Enhancing the Scalability of Blockchain Networks using a Data Partitioning Technique

Basavaiah Lathamani
Department of Management Studies, JSS Academy of Technical Education, Bengaluru, India
lathamanib@jssateb.ac.in

Niranjan C. Kundur
Department of Computer Science and Engineering, JSS Academy of Technical Education, Bengaluru, India
niranjanckundur@jssateb.ac.in (corresponding author)

Chaya J. Swamy
Department of Management Studies, JSS Academy of Technical Education, Bengaluru, India
chayajswamy@jssateb.ac.in

Pavana Kumari Hanumanthaiah
Jain (Deemed to be University) SOC, Jayanagar, Bengaluru, India
pavanajagadish77@gmail.com

Praveen M. Dhulavvagol
School of Computer Science and Engineering, KLE Technological University, Hubli, India
praveen.md@kletech.ac.in

Bellary Chiterki Anil
Department of CSE (Artificial Intelligence and Machine Learning), JSS Academy of Technical Education, Bengaluru, India
anilbc@jssateb.ac.in

Received: 16 August 2024 | Revised: 15 September 2024 | Accepted: 19 September 2024

 DOI: https://doi.org/10.48084/etasr.8760

ABSTRACT

The scalability limitations of current blockchain systems slow down their broad adoption. This issue arises because transactions are processed sequentially, limiting throughput and increasing network delays. Additionally, even with advanced multicore technology, the Proof-of-Work (PoW) process is generally performed in a linear fashion. To address these challenges, this study proposes a static analysis-based data partitioning technique to enhance transaction performance and reduce network latency by allowing parallel processing of transactions, called Simultaneous Block-Level Transaction Execution in a Distributed Setting. This framework utilizes a master-slave system within a trusted node community. The master node analyzes transactions and partitions non-conflicting ones into separate groups, or shards, which are then distributed among slave nodes for parallel execution. Once transactions are completed, the community's combined computing power is used to perform PoW simultaneously. The miner subsequently broadcasts the newly created block to other network peers for validation, which can be performed either sequentially or in parallel. Validators ensure that they achieve the same state as specified in the block. Implementing this framework on a workload can result in a maximum speedup of 1.81x for miners and 1.80x for validators, with each block containing between 150 and 550 transactions and involving six community members. PoW is a consensus mechanism in which miners solve complex cryptographic puzzles to validate transactions. It ensures network security but is resource-intensive due to its high computational demands. In the proposed framework, the master node coordinates transactions, while the slave nodes process them in parallel. This approach maximizes resource utilization across nodes.

Keywords-blockchain scalability; data partitioning; parallel transaction processing; network latency; Proof of Work (PoW)

I. INTRODUCTION

As blockchain technology becomes increasingly popular across diverse industries, its scalability has emerged as a major concern. Traditional blockchain systems such as Bitcoin and Ethereum encounter substantial limitations in transaction throughput and network latency due to their sequential transaction processing design. This constraint hampers their ability to efficiently manage high transaction volumes, thereby limiting their widespread adoption and practical use in fields such as finance, supply chain management, and healthcare. Data partitioning is one promising approach to overcome these scalability challenges, which involves dividing the blockchain network into smaller, more manageable units known as shards. This method enables parallel processing of transactions, enhancing overall throughput and reducing latency. By leveraging distributed computing resources, data partitioning allows for more efficient transaction handling and network management. The proposed data partitioning technique aims to address scalability issues by implementing a static-analysis-based framework. This approach focuses on partitioning transaction data and processing it across multiple nodes in parallel. The framework, termed Simultaneous Block-Level Transaction Execution in a Distributed Setting, is designed to improve transaction throughput and reduce network latency by enabling concurrent execution of transactions within a distributed network of trusted nodes. This study evaluates the effectiveness of this data partitioning technique through a series of experiments, comparing its performance to traditional serial processing methods. By analyzing speedup and execution times across varying transaction loads and community sizes, this study provides insights into the potential of parallel execution to enhance blockchain scalability. The results aim to contribute to the ongoing efforts to optimize blockchain systems and support their broader adoption in real-world applications.

A. Blockchain Consensus Protocols

Blockchain consensus protocols are essential to ensure the integrity and security of transactions within a blockchain network. These protocols are designed to achieve agreement among distributed nodes on the validity of transactions and the state of the blockchain. Two of the most widely adopted consensus mechanisms are Proof of Work (PoW) and Proof of Stake (PoS). In PoW, miners solve complex cryptographic puzzles to validate transactions. It ensures network security but is resource-intensive due to its high computational demands. PoW is the original consensus mechanism introduced by Bitcoin. It requires network participants, known as miners, to solve complex cryptographic puzzles to validate transactions and create new blocks. This process demands significant computational resources and energy, making PoW both secure and resource-intensive. The difficulty of the puzzles adjusts over time to ensure that blocks are added to the blockchain at a consistent rate. Although PoW is effective at securing the network and preventing double-spending, its high energy consumption has raised environmental concerns. PoS offers an alternative to PoW by selecting validators based on their economic stake in the network. In PoS, participants are chosen to create new blocks and validate transactions proportionally to the amount of cryptocurrency they hold and are willing to "stake" as collateral. This method reduces the need for extensive computational work, making it more energy efficient than PoW. PoS also mitigates the risk of centralization by allowing a wider range of participants to engage in the validation process. Variants of PoS, such as Delegated Proof of Stake (DPoS) and Hybrid PoS, introduce additional mechanisms to enhance scalability, security, and decentralization. Both PoW and PoS have their advantages and limitations. PoW is renowned for its robustness and security but is criticized for its environmental impact. PoS, on the other hand, offers greater efficiency and scalability but relies on economic incentives to ensure network security. As blockchain technology evolves, new consensus mechanisms and hybrid approaches continue to be developed to balance security, scalability, and energy consumption.

The key contributions of the proposed framework are:

- Parallel Transaction Processing Framework: A framework called Simultaneous Block-Level Transaction Execution in a Distributed Setting is proposed, which enables parallel processing of transactions. This framework utilizes a master-slave architecture within a trusted node community to efficiently divide and execute non-conflicting transactions. In this framework, the master node coordinates transactions, while the slave nodes process them in parallel. This approach maximizes resource utilization across nodes.
- Reduction in Network Latency: By partitioning the data and distributing the transactional workload across multiple shards, the proposed technique significantly reduces network latency, thus enhancing overall transaction performance.
- Simultaneous Proof of Work (PoW) Execution: The proposed framework leverages the collective computing power of the community to perform PoW concurrently, leading to faster block creation and validation.
- Scalability Demonstration: The practical application of the proposed framework demonstrates a maximum speedup of 1.81x for miners and 1.80x for validators, showcasing the scalability benefits of the data partitioning approach.

The scalability of blockchain networks is a significant research area, and many approaches have been proposed to address this issue. Traditional blockchain systems such as Bitcoin and Ethereum use a serial transaction execution model, which limits transaction throughput and increases network latency. As blockchain applications expand in sectors such as supply chain management, banking, and healthcare, improving scalability is increasingly important. In [1], a comprehensive analysis of the fundamental components of sharding blockchain systems was provided, discussing how sharding can dynamically partition nodes within a blockchain network into shards, each capable of computation, storage, and processing. This systematic study identified essential components such as epoch randomness, node assignment, and shard reconfiguration parameters, laying the groundwork for constructing effective sharding systems. In [2], the concept of sharding was extended to permissioned blockchain systems. The primary goal was to create a blockchain system capable of handling transaction volumes comparable to major cryptocurrencies such as Bitcoin and Ethereum. This study addressed the limitations of blockchain networks restricted to cryptocurrencies and the inability of consensus procedures to scale effectively. The proposed sharded blockchain system was tested on both a local cluster and the Google Cloud Platform, demonstrating significant improvements in scalability and transaction handling. In [3], Rapid Chain was introduced, a Byzantine-resistant public blockchain technology that implements full sharding. Rapid Chain divides the network into smaller committees, each maintaining distinct ledgers and independently processing transaction blocks. This approach was compared with other sharding-based protocols such as Elastic and Omni ledger, showing that it could scale to network sizes of up to 4000 nodes without significant performance degradation. In [4], a dynamic blockchain sharding system was proposed, capable of rotating participants within a shard, altering shards, and creating new shards. This innovation addresses the limitation of existing protocols that cannot change the number of shards. This system includes security measures to prevent shard takeovers by malicious nodes and introduces a shard committee rotation strategy to reduce bribery. This approach, tested on the Colla Chain blockchain, demonstrated linear performance improvements with an increasing number of shards. Thorough and systematic analyses of blockchain sharding approaches were conducted in [5-8], highlighting critical elements and challenges in sharding schemes. Various methods for generating epoch randomness and handling cross-shard transactions were discussed in [9-11]. In [12], potential research directions were identified and several well-known blockchain sharding methods were reviewed, providing a comprehensive overview of state-of-the-art sharding research. The diverse approaches and innovations discussed in these works form the basis for further advances to improve the performance and efficiency of blockchain networks [13-14]. The proposed approach builds on these foundations by proposing a static analysis-based data partitioning technique to further improve the scalability of blockchain systems through efficient parallel transaction processing and consensus mechanisms [14-16].

II. METHOD

Figure 1 illustrates a data partitioning architecture in a blockchain system, highlighting the communication and consensus mechanisms among multiple shards. Three shards are shown, each represented by a horizontal chain of blocks: Shard 1, Shard 2, and Shard 3. Each shard maintains its independent blockchain. Within each shard, a consensus algorithm ensures agreement among shard members on the blockchain's state, shown by arrows leading to the chain of blocks within each shard. Intra-shard communication, indicated by blue arrows, occurs among shard members to maintain consensus and synchronize the shard's blockchain. Cross-shard communication, represented by green arrows, coordinates and shares information between different shards, ensuring proper management of transactions involving multiple shards. Each shard contains multiple members, depicted as icons with screens, who participate in the consensus process. Additionally, a coordinator within each shard, represented by a larger icon, plays a key role in managing the shard's operations. The blocks within each shard, shown as connected squares, contain the transactional data specific to that shard, with blue, purple, and orange colors distinguishing Shards 1, 2, and 3, respectively. The legend on the right side of the diagram explains the symbols used: green arrows for cross-shard communication, blue arrows for intra-shard communication, double lines for the consensus algorithm, an icon for the coordinator within a shard, icons for shard members participating in the shard, and block icons representing the individual blocks within each shard's blockchain. This diagram effectively demonstrates how sharding can partition a blockchain network into smaller, more manageable pieces, each capable of processing transactions independently while remaining part of the larger network through cross-shard communication.

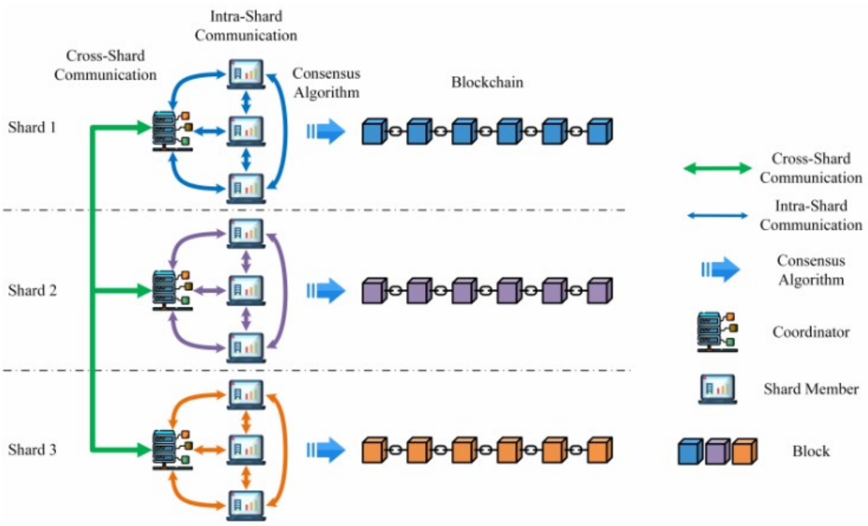

Fig. 1. Data partitioning in a blockchain network.

A. Community Functioning as Both Miner and Validator

Figure 2 illustrates the process in which the community functions as both the miner and the validator, with these roles being executed in parallel. The process begins with the master node gathering a block of transactions from the queue of pending transactions. Using static analysis, the master node identifies which transactions are dependent and which are independent. To facilitate parallel execution, independent transactions are organized into separate shards, which are then assigned to slave nodes for processing. Each slave node calculates the new state for the transactions within its assigned shard and sends the results back to the master node. For transactions within a shard that are dependent, they are executed sequentially. The master node then aggregates the results from all the slave nodes to finalize the state of the block. This parallel execution process allows for efficient handling of transactions, reducing overall processing time and improving system performance.

In the PoW phase, the task of finding a valid block hash that meets the required difficulty level is assigned to the slave nodes. The master node sends the block along with different nonce ranges to the slaves, who then work on computing the block hash. Once a slave node finds the correct hash, the master node updates its local copy of the block chain and broadcasts the block, along with the PoW, to other peers in the network. When the community acts as a validator, it verifies the proposed block by re-executing its transactions to ensure the accuracy of the PoW. The validator node checks that the miner has used sufficient computational resources to find the correct PoW, confirming the block's validity.

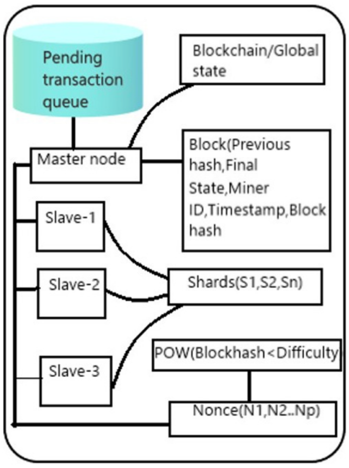

Fig. 2. Community functioning as both miner and validator.

III. RESULTS AND DISCUSSION

Enhancing blockchain network scalability through the proposed data partitioning technique yielded promising results. Implementing the proposed framework demonstrated significant performance improvements, with miners achieving a maximum speedup of 1.81x (81%) and validators achieving a maximum speedup of 1.80x (80%) when utilizing five slave nodes. The proposed framework was tested with blocks containing between 150 and 550 transactions, and various ratios of contract to monetary transactions (1:1, 1:2, 1:4, 1:8, and 1:16). The results showed that speedup increased linearly with the number of slave nodes, reflecting the efficiency of the framework in handling larger transaction volumes. Moreover, blocks with a higher number of transactions showed a greater speedup compared to those with fewer transactions, highlighting the framework's ability to scale effectively. However, it was observed that blocks with more contract transactions took longer to execute, pointing to the inherent complexity of processing smart contracts compared to simple monetary transactions. These findings confirm that the data partitioning technique enhances throughput and reduces latency, aligning well with the objectives of improving blockchain performance. Future research may focus on further optimizing this approach, potentially by exploring more decentralized methods or improving the handling of complex contract transactions.

A. Performance Analysis

The performance analysis of the proposed data partitioning technique for blockchain scalability involved evaluating how effectively the framework improves transaction processing and reduces latency. To assess performance, several key metrics were measured:

- Speedup Measurement: The primary metric to evaluate performance was speedup, calculated as the ratio of execution time for serial processing to that for parallel processing using the data partitioning framework. Speedup was measured for both miners and validators. For miners, using five slave nodes resulted in a speedup of up to 1.81x. The validators achieved a similar speedup of 1.80x. These improvements demonstrate the framework's effectiveness in accelerating blockchain operations through parallel execution.
- Impact of Community Size: The analysis showed that increasing the number of slave nodes in the community improved the performance linearly. As the size of the community grew from 2 to 5 slave nodes, the speedup increased correspondingly, indicating that the framework scales efficiently with the number of parallel processing units. This linear relationship underscores the framework's capability to leverage additional computational resources for enhanced throughput.
- Transaction Load Testing: The framework was tested with blocks containing 150 to 550 transactions, incorporating various ratios of contract to monetary transactions. The results indicated that the framework effectively handled different transaction loads, with speedup improvements becoming more noticeable as the number of transactions per block increased. This suggests that the technique is robust and adaptable to varying transaction volumes.
- Execution Time Variation: The execution time for block formation was analyzed for both parallel and serial processing. Parallel execution consistently reduced block formation time compared to serial processing, demonstrating the efficiency of the data partitioning approach. However, blocks with a higher proportion of

contract transactions experienced longer execution times, reflecting the added complexity associated with smart contracts.

Overall, the performance analysis confirms that the data partitioning technique enhances blockchain scalability by improving transaction processing speed and reducing latency. The results highlight the effectiveness of the proposed framework in managing increased transaction volumes and its potential for future optimization.

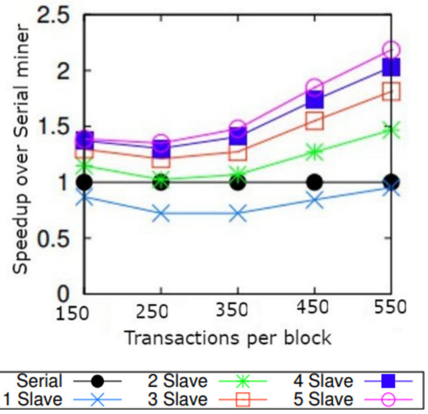

Fig. 3. Miner and validator average speed increase (without mining) for transaction execution.

Figure 3 presents the performance analysis for both miners and validators, using serial execution time as the baseline for comparison. The results reveal that utilizing five slave nodes within the community achieves a speedup of around 1.8 times. This speedup improves linearly with the increase in the number of slave nodes from 2 to 5. Additionally, the analysis shows that as the number of transactions per block increases, the speedup effect becomes more pronounced. These findings demonstrate that parallel execution notably improves processing efficiency, with greater speed improvements observed as additional resources are integrated into the system.

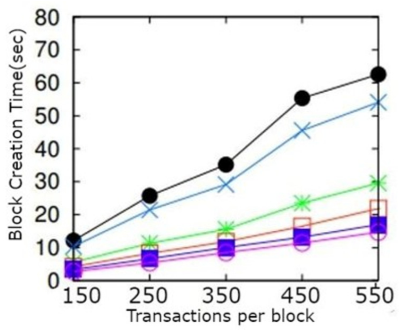

Fig. 4. Block creation duration as measured by miners on average.

Figure 4 shows that the time required to form a block decreases as the number of slave nodes increases. For a given workload, as the transaction count within a block grows, the mining process becomes more complex and time-consuming. However, the addition of more slave nodes helps mitigate this complexity by distributing the workload, thereby reducing the overall block formation time.

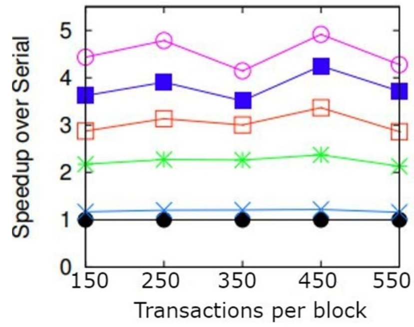

Fig. 5. Speedup of parallel over serial mining for average block creation.

Figure 5 illustrates that as the number of transactions in a block increases, communities with more members experience a greater speedup compared to those with fewer members or serial execution. This suggests that larger communities are more efficient at handling higher transaction volumes, significantly enhancing processing speed and performance as the transaction count increases. Table I indicates that with a community of five slave nodes, the miner achieved a maximum speedup of 1.81x, while the validator achieved a maximum speedup of 1.80x.

TABLE I. ESTIMATED WORKLOAD SPEEDUP WITH CONCURRENT VALIDATOR AND MINER (WITHOUT MINING)

Average speedup (average execution time for the workload)	**Number of transactions per block**					
	Miner/ Validator	**150**	**250**	**350**	**450**	**550**
Serial Execution	Miner Validator	1.00 1.00	1.00 1.00	1.00 1.00	1.00 1.00	1.00 1.00
One slave	Miner Validator	0.850 0.825	0.701 0.702	0.701 0.720	0.835 0.834	0.935 0.941
Two slaves	Miner Validator	1.150 1.063	1.021 0.965	1.068 1.023	1.264 1.224	1.451 1.425
Three slaves	Miner Validator	1.291 1.170	1.205 1.123	1.263 1.199	1.520 1.461	1.614 1.598
Four slaves	Miner Validator	1.350 1.212	1.298 1.119	1.400 1.301	1.616 1.620	1.720 1.710
Five slaves	Miner Validator	1.376 1.244	1.356 1.232	1.474 1.365	1.723 1.721	1.814 1.802

IV. CONCLUSION

The proposed static-analysis-based data partitioning technique addresses the scalability challenges faced by blockchain networks. By utilizing distributed resources through mining pools, the proposed approach aims to improve transaction throughput, measured by the average speedup compared to serial execution, and reduce network latency, measured by average end-to-end block generation time. The proposed framework was introduced as Simultaneous Block-Level Transaction Execution in a Distributed Setting, which facilitates parallel transaction processing across multiple trusted node communities. The experimental results showed significant performance enhancements with this framework. Specifically, using only five slave nodes, up to 1.81x (81%) speedup was achieved for miners and 1.80x (80%) speedup was achieved for validators. The framework was evaluated with transaction loads ranging from 150 to 550, including different ratios of contract to monetary transactions, using real Ethereum transactions for validation. The results show that speedup increases with the number of transactions per block and scales linearly with community size. However, blocks with a higher number of contract calls exhibited longer execution times. In the proposed framework, the master node coordinates transactions, while the slave nodes process them in parallel. This approach maximizes resource utilization across nodes. Future research should focus on replacing the master-slave architecture with a more decentralized approach within the community to further improve scalability and efficiency.

REFERENCES

[1] Y. Liu *et al.*, "Building blocks of sharding blockchain systems: Concepts, approaches, and open problems," *Computer Science Review*, vol. 46, Nov. 2022, Art. no. 100513, https://doi.org/10.1016/j.cosrev.2022.100513.

[2] H. Dang, T. T. A. Dinh, D. Loghin, E. C. Chang, Q. Lin, and B. C. Ooi, "Towards Scaling Blockchain Systems via Sharding," in *Proceedings of the 2019 International Conference on Management of Data*, Amsterdam Netherlands, Jun. 2019, pp. 123–140, https://doi.org/10.1145/3299869.3319889.

[3] M. Zamani, M. Movahedi, and M. Raykova, "RapidChain: Scaling Blockchain via Full Sharding," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, Toronto, Canada, Oct. 2018, pp. 931–948, https://doi.org/10.1145/3243734.3243853.

[4] D. Tennakoon and V. Gramoli, "Dynamic Blockchain Sharding," in *5th International Symposium on Foundations and Applications of Blockchain 2022 (FAB 2022)*, Berkeley, CA, USA, 2022, https://doi.org/10.4230/OASIcs.FAB.2022.6.

[5] G. Wang, Z. J. Shi, M. Nixon, and S. Han, "SoK: Sharding on Blockchain," in *Proceedings of the 1st ACM Conference on Advances in Financial Technologies*, Zurich Switzerland, Oct. 2019, pp. 41–61, https://doi.org/10.1145/3318041.3355457.

[6] P. M. Dhulavvagol, V. H. Bhajantri, and S. G. Totad, "Blockchain Ethereum Clients Performance Analysis Considering E-Voting Application," *Procedia Computer Science*, vol. 167, pp. 2506–2515, Jan. 2020, https://doi.org/10.1016/j.procs.2020.03.303.

[7] C. Decker, J. Seidel, and R. Wattenhofer, "Bitcoin meets strong consistency," in *Proceedings of the 17th International Conference on Distributed Computing and Networking*, Singapore, Jan. 2016, pp. 1–10, https://doi.org/10.1145/2833312.2833321.

[8] P. M. Dhulavvagol and S. G. Totad, "Performance Enhancement of Distributed System Using HDFS Federation and Sharding," *Procedia Computer Science*, vol. 218, pp. 2830–2841, Jan. 2023, https://doi.org/10.1016/j.procs.2023.01.254.

[9] P. S. Anjana, S. Kumari, S. Peri, S. Rathor, and A. Somani, "An Efficient Framework for Optimistic Concurrent Execution of Smart Contracts," in *2019 27th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP)*, Pavia, Italy, Feb. 2019, pp. 83–92, https://doi.org/10.1109/EMPDP.2019.8671637.

[10] A. Zhang and K. Zhang, "Enabling Concurrency on Smart Contracts Using Multiversion Ordering," in *Web and Big Data*, Macau, China, 2018, pp. 425–439, https://doi.org/10.1007/978-3-319-96893-3_32.

[11] P. M. Dhulavvagol, S. G. Totad, and N. Bhandage, "Topic Based Partitioning for Selective Search Using Sharding Technique," in *2022 International Conference for Advancement in Technology (ICONAT)*, Goa, India, Jan. 2022, pp. 1–5, https://doi.org/10.1109/ICONAT53423.2022.9726020.

[12] S. S. Taher, S. Y. Ameen, and J. A. Ahmed, "Advanced Fraud Detection in Blockchain Transactions: An Ensemble Learning and Explainable AI Approach," *Engineering, Technology & Applied Science Research*, vol. 14, no. 1, pp. 12822–12830, Feb. 2024, https://doi.org/10.48084/etasr.6641.

[13] I. Purdon and E. Erturk, "Perspectives of Blockchain Technology, its Relation to the Cloud and its Potential Role in Computer Science Education," *Engineering, Technology & Applied Science Research*, vol. 7, no. 6, pp. 2340–2344, Dec. 2017, https://doi.org/10.48084/etasr.1629.

[14] E. Aruna and A. Sahayadhas, "Blockchain-Inspired Lightweight Dynamic Encryption Schemes for a Secure Health Care Information Exchange System," *Engineering, Technology & Applied Science Research*, vol. 14, no. 4, pp. 15050–15055, Aug. 2024, https://doi.org/10.48084/etasr.7390.

[15] A. Barredo Arrieta *et al.*, "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," *Information Fusion*, vol. 58, pp. 82–115, Jun. 2020, https://doi.org/10.1016/j.inffus.2019.12.012.

[16] B. Mahbooba, M. Timilsina, R. Sahal, and M. Serrano, "Explainable Artificial Intelligence (XAI) to Enhance Trust Management in Intrusion Detection Systems Using Decision Tree Model," *Complexity*, vol. 2021, no. 1, 2021, Art. no. 6634811, https://doi.org/10.1155/2021/6634811.