Scalability Encumbrances in Blockchain Technology using Fuzzy Logic

D.A. Aina
Department of Computer Science,
College of Computing,
McPherson University, Seriki Sotayo,
Ogun State, Nigeria

A.O. Ogunjobi
Department of Computer Science,
College of Natual Sciences,
Federal University of Agriculture, Abeokuta,
Ogun State, Nigeria

A.M. Falade
Department of Software Engineering,
College of Computing,
McPherson University, Seriki Sotayo,
Ogun State, Nigeria

O.A. Odugbesan
Department of Computer Science,
College of Computing,
McPherson University, Seriki Sotayo,
Ogun State, Nigeria

ABSTRACT
Blockchain technology has emerged as a transformative innovation with applications across finance, healthcare, supply chain, and education. However, one of its persistent challenges is transaction speed and scalability, which limit its efficiency and adoption in high-volume environments. Existing research has explored various approaches, including consensus algorithm optimization, sharding, and off-chain solutions, but these methods often face issues of complexity, high cost, or limited adaptability. In this study, a fuzzy logic model is proposed to enhance transaction speed and improve scalability on blockchain networks. Using a dataset sourced from Kaggle, the research employed data collection, cleaning, and preprocessing, followed by the application of fuzzy logic through a triangular membership function that incorporates three key parameters: transaction fee, transaction speed, and block density. The model was implemented in Python using Scikit-fuzzy, supported by NumPy, Pandas, and Matplotlib for computation, data manipulation, and visualization. Results demonstrate that the fuzzy logic approach provides a flexible and efficient mechanism for decision-making within blockchain transactions, outperforming conventional analytical methods in adaptability and ease of implementation. The findings indicate that integrating fuzzy logic into blockchain systems can streamline transaction processing, enhance scalability, and support the design of more efficient consensus mechanisms. This work contributes to the growing body of knowledge on intelligent blockchain optimization and provides a foundation for further research integrating fuzzy logic with advanced consensus methods.

Keywords
Blockchain technology; Fuzzy logic; Transaction speed; Scalability; Triangular membership function; Python; Consensus algorithms

1. INTRODUCTION
Lotfi Zadeh invented fuzzy logic in the sixties and it is said to be sort of many valued logics, it stemmed from the fuzzy set theory. It is meant to deal with reasoning that is not exact but is in the vicinity. Unlike classical logic where the variables could simply be settled to be true or false, fuzzy logic variables have a truth value typical of between 0 and 1. This characteristic makes computers to act like human; thus, it is applicable in various sectors where linear thinking is irrelevant. Fuzzy logic is different from the classical Boolean logic since it has values in between an interval of 0 and 1. It is used to deal with at least one involvement within the concept of partial truth where something is a little true and a little false. Fuzzy logic models also commonly referred to as fuzzy sets are calculating tools that might capture vague and imprecise data. On this estimate, these models can handle information that is often vague and uncertain (Martani *et al.,* 2024). Blockchain technology is known for decentralization and its unchangeable ledger systems and inevitably introduces problems with scalability, transaction throughput, and resource consumption. These difficulties can be solved through the integration of fuzzy logic in blockchain which bring in smart and adaptive consensus. Fuzzy logic could be applied to enhance the choice between main resources, to organize the load of transactions more efficiently, and to enhance the decision-making process which performs within the consensus algorithms. Organization and transaction can also be improved with the help of the fuzzy logic to score fee in correlation to the state of the network (Jaoude and Saadé, 2019). For instance, the system may increase the charges charged for a faster processing time during the congestion of the network and reduce the charge during the least network congestion. Through this integration, it is possible to optimize the blockchains' performance by looking for easier, and faster, transactions (Alazzawi, 2021).

According to previous researches one major issue faced by public blockchains is the issue of the speed of transactions and it limits the efficiency in handling increasing volumes. Several factors affect the performance of the blockchain systems which includes the following; transaction fees and block density. These forms of interactions are very difficult or even impossible to quantify conventionally, which makes traditional analytical tools ineffective in modeling these real-world dynamics. It is for this reason that enhanced approaches are required in order to interpret and assess these interactions (Martani *et al*., 2024).

The purpose of this research is to develop a platform that leverages fuzzy logic to provide users with a smooth and efficient medium for conducting blockchain transactions. Objectives of this study are to:

i. Determine a blockchain technology that helps transactions to be much smoother and faster;

ii. identify and analyze the key challenges hindering the scalability in blockchain technology;

iii. enhance the scalability of blockchain.

Significance of Study

The speed of transactions is one of the major limitations of current blockchain systems among others. This study is aimed at analysing and increasing efficiency in processing transaction speeds. Better rates for the processing activity would, therefore, improve user experience and make blockchain more useful to different scenarios. Fuzzy logic's integration with blockchain technology is comparatively new. This study demonstrates the use of fuzzy logic as a flexible and adaptable method to problem-solving when dealing with uncertainty and imprecision in blockchain transactions. This invention may stimulate more study and advancement in the area.

2. LITERATURE REVIEW

Introduction to Fuzzy Logic

Fuzzy logic was first proposed by Zadeh (1965) of the University of California at Berkeley. In the year 1975 he then expanded the work in the paper that proposed the concept of *linguistic variables* also called fuzzy sets. Abdel-Aleem et al. (2017) implemented fuzzy and adaptive neuro fuzzy systems in optimization of production inventory problem. Demey and Wolff (2016) determined a model-based searching strategy for inventory management system. Fuzzy logic is a type of many-valued logic in which the truth value of variables might be any real integer from 0 to 1. It is used to manage the concept of partial truth, in which the truth value might range from totally true to completely false (Taylor, 2021).

Fuzzy Sets

It can be seen in classical sets that there is no uncertainty, therefore they have crisp boundaries. A classical set could be stated as: $A = \{x \mid x > 4\}$ but a fuzzy set is an extension of a classical set since uncertainty occurs thereby making boundaries vaguely specified Zadeh (1965).

Fuzzy Logic Operators

Fuzzy Logic replaces traditional Boolean operators (AND, OR, NOT) with fuzzy operators and the commonly used operators are:

MIN for AND
MAX for OR
1-x for NOT
x*y for AND
1- (1-x) *(1-y) for OR

Fuzzy logic operators are mainly used to handle imprecise and vague information in decision making processes, allowing for more nuanced and human-like reasoning compared to traditional binary logic (Liu, 2017).

Satoshi Nakamoto wrote for the creation of Bitcoin in 2009 and initiated the use of block chain. Bitcoin aimed at becoming a near-mission substitute to fiat money, a safe, worldwide, decentralized currency that may be exchanged for other goods and services. In the first year, the currency was worth $0. In 2018, it has a total market capitalization of $126 million. Nakamoto constructed upon the work of his predecessors. Although Stuart Haber and W. Scott Stornetta had started developing a cryptographically protected chain of blocks prior to the existence of bitcoin, the first blockchain wasn't fully conceived until Nakamoto's conception in 2008. The Bitcoin whitepaper was made available for download and became open source, enabling anybody with an inclination to expand upon the programming already in place. As a result, other "altcoins" cryptocurrencies that aimed to outperform Bitcoin were released. Namecoin and Litecoin were two well-known early adopters of the Bitcoin architecture (Alkhdour *et al*., 2024). Fuzzy logic was first introduced by Lotfi Zadeh in 1965 in the context of the theory of fuzzy sets. A fuzzy set assigns a degree of membership, usually a real number within an interval, to elements of a universe. Fuzzy logic is a method of processing variables that enables the consideration of multiple possible truth values for the same variable. It aims to address problems with ambiguous and imprecise data, utilizing heuristics to derive a range of accurate conclusions (Rani *et al.,* 2024).

Blockchain

Blockchain is a structural formation based on blocks, each of which contains some data with respect to transactions. Hardware and software requirements have to be known in order to be able to explain how the data in blockchain is arranged and where it is stored. A blockchain is a spread record or database that is copied to all computers in a particular computer network. Despite their usage in other fields, they are most famous for acting out the crucial role of providing the cryptocurrency systems with a secure and decentralized way of record-keeping. Blockchain safeguards the transactions and guarantees that information can only be inputted with no alteration by employing cryptographic approaches. The blockchain systems are safe if and only if, the persons designing the protocols have enough know-how of cryptography. Smart contracts refer to automated contracts that are basically written on a blockchain. It facilitates automated transactions and eliminates the additional dealer’s middlemen. Ethereum Smart Contracts need to be programmed in a language, one of which is Solidity, a language dedicated to creating smart contracts (DeNio and Ludwig, 2021).

The world is linked frequently than it was in the past and all this we have the Internet to thank for. Most of the existing computer-based digital resources like databases servers and other various systems and services are now connected over the Internet for the last few years. Such connectivity helps a lot in offering many new features but the security aspect of unauthorized entry is also introduced. To avoid this, access control systems are used. From the concept of computer security, access control systems are employed to regulate the access that is granted to specific data or material. In general, resource owners’ solution is to implement their own AC systems or pay for such services to a reliable third party (Yadav and Shevkar, 2021).

Even so, these solutions have problems all these mentioned solutions, which have the upper advantages, have problems. It becomes a stress to the resource owners, for you have to initiate, implement and regulate your own access control configuration. Hence, a number of owners of resources do not bother to perform the method themselves but get a reliable third party to do it. However, this is done with the danger of having a single point of failure, and leakage of personal details since only the third party has the ability to grant access to the resources. So, in the more recent literature, there are introduced systems of access control based on the blockchain that claim to address all of these problems.

Blockchain is a relatively young technology that was brought up in the white paper for Bitcoin cryptocurrency designed in 2008 by Satoshi Nakamoto. There is no information who Nakamoto is, maybe it is a team of developers or only one developer. The concept of a blockchain mainly revolves around it being a digital ledger or a database that is installed in the nodes of a computer network. This means that it is actually decentralized, and the information of the network is distributed to all the nodes hence eliminating the provision of the one central control point (Ismail and Materwala, 2019).

Apparently, the blockchain access control systems are not similar to the traditional ones in the way that the latter grants the final control to the holder of a certain resource. The owners of resources can decide whom among the other players have the right to access the resources owned by the owner. It means that, there are multiples of authorizations to certain resources. It is evident that users who have access to the resources could transfer this right of access to other users if the policies that originated from the owner of the resource are cared for. Therefore, the transfer of resource rights is highly transparent since the information is relayed on the blockchain (Biswas *et al.,* 2019).

Scalability

Scalability as defined in the general sense is the ability of the blockchain system to handle increased traffic in terms of throughput and transaction numbers without compromising on the security of the network. It is an issue meaningful because as the number of blocks increases along with the ledger, the network becomes larger and more complex. As such, there could be issues of scale with a larger amount of transactions – that is processing time and cost will increase. Scalability, therefore, refers to a system's, a networks or a process's ability to handle work or resources in larger quantities without necessitating radical adjustive measures. As to scalability, it means the ability of a system to increase and develop in response to increased load intensity without negatively influencing its work results and quality of service. The scalability is in many cases crucial in many fields where the development and expansion are usual objectives, for example in the technologies, business, infrastructures (Mehbodniya *et al.,* 2021).

Scalability in technology

Usually, scalability in technology addresses the ability of a given system's hardware, database or a software application to accommodate the increasing loads or user traffic without compromising on the level of performance. Likewise, when there is increase in the usage by the users such as high hours, the web application should be able to handle this without necessarily slowing or crashing. This is about the capability that any computerized program or physical computer gadget or software or any other tool (hardware or software) has to retain functionality even when it has been expanded or shrunk to meet a user's demand. The rescaling is often done to a higher volume or size. The displacement or shifting is normally done upwards. The concept of rescaling applies to a product that is a line of computer systems that equipped with different ranges of RAM, storage, and many other parts. Otherwise, the scalable item has to be migrated to another context, for instance, an operating system (Matina, 2024).

Basic Principles of Blockchain

Blockchain is essentially a type of distributed ledger. Distributed ledgers are databases that can be synchronized and shared across multiple sites, geographies, or institution. A distributed ledger employs a cryptographic log with all the records of the data in it. It is recorded and updated by the various stakeholders who are involved in the actualization of the distributed ledger while all the participants have an ability to access it. This also implies that no party ever has control over the information or the data. Previously, blockchain is the technique to achieve the Distributed Ledger Technology, while not all DLTs are Blockchain. As it will be seen in a distributed ledger technology, the data structure does not have to be of blocks necessarily. From an analysis of the flow of data, it only undergoes verification once it is inputted to the chain and not when new blocks are added to the chain. This difference is merely a distinction between the two, and that depicts how closely related the two are (Taylor, 2020).

That was the main idea when Satoshi Nakamoto introduced Bitcoin as such type of distributed peer-topeer ledger. However, there are also some other principles that are important and they include; transparency with anonymity. Meaning, any operation on the blockchain needs to be registered by various nodes in the chain. Thus, all the nodes of a network can view information and what participants are exchanging in the course of transferring data. As a result, this makes the handling of the block chain irrevocable in its nature. Also, each and every transaction will be recorded permanently on the block because each block is connected with another block consequently forming a chain. Finally, the blockchain should be programmable. Thus, the users can create their own algorithms and smart contracts to comply with the agreed-upon condition. Smart contracts are contracts that are uploaded on the blockchain, and once both the parties sign it the contract will be executed automatically. If blockchains which hold smart contracts are programmable then anyone can code their own. Smart contract plays a crucial role in the access control based on blockchain since the permissions and conditions can be monitored and can also be enforced (Rani, 2024).

Structure of the blockchain

Blockchain's structure, as the name implies is extremely straightforward. It's a chain of blocks and the blocks are slightly complicated. Blocks can simply be described as a bundle of transactions organized together.

The below block has several keys. The hash value is basically the individual blocks ID that is made based on the data in the block. All the blocks point to the previous blocks hash value, thus making a chain. The only block that does not point to the previous is the genesis block. The genesis block is the first hardcoded block on the chain, and it often has some special value such as zero in the place of the previous blocks hash (Alazzawi, 2021).

Connection happening with previous blocks hash.

Merkle root is the hash of all the transactions in the block. It is formed by placing two transactions together and hashing them. This process is repeated until only one hash remains. The benefit of combining all the transactions is that only verifying Merkle root is necessary to verify the whole block. Nonce is a value that is included in the block if the consensus method in the blockchain needs it (Alazzawi, 2021).

Blockchain Trilemma

Ethereum co-founder Vitalik Buterin coined the term "blockchain trilemma," which characterizes the difficulty of concurrently attaining three essential characteristics in blockchain networks: scalability, security, and decentralization (Mehbodniya *et al.,* 2022).

a) Blockchain Technology?

Blockchain is a public ledger distribution that runs in a distributed manner and enables the reports of transactions through computers safely and in an open manner. Blockchain, which can be traced back to 2008 amid the creation of Bitcoin based on the artwork of Satoshi Nakamoto by an unknown person or a group of people, has diversified into a versatile tool applicable in areas other than financial activities. Blockchain is a digital database platform that is spread across different computers with the records of data received in blocks forming a chain. Data captured under this method is not easily amendable, hackable or manipulable. Since changing or eliminating the chain is virtually impossible without the network's consent, there is little time drift. Blockchain is an honest ledger dispersed over a large number of computers that records multiple transactions without compromising security (Yadav and Shevkar, 2021). As every transaction is segmented to blocks and associated time-wise, the information is safe and cannot be manipulated. The nodes in the network, that is the specific computers, validate and maintain the blockchain with methods such as the Proof of Work or the Proof of Stake. Information security in the fiscal platform is ensured by cryptography which also makes transactions secure in terms of security, integrity and authentication. This is because all the probable transactions that can take place are viewed on an open platform that is accessible to **(i)** all the parties. It makes security better because of the reduction of the single place of failure and enhancing the system's ability to resist cyber threats. Bitcoin is not the only application of blockchain; other applications are voting, supply chain, finance, healthcare, and property. At the same time, with the help of blockchain solutions, several enterprises experience a revolution in data handling and implementation of deals (Jaoude and Saadé, 2019).

b. Scalability in blockchain

Scalability in the context of block chain simply refers to the ability of the network to accommodate a large number of transactions or users as time passes and does so without having a negative impact on the operation of the network. This is important since the system has to remain fast and scalable should more users adopt the blockchain networks and should there be more transactions made. As a start, what do people understand by blockchain scalability, first? When it comes to the context of blockchain, then the frequency at which a transaction is affected is regarded as scalability. Thus, scalability is the capability of a Blockchain network to expand in the number of transactions and database so as also the nodes added, without compromising the key features that define the network, namely, security, decentralization and consensus. Also, the effectiveness of a blockchain network is defined by the number of transactions per second (TPS) it can process to meet the expectations. Networking is a factor that has impact on the scalability of block-chain; other parameters are finality, through put, cost and capacity, confirmation time (Biswas et al., 2019).

Scalability is one of the most important aspects of any business. Scalability is also a component as the size and the structure of a blockchain network increase each time a transaction is included into the network. Thus, to meet the increasing requirements, the blockchain network needs to process numerous transactions at a high speed and a large number. Sluggish payment processing, high costs, and a poor impression are the effects of a network that cannot deal with transactions' demand or specifications. The applicability and functionality of the blockchain networks can be affected by slow processing times and high costs associated with the transactions involved especially for operations that require a large number of transactions, such as supply chain and DeFi. In this regard, scalability is a crucial determinant of the possible expansion of blockchain (Matani, 2024).

Application of scalable encumbrances in blockchain technology

Improved Automation: Fuzzy logic is a fuzzy rulebased methodology which makes it possible to make more intelligent and automated decisions about encumbrances, which lowers the need for human intervention.

Flexible Transaction Conditions: Fuzzy logic enables the development of more intricate and sophisticated locking scripts, or encumbrances, for blockchain transactions. As a result, there is more freedom to define the circumstances under which a transaction output may be used.

Enhanced Privacy: When opposed to strict locking scripts, fuzzy encumbrances offer an extra degree of privacy by obscuring the exact circumstances for spending a transaction output (DeNio & Ludwig, 2021).

Limitation of scalable encumbrances in blockchain technology

Complexity: Fuzzy logic implementation for encumbrances increases the blockchain system's complexity, which can affect performance and make it more difficult to comprehend and audit.

Standardization: The application of fuzzy logic to blockchain encumbrances is not standardized, which can impede adoption and interoperability between various blockchain platforms.

Interpretability: It may be difficult to grasp the fuzzy rules and membership functions employed in fuzzy encumbrances, which makes it difficult to confirm the accuracy and security of the conditions (Martina, 2024).

Role of Fuzzy Logic in Enhancing Scalability **Fuzzy Logic in Cloud-Based Architectures:** Fuzzy logic deals with ambiguity and uncertainty in data, enabling intelligent and adaptable systems in Internet of Things applications. A "Fuzzy as a Service (FaaS)" architecture that runs in the cloud has been developed by researchers to convert fuzzy logic systems into services that are available from anywhere in the world. Using fuzzy logic systems becomes more cost-effective while maintaining transparency, system neutrality, load distribution, and resource allocation.

Fuzzy Logic for Improved Modeling and DecisionMaking: Control systems that are more adaptable and flexible and that can change with the environment in Internet of Things environments can be created with fuzzy logic. IoT systems can be helped by fuzzy logicbased decision-making when managing ambiguity and uncertainty, like in traffic control applications. Additionally, studies have demonstrated that fuzzy logic, as opposed to linear combinations, can produce models for processing Likert-scale data that suit data better

(Alkhdour, 2024). The figure 1 below shows the diagram of distributed ledger and centralized ledger, figure 2 shows the block's structure, figure 3 shows the chain of blocks, figure 4 shows the blockchain trilemma, figure 5 shows the diagram of the blockchain technology and figure 6 shows how the blockchain technology works.

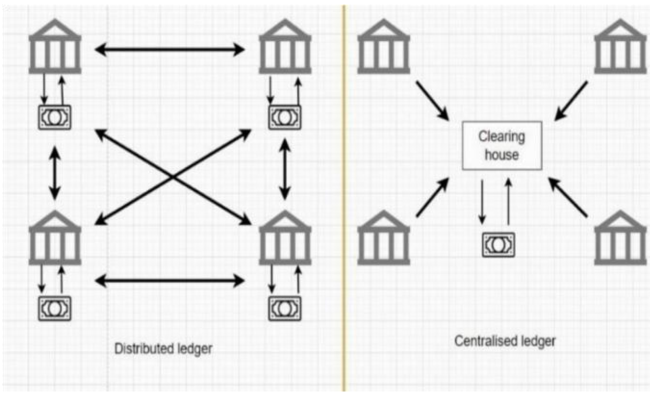


Figure 1: Distributed Ledger vs Centralized Ledger (Hilary, 2022)

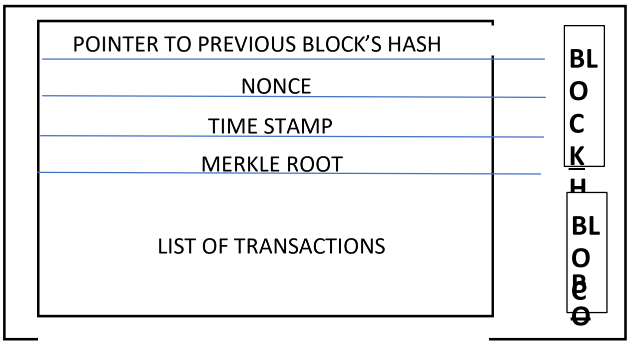


Figure 2: The Block's Structure (Raj, 2021)

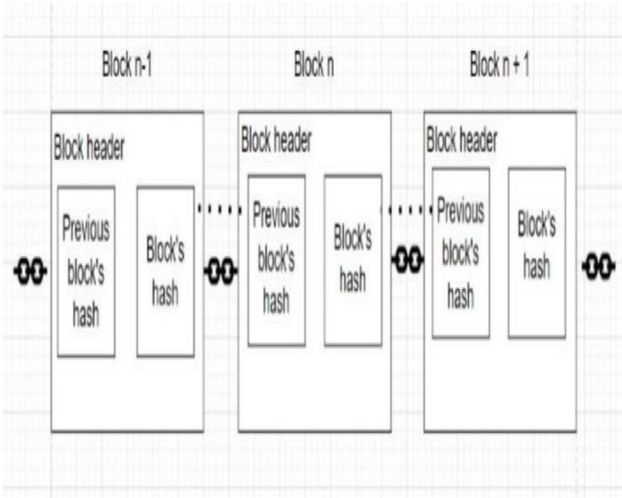


Figure 3: The Chain of Blocks (Raj, 2021)

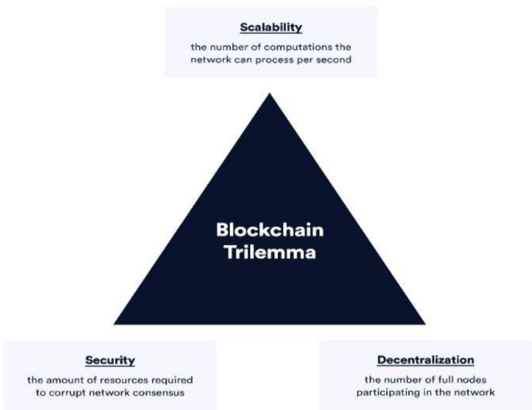


Figure 4: Blockchain Trilemma (Morpher, 2024)

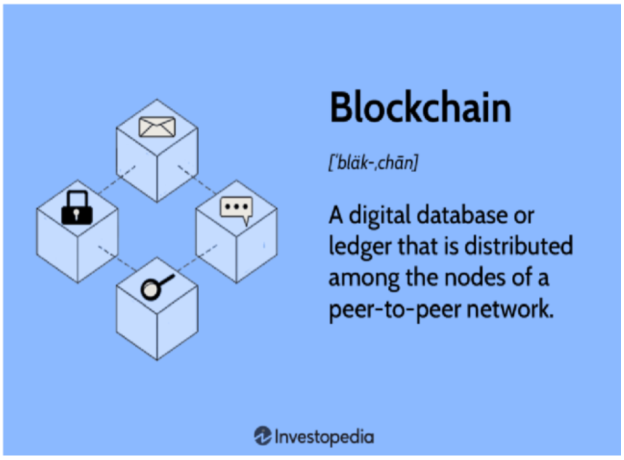


Figure 5: Blockchain Technology (Anandan and Deepak, 2022)

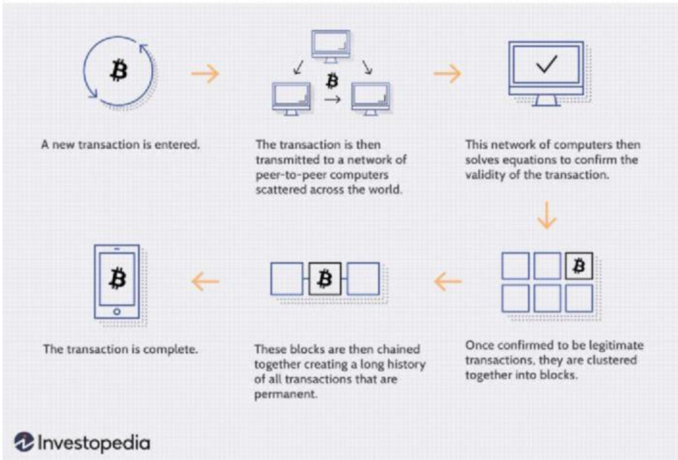


Figure 6: How Blockchain works (Anandan and Deepak, 2022)

Related Works

Accounting and auditing at the time of blockchain technology: A research agenda (Schmitz, J., Leoni, G. 2021)

Objective: The goal is to investigate the effects of blockchain technology on the accounting and auditing professions. The publication attempts to examine emerging trends in academic research and professional reports on blockchain technology in accounting and auditing. Furthermore, the research aims to give insights into the practical implications of blockchain technology for accountants and auditors, with a focus on governance, transparency, trust, continuous auditing, smart contracts, and the changing responsibilities of industry experts.

Methodology: Created a current, and potential Blockchain technology that can be use in business, specifically in accounting and in cybersecurity. We relate Blockchain uses to current concerns within cybersecurity and accounting.

Result: Based on the findings, four main topics have been identified: smart contract applications; blockchainenabled continuous audits; trust challenges in the blockchain ecosystem; governance; and a paradigm shift in the responsibilities of accountants and auditors. With regard to the implications of blockchain technology for the accounting and auditing profession, several topics arose from the examination of professional papers and scholarly literature.

Gaps: The research gap that has been highlighted is the low understanding and involvement of professionals and academics in the technological components of blockchain technology.

1) ***Implementation of blockchain technology in accounting sphere (Kwilinski, 2019)*** Objective: The aim of this research is to examine the application of Blockchain Technology in the accounting domain, emphasizing its benefits, obstacles, and consequences for accounting methodologies.

Methodology: The aim of this research is to examine the application of Blockchain Technology in the accounting domain, emphasizing its benefits, obstacles, and consequences for accounting methodologies.

Result: According to the report, there are a number of benefits to implementing blockchain technology in accounting, including enhanced security, efficiency, transparency, and process automation. It lowers the possibility of accounting errors, facilitates quick and safe document management with customers and contractors, and offers a single database for all accounting levels.

Gap: Even though blockchain technology has many benefits for accounting, further study is necessary to solve the obstacles and restrictions that come with putting it into practice.

How will blockchain technology impact auditing and accounting: Permissionless versus permissioned blockchain (Liu and Wu, 2023)

Objective: The main aim is to explore the impact of blockchain technology on the fields of auditing and accounting. They focus on comparing permissionless blockchains (public, decentralized) with permissioned blockchains (private, controlled) to assess how each type of blockchain could influence auditing and accounting practices.

Methodology: The authors created two types (i.e., permissionless and permissioned) of blockchain and lay out their technological features. They later showed implications of blockchain to auditing and elaborate on opportunities and challenges of the two types of blockchain to auditors

Result: The study comes to the conclusion that auditing and accounting might undergo a substantial transformation thanks to blockchain technology, both permissionless and permissioned, as it would increase efficiency, decrease fraud, and improve transparency.

Gap: The legal environment surrounding blockchain technology is still hazy and uneven in many states. This makes it difficult for auditors and accountants to make sure that different legal and regulatory standards are met.

A comprehensive review on blockchain scalability (A. Matani, A. Sahafi, and A. Broumandnia)
Objective: Globally, a growing number of disciplines are showing interest in blockchain technology as a distributed, impenetrable data ledger. Scalability becomes a difficult problem when considering the blockchain's ever-increasing transaction data and node count.

Methodology: The process outlined in the PDF file entails systematically reviewing the scalability options now available in blockchain technology. It comprises grouping the solutions into five primary groups, contrasting them according to scalability measures, talking about their benefits and drawbacks, and suggesting future lines of inquiry to improve blockchain scalability. Result: This research offers a thorough grasp of the many facets of blockchain scalability as well as the solutions that are now accessible. Lastly, it is stated that future research areas and unresolved concerns in each category should spur more attempts to increase blockchain scalability.

Gap: The necessity for a more thorough comparison and study of blockchain technology scalability solutions is the gap in the PDF file that has been found. Although the paper classifies several methods and weighs their benefits and drawbacks, more research and assessment of newly developed scaling mechanisms might be conducted to meet the changing requirements of blockchain systems.

A systematic review on blockchain scalability (Asmaa et al., 2023)
Objective: The aim is to ascertain, categorize, and evaluate the diverse methodologies and resolutions put forth to tackle the scalability issues in blockchain technology.

Methodology: Choosing pertinent studies from databases, establishing inclusion and exclusion criteria, developing research questions, and following systematic review procedures to evaluate and interpret the research results were all part of the methodology of the systematic review on blockchain scalability.

Result: Publication information, application domains, scalability problems, underlying causes, and suggested fixes were among the important data pieces that were taken out of the chosen papers and combined to give a thorough summary of blockchain scalability research.

Gap: The scalability solutions put out in the literature cannot be compared or evaluated using any widely recognized framework or standard. It is challenging to evaluate various techniques' efficacy and efficiency in an unbiased manner due to the absence of standards.

3. METHODOLOGY

3.1 Methodology Overview

This study proposes a fuzzy logic-based framework for enhancing blockchain scalability by optimizing transaction processing under varying network conditions. The framework employs a Mamdani Fuzzy Inference System (FIS) with triangular membership functions to model the uncertainty associated with blockchain transaction processing. The fuzzy logic system evaluates blockchain performance using three key scalability parameters: **transaction fee**, **block density**, and **transaction speed**.

Unlike conventional blockchain systems that rely on deterministic decision-making, the proposed approach incorporates fuzzy reasoning to improve transaction prioritization and network resource allocation. The fuzzy inference system analyzes the interaction among the selected parameters and determines the expected transaction speed under different network conditions. The implementation was carried out using Python programming language with the Scikit-Fuzzy library, while Pandas and NumPy were used for data preprocessing and numerical computation.

The methodology consists of six major stages:

- i Problem identification
- ii Data collection and preprocessing
- iii Design of the fuzzy inference system
- iv Fuzzification of blockchain parameters
- v Rule-based fuzzy inference
- vi Defuzzification and performance evaluation

The overall methodology adopted for this study follows the Waterfall Software Development Model because each phase depends on the successful completion of the preceding phase.

3.2 Waterfall Development Methodology

The Waterfall model provides a systematic and sequential framework for designing and implementing the proposed blockchain scalability model. The methodology consists of the following phases:

3.2.1 Requirements Analysis

This phase involved identifying the major scalability challenges affecting blockchain technology through extensive literature review. Three parameters were identified as the major determinants of scalability. Transaction fee, Block density, and Transaction speed. These parameters constitute the inputs and output of the fuzzy inference system.

3.2.2 System Design

During this phase, the fuzzy logic architecture was designed. Three linguistic variables were defined:

Input Variables are transaction Fee and Block Density

Output Variable is Transaction Speed

Each variable was partitioned into three linguistic terms using triangular membership functions.

3.2.3 System Implementation

The fuzzy inference model was implemented using Python. The implementation employed:

i Python 3.x
ii Pandas
iii NumPy
iv Scikit-Fuzzy
v Matplotlib

The blockchain dataset was loaded into Python, cleaned, normalized where necessary, and passed into the fuzzy inference system.

3.2.4 Integration and Testing

The fuzzy logic controller was integrated with the blockchain transaction dataset and tested under varying transaction conditions to determine whether the proposed framework could correctly classify transaction speeds.

3.2.5 Deployment

The validated model was deployed for blockchain transaction analysis.

3.2.6 Maintenance

The fuzzy rule base can easily be updated whenever blockchain parameters change.

3.3 Fuzzy Logic Model

The proposed framework employs a **Mamdani-type Fuzzy Inference** System **(FIS)** because of its simplicity, interpretability, and suitability for modelling uncertain environments such as blockchain networks.

The fuzzy inference system consists of four major components:

i Fuzzification
ii Rule Base
iii Inference Engine
iv Defuzzification

Figure 3.1 illustrates the architecture of the proposed fuzzy inference system.

3.4 Input and Output Variables

The proposed model uses two input variables and one output variable.

Input Variable 1: Transaction Fee

Transaction fee represents the amount paid by blockchain users to miners or validators for transaction confirmation.

The linguistic variables are:Low , Medium , High

Input Variable 2: Block Density

Block density represents the number of transactions contained within a block. The linguistic variables are

- Sparse
- Moderate
- Dense

Output Variable: Transaction Speed

Transaction speed represents the efficiency with which transactions are confirmed.

The linguistic variables are Slow, Average and Fast

3.5 triangular membership function

There are few triangular membership functions that I would be showing below:

i. Transaction fee

I used it in the transaction fee and I was able to know which was faster, medium and which was slow txn_fee ['low'] = fuzz.trimf(txn_fee.universe, [0, 0, 0.5]) txn_fee['medium'] = fuzz.trimf(txn_fee.universe, [0,

0.5, 1]) txn_fee['high'] = fuzz.trimf(txn_fee.universe, [0.5, 1, 1])

ii. Block Density

I also used the trimf in the block density and the code would be shown below:

block_density['sparse'] = fuzz.trimf(block_density.universe, [0, 0, 2500])
block_density['medium'] = fuzz.trimf(block_density.universe, [0, 2500, 5000])
block_density['dense'] = fuzz.trimf(block_density.universe, [2500, 5000, 5000])

Why did I use Triangular Membership Function; It is simple to define since only three parameters are needed (start, peak and end points).

 It is easy to understand and visualize which helps when defining the fuzzy logic rules.

Tools used *in analyzing the data*

Python Programming language: This programming language was used because it is mostly used in data analysis and it has a rich set of libraries that is easy to use.

Pandas: this was used to load my dataset and to perform operations like filtering and aggregating data.

NumPy: This was used for numerical operations because it handles large arrays and matrices efficiently.

Scikit-Fuzzy: This was used to provide tools that create fuzzy variables and define membership functions. *techniques used in analyzing data*

(i) **Fuzzy Logic:** this was used to classify the speed of transaction (slow, average, fast) based on the fuzzy variables like the block density and transaction fee. (ii) **Membership Functions:** This was used to define the fuzzy sets for each of the variables. # Define membership functions for txn_fee txn_fee['low'] = fuzz.trimf(txn_fee.universe, [0, 0, 0.3]) txn_fee['medium'] = fuzz.trimf(txn_fee.universe, [0.2,

0.5, 0.8])

txn_fee['high'] = fuzz.trimf(txn_fee.universe, [0.7, 1, 1]) (iii) **Fuzzy Rules:** This is used to establish a set of guidelines to decide the node label based on block density and transaction fee. # Define fuzzy rules

rule1 = ctrl.Rule(txn_fee['low'] & block_density['sparse'], txn_speed['slow'])
rule2 = ctrl.Rule(txn_fee['low'] & block_density['moderate'], txn_speed['slow']) rule3 = ctrl.Rule(txn_fee['low'] & block_density['dense'], txn_speed['average'])
rule4 = ctrl.Rule(txn_fee['medium'] & block_density['sparse'], txn_speed['average'])
rule5 = ctrl.Rule(txn_fee['medium'] & block_density['moderate'], txn_speed['average'])
rule6 = ctrl.Rule(txn_fee['medium'] & block_density['dense'], txn_speed['fast'])
rule7 = ctrl.Rule(txn_fee['high'] & block_density['sparse'], txn_speed['fast'])
rule8 = ctrl.Rule(txn_fee['high'] & block_density['moderate'], txn_speed['fast'])
rule9 = ctrl.Rule(txn_fee['high'] & block_density['dense'], txn_speed['fast'])

i. Challenges Hindering Scalability in Blockchain Technology

After undergoing several reviews, issues detected was the issue of block density and transaction speed. When there is congestion on the blockchain technology users tend to increase the transaction speed in order to prioritize their transaction. When the size of blocks decreases, they have a few transactions that can be included, thus creating congestion and making it difficult for people to complete their transactions on time during moments when many transactions are performed between willing buyers and sellers. This in turn slows down the transaction speed as users are made to wait longer before their transactions become part of a block.

ii. Mathematical Foundation of Fuzzy Logic

Fuzzy logic extends classical Boolean logic by allowing variables to have degrees of membership ranging from 0 to 1 rather than binary values of 0 or 1. A fuzzy set AAA over a universe of discourse XXX is represented as

$$A=\{(x,\mu A(x))|x\in X\},$$

where μA(x) is the membership function that assigns each element a degree of membership within the interval [0,1]. This capability enables fuzzy logic to effectively model uncertainty and support intelligent decision-making.

iii.Membership Function

In fuzzy logic, a membership function is a curve that specifies how each point in the input space is translated to a membership value, also known as the degree of membership, which ranges from 0 to 1. Membership function is given the title of μ. The input area is sometimes called the universe of discourse. μA(x) in A = {x, μA (x) | x Є X} is called the membership function of x in A. The membership function maps each element of X to a membership value between 0 and 1 (Mehmed, 2003). Membership functions allow for the graphical representation of a fuzzy set.

iv. Fuzzification

Fuzzification is the process of assigning a system's numerical input to fuzzy sets with varying degrees of membership. This level of membership might be anywhere within the range [0,1]. If it is zero, the value does not belong to the provided fuzzy set; if it is one, the value fully belongs to the fuzzy set. Any value between 0 and 1 represents the degree of uncertainty about the value's membership in the set. These fuzzy sets are often defined in words, therefore assigning the system input to fuzzy sets allows us to reason about it in a linguistically reasonable way.

v. Triangular Fuzzy Number

A triangular membership function as shown in figure 8 is identified by three parameters {a, b, c} and it is as follows:

The parameters {a, b, c}, with a < b < c, determine the x coordinates of the three corners of the underlying triangular membership function (Mehmed, 2003). The Figure 7 below shows the diagram of a waterfall methodology life cycle, the figure 8 below is the diagram triangular membership function, figure 9 shows the flowchart of a fuzzy logic algorithm.

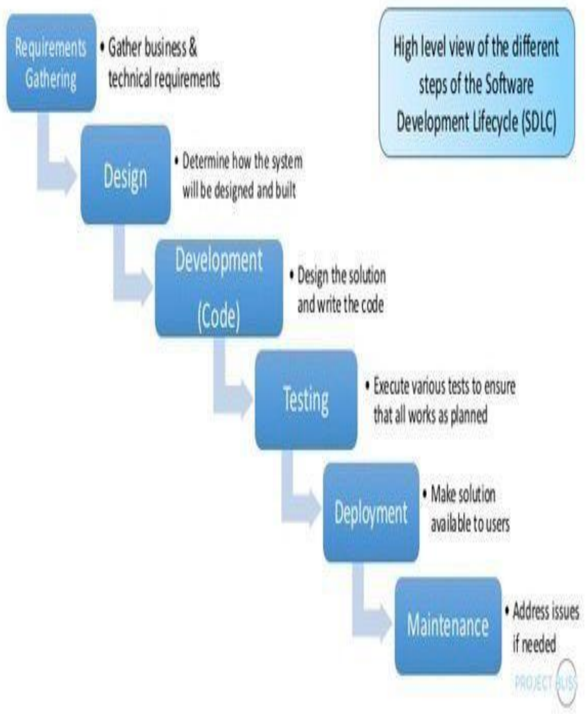
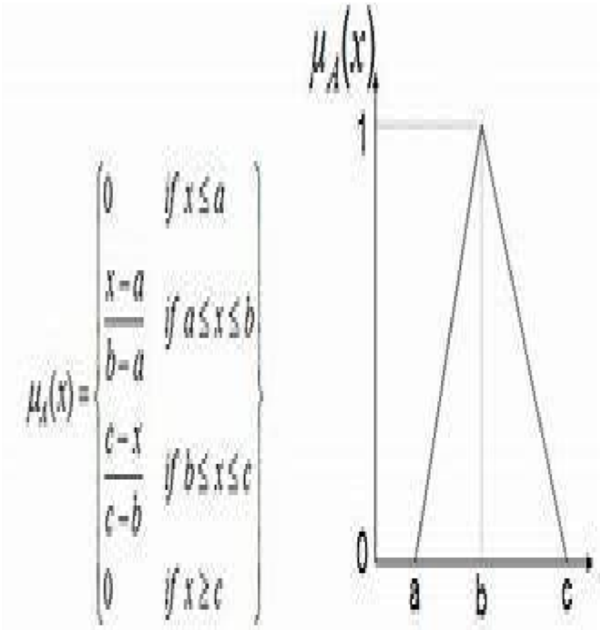


Figure 7: Waterfall Methodology Life Cycle (Glass and Vessey, 2002)



$$\mu_A(x) = \begin{cases} 0 & if\ x \leq a \\ \frac{x-a}{b-a} & if\ a \leq x \leq b \\ \frac{c-x}{c-b} & if\ b \leq x \leq c \\ 0 & if\ x \geq c \end{cases}$$

Figure 8: Triangular Membership Function (Percy, 2024)

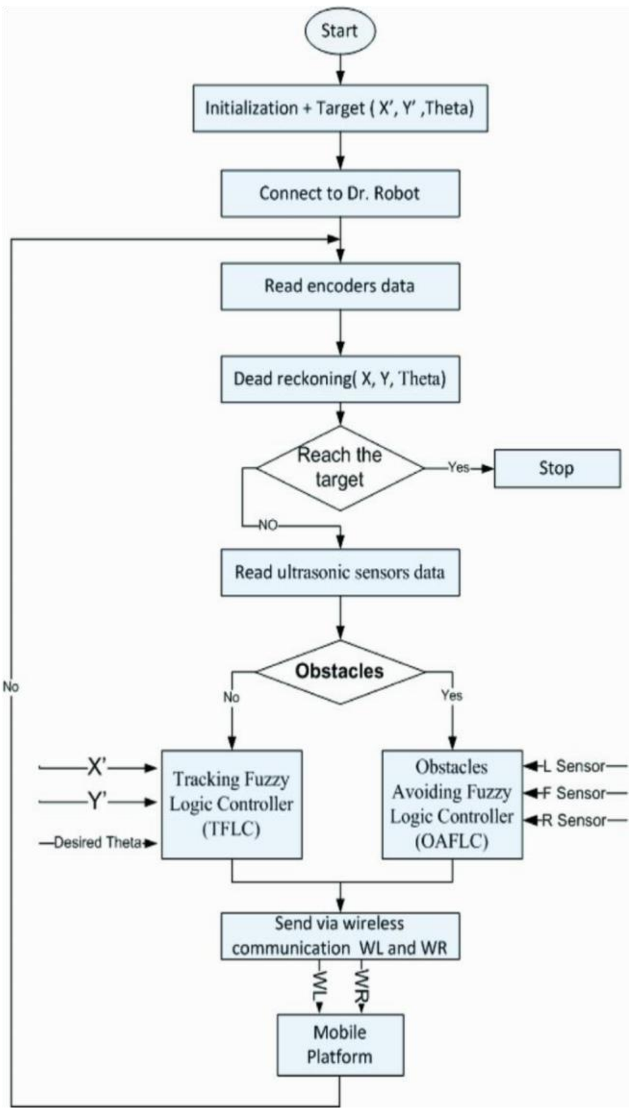


Figure 9: Flowchart of a Fuzzy Logic Algorithm (Soltani et al., 2022)

4. DATA ANALYSIS AND RESULT

Introduction

This chapter uses fuzzy logic approaches to describe the analysis and findings of the blockchain transaction dataset. Measuring transaction speed within the blockchain and figuring out how fast transactions happen in different places is the main goal. The scikit-fuzzy library was utilized in the Python implementation of the fuzzy logic system to manage the operations of fuzzification, rule evaluation, and defuzzification.

4.1. System Specification

The fuzzy logic-based blockchain transaction analysis system's requirements, architecture, and features are described in the system specification. The fuzzy logicbased blockchain transaction analysis system's requirements, design, and functionality are all thoroughly outlined in the system specification. The system is built to be dependable, scalable, and efficient by adhering to the waterfall model's organized methodology, which also offers insightful information on blockchain transaction speeds.

4.2. Hardware requirements

The minimum hardware requirements for the fuzzy expert system are:

(i) Intel core i5 or higher processor (ii) 8GB or higher (RAM)

(iii) 256GB SSD or higher (Storage)

(iv) Windows 10, macOS, or Linux (OS)

(v) 1280x720 resolution or higher (Display)

4.3. Software requirements

The software requirements for the fuzzy expert system are:

(i) **Python:** Version 3.6 or higher

(ii) **Python Libraries:** Pandas, Numpy, Matplotlib and Scikit-fuzzy

(iii) **IDE/Code Editor:** Jupyter Notebook and other Python-compatible IDE.

4.4. Result Analysis and Evaluation

The proposed system was evaluated using benchmark performance metrics based on transaction speed, transaction fee, node label, and block density.

4.5 Performance metrics

The performance of the applications was measured using Transaction Speed, Transaction Fee, Node Label, and Block Density. Node label is a fuzzy variable that represents the categorization of transaction speeds based on the inputs of transaction fee and block density. It is the output of the fuzzy logic system. Transaction speed refers to the rate at which transactions are processed and confirmed in the blockchain network. Transaction fee is the cost that users pay to have their transactions processed and included in a block by miners. Block density represents the percentage of space utilized in a block relative to its maximum capacity. To evaluate the applications, the applications were test run against the programs of the performance evaluation metrics.

Blockchain technology applications test

The testing was done on an Intel core i5 Windows 10 laptop. The rationale behind this evaluation is to determine the blockchain technology application using fuzzy logic is best suited for scaling encumbrances. The system was implemented with the aid of Python, Pandas, Scikit-Fuzzy and NumPy. The performance of the system was evaluated in terms of Node Label, Transaction Fee, Transaction Speed, and Block Density. The Node Label programming is shown below: for index, row in df.iterrows(): try:

node_label_sim.input['txn_fee'] = row['TxnFee(ETH)']
node_label_sim.input['block_density'] = row['Block Density (%)']
node_label_sim.compute()
df.at[index, 'fuzzy_node_label'] = node_label_sim.output['node_label'] except ValueError as e:
print(f"Skipping row {index} due to error: {e}")
df.at[index, 'fuzzy_node_label'] = np.nan The Block Density Program is shown below:

rule1 = ctrl.Rule(txn_fee['poor'] & block_density['poor'], node_label['slow'])

rule2 = ctrl.Rule(txn_fee['average'] & block_density['average'], node_label['average'])

rule3 = ctrl.Rule(txn_fee['good'] & block_density['good'], node_label['fast'])

# Control system rule2,
node_label_ctrl = ctrl.ControlSystem([rule1, rule3])

node_label_sim = ctrl.ControlSystemSimulation(node_label_ctrl) The Transaction Fee program is shown below
# Antecedent/Inputs txn_fee = ctrl.Antecedent(np.arange(0, 1.1, 0.1), 'txn_fee') block_density = ctrl.Antecedent(np.arange(0, 10001, 1), 'block_density') # Consequent/Output node_label = ctrl.Consequent(np.arange(0, 101, 1), 'node_label')
# Auto-membership function population txn_fee.automf(3) # 3 membership functions for txn_fee: 'poor', 'average', 'good'
block_density.automf(3) # 3 membership functions for block_density: 'low', 'average', 'high' # Custom membership functions for the node label node_label['slow'] = fuzz.trimf(node_label.universe, [0, 0, 50])
node_label['average'] = fuzz.trimf(node_label.universe, [0, 50, 100])
node_label['fast'] = fuzz.trimf(node_label.universe, [50, 100, 100])
The Transaction Speed program is shown below import matplotlib.pyplot as plt
plt.figure(figsize=(10, 6)) plt.hist(df['fuzzy_node_label'].dropna(), bins=30, alpha=0.7, label='Fuzzy Node Label') plt.xlabel('Node Label') plt.ylabel('Frequency') plt.title('Distribution of Fuzzy Node Labels') plt.legend() plt.show()

From the above shows how, the applications were evaluated and compared using the performance evaluation. Below figure 10 shows the Histogram Representation of the Distribution Transaction Speed, figure 11 shows the Performance Evaluation of Panda Data, figure 12 shows the Programming Pictorial Representation of the Performance Evaluation of Fuzzy Variables, figure 13 shows the Distribution of Fuzzy Node Labels, figure 14 shows the Programming Pictorial Representation of Performance Evaluation of Fuzzy Rules, figure 15 shows the Programming Pictorial Representation of Transaction Fee, Block Density and Node Label, and figure 16 shows the Graphical Representation of Transaction Speed, Block Density, and Transaction Fee of comparison between the applications.

Distribution of Fuzzy Transaction Speed *Figure 4.1.* Distribution of fuzzy transaction speeds. The histogram illustrates two dominant clusters: a low-speed cluster (18–23 units) and a high-speed cluster around the fuzzy-defined optimal point (49–51 units). The results demonstrate the model's ability to identify smooth transaction states while minimizing intermediate

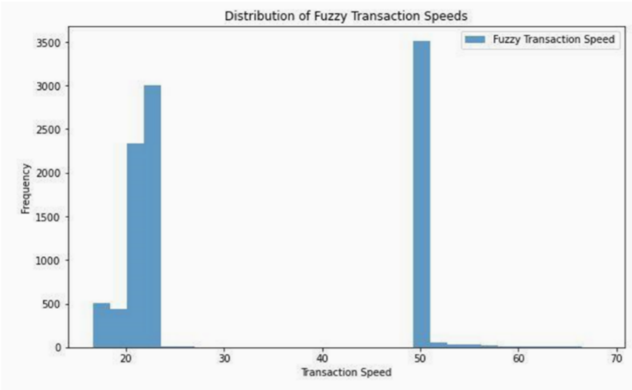


Figure 10: Distribution of Fuzzy Transaction Speeds

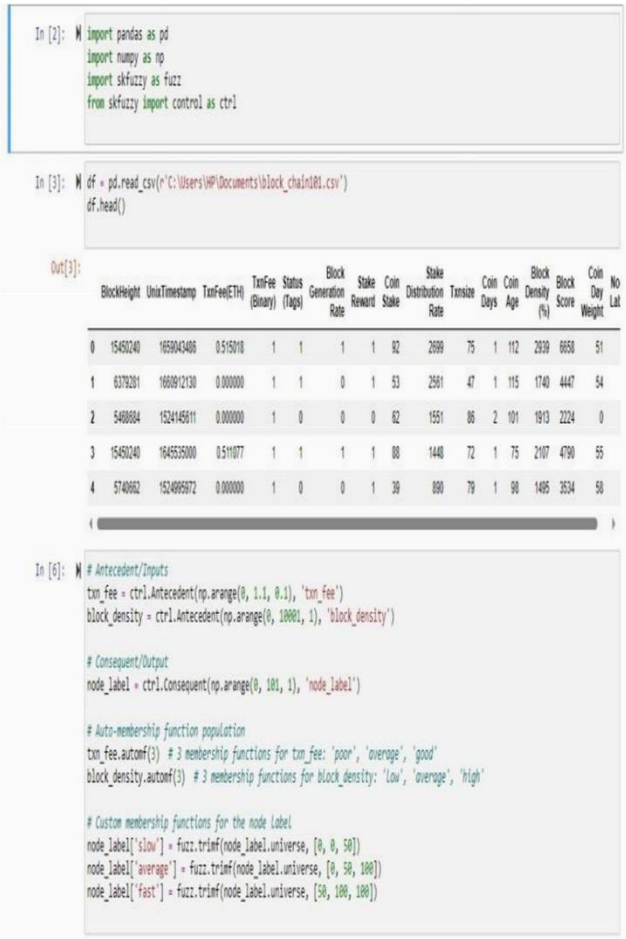


Figure 11: Performance Evaluation of Panda Data

```
In [15]: # Define fuzzy variables
txn_fee = ctrl.Antecedent(np.arange(0, 1.1, 0.1), 'txn_fee')
block_density = ctrl.Antecedent(np.arange(0, 5001, 1), 'block_density')
txn_speed = ctrl.Consequent(np.arange(0, 101, 1), 'txn_speed')

# Define membership functions for txn_fee
txn_fee['low'] = fuzz.trimf(txn_fee.universe, [0, 0, 0.3])
txn_fee['medium'] = fuzz.trimf(txn_fee.universe, [0.2, 0.5, 0.8])
txn_fee['high'] = fuzz.trimf(txn_fee.universe, [0.7, 1, 1])

# Define membership functions for block_density
block_density['sparse'] = fuzz.trimf(block_density.universe, [0, 0, 2000])
block_density['moderate'] = fuzz.trimf(block_density.universe, [1500, 2500, 3500])
block_density['dense'] = fuzz.trimf(block_density.universe, [3000, 5000, 5000])

# Define membership functions for txn_speed
txn_speed['slow'] = fuzz.trimf(txn_speed.universe, [0, 0, 50])
txn_speed['average'] = fuzz.trimf(txn_speed.universe, [25, 50, 75])
txn_speed['fast'] = fuzz.trimf(txn_speed.universe, [50, 100, 100])

# Visualize the membership functions
txn_fee.view()
block_density.view()
txn_speed.view()
```

Figure 12: Programming Pictorial Representation of the Performance Evaluation of Fuzzy Variables

Distribution of fuzzy node labels

Figure 10 Distribution of fuzzy node labels. The histogram presents a bimodal frequency distribution, featuring two distinct clusters: (1) a concentrated low label cluster spanning 15–25, where frequencies peak at approximately 2,700 and taper off gradually as labels increase; and (2) a dominant high-label cluster centered at the fuzzy-defined upper bound (49–51), with frequencies exceeding 3,600. Intermediate label values (26–48) show negligible occurrence, creating a sharp gap between the two clusters. These results highlight the model's ability to clearly demarcate discrete node label states while minimizing ambiguous or transitional values, reinforcing its effectiveness in distinguishing well-defined fuzzy categories.

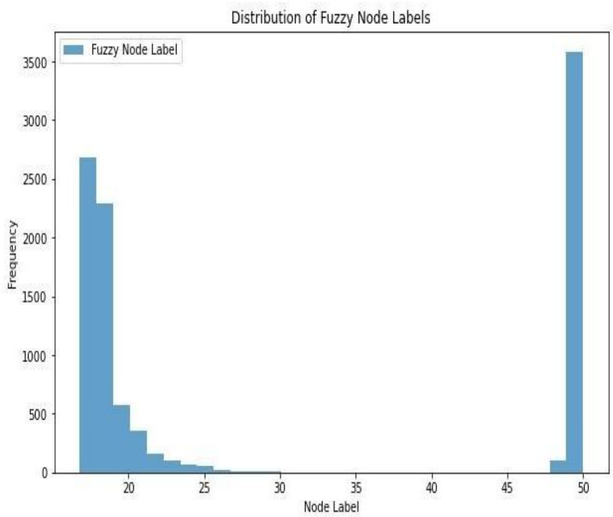
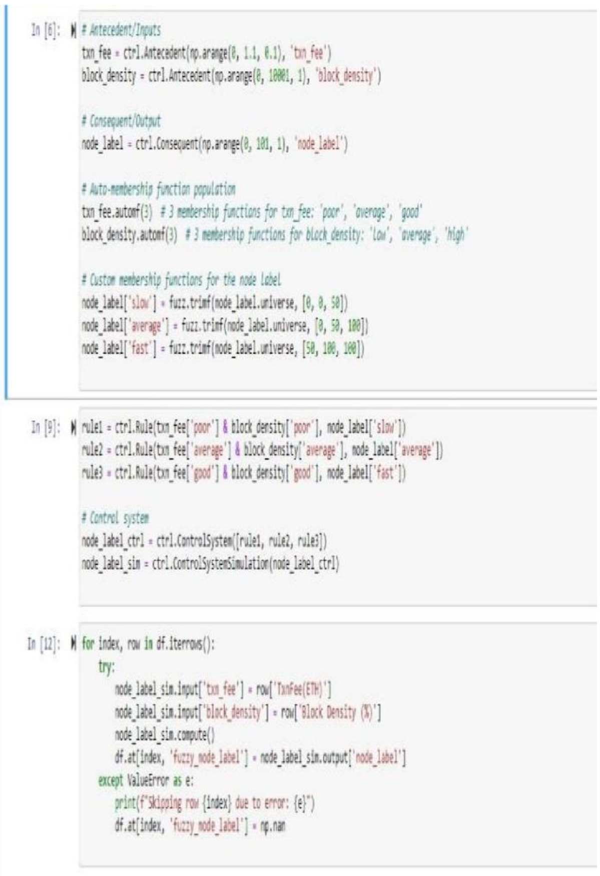


Figure 13: Distribution of Fuzzy Node Labels

```
In [28]: # Define fuzzy rules
rule1 = ctrl.Rule(txn_fee['low'] & block_density['sparse'], txn_speed['slow'])
rule2 = ctrl.Rule(txn_fee['low'] & block_density['moderate'], txn_speed['slow'])
rule3 = ctrl.Rule(txn_fee['low'] & block_density['dense'], txn_speed['average'])
rule4 = ctrl.Rule(txn_fee['medium'] & block_density['sparse'], txn_speed['average'])
rule5 = ctrl.Rule(txn_fee['medium'] & block_density['moderate'], txn_speed['average'])
rule6 = ctrl.Rule(txn_fee['medium'] & block_density['dense'], txn_speed['fast'])
rule7 = ctrl.Rule(txn_fee['high'] & block_density['sparse'], txn_speed['fast'])
rule8 = ctrl.Rule(txn_fee['high'] & block_density['moderate'], txn_speed['fast'])
rule9 = ctrl.Rule(txn_fee['high'] & block_density['dense'], txn_speed['fast'])

# Create control system and simulation
txn_speed_ctrl = ctrl.ControlSystem([rule1, rule2, rule3, rule4, rule5, rule6, rule7, rule8, rule9])
txn_speed_sim = ctrl.ControlSystemSimulation(txn_speed_ctrl)

In [ ]: # Apply fuzzy logic to each transaction
df['fuzzy_txn_speed'] = np.nan

for index, row in df.iterrows():
    try:
        txn_speed_sim.input['txn_fee'] = row['TxnFee(ETH)']
        txn_speed_sim.input['block_density'] = row['Block Density (%)']
        txn_speed_sim.compute()
        df.at[index, 'fuzzy_txn_speed'] = txn_speed_sim.output['txn_speed']
    except ValueError as e:
        print(f"Skipping row {index} due to error: {e}")
        df.at[index, 'fuzzy_txn_speed'] = np.nan  # Handle error appropriately

# Display the results
print(df[['TxnFee(ETH)', 'Block Density (%)', 'fuzzy_txn_speed']].head())

In [31]: # Plot the distribution of fuzzy transaction speeds
plt.figure(figsize=(10, 6))
plt.hist(df['fuzzy_txn_speed'].dropna(), bins=30, alpha=0.7, label='Fuzzy Transaction Speed')
plt.xlabel('Transaction Speed')
plt.ylabel('Frequency')
plt.title('Distribution of Fuzzy Transaction Speeds')
plt.legend()
plt.show()
```

Figure 14: Programming Pictorial Representation of Performance Evaluation of Fuzzy Rules



```
In [6]:  # Antecedent/Inputs
         txn_fee = ctrl.Antecedent(np.arange(0, 1.1, 0.1), 'txn_fee')
         block_density = ctrl.Antecedent(np.arange(0, 10001, 1), 'block_density')

         # Consequent/Output
         node_label = ctrl.Consequent(np.arange(0, 101, 1), 'node_label')

         # Auto-membership function population
         txn_fee.automf(3)  # 3 membership functions for txn_fee: 'poor', 'average', 'good'
         block_density.automf(3)  # 3 membership functions for block_density: 'low', 'average', 'high'

         # Custom membership functions for the node label
         node_label['slow'] = fuzz.trimf(node_label.universe, [0, 0, 50])
         node_label['average'] = fuzz.trimf(node_label.universe, [0, 50, 100])
         node_label['fast'] = fuzz.trimf(node_label.universe, [50, 100, 100])

In [9]:  rule1 = ctrl.Rule(txn_fee['poor'] & block_density['poor'], node_label['slow'])
         rule2 = ctrl.Rule(txn_fee['average'] & block_density['average'], node_label['average'])
         rule3 = ctrl.Rule(txn_fee['good'] & block_density['good'], node_label['fast'])

         # Control system
         node_label_ctrl = ctrl.ControlSystem([rule1, rule2, rule3])
         node_label_sim = ctrl.ControlSystemSimulation(node_label_ctrl)

In [12]: for index, row in df.iterrows():
             try:
                 node_label_sim.input['txn_fee'] = row['TxnFee(ETH)']
                 node_label_sim.input['block_density'] = row['Block Density (%)']
                 node_label_sim.compute()
                 df.at[index, 'fuzzy_node_label'] = node_label_sim.output['node_label']
             except ValueError as e:
                 print(f"Skipping row {index} due to error: {e}")
                 df.at[index, 'fuzzy_node_label'] = np.nan
```

Figure 15: Algorithm for Transaction Fee, Block Density and Node Label

Graphical Representation of Transaction Speed, Block Density, and Transaction Fee of Comparison Between the Applications

Figure 16, 17 and 18 shows the Membership functions for fuzzy variables in the transaction processing model which are transaction fee, block density, transaction speed respectively. The three panels illustrate how numerical inputs are mapped to linguistic categories via trapezoidal or triangular membership functions, enabling flexible reasoning about ambiguous system states.

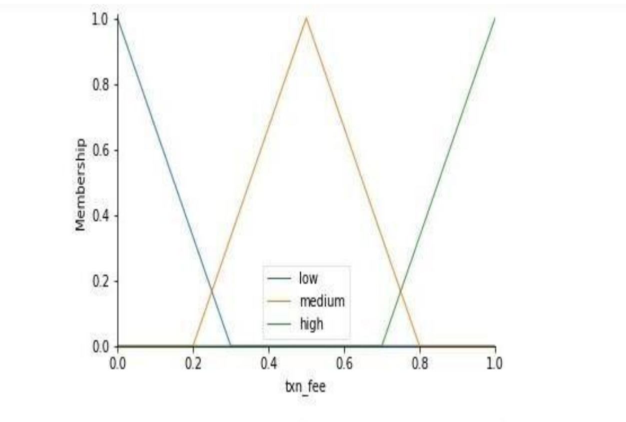


Figure 16: Transaction Fee (txn_fee)

The x-axis in figure 16 represents normalized transaction fee values (0–1), and the y-axis denotes membership degree (0 = no belonging, 1 = full belonging). Three fuzzy sets are defined: Low: A decreasing linear function (full membership at 0, drops to 0 at 0.2), capturing "very low fees." Medium: A triangular function peaking at 0.5 (spanning 0.2–0.7), representing "moderate fees." High: An increasing linear function (starts at 0.7, reaches full membership at 1), defining "high fees."

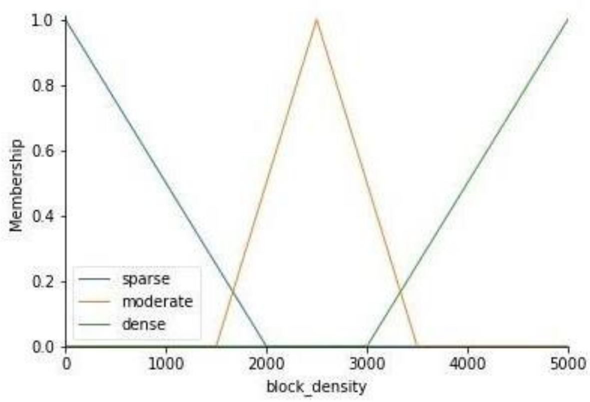


Figure 17: Block Density (block_density)

The x-axis in figure 17 shows block density values (0– 5000, e.g., blocks per interval), and the y-axis is membership degree. Three fuzzy sets are used:

Sparse: A decreasing linear function (full membership at

0, drops to 0 at 1000), indicating "low block density."

Moderate: A triangular function peaking at 2500

(spanning 1000–4000), capturing "moderate block density."

Dense: An increasing linear function (starts at 4000, reaches full membership at 5000), representing "high block density."

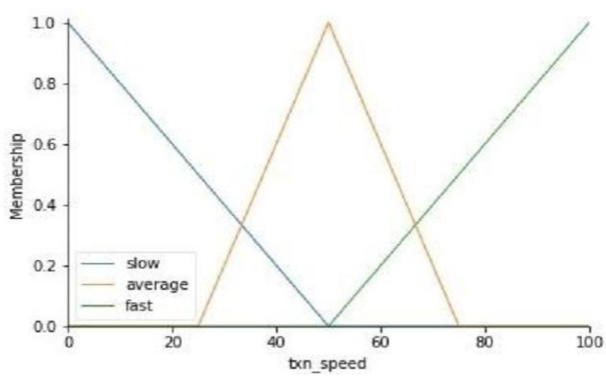


Figure 18: Transaction Speed (txn_speed)

The x-axis in figure 18 displays transaction speed values (0–100, e.g., transactions per second), and the y-axis is membership degree. Three fuzzy sets are defined:

Slow: A decreasing linear function (full membership at

0, drops to 0 at 40), capturing "slow transactions."

Average: A triangular function peaking at 60 (spanning 40–80), representing "average-speed transactions."

Fast: An increasing linear function (starts at 80, reaches full membership at 100), defining "fast transactions."

Significance

These membership functions translate numerical data into human-interpretable linguistic categories (e.g., "high fee," "dense block”), which are combined via fuzzy rules to make decisions (e.g., optimizing transaction prioritization). The triangular shape for “medium” sets balances sensitivity to moderate values, while linear sets for “extreme” categories (low/high, sparse/dense, slow/fast) enforce clear boundaries, ensuring the model handles ambiguity robustly. This structure is critical for modeling complex, real-world systems (e.g., blockchain networks) where precise thresholds are impractical Discussion

The performance evaluation results obtained above that the applications were comparable to find out the best application for scaling encumbrances. It is noteworthy to say that the application is a good secured dark web intrusion detection system within a localize area. However, the Pandas, NumPy, and Scikit-Fuzzy adopted in this study were used in the blockchain technology applications. The Scikit-Fuzzy application with python has an obvious advantage in all the performance evaluation metrics and it is also easy to implement. It is noteworthy to say that the application is a good for scaling encumbrances. Blockchain technology can be integrated with fuzzy logic to resolve its long-term scalability issues, which is one of the main reasons that have affected its adoption on a large scale. Fuzzy logic helps in reasoning that is approximate as opposed to fixed and exactness and this makes it possible to be used in managing uncertainties and imprecisions inherent in blockchain transactions. In the context of blockchain transactions, fuzzy logic provides an adaptable approach for handling associated ambiguities. The consensus algorithms will achieve more informed decisions by using fuzzy logic integration which will optimize transaction speeds thus bettering system efficiency. A fuzzy logic model was developed in this project to analyze and improve the speed of transactions on the blockchain. Three key parameters were grouped using triangular membership function: transaction fee, transaction speed, and block density. This implementation employed Python programming language with such libraries as Matplotlib, NumPy, Pandas as well as Scikit-fuzzy for data manipulation and algorithm development. It was found that fuzzy logic applied to the blockchain can highly increase efficiency of transactions. The fuzzy-logic-based model demonstrated its ability to manage transaction loads and optimize decision making within consensus algorithms in simplified manner. This makes blockchain scale better through smoother and faster transactions. In addition, this investigation presents considerations for the efficacy of using fuzziness as a solution to scalability problem. By dynamically adjusting transaction fees depending on network congestion, the system can maintain optimal performance even under varying load conditions. This adaptive approach is important for future development of blockchain technology so that they can be able to scale with demand properly. Thus, it goes without saying that integration of fuzzy logic into blockchain technology is a possible resolution of scalability problems therein. It provides an intricate way of handling complexities in managing transactions resulting in fast and smooth transactions.

5. SUMMARY

This study explores how fuzzy logic can be integrated with blockchain technology to enhance scalability, with a focus on improving transaction speeds. The main drive is that existing blockchain systems have inherent limitations in the transaction throughput that causes delays and inefficiencies. In this study, they are expected to adopt fuzzy logic by allowing for more adaptive and efficient consensus mechanism to be in place. Triangular membership functions were used as part of research methodology to evaluate block density, transaction fees and speed of transactions. The results show that fuzzy logic can greatly enhance the speed and efficiency of blockchain transactions such as through flexibility or ability for addressing scalability problems. Any future work should also look at what happens when fuzzy logic is combined with other consensus mechanisms like proofof-stake together with real-world implementation and testing aimed at proving that they system works efficiently as meant by its designers. Researchers are encouraged by this study to continue investigating the potential role of fuzzy logic in advancing blockchain technology while urging further research for practical applications.

5.1 Conclusion

Through the application of fuzzy logic approaches to transaction speeds, which offers a solid framework for improving blockchain performance without sacrificing its decentralized structure. The study comes to conclusion that employing fuzzy logic systems, like Python’s ScikitFuzzy, is simpler to construct and to also deliver a better performance. Consequently, the use of the fuzzy logic approaches on the transaction speeds creates a good framework to improve the performance of blockchain without altering the decentralized nature of the technology. The outcomes presented in this research reveal that not only is the use of models, including Python’s Scikit-Fuzzy, easier to integrate, but they also provide a higher level of performance. Hence through the integration of fuzzy logic the study is able to solve some of the known challenges of blockchain scalability and provides an actual solution to ease out and speed up the transactions. This approach also guarantees the growth to higher volumes of transactions as are made possible by the utilization of Blockchain technology hence opening the room for even more possibilities in the usage of the technology.

5.2 Recommendation

i. Promote the continuation of research in differed fuzzy logic models and their suitability throughout the multiple settings of blockchain implementation.
ii. I encourage blockchain developers and companies to think about adding fuzzy logic to their systems in order to improve speed and scalability of transactions.

Future works

i. **Security Implications:** It's critical to look into the security ramifications of incorporating fuzzy logic into blockchain technology. It should be a top concern to make sure that the flexibility that fuzzy logic brings about doesn't jeopardize the security that blockchain systems are built with.
ii. **Examining diverse Fuzzy Logic Models:** To find the model that performs the best in diverse blockchain situations, further study might look into using alternative fuzzy logic models, such as Sugeno or Mamdani. iii. **Improved Methods of Gathering and Analyzing Data:** The fuzzy logic models might be improved in terms of accuracy and efficiency by utilizing more complex analytical methods like machine learning and more advanced data collecting approaches.

6. REFERENCES

[1] Ahmad, S. and Bharti, A. K. 2019. Blockchain scalability enhancement using parallel miners selection. *International Journal of Innovative Technology and Exploring Engineering* 8, 12, 4357.

[2] Alazzawi, M. Q. 2021. Blockchain and cyber security. *ResearchGate*. August 1, 2021.

[3] Alkhdour, T., Almaiah, D., Ali, A. and Lutfi, A. 2024. Revolutionizing healthcare through fuzzy logic authentication: Unleashing blockchain brilliance. *Journal of*

Theoretical and Applied Information Technology, 1694–1695.

[4] Arora, P., Kushwaha, A. S. and Kumar, G. 2018. Designing optimized scalable framework for blockchain-based applications. *Journal of Emerging Technologies and Innovative Research* 5, 10, 129–132.

[5] Biswas, S., Sharif, K., Li, F., Nour, B. and Wang, Y. 2019. A scalable blockchain framework for secure transactions in IoT. *IEEE Internet of Things Journal* 6, 3, 4650–4659.

[6] Blondeel, M., Schockaert, S., De Cock, M. and Vermeir, D. 2014. Fuzzy autoepistemic logic and its relation to fuzzy answer set programming. *Fuzzy Sets and Systems* 239, 51–80.

[7] DeNio, J. A. and Ludwig, S. A. 2021. Improving transaction speed and scalability in blockchain systems. In *Proceedings of the 2021 International Conference on Compilers, Architecture, and Synthesis for Embedded Systems (CASES).*

[8] Ganaie, F. R. 2023. Application of fuzzy logic in artificial intelligence. *International Journal for Research in Applied Science and Engineering Technology* 11, 4, 2350–2356.

[9] 2015. Hybrid image compression based on fuzzy logic technology. *International Journal of Engineering Research and General Science* 3, 2, 799–801.

[10] Ismail, L. and Materwala, H. 2019. A review of blockchain architecture and consensus protocols: Use cases, challenges, and solutions. *Preprints.*

[11] I, V., Nagesh Babu, J. and G. J., P. 2020. Survey on blockchain: Backbone of cryptocurrency. *International Journal for Research in Applied Science & Engineering Technology.*

[12] Jaoude, J. A. and Saadé, R. G. 2019. Blockchain applications: Usage in different domains. *IEEE Access.*

[13] Kacprzyk, J. and Zadrożny, S. 2001. Computing with words in intelligent database querying: Standalone and Internet-based applications. *Information Sciences* 134, 71–109.

[14] Liu, M., Wu, K. and Xu, J. 2017. *How will blockchain technology impact auditing and accounting? Permissionless vs. permissioned blockchain.*

[15] Matani, A., Sahafi, A. and Broumandnia, A. 2024. A comprehensive review on blockchain scalability. *Journal of Electrical and Computer Engineering Innovations* 12, 1, 187–216.

[16] Mehbodniya, A., Webber, J. L., Rani, R., Ahmad, S. S., Wattar, I., Ali, L. and Nuagah, S. J. 2022. Energy-aware routing protocol with fuzzy logic in industrial Internet of Things with blockchain technology. *Wireless Communications and Mobile Computing* 2022, 1–15.

[17] Mitra, M. and Chowdhury, A. 2020. A modernized voting system using fuzzy logic and blockchain technology. *International Journal of Modern Education and Computer Science* 12, 3, 17–25.

[18] Mohamed, T., Kasa, A. and Taha, M. R. 2021. Fuzzy logic system for slope stability prediction. *International Journal on Advanced Science, Engineering and Information Technology.*

[19] 2018. *Advances in Fuzzy Logic and Technology 2017. Advances in Intelligent Systems and Computing.*

[20] Moraga, C. 2005. An engineering view of fuzzy logic. *Facta Universitatis, Series: Electronics and Energetics* 18, 2.

[21] Ralescu, A. L. 1994. *Applied Research in Fuzzy Technology.* Kluwer Academic Publishers, Dordrecht, The Netherlands.

[22] Rani, K. L. F. C. and Anuradha, M. P. 2024. Fuzzy logic-based mining strategy for transaction congestion management in blockchain networks. *Indian Journal of Science and Technology* 17, 19, 1968–1982.

[23] Sharma, D. and Garg, S. 2016. A critical study of fuzzy logic systems and its applications. *International Journal of Trend in Scientific Research and Development* 1, 1, 56–57.

[24] Singh, H., Gupta, M. M., Meitzler, T., Hou, Z. G., Garg, K. K., Solo, A. M. G. and Zadeh, L. A. 2013. Real-life applications of fuzzy logic. *Advances in Fuzzy Systems* 2013, 1–3.

[25] Suganya, M. and Jayanthi, K. 2019. Fuzzy logic control. *International Journal of Scientific Research in Science, Engineering and Technology* 6, 2, 539–545.

[26] Taylor, P. J., Dargahi, T., Dehghantanha, A., Parizi, R. M. and Choo, K. K. R. 2020. A systematic literature review of blockchain cyber security. *Digital Communications and Networks.*

[27] Tundjungsari, V. and Wardhana, N. A. D. 2022. Blockchain-based music metadata copyright protection using fuzzy logic. *International Journal of Applied Engineering & Technology*, 101–105.

[28] Yadav, J. and Shevkar, R. 2021. Performance-based analysis of blockchain scalability metric. *Tehnički Glasnik* 15, 1, 133–142.