSO approach.

C- Select the TC members which have the highest rank according to the values of probability function.

We give the flowchart of this procedure in Figure 3.7.

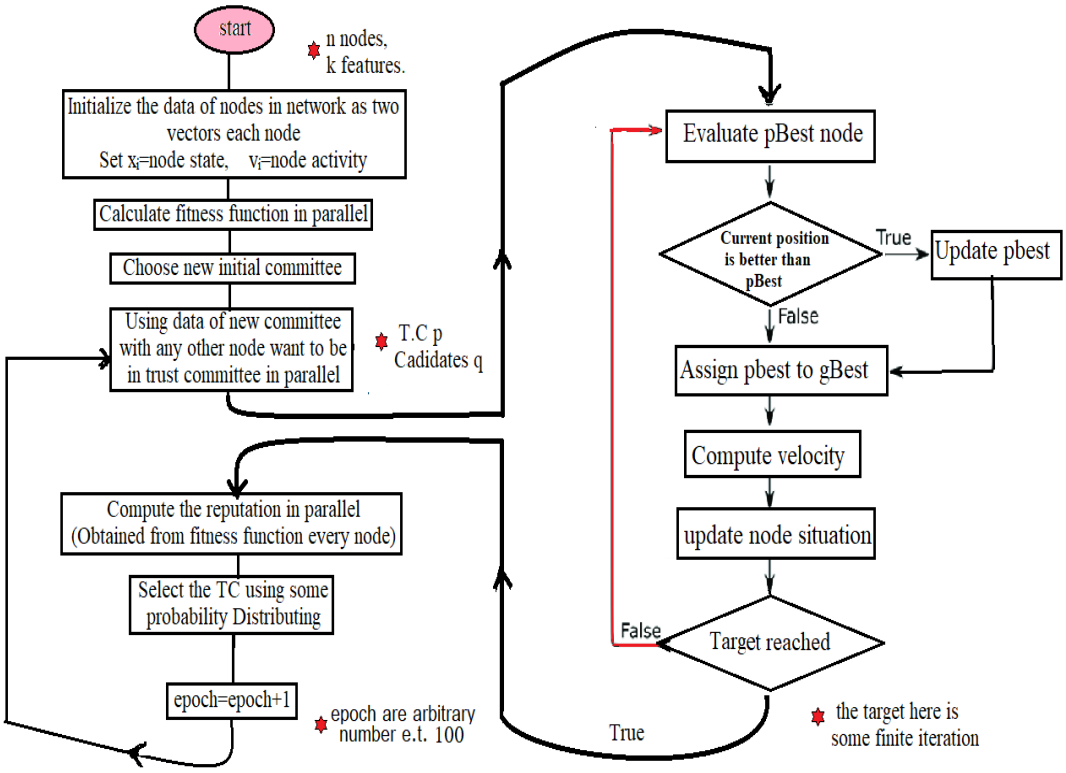

Figure 3.7: Flowchart of PSO based TC selection algorithm

3.1.4 Experimental Result and Discussion

The proposed algorithm for selecting the trust committee (TC) was implemented in Python 3.2 programming language on PC with Intel(R) Core(TM) i5-42000M CPU 2x2.50 Ghz. It's more efficient to work through techniques that have proven their efficiency and applicability in many applied fields like heuristic techniques, that remain preferred over traditional methods since it performs better than pure blind search. In this section, the numerical report and results obtained by applying the PSO algorithm are discussed. Note that PSO is considered as one of the machine learning methods.

The proposed mechanism in the consensus protocol of a blockchain is a system for monitoring active members within the network then gathering a group of participants to choose them as leaders in order to accomplish the work of consensus protocol. So the most important ad-

vantages of the proposed method is it can be implemented in coordination with all protocols consensus.

Swarm Size: We started by selecting a swarm of $n = 100$ particles. The size of the swarm is the same size as the number of participants in the blockchain network. The main focus of the experiments was to calculate the entitlement of each node depends in the input vector of $(k \times 1)$ size. In our practical tests we set $k = 4$. It can be more or less, it depends on the approved characteristics for each participant. Each feature x_{ij} of the node i in the swarm is initiated to a real number in the interval $[0, 1]$ for $i = 1, 2, \ldots, 100$ and $j = 1, 2, \ldots, 4$. Only for the last index $j = 5$, x_{ij} takes an arbitrary value from the set $\{0, 1\}$.

Parameter Setting: During experiments, we fixed the inertia weight c_1, which controls the impact of the previous history of velocities on the current one as

$$c_1 = 0.5$$

and the acceleration constants as

$$c_2 = 1 \text{ and } c_3 = 2.$$

Nevertheless, r_1 and r_2 are random real numbers drawn from U(0, 1).

Fitness Function: We need an appropriate tool to compare each member in the network with the rest of the members, or to decide how close the nodes to the ideal member, which is also known as the fitness function of the swarm. There are many functions for this purpose, including the mean square error, the fuzzy representation, and the distance function. Then, after some practical and programming experiments, it is seen that the distance function is a suitable metric according to mathematical computation

$$f(x) = \|x\| = (\sum_{i=1}^{k} x^2)^{\frac{1}{2}} .$$

What distinguishes this function from the others is that it is continuous, but not differentiable. The current technology does not need the property of differentiability. It can additionally be used on discrete (statistics) data seamlessly.

Trust Committee Selection

a- Initial Committee

We assigned 10 members of the most ranked participants from the swarm network of size 100 as the initial members of the trust committee (TC). The efficiency of a node is calculated primarily based on the recreation of every node and based totally on the attributes it possesses

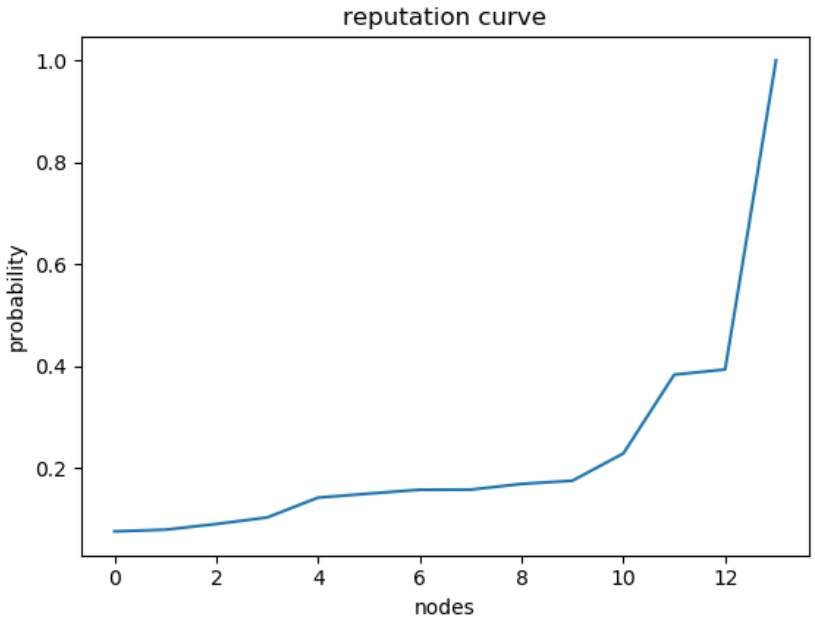

Figure 3.8: Probability scores vs size of TC

throughout the fitness function. The percentage attributed to every node is calculated by using the general probability rule:

$$P(node\ in\ trust\ committee) = \frac{fittness\ of\ node}{total\ fittness\ of\ nodes}.$$

We simulated the initial selection of k members randomly from a set of 100 participants and present the lowest probability scores required to be a TC member in Figure 3.8. It is seen that the required probability score increases with respect to the size of TC.

Figure 3.8 shows the opportunity for network nodes to belong to trust committee in an organized manner in ascending order. To finish the first stage of installation, the most qualified participants are taken from highest one by one, to the extent that the required and sufficient number of administer consensus protocols is completed.

b- Update Committee (Epoch)

Periodically, the trust committee is updated at every s blocks. We assigned $s = 100$ in our simulation. We note that the selected TC can not dominate the system as the new candidates have score 1 and existing old members have score 0. We simulated the system working with regular update of TC, one instance of this simulation is presented in Figure 3.9. It is seen in this figure that 4 new members among 5 candidates are selected for TC.

One of our fears is that this committee will dominate the administration of the protocol in a permanent way and challenge the decentralized feature of the blockchain network. The

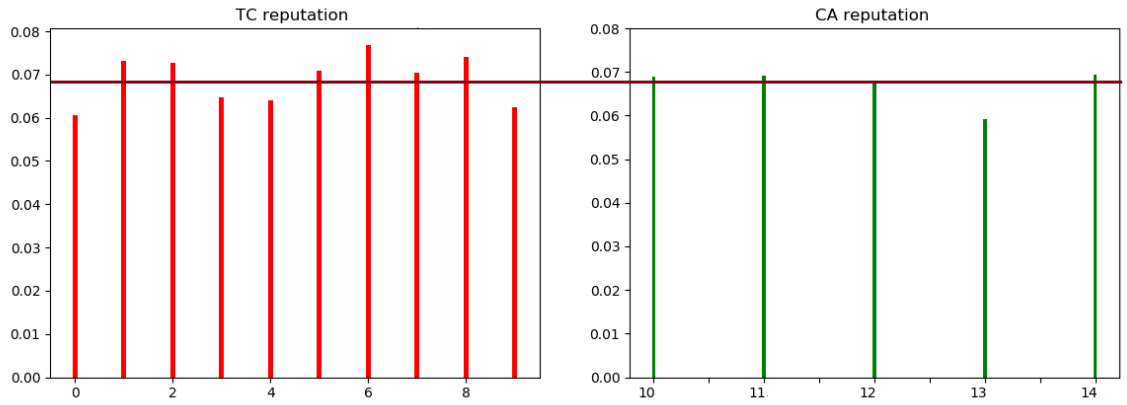

Figure 3.9: Selection of new trust committee (TC) among 10 exiting and 5 new candidates (CA)

members of the committee compete with the new candidates. The competition is managed via the PSO algorithm by monitoring the activity and efficiency of the competitors. After that, the decision is taken and the members of the trust committee are selected, as it was done previously in the first step.

Termination Criteria and TC session

Measures of achievement differ from one algorithm to another, and according to the application of which algorithm is employed. Here the stop criteria for the selection of the committee is the determination of the number of iterations. Limits are also set for the last activity for each participant which we represented with the velocity vector:

$$\text{The max iteration number } T_{max} = 3.0$$

$$\text{Maximum limited velocity } V_{max} \in (0, 1)$$

.

3.1.5 Summary and Discussion

In the blockchain system, the participating users must agree on what is added to the ledger based on a set of pre-defined standards. In particular, a consensus must be reached on the majority of nodes in the network, which must be implemented flawlessly through the monitoring and screening of active participants. Reaching to consensus via a trust committee (TC) is one of the solutions. A novel method is offered in this section for TC selection employing

particle swarm optimization (PSO). Simulations are implemented to test the efficiency and the applicability of developed method. After several tests, the model demonstrated effectively selects TC and updates it regularly. It could be a good future work to choose better coefficients in the velocity update equation with new characteristic properties.

3.2 GWO-Based Reinforcement Security when Selecting Blockchain Trust Committee

It is challenging to implement blockchain systems with less decentralized computing capacity. Choosing a trust committee (TC) is one of the most effective options for blockchain scaling. This aspect was discussed in Section 3.1. However, protecting the committee from attackers remains the paramount goal. A decentralized distribution property causes the blockchain to agree with swarm intelligence in a decentralized scheme. Through Grey Wolf Optimization (GWO)-based on contract behaviour updating, we plan to create a nature-inspired system to eliminate suspicious individuals from panellists based on their bad reputation. The idea of pooling the nominees of the blockchain participants and the performance of the outgoing leader committee during the previous rounds of the consensus mechanism to banish the worst and provide the good to enter the competition to be joined by the PSO monitoring system.

In order to accomplish the objectives of selecting the updated trust committee and providing an option for all network members to join at any moment, in general, is insufficient. To preserve the security of the blockchain network, it is still necessary to be wary of suspicious nodes. We utilize a straightforward strategy inspired by bio-dynamic systems to redirect the trust committee's attention away from the assaulting nodes. We improve the selection of honest nodes or participants by excluding poorly personalized nodes from the resultant algorithm to detect those destructive nodes. We suggested an unsupervised machine learning technique to solve the current challenge in this work by utilizing a Grey Wolf Optimization (GWO) technique. The idea is to determine a node's bad reputation rank based on its existing risk properties and exclude them from the consensus committee. In the next subsection, we describe the GWO approach. In Section 3.2.2 we briefly propose the GWO algorithm for diagnostics and elimination of attackers. Finally, the discussions of the obtained results are given in the rest of the Section.

3.2.1 Grey Wolf Optimizer Algorithm

Similar to PSO, GWO is also inspired from nature, more precisely it is inspired by the way grey wolves hunt. It is a crucial swarm intelligence solution [73] proposed in [74]. Grey wolves have a rigid social structure. This hierarchy allows them to stay stable and help each other when hunting. Grey wolves seek their prey under the alpha wolf's guidance. The rank of the wolves according to their ferocity is given in Figure 3.10. Occasionally, the alpha wolf will enlist the assistance of beta wolves, the next level below the alpha. Then gamma and delta wolves line up. While gamma wolves assist beta wolves, delta wolves are considered slaves. The remaining wolves command deltas and the leader wolf commands all of the other wolves. Grey wolves approach and encircle their victim as depicted in Figure 3.12.

The grey wolves division and assigning work are very similar to how people divide labour, which is an essential component of our society [75]. As with humans, who divide complex operations (e.g., manufacture engineering, exploration under challenging situations, or a federal investigation) into simpler subtasks, alpha grey wolves assign tasks to pack members who have demonstrated aptitude and expertise in efficiently performing them. It is thought that GWO is particularly effective since it appears to replicate highly effective patterns associated with human teams and their behaviours/strategies. From another angle, the process of initially encircling and then assaulting the prey is analogous to the growth and subsequent reduction of design space.

S. Mirjalili et al. [74] devised the GWO method in such a way that the solution with the most excellent fit is regarded to be the alpha group (α). As a result, the best solutions are designated as beta (β) and gamma (γ) for the second and third groups. The remainder of the solutions is called deltas (δ). In the GWO algorithm, wolves α, β, and γ direct the hunt (optimization). The δ wolves immediately pursue these three individual pack wolves [74]. It is graphically seen in Figure 3.10 that the order of dominance decrease in the same order as alpha, beta, gamma, and delta.

Mathematical Modeling of Grey Wolf Pack Hunting [74]

The GWO algorithm is influenced by α, β, and γ through its (optimized) hunting. These three wolves are chased by the δ ones. The following mathematical model describes how grey wolves encircle their target during hunting:

$$D = | C.X_p(t) - X(t) | \tag{3.2}$$

$$X(t+1) = X_p(t) - A.D \tag{3.3}$$

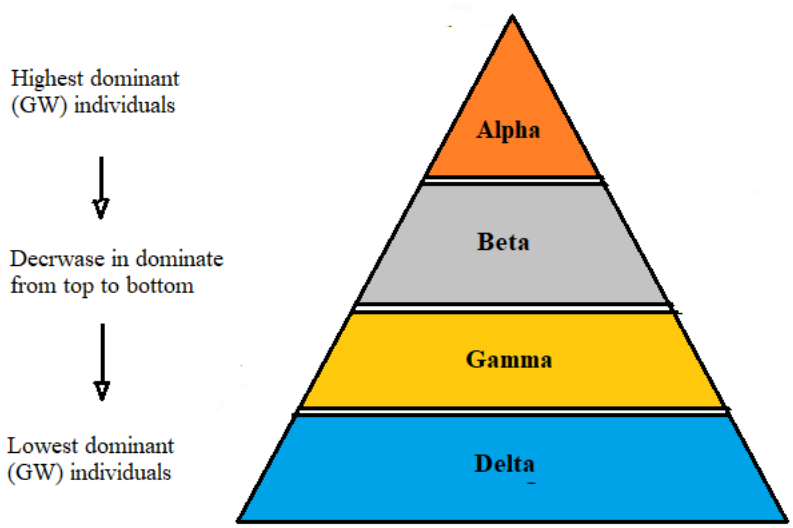

Figure 3.10: Hierarchy structure grey wolf dominance [4].

Where t reveals the recent iteration, C and A show the calibration coefficient vectors, X_p is the vector representing the target position, and X is the vector representing the grey wolf position. The following procedure is used to compute and correct the vectors A and C:

$$A = 2\ a.r_1 - a \tag{3.4}$$

$$C = 2\ .r_2 \tag{3.5}$$

Over iterations, the elements of vector a are reduced linearly from two to zero, and r_1 and r_2 are uniformly distributed randomized vectors in the domain $[0, 1]$.

By altering the weights of A and C vectors, it is possible to achieve many locations surrounding the best agent relative to the current position. It's worth noting that the random vectors r_1 and r_2 enable wolves to reach any point between the spots depicted in Figure 3.12, and then by applying Eqs. (3.2) and (3.3). Hence, a grey wolf can correct its place within the target's space in any random location.

One of its attractive features can locate and encircle prey. Grey wolves can recognize and encircle prey even if they have no idea where the optimal location is in an abstract (prey) search space. Mirjalili et al. presume that alpha (the most acceptable solution candidate), beta, and gamma have superior familiarity with the target's likely location to recreate the hunting behaviour mathematically. As a result, the top three search agents save the solutions discovered thus far and compel other search agents (including Delta) to upgrade their sites among the top search agent sites. In this regard, the following formulas have been proposed

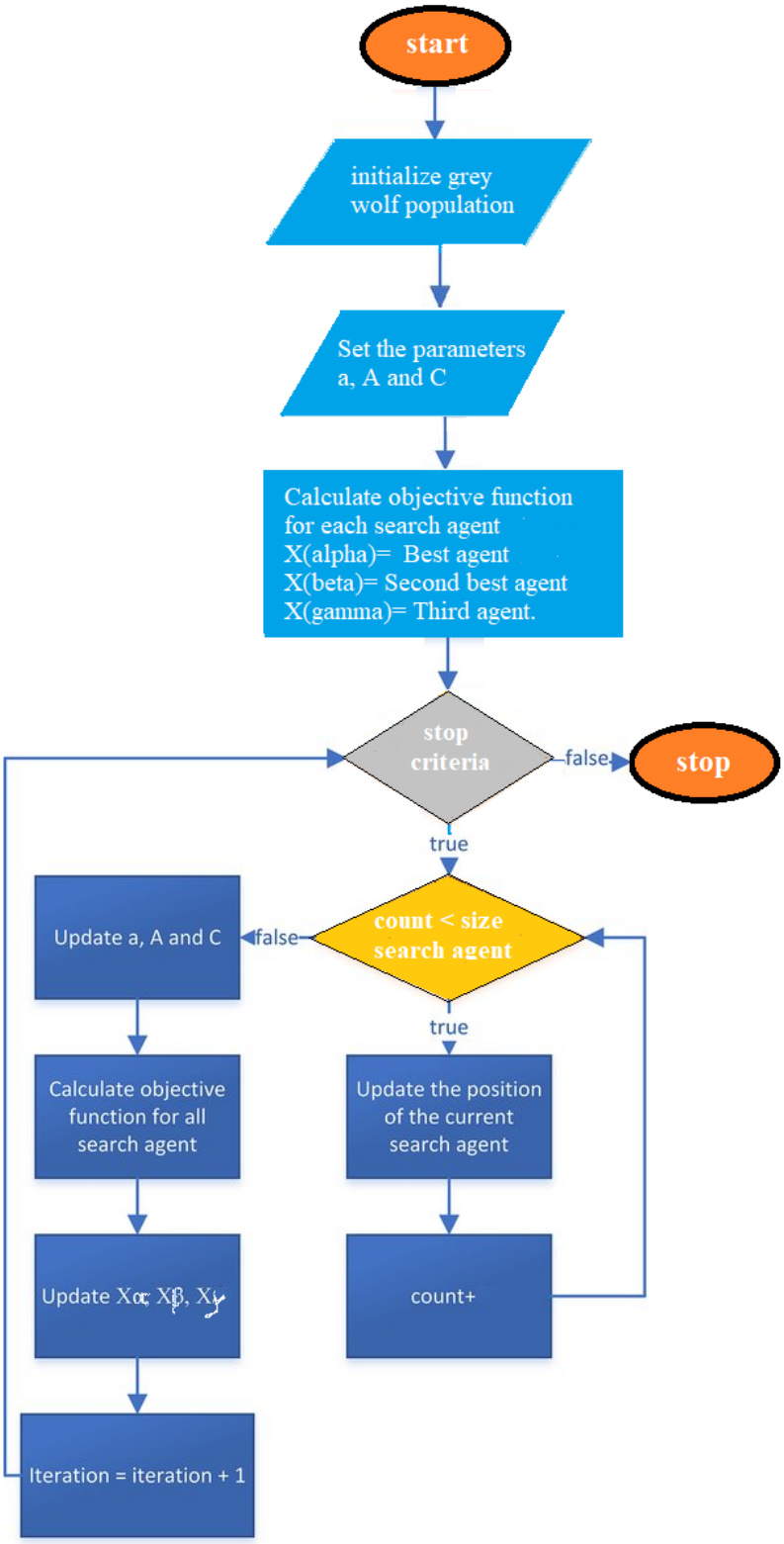

Figure 3.11: Flowchart Grey Wolf Optimizer

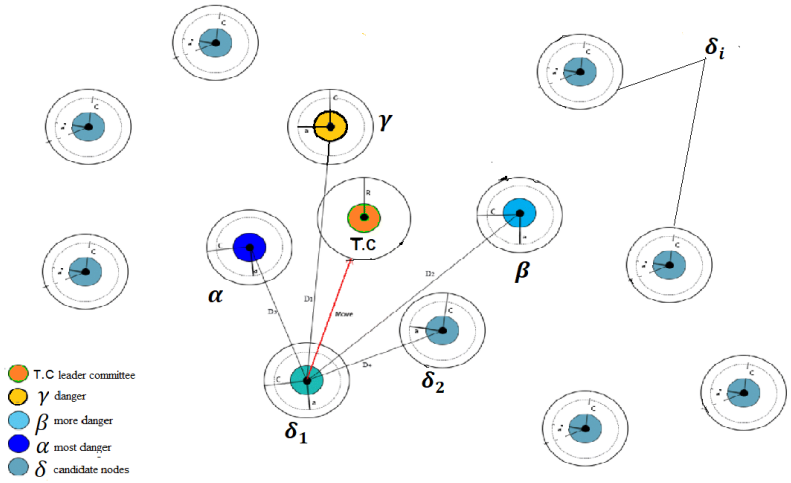

Figure 3.12: Position updating in GWO to attack

in [74]:

$$D_\alpha = |C_1 X_\alpha - X|;\ D_\beta = |C_2 X_\beta - X|;\ D_\gamma = |C_3 X_\gamma - X| \tag{3.6}$$

$$X^1 = X_\alpha - A_1.D_\alpha;\ X^2 = X_\beta - A_2.D_\beta;\ X^3 = X_\gamma - A_3.D_\gamma \tag{3.7}$$

$$X(t+1) = \frac{X^1 + X^2 + X^3}{3} \tag{3.8}$$

Figure 3.12 illustrates how a search agent adjusts its position in a two-dimensional search space in response to relying on the fiercest attackers alpha, beta and gamma. For this reason, by employing the dynamic of GWO, we try to find the most dangerous members and exclude them from entering the competition and thus joining the TC. As can be seen, the ultimate position will be random within a circle created by the α, β, and γ positions in the search space. In other words, alpha(α), beta(β), and gamma(γ) estimate the target's location, while further wolves randomly update their situations surrounding the target.

3.2.2 Trust Committee Protection Against Attackers

In a simulation case, the security risks associated with a trust committee will be handled by attackers. We consider the prey to be the leading committee that will be formed, and the predatory attackers (wolves) in the blockchain community are the attackers, i.e., nodes with improper intentions.

The PSO algorithm in the previous item chose the new committee from a competition between the previous committee and the candidates wishing to join the committee. The input

was from the members (last TC (p) + candidates (q)), and the output was an attempt to classify and identify some active nodes. The security protection performed by the SGWO technology precedes the PSO by making a primitive filter to (p + q) and removing the dangerous nodes and then submitting the remaining members to the PSO algorithm to select. We convert the terminology of the GWO algorithm to blockchain technology in Table 3.2 below.

Table 3.2: Adaptation of GWO to Blockchain

Variable	GWO	Blockchain
X_{α}^{t}	Best hunter at time t	Most danger attacker t-th iteration
X_{β}^{t}	Second best hunter at time t	More danger attacker t-th iteration
X_{γ}^{t}	Third best hunter at time t	Danger attacker t-th iteration
X_{δ}^{t}	General grey wolf candidate t	General nodes t-th iteration
$D_i, i = \alpha, \beta, \gamma$	Encircling prey	Approach factor in $[t-1, t]$
$A_i.i = 1, 2, 3$	$\|A\| < 1$ exploitation parameter	Score of Attacking for TC
	$\|A\| > 1$ exploration parameter	Score of Attacking for rest nodes

Our approach starts by picking hunters (grey wolves); each represents node riskiness in the blockchain network, which is a possible solution. GWO then assists in selecting the grave and more dangerous nodes by monitoring their behaviour. Subsequently, we easily select the participants (examined nodes) who possess enough lousy reputation to be considered a blocking to join the trust committee (TC). The GWO approach's objective function is shown in Figure 3.12 and technique is shortly shown in Figure 3.13 which we precisely follow. The algorithm with the flowchart will be explained in detail in the following section. In search space, each node given by the vector possess concluded its bad taints:

$$x_i = \begin{bmatrix} x_{i1} \\ x_{i2} \\ x_{i3} \\ x_{i4} \\ x_{i5} \end{bmatrix} = \begin{bmatrix} \text{Dormant service} \\ \text{1-Response time} \\ \text{1-Rate of success} \\ \text{1-Amount of stake} \\ \text{1-Type (new or old)} \end{bmatrix}$$

Dormant service refers to an account that has been inactive for a long time, save for posting interest. Belong time of a node is the weighted amount of time it exists in the blockchain

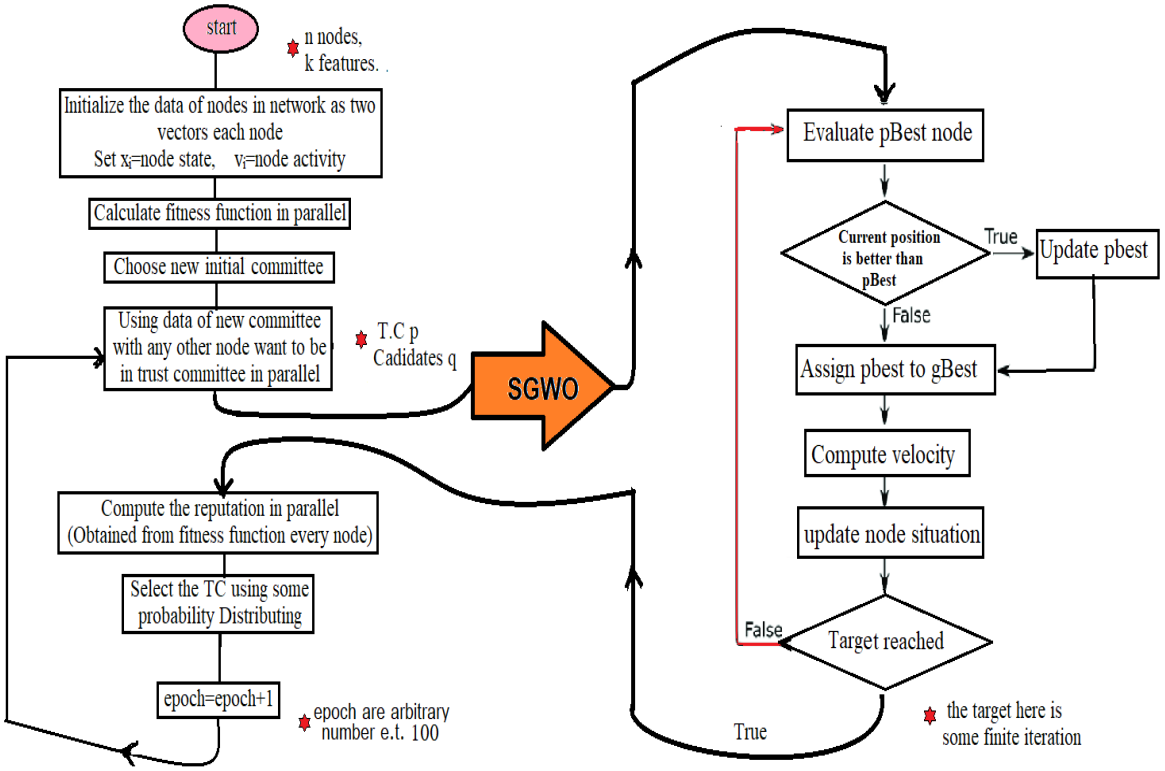

Figure 3.13: Flowchart of security protection of TC selection utilize SGWO

network. Response time of a node is the weighted amount of average time it reacts for the approval of the transactions as mentioned in Section 3.1. The success rate measures the ratio of the node's successful approval of transactions. The amount of stake is the weighted amount of coins/tokens the node has. Finally, the type of a node is one if it is a new candidate for TC; 0 otherwise.

At each round, k-th risk status for each node is updated as mentioned above matrix: confirm transactions and keep copies of confirmations, and participate in the development of new blocks in the chain, for which nodes are rewarded with an amount of stake. Their reaction to given assignments updates their rely time on the transaction, success rate, and others.

The more variables $x_{i6}, \ldots, x_{ik}$ we can add, the more accurate outcome we get. Initially, we choose the values of each danger at random from the range $[0, 1]$. At each update time, we normalize the value to a range between 0 and 1 among all blockchain nodes. We note that a node is assigned whether it is a committed member or not; hence, it is only updated when its membership is changed to incorporate data about recent actions. For its j-th faulty, each node i has a specific state x_{ij}^t at time t. The j-th faulty of node i fluctuates with time depending on the type and amount of its harmful activity; hence, the difference from the scenario they

Algorithm 1 Grey wolf optimization based security (SGWO)

1: **Input:** Trust committee with candidate members.

2: **Output:** Recognize and ban the most harmful members (attackers).

3: Initialize the population mimic for the nodes $X_i,\ (i = 1, 2, \ldots, p + q)$

4: Setting a, A_i and C_i.

5: Calculate the (malignity) reputation via fitness values Eq. (3.9) of each node (wolf)

6: Diagnose attackers: setting for almost dangers. X_α = the best (highest) fitness value as the alpha node, X_β = the second-best fitness value as the node, X_γ = the third-best fitness value as the node, save the rest of nodes in X_δ.

7: **while** not (stopping criteria) **do**

8: Update the attitude of the candidate node (delta) Eqs. (3.6), (3.7) and (3.8)

9: Correcting the parameters a, A_i and C_i Eqs. (3.4) and (3.5).

10: Evaluate candidates via calculate the fitness values via Eq. (3.9) of all wolves and sort them in addition to X_α, X_β, X_γ

11: $t = t + 1$

12: **return** Step 6.

had which was measured and stored in a vector X_i consisting of i for $i = 1, 2, 3$ using Eqs. (3.2) and (3.3).

Steps in our methodology

Step 1: Simulation

Each node in the family members $p + q$ of candidate nodes in the network are randomly sampled where p is the number of members in the last version in TC and q is the number of the candidates in general.

Step 2: Insert Data

We have $p+q$ nodes in search space. Every member has a position containing 4 or k quantitative (dimension) values representing given bad trait per node in metric space representation. Nevertheless, the number of characteristics as flaws can be incremented in demand. We describe each node i as a vector X_i based on its riskiness $\{x_{i1}, x_{i2}, x_{i3}, x_{i4}, x_{i5}\}$, and each component corresponds to an attribute in the vector given above.

Step 3: Implementation

A- Calculate malignity by evaluating fitness function $f(x)$ as in the GWO approach:

$$f(x) = \|x\|_2 = (\sum_{i=1}^{k} x_i^2)^{\frac{1}{2}} \tag{3.9}$$

B- Rank the candidates (p+q) in general according to risk. She was assigned by blocking the riskiest contract (α), the riskiest two (α, β), or the riskiest three (α, β, γ).

C- After we exclude (α), the riskiest two (α, β), or the riskiest three (α, β, γ) the rest of the members send to PSO algorithm As illustrated by the flowchart of the algorithm in Figure 3.13.

Epoch:

Blockchain runs finite number s of iterations (blocks) $t = rs + 1, rs + 2, \ldots, rs + s$ that is sufficient to recognize nodes in TC and the rest of members in whole blockchain network at each round $r = 0, 1, 2, 3 \ldots$. The situation X_i^t with the position values x_{ij}^t of each node i for the j-th risk and activation are updated at each $t = 1, 2, 3, \ldots$ This is a competition among the new candidates for TC and existing nodes and we repeat step three times.

3.2.3 Result and Discussion

The proposed mechanism in the consensus protocol of a blockchain is a system for monitoring active members within the network then gathering a group of participants to choose them as leaders to accomplish the work of consensus protocol; although being a good alternative, it lacks some security enhancements-sensitivity of attackers to the fact that is attacking committee in order to control protocol compatibility.

Our contribution here is to use machine learning to examine the attacker's negative behaviour and malicious intent using the GWO algorithm. In coordination with all protocols consensus, we used a GWO algorithm to assess potentially harmful participants and eliminate them from consideration for membership on the leading committee. Then the system prevents them from joining the TC. Thus, the primary advantage of the suggested method is that it may be used in conjunction with the PSO algorithm before the responsibility committee designing consensus protocols. This is done for the purpose of enhancing the security by excluding undesirable members.

One of the issues is that the selected committee will permanently control the protocol's management, eroding the blockchain network's decentralized nature. This scenario occurs if the committee exploits some attacks and coerces committee members into adopting new candidates. The GWO algorithm controls competition by observing competitors' unwanted and

ineffective behaviour. Following that, a choice is reached, and the TC members are chosen. This will prevent the algorithm from identifying the node as compromised by attackers.

Pick Size

There is flexibility in choosing the total number of herds for mimics customized to meet the user's needs. Our work is governed by the number of each committee member, which requires adjusting (p) with candidates from the network and their number (q). We bind each participant or node to the grey wolf in pack member to finish the algorithm's operation to achieve discrimination. Then each faulty x_{ij} of the node i in the pack is initiated according to the vector of risk analysis as mentioned in Section 3.2.2

Parameter Setting

Many parameters must be constantly adjusted. Drop parameter a from 2 to 0 to emphasize exploration and exploitation throughout the search phase. If a is more than one, the candidate solutions diverge from the target, whereas a is greater than zero, the candidate solutions converge to the target. This process is repeated until the stop conditions are met, at which point the GWO algorithm is terminated [74].

When the target stops moving, wolves attack it to complete the attack. This is shown by reducing a from 2 to 0 during iterations. As a declines, so does A. As a result, $|A| < 1$ forces the attacker to combat the target. $|A| > 1$ diverts away from the target in search of a better target. In this situation, sharding the blockchain or having more than one confidence committee indicates more than one target.

The C parameter vector consists of a random value chosen from the interval $[0, 1]$. If $C > 1$, it helps to place some more weight on the target, making it difficult for attackers to find it: Emphasize $C < 1$ Reduce the emphasis (reduce importance).

Fitness Function

We require a tool capable of comparing each network member to the rest of members, which is also referred to as a pick fitness function. Fitness function is objective that how close a given design solution is to meet the set goals. It is used to help simulations find the best possible designs. Then, investigations determined that a distance function is an appropriate unit of measurement, according to mathematical computation

$$f(x) = \|x\| = (\sum_{i=1}^{k} x_i^2)^{\frac{1}{2}}$$

Termination Criteria

The indicators used to assess the achievement vary between algorithms. Mostly, it depends

on the application for which algorithm is used. The number of iterations is used as the minimum threshold for selecting the committee in this case. Additionally, limits are established for the participant's final activity by testing for a function value convergence. When all of the function values are sufficiently near in some way, the iteration is completed. Perhaps the maximum number of iterations is the best choice: The iteration ends when the stated iteration limit or function evaluation is exceeded. Composite criteria are occasionally utilized.

3.2.4 Summary and Discussion

In a blockchain system, users must agree on adding to the ledger based on a predefined set of criteria via consensus protocol achieved on most network nodes. This must be done flawlessly by monitoring and screening active users. Consensus-building using a responsible committee (TC) is one option that considers the committee's security. Using the Grey Wolf optimizer (GWO) algorithm, this section presents a novel way to exclude nodes attempting to assault a trust committee (TC). The developed method is then subjected to simulations to determine its efficiency and applicability. The model depicted effectively picks and updates the TC. Additionally, the committee's security-enhanced its reliability by identifying attacking nodes and prohibiting them from causing any damage to the committee.

4 COMBINATORIAL TOPOLOGY BASED MACHINE LEARNING FOR BLOCKCHAIN SHARDING

4.1 Introduction

A public blockchain is a network that anyone can access. Anyone may submit transactions and expect them to be included if they are genuine, and anyone can participate in the consensus process [36]. The network can become crowded, slowing the process (latency). Sharding is one of the strategies being studied for achieving latency-free scaling [41]. However, if the shards are overloaded, finding an optimum value for the number of shards would be nice. Here, we need to balance the available computation power by dividing it into several smaller committees called shards. The growth of their number is proportional to the strength of the total computational power. These committees have a fixed number of members, say m, and they internally run a classic Byzantine consensus protocol to agree on a single value. At this stage, our role is neither in the infrastructure nor in the blockchain structure itself. We only work on the division and management of tasks for the existing members of the system [73]. This study focuses on the identification and evaluation of the optimal sharding principle. Indeed, the idea of blockchain architecture from the aspects of combinatorial topology is given. We first deal with a data distribution system enriched with blockchain networks, using topological data analysis (TDA) to get a graph network. Then, we study the combinatorial structure of these data using the simplicial complex. By doing so, we have its combinatorial topology and persistent homology Betti number. Indeed, this helps us introduce an unsupervised machine learning tool to analyze the system and fit the shard size to the simulation results and attempt to construct a mathematical model following TDA. The Linear Programming Problem (LPP) is built and solved using the Dual-Simplex method to determine the optimal shard size. The sharding operation is then used, which helps research fundamental computational issues in blockchain technology such as consensus, Byzantine fault tolerance, and self-stabilization. Moreover, successful sharding could be used easily in various fields, making the processes faster.

4.1.1 Sharding

Sharding is a database partitioning technique that blockchain firms employ to increase their scalability and handle more transactions per second. Sharding divides the whole network of

a blockchain corporation into smaller segments called "shards". Each shard is built from its own data, making it separate and self-contained compared to other shards [73].

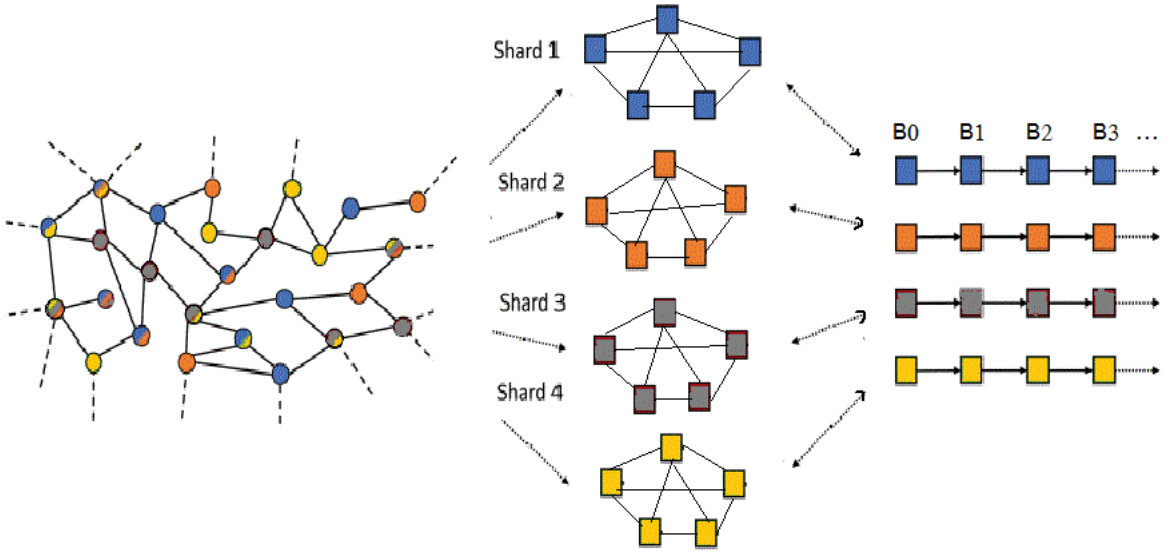

Figure 4.1: Graphic representation of scaling with sharding

Sharding is also a strategy for segmenting a network's workload, which can help reduce latency while also allowing the blockchain to process a greater volume of transactions. In database systems, sharding is based on a crash-failure model, in which a problematic node ceases to send and react to queries. This assumption has three significant ramifications [22]:

- The coordinators who are in charge of coordinating protocols are entirely trustworthy.
- The process of creating a shard is straightforward. For instance, a node's location can be used to assign it to a shard.
- To achieve high performance, efficient consensus procedures that account for the crash-failure model can be used.

4.1.2 Topology

Definition 4.1. *The study of qualitative features of particular objects (topological space) which are invariant under a specific type of transformation (continuous map), particularly those properties that are invariant under a specific kind of equivalence (homeomorphism) [5].*

Figure 4.2 shows the Königsberg Seven Bridges is a famous mathematical problem. Its negative solution by Euler in severity country established graph theory and topology.

Figure 4.2: The Seven Bridges of Königsberg, a problem solved by Leonard Euler (1736) [5]

Having the basic understanding of topology, the natural question that arises here is: how can it assist us in comprehending our data? Topology has many qualities that make it very useful when analyzing data. For example;

- Changes under small deformations can be easily traced using topology. Therefore, it is much easier to deal with noise.
- Coordinates are no longer a concern, or at least they are not that significant, because the topological properties that are analyzed are no longer affected by them.
- Objects can be examined by using a graph, as in Figure 4.3 or a simplified representation, called simplicial complexes (which will be described later), preserving the desired topological characteristics.

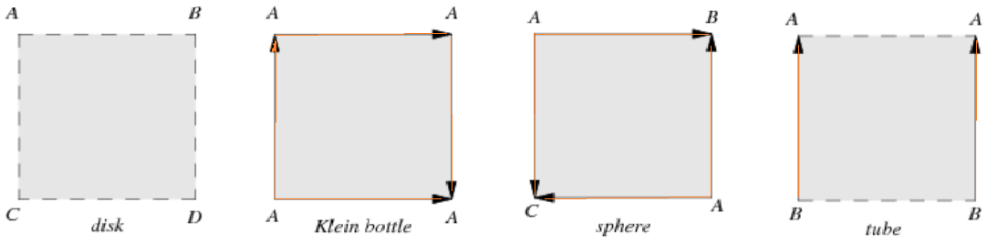

Figure 4.3: Connection of parallel lines to construct different shapes.

Topology can be used to abstract an object's underlying connectedness while disregarding its detailed shape [76]. The pictures above, for example, depict the connection of a variety of topologically diverse surfaces. Parallel lines are drawn in solid to connect parallel

edges with the orientation indicated by arrows so that corners labeled with the same letter correspond to the same point, while dashed lines represent free edges. Thus, how can we incorporate topology into our distributed system? To begin with, we must introduce certain related concepts.

4.1.3 Network Topology

The term "network topology" refers to the physical or logical arrangement of the network's nodes, devices, and connections as shown in Figure 4.4 as its types. Consider a network to be a city, with the topology serving as the road map. Just as there are numerous ways to organize and manage a city—for example, ensuring that roads and boulevards allow movement between the busiest parts of town—there are multiple methods to organize a network. Each offers several advantages and downsides, and depending on the business's needs, different arrangements may provide a higher level of connectivity and security [77].

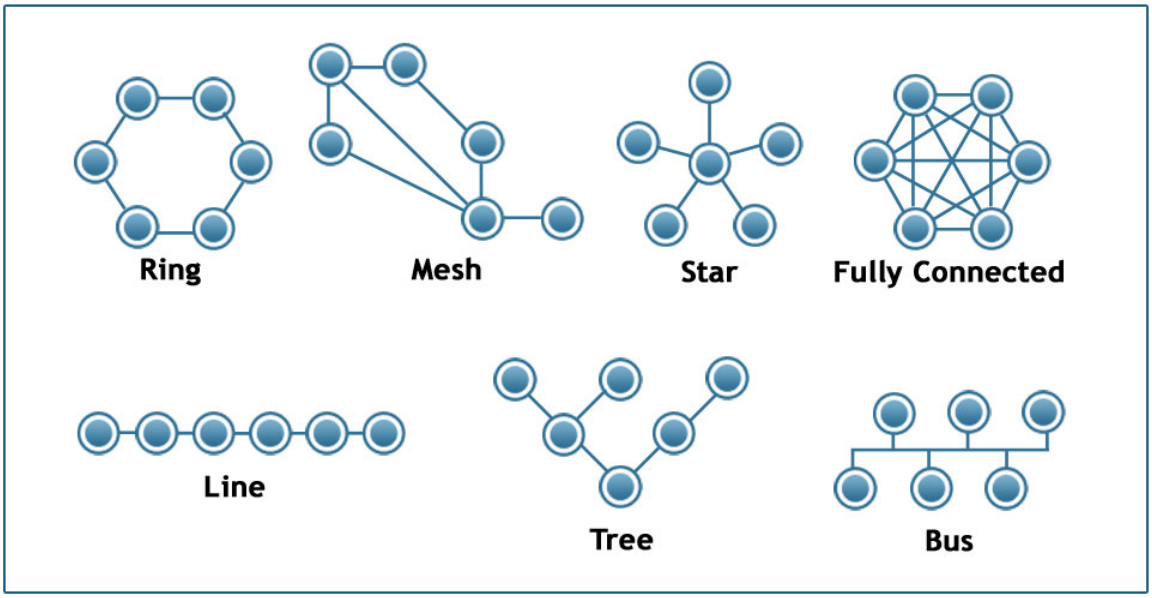

Figure 4.4: The Various Types of Network Topology [6]

4.1.4 Topological Data Analysis (TDA)

TDA attempts to provide sound mathematical, statistical, and computational approaches for inferring, analyzing, and exploiting the complex topological and geometric structures underlying data that are frequently represented as point clouds in Euclidean or more generic metric

spaces [78].

The relationship between data and shape or structure can be obtained by comprehensively measuring the shape of data sets using algebraic topology. For example, given a finite collection of data points $P_{i=1}^{N}$, we want to understand the shape of the data. What does this mean? What is the data's relationship and structure? We want to develop some tools to capture qualitative information about the shape of the data. We want this information to be robust against noise.

Figure 4.5: Topological Data Analysis Pipeline a and b [7]

a.First approximate the unknown space X in a combinatorial structure n.

b. Then compute topological invariants of n.

This procedure is done through the following steps: Through TDA we construct a higher-dimensional structure on our point cloud via simplicial complexes. Then, we analyze this family of nested complexes with persistent homology. That precisely displays the Betti numbers in graph form.

4.1.5 Persistent Homology (PH)

It is a technique used in TDA to investigate qualitative data characteristics that persist across multiple scales. It is resistant to perturbations in the input data, dimension-and coordinate-independent and gives a concise representation of its qualitative characteristics [79]. Then, PH is an algebraic method of discerning the topological features (e.g. components, holes, graph structure, etc.) of data that means the set of points with a metric that distances between pairs of points. To find the PH, first a simplicial complex must represent the space. Filtration of the simplicial complex, a nested sequence of rising subsets, corresponds to a distance function on the underlying space. The definition of **homology** is a bit too technical, counting the connected components or lines (1-Simplex), holes (2-Simplex), and voids (3-Simplex)

of a simplicial complex. Moreover, homology is computable via linear algebra [80].

How to Read Betti Numbers: Betti numbers refer to the number of independent (suitably defined) objects in that dimension that have no boundaries. Betti numbers are a sequence of numbers indicating how many holes an object has in each size or dimension.

Like simplexes, they come in different dimensionalities; a simple example is given in Figure 4.10. The Betti numbers represent the number of $(n-D)$ holes in space. They are an essential topological feature of the complex, and they are determined by analyzing the simplicial complex.

$$\text{Betti number}(\beta_i) = \{\beta_0, \beta_1, \beta_2, \beta_3, \beta_4, \ldots\}$$

Betti 0 (β_0): The number of components that are interconnected. If an edge connects everything, β_0 is 1. If there are two groups of corresponding edges, β_0 is 2. If n points are not connected (discrete), β_0 is n.

Betti 1 (β_1): It represents the number of holes in a surface. For example, for a 2-Complex $(k = 2)$, a triangle has 3 edges and 1 hole. Therefore, each triangle in the complex creates a hole.

However, if the 3-Complex is used, triangles are considered "filled in", meaning there will be no more holes $(\beta_1 = 0)$. There may be voids in the volume of a tetrahedron β_2.

Betti 2 (β_2): It is the voids (Empty volumes).

Remark 4.2. It does not always make sense to consider the higher-order Betti numbers, depending on the data. For example, if the data is known to represent a $2D$ surface, β_2 to β_n may not be of value.

4.1.6 Simplicial complex

We start by constructing a simplicial complex from the points of data clouds. In TDA, a simplicial complex is an n-dimensional triangular structure composed entirely of the original data. The complex can then be used to investigate homology and other topological properties. The complex is made up of many simplexes that are linked together.

The k in a k-simplex denotes its dimension. For example, a 0-Simplex is simply a point. Two connected points make a 1-simplex (edge). 3 connected points form a 2-Simplex (a triangle). 4 fully connected points form a 3-Simplex (tetrahedron).

n connected points form an $(n-1)$-Simplex. The simplicial complex's dimensionality can be much higher than the original data, depending on how the data is connected and the type

of complexity. For example, we can form a tetrahedron with $4\ (2-D)$ points see Figure 4.6. Connectivity is determined by the type of complex building method used.

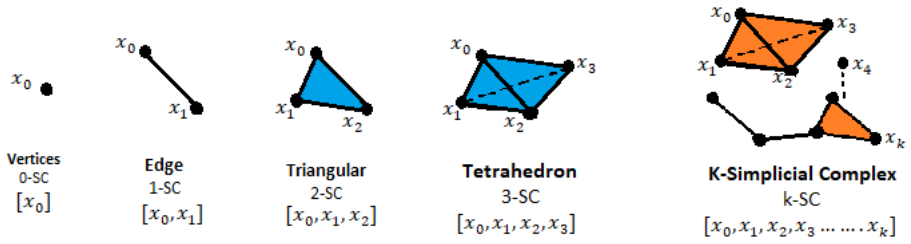

Figure 4.6: k-simplicial complex

Different complexes appear at different ϵ (threshold) values, so we track how complexes become connected with barcodes. We are looking for a structure in the sense of a graph that has the desired cluster of nodes.

4.1.7 Simplicial Complex of Data

It is used as data aggregation on existing networks to convert an active unpartitioned network to a sharded network. The following measures can be taken to accomplish this goal:

1. Create a snapshot of the ledger's current state (ideally in the form of an account balance list) and format it as a genesis block B_0.

2. Use the B_0 snapshot to bootstrap a sharded variant of the protocol.

Then, there are two main applications of simplicial complexes in data analysis: the representation of relations and the discretization of data spaces [81]. To better understand our data's structure by extracting topological information, we apply TDA by utilizing PH which is one of the critical tools for simplicial complexes. Since the late 1800s, simplicial complexes have been utilized to convert complicated topological problems into more familiar algebraic issues. With the introduction of computers, their ability to hold geometric and topological information in discrete form made them indispensable tools for image identification and, more recently, data analysis. They successfully approximated the topology of the space underlying the data collection.
We have a finite collection of data points that have no intrinsic shape. To deal with this,

- Calculate the difference (or dissimilarity) between two data points.This can be very

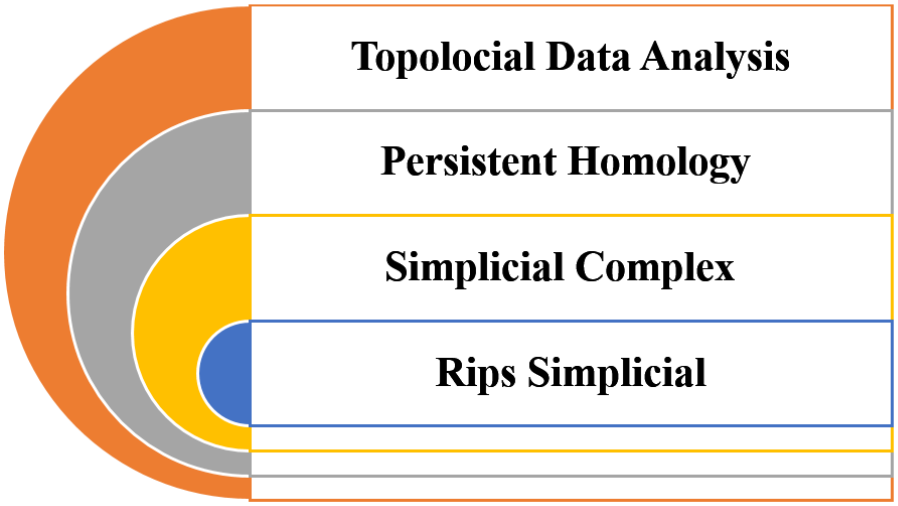

Figure 4.7: Pipeline of topological data analysis

general, e.g., color gradation $d(black; black) = 0$, $d(black; white) = 1$: we get a finite metric space.

- Given an ϵ scale parameter, add a k-simplex of σ if each of σ's vertices is within an ϵ distance.
- Depending on the scale parameter, we get different simplicial complexes.
- Calculate the homology of the different complexes. How does it vary with the scale parameter?

The main prototype of the simplicial complex is the following type [82]:

4.1.8 Rips complex

The Rips complex is a type of data simplicial complex. Formally, we have it for any metric space (M, d).

Definition 4.3. *The Vietoris-Rips complex of (M; d) at threshold ϵ is the simplicial complex VR(M; ϵ) whose*

1. Vertices are points in M.
2. A k + 1-tuple of distinct points $x_0, x_1, \ldots, x_k$ spans a k-simplex precisely when $d(x_i; x_j) \leq \epsilon$ for $i; j = 0, \ldots, k$.

Note that the Rips complex is massive if M has many points. However, it is preferable over the remainder of the class because it just involves verifying if the distances between each pair of data points are less than a given threshold, and there are $\binom{n}{2}$ such matching.

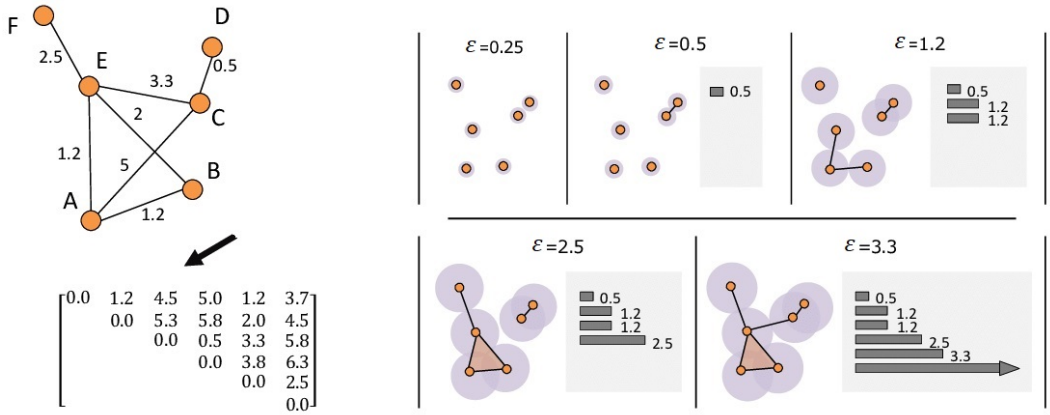

Figure 4.8: Illustrative sketch the Vietoris-Rips complex

Remark 4.4. The Rips-construction is applied to finite metric spaces. We can understand the data shape by looking at the Rips complex at varying scales (different ϵ). We want to develop some tools to capture qualitative information from the body of data. We indeed wish that this information was robust against noise. By increasing the parameter ϵ, the simplicial complex will grow by adding new fractures and homology properties.

4.1.9 Filtration

Each simplicial complex is a sub-complex of the next. The sequence of a simplicial complex is called filtration. We apply homology to filtration, which results in an algebraic structure known as a persistent module.

$$H_i(C_1) \to H_i(C_2) \to H_i(C_3)\,,$$

where C_i is a vector space. Persistent homology module is:

$$M = H_i(C_1) \oplus H_i(C_1) \oplus H_i(C_3)$$

Module M decomposes into a direct sum of interval modules ${M_j}^I$, each of which corresponds to a bar in the barcodes

$$M = \oplus_j {M_j}^I\,.$$

4.2 The procedure for our approach

We shall explain the fundamental operations in this section. Let us examine the procedures in greater detail:

1. ***Extract the data and envision data as a point cloud***: Point cloud data is represented in R^d Euclidean space. Each node is a collection of d features related to participants in blockchain in vector form.

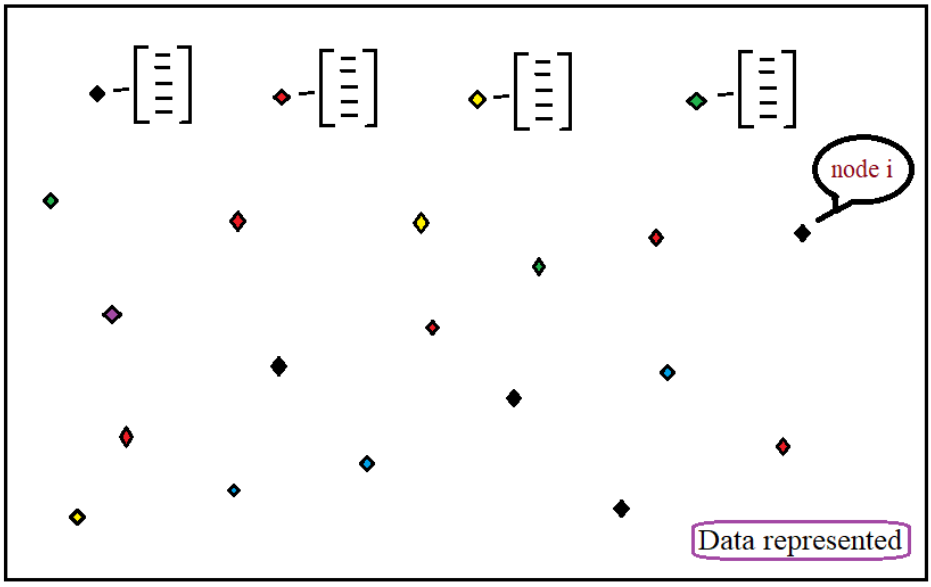

Figure 4.9: Point cloud data

2. ***Create connections between proximate and the combinatorial topology representation***: In a combinatorial structure, approximate the unknown space X as closely as possible. Firstly, we define the closeness function to build the simplicial complex. To compute the measure of the nearest among each other vertices in cloud data, we give the closeness of the node to connect them. For this purpose, it has been proposed:

$$cov(x_i, x_j) = w, w \in [-1, 1] \qquad \text{or,}$$

$$cov(x_i, x_j) = |w|, w \in [0, 1]].$$

This also represents the scope of the threshold ϵ such that we set $\epsilon = w$.

3. ***Determine topological structure of data in the simplicial complex***: One of the essential tools in persistent homology, as in the Figure 4.10 and Figure 4.6 is the simplicial complex—ideas of the simplicial complex used here to give graph homology from a discrete set of N data points.

4. ***Estimate persistent homology of the topological properties of the Betti number*** (β_k): Analysis and description of the topological qualities of persistent homology for our data representation in blockchain involve identifying these spaces. From one space to another, this procedure is done by measuring their link by computing the k-dimensional hole as illustrated in figure 4.10, then examining the Betti numbers.

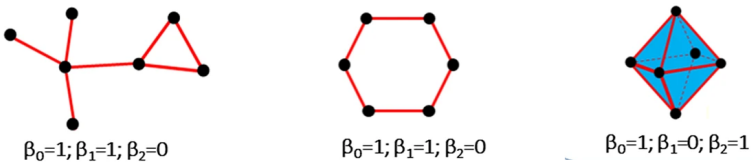

Figure 4.10: Topological invariant betti analysis

5. ***Build the Mathematical programming to fit the shard size and solve the problem using the Dual-simplex technique***: Throughout the preceding steps, Betti numbers are utilized to denote topological spaces. We acquire some graph structure as a type of homology after exhibiting the topological features based on the connectedness of k-dimensional simplicial complexes. Additionally, we provide some sharding with the size of each shard based on the threshold and number of k-dimensional holes. As a result, we have provided the framework of our parallel blockchain transaction system. The approach is based on mathematical programming and is used to determine the optimal number of shards by treating it as a linear programming problem and obtaining information about the multiple dimensions of the Betti number and k-SC. This optimization is approached using the dual-simplex technique. The following is a basic description of the LPP that is being proposed:

   The objective function

$$Min\#(shard) = C^T\beta\,,$$

   is subject to

$$A\beta \leq b$$

$$\beta_0, \beta_1, \beta_2, \beta_3 \geq 0\,.$$

More comprehensively our objective function that may be examined is the number of shards required to fragment the network according to the number of members and the link between the members that we deduced in TDA. Mathematically the following can be considered as the objective function (O.F)

$$Min\#(shard) = \sum_{(i=0)}^{k} c_i \beta_i \,,$$

which is subject to constrained by the information about the change in the topological properties by increasing the correlation coefficient (ϵ) between nodes.

$$A\beta \leq b$$

$$\beta_0, \beta_1, \beta_2, \beta_3 \geq 0 \,.$$

Here $C = [c_i]$ a vertical cost parameter vector that scores the values of the decision variables β_i. We choose C when specific epsilon (ϵ^*) makes all beta values close together, because of increasing the value of ϵ of the β_0 decreasing to one in general and high Betti numbers β_i increase starting from zero for any geometric graph constructing by simplicial complex.

The coefficient matrix A is usually taken for different cases as a constraint for variables in the objective.

In our issue b, the bound of constraint (right side of an equation $A\beta \leq b$) is inserted by mean square for β_j and k-simplicial simplex at ϵ_j with β^* and SC^* respectively at ϵ^*. Measure the distance of each state from the state we chose for the objective function. Between each case, it is constrained to a near-optimal situation. Then the vector $b = [b_j]$ is MSE where j is number of bounds determine through (how many different value to ϵ_j) each b_j will be calculated by

$$b_j = (\sum_{(i=0)}^{k} \beta_i - \beta^*) - (\sum_{(i=0)}^{k} SC_i - SC^*), \tag{4.1}$$

we have β^* and SC^* at ϵ^*

4.3 Practical Experimental Results and Challenges

When it comes to the byzantine environment, sharding is a well-known open problem [33], especially for the appropriate number of shards. We have shown that combinatorial TDA

tackle is secure and cost-effective even when confronted with byzantine adversaries, allowing transaction throughput to rise practically linearly with network processing capacity. We will explore the simulation for our proposed technique, detail the difficulties encountered in this study, and present the first solution to sharding in a partially synchronized environment. Forming shards in a blockchain system is usually more compound than in a distributed database:

- The nodes must be distributed fairly and randomly among the committees.
- Each committee's size and number of shards must be carefully chosen to create a compromise between performance and security.
- Finally, periodic committee assignments must be undertaken to prevent an adaptive attacker from compromising most nodes in a committee.

As such, this section will discuss our method for utilizing TDA, visualizing blockchain properties using combinatorial topology, and developing an unsupervised machine to handle these obstacles and solve them. To begin with, an internal blockchain network for containers was developed, as illustrated in the simulation Figure 3.5 and the data scheme representation for blockchain in Figure 4.9. 100 nodes will be used in the simulation. Each one has four randomly generated features generated using MATLAB simulation code, which attempt to approximate the values as closely as the blockchain can within the logical range of the BC data. In general, coding is accomplished by the use of TDA, MSE, and the Dual-simplex approach.

4.3.1 Returning to Simplicial complex

Persistent homology exists when the point cloud's homology remains constant throughout time as a result of the persistence of its topological properties in this space. To reinstate the simplicial complex into our approach, we increase the scale parameter ϵ, which controls the simplicial complex's regular modification. For each species, we may determine the homology groups and Betti numbers.

We have seen in Table 4.2 in betti curve as shown in Figure 4.12 that the $k - holes$ close to each other at the range when ϵ fall in $[0.1, 0.2]$, so we focus the analysis and step size in this line to give more suitable concrete k-holes and to decrease the mean square error as in Table 4.2

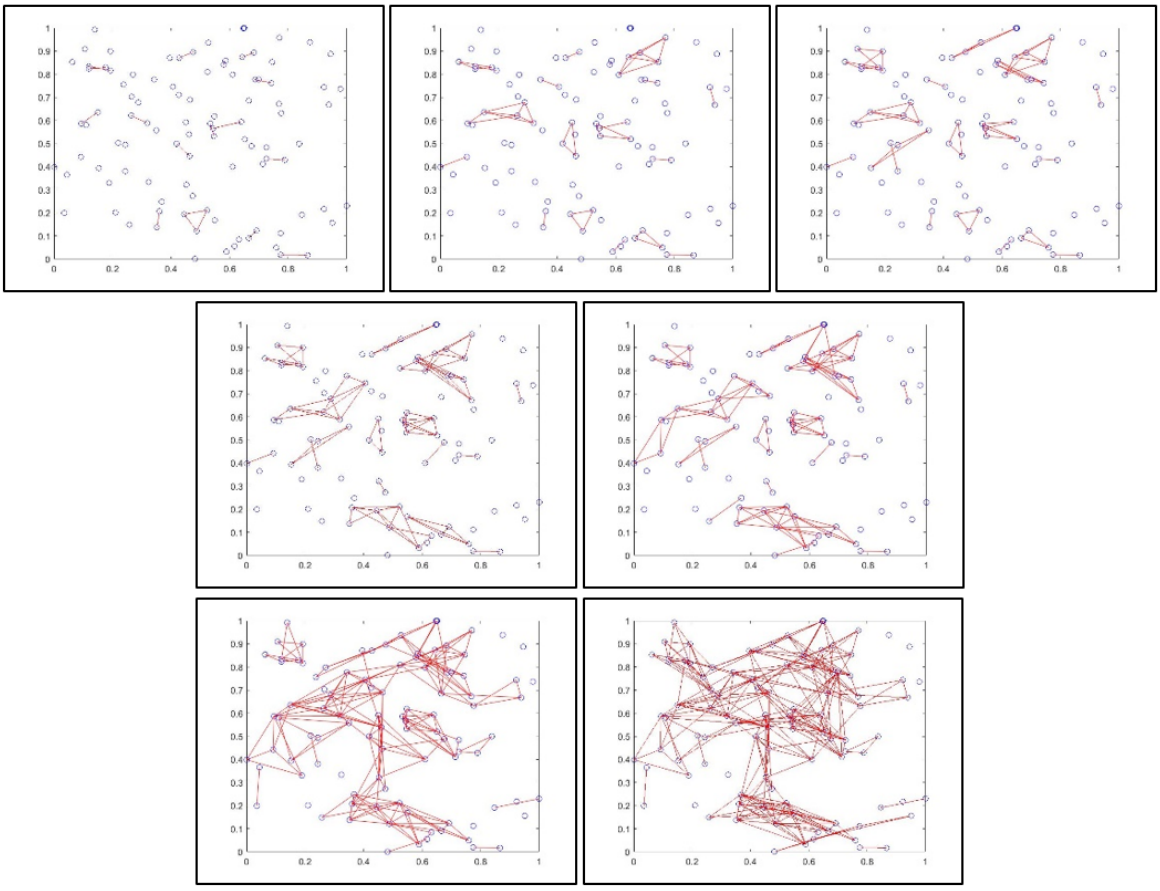

Figure 4.11: **Filtration**: Stages Simplicial model of the data with varying scale parameters ϵ in $\{0.1, 0.12, 0.14, 0.15, 0.16, 0.18, 0, 20\}$.

We attempt to include k-simplicial complex into the model as a decision variable to ensure that no information is omitted from the description and analysis. The epsilon effect, as measured by mean square error (MSE), is dependent on the k-simplicial complex. The MSE between the k-simplicial complex in the objective function row's coefficient and the k-simplicial complex for various epsilon values. Then, the MSE is set as a vector of b.

4.3.2 Generational distribution

In order to build a secure shard one needs a random integer, m, to feed the nodes for committee assignment. The nodes compute the betti number given m, which is the invariant perspective of blockchains and data distribution systems' topological characteristics. The recommended strategy begins by determining the number of shards, and then demonstrates the complementary method for connecting each node to its associated shard. The suggested algorithm is re-educated for each period by restructuring all of the previously specified sub-

Table 4.1: Numerical TDA simulation result in sample of $n = 100$ nodes and $d = 4$ features and $k = 0, 1, 2, 3$ dimension

ϵ	$Betti_0$	$Betti_1$	$Betti_2$	$Betti_3$	$\#0 - SC$	$\#1 - SC$	$\#2 - SC$	$\#3 - SC$
0	100	0	0	0	100	0	0	0
0.05	100	0	0	0	100	0	0	0
0.1	83	1	0	0	100	18	1	0
0.15	52	44	16	2	100	92	60	18
0.2	15	207	248	196	100	292	454	444
0.25	3	551	1365	2127	100	648	1912	3490
0.3	2	1125	5690	3000	100	1223	6815	23058

Table 4.2: depending on $c_i \geq c_{(i+1)}$, the focus the range $0.1 \leq \epsilon \leq 0.2$ with density analyze and MSE calculating by Eq (4.1)

epsilon	Betti-0	Betti-1	Betti-2	Betti-3	#0-SC	#1-SC	#2-SC	#3-SC	MSE
0.1	83	1	0	0	100	18	1	0	48.169
0.12	70	14	3	0	100	44	17	3	33.0832
0.14	61	25	5	0	100	64	30	5	21.5232
0.15	52	44	16	2	100	92	60	18	(O.F).Coef.
0.16	46	60	30	6	100	114	90	36	20.664
0.18	24	128	106	55	100	204	234	161	125.7666
0.2	15	207	248	196	100	292	454	444	306.8843

jects.

4.3.3 Shard Size m

In our context, learning refers to the algorithm for sharding a data structure based on its topological properties and the strength of persistent homology (Betti Number). We derive it using the data's combinatorial topological representation. Following that, we sample a network with n nodes to imitate the system's operation. As a result, we depict the design system with critical data information based on the persistent analysis homology (Betti Number) and the scale control parameter ϵ. Then, using mathematical programming, the proper #(shard) is calculated using β_k and k-SC, just as is done in an LPP. The dual-simplex approach is used to solve this optimization challenge. The following is a more detailed description of the

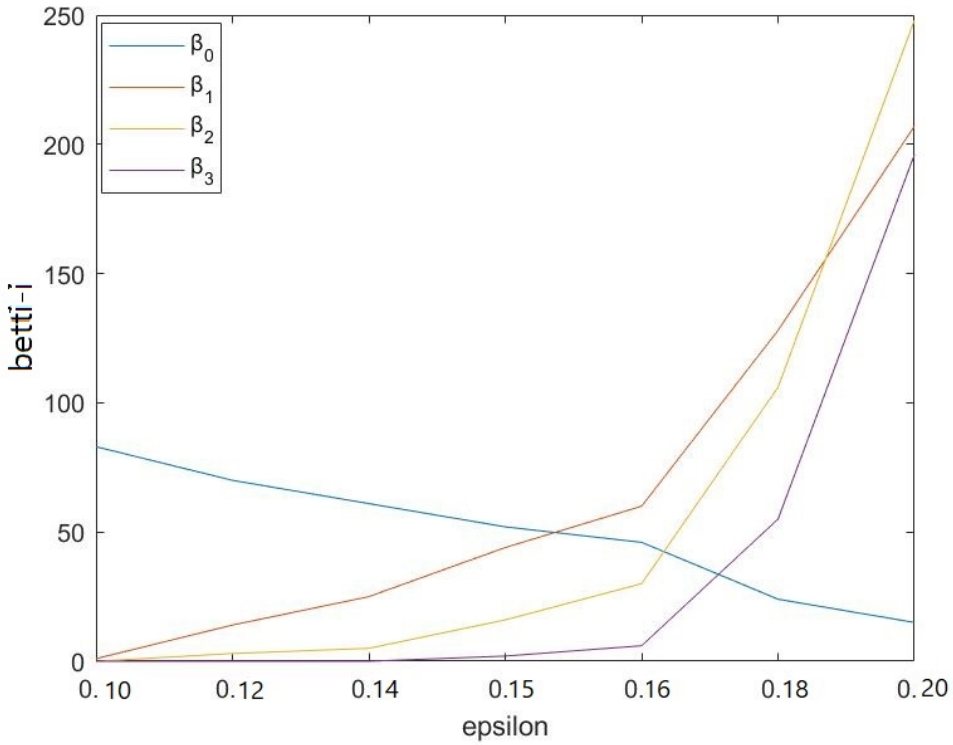

Figure 4.12: Topological invariant betti analysis

issue on an example:

The objective function

$$Min\#(shard) = \sum_{(i=0)}^{k} c_i \beta_i \,,$$

which is subject to

$$A\beta \leq b$$

$$\beta_0, \beta_1, \beta_2, \beta_3 \geq 0.$$

For the objective purpose, we can easily decide the number of shards by multiplying every β_i by a scale parameter according to their importance here. We have $c_i \geq c_{(i+1)}$ and each one possesses from the node into the main betti choosing: scale factor ϵ.

In our numerical case detailed in Table 4.2, we construct the model-based on the results founded by TDA.

$$min \quad z = 52\beta_0 + 44\beta_1 + 16\beta_2 + 2\beta_3$$

such that

$$83\beta_0 + \beta_1 \leq 48.169$$

$$70\beta_0 + 14\beta_1 + 3\beta_2 \leq 33.0832$$

$$61\beta_0 + 25\beta_1 + 5\beta_2 \leq 21.5232$$

$$48\beta_0 + 60\beta_1 + 39\beta_2 + 2\beta_3 \leq 20.664$$

$$24\beta_0 + 128\beta_1 + 196\beta_2 + 55\beta_3 \leq 125.7666$$

$$15\beta_0 + 207\beta_1 + 148\beta_2 + 196\beta_3 \leq 306.8843$$

$\beta_0, \beta_1, \beta_2, \beta_3 \geq 0$

The objective function coefficients, vector $C = [52\ \ 44\ \ 16\ \ 2]$ have β_i weights at epsilon $\epsilon^* = 0.15$. The Betti numbers for different ϵ will build the constrained. The prototype of proceeding mathematical programming is LPP, and the standard method to solve this problem is simplex methods. The simplex technique is dimensionally less objective and few constrained. Therefore, we use Dual-simplex methods, which are more efficient for large-scale linear optimization problems. Now the numerical result for the case of study is given using MATLAB simulations to solve the aforementioned mathematical programming problem:

- The magnitude of the affected decision variables (k-holes) for the objective

$$\beta_0 = 0.3087,\ \beta_1 = 0.1077,\ \beta_2 = 0,\ \beta_3 = 0$$

- The optimal recommended shard size 21, which is our main goal.

4.4 Shard Reconfiguration

As discussed in Section 4.3.2, all individuals are unified during each cycle of time (period). The data is reloaded into the machine to dynamically divide it into appropriate shard sizes using the exact mechanism as the TDA machine technology.

While disassembling the blockchain into shards, care should be taken to ensure that it contains a sufficient number of nodes to avoid jeopardizing the individual components' security and performing BFT or other operations.

It is a sensitive point that identifies the optimal amount of shards and nodes within the shard. If the blockchain's primary aim is to act as a currency system, it can support a low level of interoperability as long as there is adequate liquidity. Nonetheless, given the necessity of the blockchain, its operating system, and its flexibility in managing numerous applications, most notably financial transfers, we need to accelerate and improve the blockchain's efficiency in

terms of security, data distribution, information preservation, and information transfer speed. While disassembling the blockchain (preamble to sharding), care should be made to guarantee that it contains a sufficient number of nodes to avoid jeopardizing the individual components' security and ensuring that consensus mechanisms perform like BFT. The greater the number of nodes, the greater the possibility that honest nodes will be included in the system. However, as the number of nodes increases, BFT performance decreases. As a result, the number of nodes in each piece can be calculated following the chosen consensus process.

For instance, the BFT technique can be used to determine the number of nodes contained

C.P.	BFT	.	.	.	Advance BFT	.	.	Our approach
Meth.	NEO	EOS	Ripple	HyperLedger	HoneyBadger	ByzCoin	LinBFT	TDA
N_{nodes}	7	21	32	16	64	> 100	> 200	dynamic
Ref.	[83]	[84]	[54]	[85]	[47]	[50]	[86]	[current]
N_{shards}	$n/7$	$n/21$	$n/32$	$n/16$	$n/64$	$n/100$	$n/200$	*dynamic*

Table 4.3: Number of shards and nodes in a shard in a BFT technique

in each fragment. As such, they are to NEO 7 nodes, EOS 21 nodes, and so forth. Methods utilizing more advanced BFT can be used successfully with node numbers that are more important. For instance, HoneyBadger can operate properly with 64 nodes, as described in Table 4.3. As the number of components rises, the number of concurrent processes increases as well. As a result, the number of components can be defined by the number of nodes. Our technique maintains a dynamic equilibrium between the number of shards and the content of their nodes.

4.5 Properties of the scheme

4.5.1 Reducing the size of malicious nodes

As we can see, the suspicious nodes are somewhat similar to one another, which groups them together for the same purpose. This allows a problem to be confined and controlled, or dealt with as the attacking nodes. As a result, it cannot be connected to any other nodes. To minimize its efficacy, it is advisable to apply it with a high-spec shard.

A portion of the network has been hacked as a result of fraud or a hostile assault. Since all of the participants in a given shared network maintain one copy of the ledger's transactions, the shared network participants can determine what was altered by the fraudsters. The first point outlines the majority of the preventative measures necessary to overcome the mystique of the malicious node.

4.5.2 Comparison

Assume that we view the new machine learning technique as a technology that functions on the same premise as split or cluster techniques. The proposed protocol outperforms the K-means clustering methodology, which is considered to be one of the most fundamental and widely used clustering strategies.

For several critical reasons:

The first is to obviate the need for several calculations between vectors and at each step. The second reason is that we are aware of and regulate the number of appropriate partitions. The disadvantage is the presence of some nodes in several shards. That is a positive point, because we regard dividing into shards as partial or fuzzy on one hand, because we take the $(k = 0, 1, 2, 3)$ hole into account. On the other hand, this contract enables us to inform the final central committee between shards about the transfer activities and to include them into the blockchain as a fund. Thirdly, our technique allows for a factor of ϵ controlling over the number of shards.

4.6 Summary and Discussion

Scalability is a concern for blockchain technology, as the networks may not be able to handle the increasing number of data and transactions as more businesses join the platform. Sharding is a method of dividing the network burden into portions in order to achieve latency-free scalability, which allows the blockchain to process more transactions. The TDA of blockchain is investigated in this study and we report how to determine the quantity of the network's connection and interdependence between nodes using simplicial complex (SC) computation which indeed describes the network's topological properties (persistent homology (PH)). We next attempt to model this relationship in order to have the LPP and solve it using recognized methods. Also in light of network modification the correct number of shards (multi mini-network) is recognised. Hence, the most important step, namely the de-

termination of the appropriate shard number, is completed before starting a sharding procedure.

5 CONCLUSION

The findings of this study are crucial because blockchain technology can boost the speed and flow of labour. Each peer-to-peer node validates each block independently under the current implementation of blockchain technology. The scaling bottleneck is striking a balance between the size of each block and the planned time necessary to produce it while allowing for network latency. Our findings of blockchain's complete updating prognosis are typically consistent with scalability. We added two new scaling options to the blockchain to address scaling challenges caused by rising transaction demand per second.

The blockchain system requires that all participants agree on what should be added to the ledger based on a set of predefined rules. In particular, with scaling solutions they may reach a consensus on most nodes in the network, which must be implemented flawlessly through active participant monitoring and screening. One method is to reach a consensus through the use of a trust committee (TC). We introduce TC in this segment, a novel method for identifying trustworthy committees (TCs) that use PSO. In order to test the efficiency and applicability of this proposed method we use the simulations. After multiple tests, it was determined that the model effectively picks and updates TC. It could be an excellent future project to find better coefficients with new distinctive characteristics for the velocity update equation.

Consensus-building via a responsible committee is one technique that considers committee security. Using the Grey Wolf optimizer (GWO) algorithm, consequently we first revealed an innovative strategy for excluding nodes that attack a trust committee (TC) node. The illustrated model efficiently picks and changes the TC. By identifying hostile nodes and barring them from entering the committee, the committee's security enhanced its reliability.

Scalability is an issue with blockchain technology as the networks may struggle to handle the increasing number of data and transactions as more businesses join the platform. The second component of the research problem was sharding; sharding is a technique for partitioning the network's load into smaller pieces in order to achieve latency-free scalability, allowing the blockchain to process more transactions. The simplicial complex (SC) calculation is used to define the topological aspects of the blockchain network, such as persistent homology (PH), and to determine the network's interaction and interdependence between nodes. Then, using these aspects, we attempted to model this relationship in order to obtain and solve the LPP problem using well-known methods, as well as to calculate the integer number of multi-mini-

networks (shards) in the network in response to network changes. This concludes the most essential stage prior to the participation phase, which is determining the ideal shard number.

5.1 Future work

- Based on our extensive study and in addition to what was mentioned in Section 2.5, more platforms can be produced for blockchain. It is possible to find many new features that illustrate the blockchain platform invented for different applications in several sectors and required fields. Developing sides include improvement in interfaces and access, cryptography, distributed consensus and network structure.

- Blockchain technology simulation of our methods in an enterprise environment would be good future work. They provide a multitude of alternatives in a variety of application settings. Industries, the finance sector, trading, and supply chains are all areas of interest, some of them are mentioned in Table 2.1.

- Another direction of the research area could be that the blockchain technology can be implemented within the supply chain and Internet of Things.

- Directed Acyclic Graph(DAG) could be studied as it serves scaling in Ethereum type of blockchain.

- Applying fuzzy artificial neural networks in the sharding of the blockchain as in Chapter 4 would be a good future work, where scaling the blockchain with fuzzy logic by estimating the nodes' reputation using a fuzzy representation could be used.

References

[1] C Vijai, SM Suriyalakshmi, and D Joyce. The blockchain technology and modern ledgers through blockchain accounting. *Adalya Journal*, 8(12), 2019.

[2] https://www.n-ix.com/deep-learning-vs-machine-learning/.

[3] https://medium.com/@iamterryclark/swarm-intelli-eb5e46eda0c3.

[4] Kutaiba Sabah Nimma, Monaaf DA Al-Falahi, Hung Duc Nguyen, SDG Jayasinghe, Thair S Mahmoud, and Michael Negnevitsky. Grey wolf optimization-based optimum energy-management and battery-sizing method for grid-connected microgrids. *Energies*, 11(4):847, 2018.

[5] Rob Shields. Cultural topology: The seven bridges of königsburg, 1736. *Theory, Culture & Society*, 29(4-5):43–57, 2012.

[6] https://www.dnsstuff.com/what-is-network-topology.

[7] Afra Zomorodian. Topological data analysis. *Advances in applied and computational topology*, 70:1–39, 2012.

[8] Ahmet Bugday, Adnan Ozsoy, Serdar Murat Öztaner, and Hayri Sever. Creating consensus group using online learning based reputation in blockchain networks. *Pervasive and Mobile Computing*, 59:101056, 2019.

[9] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system bitcoin: A peer-to-peer electronic cash system. *Bitcoin. org. Disponible en https://bitcoin. org/en/bitcoin-paper*, 2009.

[10] Markus Jakobsson and Ari Juels. Proofs of work and bread pudding protocols. In *Secure information networks*, pages 258–272. Springer, 1999.

[11] Alex Biryukov, Daniel Feher, and Dmitry Khovratovich. Guru: Universal reputation module for distributed consensus protocols. Technical report, University of Luxembourg, 2017.

[12] Adam Back et al. Hashcash-a denial of service counter-measure. 2002.

[13] John R Douceur. The sybil attack. In *International workshop on peer-to-peer systems*, pages 251–260. Springer, 2002.

[14] Md Sadek Ferdous, Mohammad Jabed Morshed Chowdhury, Mohammad A Hoque, and Alan Colman. Blockchain consensus algorithms: A survey. *arXiv preprint arXiv:2001.07091*, 2020.

[15] Damilare Peter Oyinloye, Je Sen Teh, Norziana Jamil, and Moatsum Alawida. Blockchain consensus: An overview of alternative protocols. *Symmetry*, 13(8):1363, 2021.

[16] Gavin Wood et al. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper*, 151(2014):1–32, 2014.

[17] Leslie Lamport, Robert Shostak, and Marshall Pease. The byzantine generals problem. In *Concurrency: the Works of Leslie Lamport*, pages 203–226. 2019.

[18] Fran Casino, Thomas K Dasaklis, and Constantinos Patsakis. A systematic literature review of blockchain-based applications: Current status, classification and open issues. *Telematics and informatics*, 36:55–81, 2019.

[19] Christian Berger and Hans P Reiser. Scaling byzantine consensus: A broad analysis. In *Proceedings of the 2nd Workshop on Scalable and Resilient Infrastructures for Distributed Ledgers*, pages 13–18, 2018.

[20] Marwan Jameel and Oğuz Yayla. Pso based blockchain committee member selection. In *2021 6th International Conference on Computer Science and Engineering (UBMK)*, pages 725–730. IEEE, 2021.

[21] Eleftherios Kokoris-Kogias, Philipp Jovanovic, Linus Gasser, Nicolas Gailly, Ewa Syta, and Bryan Ford. Omniledger: A secure, scale-out, decentralized ledger via sharding. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 583–598. IEEE, 2018.

[22] Hung Dang, Tien Tuan Anh Dinh, Dumitrel Loghin, Ee-Chien Chang, Qian Lin, and Beng Chin Ooi. Towards scaling blockchain systems via sharding. In *Proceedings of the 2019 international conference on management of data*, pages 123–140, 2019.

[23] Ahmet Bugday, Adnan Ozsoy, and Hayri Sever. Securing blockchain shards by using learning based reputation and verifiable random functions. In *2019 International Symposium on Networks, Computers and Communications (ISNCC)*, pages 1–4. IEEE, 2019.

[24] David Galindo, Jia Liu, Mihai Ordean, and Jin-Mann Wong. Fully distributed verifiable random functions and their application to decentralised random beacons. *IACR Cryptol. ePrint Arch.*, 2020:96, 2020.

[25] Yossi Gilad, Rotem Hemo, Silvio Micali, Georgios Vlachos, and Nickolai Zeldovich. Algorand: Scaling byzantine agreements for cryptocurrencies. In *Proceedings of the 26th symposium on operating systems principles*, pages 51–68, 2017.

[26] Silvio Micali, Michael Rabin, and Salil Vadhan. Verifiable random functions. In *40th annual symposium on foundations of computer science (cat. No. 99CB37039)*, pages 120–130. IEEE, 1999.

[27] İsmet Yurduşen Marwan Jameel, Oğuz Yayla. Combinatorial topology to develop a machine learning technique for blockchain sharding. *AIP2020, pages=–, year=2020.*

[28] Ralph C Merkle. Protocols for public key cryptosystems. In *1980 IEEE Symposium on Security and Privacy*, pages 122–122. IEEE, 1980.

[29] Ralph C Merkle. Method of providing digital signatures, 1 1982. *US Patent US*, 4309569.

[30] Georg Becker. Merkle signature schemes, merkle trees and their cryptanalysis. *Ruhr-University Bochum, Tech. Rep*, 2008.

[31] Kyle Croman, Christian Decker, Ittay Eyal, Adem Efe Gencer, Ari Juels, Ahmed Kosba, Andrew Miller, Prateek Saxena, Elaine Shi, Emin Gün Sirer, et al. On scaling decentralized blockchains. In *International conference on financial cryptography and data security*, pages 106–125. Springer, 2016.

[32] Alin Tomescu, Ittai Abraham, Vitalik Buterin, Justin Drake, Dankrad Feist, and Dmitry Khovratovich. Aggregatable subvector commitments for stateless cryptocurrencies. In *International Conference on Security and Cryptography for Networks*, pages 45–64. Springer, 2020.

[33] Loi Luu, Viswesh Narayanan, Chaodong Zheng, Kunal Baweja, Seth Gilbert, and Prateek Saxena. A secure sharding protocol for open blockchains. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 17–30, 2016.

[34] Gavin Andresen. Bitcoin improvement proposal 101, 2015.

[35] Omer Bobrowski and Matthew Kahle. Topology of random geometric complexes: a survey. *Journal of applied and Computational Topology*, 1(3):331–364, 2018.

[36] Vitalik Buterin, Jeff Coleman, and Matthew Wampler-Doty. Notes on scalable blockchain protocols (verson 0.3), 2015.

[37] Chaitanya Bapat. Blockchain for academic credentials. *arXiv preprint arXiv:2006.12665*, 2020.

[38] Dimiter V Dimitrov. Blockchain applications for healthcare data management. *Healthcare informatics research*, 25(1):51–56, 2019.

[39] Ori Jacobovitz. Blockchain for identity management. *The Lynne and William Frankel Center for Computer Science Department of Computer Science. Ben-Gurion University, Beer Sheva*, 2016.

[40] Maurice Clerc. Discrete particle swarm optimization, illustrated by the traveling salesman problem. In *New optimization techniques in engineering*, pages 219–239. Springer, 2004.

[41] Leila Ismail and Huned Materwala. A review of blockchain architecture and consensus protocols: Use cases, challenges, and solutions. *Symmetry*, 11(10):1198, 2019.

[42] Sunny King and Scott Nadal. Ppcoin: Peer-to-peer crypto-currency with proof-of-stake. *self-published paper, August*, 19(1), 2012.

[43] Thomas Kerber, Aggelos Kiayias, Markulf Kohlweiss, and Vassilis Zikas. Ouroboros crypsinous: Privacy-preserving proof-of-stake. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 157–174. IEEE, 2019.

[44] LM Goodman. Tezos: A self-amending crypto-ledger position paper. *Aug*, 3:2014, 2014.

[45] Jonah Brown-Cohen, Arvind Narayanan, Alexandros Psomas, and S Matthew Weinberg. Formal barriers to longest-chain proof-of-stake protocols. In *Proceedings of the 2019 ACM Conference on Economics and Computation*, pages 459–473, 2019.

[46] Miguel Castro, Barbara Liskov, et al. Practical byzantine fault tolerance. In *OSDI*, volume 99, pages 173–186, 1999.

[47] Andrew Miller, Yu Xia, Kyle Croman, Elaine Shi, and Dawn Song. The honey badger of bft protocols. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 31–42, 2016.

[48] Alysson Neves Bessani and Marcel Santos. Bft-smart-high-performance byzantine-faulttolerant state machine replication, 2011.

[49] Vincent Gramoli. From blockchain consensus back to byzantine consensus. *Future Generation Computer Systems*, 107:760–769, 2020.

[50] Eleftherios Kokoris Kogias, Philipp Jovanovic, Nicolas Gailly, Ismail Khoffi, Linus Gasser, and Bryan Ford. Enhancing bitcoin security and performance with strong consistency via collective signing. In *25th {usenix} security symposium ({usenix} security 16)*, pages 279–296, 2016.

[51] Rafael Pass and Elaine Shi. Hybrid consensus: Efficient consensus in the permissionless model. In *31st International Symposium on Distributed Computing (DISC 2017)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017.

[52] Ittai Abraham, Dahlia Malkhi, Kartik Nayak, Ling Ren, and Alexander Spiegelman. Solidus: An incentive-compatible cryptocurrency based on permissionless byzantine consensus. *CoRR, abs/1612.02916*, 2016.

[53] George Danezis and Sarah Meiklejohn. Centrally banked cryptocurrencies. *arXiv preprint arXiv:1505.06895*, 2015.

[54] Marcel T Rosner and Andrew Kang. Understanding and regulating twenty-first century payment systems: The ripple case study. *Mich. L. Rev.*, 114:649, 2015.

[55] David Mazieres. The stellar consensus protocol: A federated model for internet-level consensus. *Stellar Development Foundation*, 32, 2015.

[56] Zaiqing Nie, Yuanzhi Zhang, Ji-Rong Wen, and Wei-Ying Ma. Object-level ranking: bringing order to web objects. In *Proceedings of the 14th international conference on World Wide Web*, pages 567–574, 2005.

[57] Sepandar D Kamvar, Mario T Schlosser, and Hector Garcia-Molina. The eigentrust algorithm for reputation management in p2p networks. In *Proceedings of the 12th international conference on World Wide Web*, pages 640–651, 2003.

[58] Li Xiong and Ling Liu. Peertrust: Supporting reputation-based trust for peer-to-peer electronic communities. *IEEE transactions on Knowledge and Data Engineering*, 16(7):843–857, 2004.

[59] Gang Wang, Zhijie Jerry Shi, Mark Nixon, and Song Han. Sok: Sharding on blockchain. In *Proceedings of the 1st ACM Conference on Advances in Financial Technologies*, pages 41–61, 2019.

[60] Ewa Syta, Philipp Jovanovic, Eleftherios Kokoris Kogias, Nicolas Gailly, Linus Gasser, Ismail Khoffi, Michael J Fischer, and Bryan Ford. Scalable bias-resistant distributed randomness. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 444–460. Ieee, 2017.

[61] John Paul Mueller and Luca Massaron. *Machine learning for dummies*. John Wiley & Sons, 2021.

[62] TK Balaji, Chandra Sekhara Rao Annavarapu, and Annushree Bablani. Machine learning algorithms for social media analysis: A survey. *Computer Science Review*, 40:100395, 2021.

[63] Goëry Genty, Lauri Salmela, John M Dudley, Daniel Brunner, Alexey Kokhanovskiy, Sergei Kobtsev, and Sergei K Turitsyn. Machine learning and applications in ultrafast photonics. *Nature Photonics*, 15(2):91–101, 2021.

[64] Lukas Tuggener, Mohammadreza Amirian, Katharina Rombach, Stefan Lörwald, Anastasia Varlet, Christian Westermann, and Thilo Stadelmann. Automated machine learning in practice: state of the art and recent results. In *2019 6th Swiss Conference on Data Science (SDS)*, pages 31–36. IEEE, 2019.

[65] B. Grünbaum and Raj C Bose. combinatoric. *Encyclopedia Britannica.*, 2013.

[66] Gil Kalai, Isabella Novik, Francisco Santos, and Volkmar Welker. Geometric, algebraic, and topological combinatorics. *Oberwolfach Reports*, 16(3):2395–2472, 2020.

[67] Richard Bellman, Roger Fletcher, Ronald A Howard, Fritz John, Narendra Karmarkar, William Karush, Leonid Khachiyan, Bernard Koopman, Harold Kuhn, László Lovász, et al. Mathematical optimization source: en. wikipedia. org/wiki/mathematical_optimization.

[68] Richard Bellman, Roger Fletcher, Ronald A Howard, Fritz John, Narendra Karmarkar, William Karush, Leonid Khachiyan, Bernard Koopman, Harold Kuhn, László Lovász, et al. From wikiprojectmed.

[69] Leonora Bianchi, Marco Dorigo, Luca Maria Gambardella, and Walter J Gutjahr. A survey on metaheuristics for stochastic combinatorial optimization. *Natural Computing*, 8(2):239–287, 2009.

[70] Christian Blum and Andrea Roli. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM computing surveys (CSUR)*, 35(3):268–308, 2003.

[71] Chris Rorres and Howard Anton. *Elementary linear algebra: applications version*. Wiley, 1994.

[72] Russell Eberhart and James Kennedy. Particle swarm optimization. In *Proceedings of the IEEE international conference on neural networks*, volume 4, pages 1942–1948. Citeseer, 1995.

[73] Gerardo Beni and Jing Wang. Swarm intelligence in cellular robotic systems. In *Robots and biological systems: towards a new bionics?*, pages 703–712. Springer, 1993.

[74] Seyedali Mirjalili, Seyed Mohammad Mirjalili, and Andrew Lewis. Grey wolf optimizer. *Advances in engineering software*, 69:46–61, 2014.

[75] Uta Maria Jürgens and Paul MW Hackett. Wolves, crows, and spiders: An eclectic literature review inspires a model explaining humans' similar reactions to ecologically different wildlife. *Frontiers in Environmental Science*, 9:3, 2021.

[76] Joseph Muscat, David Buhagiar, et al. Connective spaces. *Mem. Fac. Sci. Eng. Shimane Univ. Series B: Math. Sci*, 39:1–13, 2006.

[77] Amir Bashan, Ronny P Bartsch, Jan W Kantelhardt, Shlomo Havlin, and Plamen Ch Ivanov. Network physiology reveals relations between network topology and physiological function. *Nature communications*, 3(1):1–9, 2012.

[78] Frédéric Chazal and Bertrand Michel. An introduction to topological data analysis: fundamental and practical aspects for data scientists. *arXiv preprint arXiv:1710.04019*, 2017.

[79] Herbert Edelsbrunner. Persistent homology: theory and practice. 2013.

[80] Afra Zomorodian. Fast construction of the vietoris-rips complex. *Computers & Graphics*, 34(3):263–271, 2010.

[81] Jean-Daniel Boissonnat and Monique Teillaud. Effective computational geometry for curves and surfaces. 2006.

[82] Cecil Jose A Delfinado and Herbert Edelsbrunner. An incremental algorithm for betti numbers of simplicial complexes on the 3-sphere. *Computer Aided Geometric Design*, 12(7):771–784, 1995.

[83] Elad Elrom. Neo blockchain and smart contracts. In *The Blockchain Developer*, pages 257–298. Springer, 2019.

[84] Ian Grigg. Eos-an introduction. *White paper. https://whitepaperdatabase. com/eos-whitepaper*, 2017.

[85] Christian Cachin et al. Architecture of the hyperledger blockchain fabric. In *Workshop on distributed cryptocurrencies and consensus ledgers*, volume 310. Chicago, IL, 2016.

[86] Yin Yang. Linbft: Linear-communication byzantine fault tolerance for public blockchains. *arXiv preprint arXiv:1807.01829*, 2018.