Computers & Industrial Engineering 170 (2022) 108327

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

Improving blockchain performance in clinical trials using intelligent optimal transaction traffic control mechanism in smart healthcare applications

Faisal Jamil [a], Shabir Ahmad [b], Taeg Keun Whangbo [b,*], Ammar Muthanna [c,d], Do-Hyeun Kim [e]

[a] Department of ICT and Natural Sciences, Faculty of Information Technology and Electrical Engineering, Norwegian University of Science and Technology (NTNU), Larsgårdsvegen 2, Ålesund, 6009, Norway
[b] Department of IT Convergence, Gachon University, Seongnam, Gyonggi-do, Republic of Korea
[c] Department of Applied Probability and Informatics, Peoples' Friendship University of Russia (RUDN University), 117198 Moscow, Russia
[d] Department of Telecommunication Networks and Data Transmission, The Bonch-Bruevich Saint-Petersburg State University of Telecommunications, 193232 Saint Petersburg, Russia
[e] Department of Computer Engineering, Jeju National University, Jejusi 63243, Jeju Special Self-Governing Province, Republic of Korea

ARTICLE INFO

Keywords:
Blockchain
Smart contract
Fuzzy logic
Transaction traffic control
Optimization
Blockchain performance

ABSTRACT

Blockchain technology has revolutionized the ways of processing and storing data in terms of reliability and security. Blockchain plays a pivotal role in transferring the processing hurdle from the client–server to a decentralized and secured platform. Blockchain is deemed to be an efficient technology in a forthcoming era that would beneficiate multifarious industries. An issue that becomes a bottleneck for blockchain-based applications is their restricted ability to process in comparison to distributed database systems. In this paper, we present intelligent traffic control using a hybrid model based on the PSO-based optimization algorithm and fuzzy logic in order to improve blockchain performance. The real-time network feedback model is designed and used as an input to control the transaction traffic across the entire network in a robust way without human intervention. In order to evaluate the effectiveness of the designed model, a clinical trial service framework as a test network is implemented on top of Hyperledger Fabric. The case study is further compared with baseline network, network with fuzzy approach, and network with optimized parameter. The experiments show that the proposed model not only enhanced the network by maximizing the network throughput and minimizing the network latency. A smart contract is implemented to automate the transaction flow as per real-time data of network conditions. An open-source blockchain framework, Hyperledger Fabric, is harnessed for implementation of the experiment environment in order to signify the potential of the proposed model. The outcome of this study indicated a remarkable increase in transaction throughput (i.e., 38.5%) and a decrease in transaction latency of 40.5%. Moreover, the proposed model can easily be integrated with other existing blockchain-based performance-enhancing tools.

1. Introduction

Nowadays, the blockchain is considered one of the prominent technology based on the distributed ledger (DL). Blockchain technology provides a secure way to submit transactions without the third party intervention (Maull, Godsiff, Mulligan, Brown, & Kewell, 2017). During the past few year many application have been developed in various domains, such as healthcare (Ibrahim, Jamil, Lee, & Kim, 2022; Jamil, Ahmad, Iqbal, & Kim, 2020; Jamil, Hang, Kim, & Kim, 2019; Jamil, Iqbal, Amin, & Kim, 2019; Jamil, Kahng, Kim, & Kim, 2021; Jamil, Qayyum, Alhelaly, Javed & Muthanna, 2021; Shahbazi & Byun, 2020), supply-chain Jamil, Hang et al. (2019), Shahbazi and Byun (2021d), smart grid (Ahmad, Ullah, Jamil, & Kim, 2020; Aziz et al., 2021; Jamil et al., 2021), Internet of things (IoT) (Jamil, Qayyum, Jamil & Kim, 2021; Shahbazi & Byun, 2021a, 2021b, 2021c, 2021e; Zaabar, Cheikhrouhou, Jamil, Ammi, & Abid, 2021), and agriculture (Jamil et al., 2022), etc. The blockchain refers to the DL of transactions where all the participating nodes retain the network. The transactions in the blockchain indicate the back-end business logic and construct the sequence of blocks stored in the ledger. Each node in the blockchain network contains the updated ledger copy maintained after every consensus mechanism.

* Corresponding author.
E-mail addresses: faisal.jamil@ntnu.no (F. Jamil), shabir@gachon.ac.kr (S. Ahmad), tkwhangbo@gachon.ac.kr (T.K. Whangbo), ammar@rudn.com (A. Muthanna), kimdh@jejunu.ac.kr (D.-H. Kim).

https://doi.org/10.1016/j.cie.2022.108327

Available online 17 June 2022

Currently, blockchain serves as a dominant practice used in several industries, such as real estate, healthcare, finance, and many others (Hameed, Barika, Garg, Amin, & Kang, 2021). Nonetheless, logically, the rate of blockchain transactions must be higher than the traditional database systems that support some level of transaction assurance. The performance factor is inspected as the critical challenge in acquiring blockchain technologies as compared to conventional centralized approaches (Latif, Hussain, Jhanjhi, Nayyar, & Rizwan, 2020). The inadequate performance factors in terms of transactions competence, such as poor scalability, less throughput, limited storage, high transaction latency, etc., can create a hurdle in the development of blockchain systems (Moin et al., 2019). For instance, the bitcoin has limited memory of 1 MB to create the block, and it requires 10 min to construct the new block. Moreover, bitcoin has a fixed transaction rate of 7 transactions per second, hindering high-frequency trading. Additionally, it requires an hour to mine a block in the bitcoin after it is confirmed. Likewise, in the case of Ethereum, it takes around 15 s to create a new block, and it can take more time depending on the network configuration. In public blockchains, such as Ethereum and Bitcoin, anonymous users can participate and perform transactions, indicating the lack of privacy and data validation. Moreover, their permissionless blockchain platform publicizes tokens to boost transaction and currency mining while lacking data privacy. The use of digital currencies can also negatively impact the transaction in terms of cost and speed.

Most researchers believe that blockchain technology is not suitable for substantial application due to its unsatisfactory performance. Permissionless blockchain, such as Ethereum and Bitcoin, consists of nodes ranging from thousands to millions, supporting cryptocurrency transactions. To establish trust among the unidentified identities, several consensus approaches, such as crash fault tolerant (CFT) (Muratov, Lebedev, Iushkevich, Nasrulin, & Takemiya, 2018), Proof of Work (PoW) (Duong, Chepurnoy, Fan, & Zhou, 2018), Practical Byzantine Fault Tolerance (PBFT) (Sousa, Bessani, & Vukolic, 2018), Proof of Stake (PoS), Proof of Burn (PoB), etc., have been used to provide user authentication and data validation. These consensus mechanisms are time-consuming for transaction achieving, which ultimately decreases the transaction throughput. Consequently, there is a need for an approach that offloads the transaction processing from the main chain to the side chain, which may significantly increase the blockchain's scalability and performance. Similarly, in the permissioned blockchain platform, all the participants are authorized and known, which may reduce the risk of privacy and security. Likewise, the trust established among the participants in the permission blockchain improved the network performance in terms of throughput since many consensuses approach, such as PBFT or CFT, support the transaction with low-cost mining.

Present advancement in the blockchain practice inventing new possibilities for intelligent applications (Mougayar, 2016). Several studies use the optimization and control approaches to address existing challenges in the blockchain. The invention of smart contract in the blockchain also support full or partial execution without human intervention (Bhardwaj et al., 2020). Besides, the smart contract also provides data security and supports an execution environment where data is not altered or tampered. In addition, the combination of control and optimization assists in creating a decentralized ecosystem. Nevertheless, existing studies related to artificial intelligence also employ blockchain framework to enable users through data model and control in order to enhance data reliability. Nonetheless, non of any studies use optimization and control in order to improve the performance of the blockchain network. In this article, we proposed an intelligent transaction traffic control using a hybrid approach based on optimization and fuzzy logic to improve the performance of the blockchain network. The real-time network feedback is acquired to control transactions' traffic based on a hybrid approach without human intervention. We have used the particle swam optimization algorithm for the optimization module to compute the optimal value of latency, send rate, and transaction throughput. Similarly, the smart contract-enabled fuzzy logic is based on the Mamdani fuzzy system used for decision-making and automatizing the transaction traffic control process without third-party involvement. Furthermore, we have considered a Clinical Trial Service as a case study to evaluate the proposed model's effectiveness and usability. For experimental analysis, we have used Hyperledger Fabric, a tool-set and framework used to implement blockchain-based applications. Lastly, the proposed case study is evaluated in terms of latency and throughput. Results indicate that the proposed optimization model considerably enhances the performance in terms of throughput and latency compared to the predefined scheme.

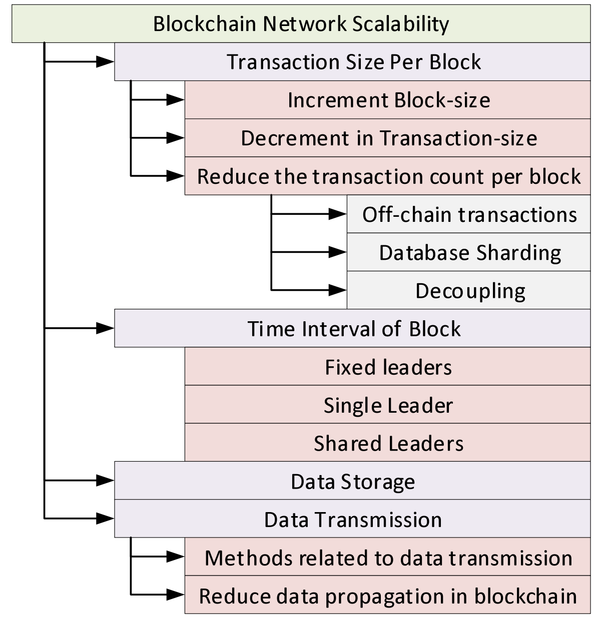


Fig. 1. Parameters for scalability of blockchain platforms.

The main contribution of this paper is followed as

- The proposed model is based on a hybrid approach based on optimization and fuzzy logic to improve the performance of the blockchain network.
- The real-time network feedback is collected to automate the transaction traffic control using particle swarm optimization and fuzzy logic.
- The proposed model is tested on a clinical trial service platform to evaluate the effectiveness and usability of the proposed hybrid model.

2. Related work

The blockchain platform's throughput depends on several factors, such as transactions size in every block, interval time of block, data transmission, and data storage. Fig. 1, shows the existing technologies for scalable blockchain platforms.

2.1. Transactions size per block

The transaction size per block is mainly depend on two factors, such as block-size increment, decrement in transaction size, and reduce the transactions number per blocks.

- **Increase the Block Size:** In Blockchain the block size is the key factor which can improve the transaction throughput by adding more transactions in every block. Nevertheless, the increase in the size of block increase the processing overhead, and propagation delay of nodes to commit transactions in the blockchain network.

- **Reduce the transaction size:** In a blockchain, decreasing the transaction size which ultimately increases the transaction in every block. The digital signature in the blockchain is used to verify the authentication of the transaction. SegWit, the blockchain platform, splits the digital signature from the transaction data and record the data at the end of the blocks in the blockchain. Thus, the transaction size is reduced, and more transactions can be accumulated in a block.
- **Reduce the Number of Transactions:** The main components which are used to enhance the blockchain throughput are transactions count, sharding, off-chain transactions, and control and decoupling management. In an off-chain transaction, the processing of crypto-currency is usually executed outside the blockchain network. Duplex micropayment channel (Khan et al., 2019) and Lightning Network (Kim, Kwon, & Cho, 2018) are implemented using the off-chain transactions. In Lightning Network, each update of the micro-payment channel requires additional information to commit to the blockchain network. Similarly, the Duplex Micro-payment provides the auto-update of the micropayment channel and supports multiple transactions. Sharding is an approach in the blockchain used to enhance the horizontal scalability of blockchain systems. The sharding nodes can be segregated into multiple shards. Furthermore, the parallel execution is also supported in sharding, where the chunk transaction is processed simultaneously. Byzantine consensus approach is used to commit the transactions by the selective set of nodes participating in the blockchain network. OmniLedger (Kokoris-Kogias et al., 2018) and Elastico (Luu et al., 2016) are the two blockchain platform based on the sharding techniques. The decoupling enables the virtualization of distributed ledger technology (vDLT) (Yu & He, 2019) provides the management of several types of services in terms of quality of service (QoS) and execution of the smart contracts. The decoupling of services can be done using technology like virtualization, where several instances of DLT are created and accessed easily.

2.2. Block time interval

The block time is a time interval at which the block is thoroughly mined. The expected time to mine a block in bitcoin is approximately 10 min whereas, in the case of ethereum, it takes 10–20 s to mine a block in a blockchain. The block generation in the blockchain comprises two operations, i.e., leader election and transaction generalization, where each leader node validates transactions to generate a new block. The leader election approach further segregated into three groups, such as fixed leaders (Xie et al., 2019), single leader (Abraham et al., 2017), and collective leaders (Xie et al., 2019).

2.3. Data storage

The main purpose of data storage is to store a large amount of data. In a blockchain, the distributed ledger storage system is integrated with the traditional database to address the existing issues related to storing a huge amount of data. In Danish, Zhang, and Jacobsen (2020), the author presented an approach that uses a distributed ledger technology where the un pre-processed data is recorded using the off-chain Distributed Hash Table (DHT). The main aims of this approach are to store the reference of data in the blockchain, which is encrypted using the SHA-256. Desema (Hasan & Salah, 2018) an ethereum based InterPlanetary File System (IPFS) that provides distributed marketplace system capable of storing a huge amount of data. BigchainDB (McConaghy et al., 2016) a blockchain platform that integrates the distributed ledger with the existing database to enhance the database volume for storing data. The BigchainDB stores data in two formats, i.e., database S (store transactions) and database C (store the block of data generated after committing the transactions).

2.4. Data transmission

Blockchain technology is used to share the data securely amongst the participants. To transmit data between the users effectively while considering parameters like network bandwidth and resource utilization. Mistry, Tanwar, Tyagi, and Kumar (2020). Table 1 presents the performance comparison of the existing blockchain platform in terms of latency and throughput. In contrast, Table 2 shows the critical analysis of existing state-of-arts technologies for a scalable blockchain system.

As mentioned below, these blockchain systems are either permissionless or not available open-source therefore the researcher community cannot modify or upgrade for their use. Moreover, most of the literature review approaches mainly focus on data transmission, block interval time, number of transactions, and data storage capacity for a scalable blockchain system. Nonetheless, none of any discussed systems use transaction traffic control to improve blockchain performance in terms of throughput and latency. Furthermore, most of the systems discussed above use the native crypto-currency, which degrades the blockchain performance during transaction processing. However, in the proposed model, we consider parameters such as block size, block frequency, ordering service, client size, organization size, transport layer security, transaction size, and interval time in order to improve the blockchain performance in terms of latency and throughput. To the best of the authors' knowledge, there is no functional intelligent transaction traffic control using a hybrid approach based on optimization and fuzzy logic for scalable blockchain.

3. Transaction traffic control based on optimization and fuzzy inference system

3.1. Overview of conceptual system architecture

The proposed system conceptual architecture is presented in Fig. 2. The designed blockchain network consists of several nodes, where the smart contract in the host environment stores and maintains the distributed ledger copy of the entire network. The smart contract contains transactions invoked by clients who submit the transaction. The smart contract-enabled fuzzy controller is used to control the transaction traffic in the entire network automatically. The proposed fuzzy controller encompasses transaction control and fuzzy inference system.

Similarly, the optimization module is executed recursively by analyzing the benchmark results acquired from the blockchain and finding the optimal solution. The real-time optimum benchmark results are observed and disseminated to the fuzzy controller-enabled smart contract. The fuzzy controller enabled smart contracts to analyze and decide the transaction received based on the computed control commands. The entire blockchain network is established based on the consensus algorithm, and the response of the transaction execution is sent directly to clients.

3.2. Hybrid optimization and fuzzy logic approach for transaction traffic control

The proposed hybrid architecture consists of an administrator, blockchain adopter, optimization module, smart contract enabled fuzzy inference system, benchmark database, and the blockchain network as shown in Fig. 3.

The blockchain network comprises several nodes, where each node holds a smart contract and distributed ledger replica. The network administrator is responsible for configuration network files and benchmarks to perform transaction evaluation. The network configuration file determines the network connection requirement and system underperformance. Similarly, the user test file and performance benchmark workload can be described using the benchmark configuration file. The blockchain adaptor is responsible for starting and initializing

Table 1
Performance comparison of existing blockchain models.

Approach	Nodes	Latency (s)	Throughput (TPS)
OmniLedger (Kokoris-Kogias et al., 2018)	1800	20	6000
SegWit (Mechkaroska, Dimitrova, & Popovska-Mitrovikj, 2018)	1320	25	17–23
Bitcoin-NG (Das, 2021)	1000	600	Limited
Elastico (Luu et al., 2016)	1600	711	Four times higher than bitcoin
ByzCoin (Xie et al., 2019)	1008	15–20	Two times higher than bitcoin
BigchainDB (McConaghy et al., 2016)	32	15–20	1000
Solida (Abraham, Malkhi, Nayak, Ren, & Spiegelman, 2016)	1000	23.6	30

Table 2
Critical analysis of existing technologies for scalable blockchain systems.

Parameters	Years	Approaches	Consensus algorithm	Permissioned	Attack percentage	Immutability	Transaction finality	Open-source
Data Transmission	2016	Xtreme Thinblocks (Xie et al., 2019)	PoW	✗	0.51	High	Probabilistic	✓
	2018	GeeqChain (Conley et al., 2019)	Proof of Honesty	✗	0.42	High	Deterministic	✓
	2016	Compact Blocks (Mišić, Mišić, & Chang, 2020)	PoW	✗	0.51	High	Probabilistic	✓
	2017	Cardano (Seijas, Nemish, Smith, & Thompson, 2020)	Ouroboros	✗	0.51	High	Probabilistic	✓
Block Interval Time	2016	Hyperledger Fabric (Cachin et al., 2016)	PBFT	✓	0.33	High	Deterministic	✓
	2016	BycCoin (Kogias et al., 2016)	PBFT	✗	0.25	High	Deterministic	✓
	2016	Bitcoin-NG (Eyal, Gencer, Sirer, & Van Renesse, 2016)	PoW	✗	0.51	High	Probabilistic	✗
	2017	Solida (Abraham et al., 2016)	Byzantine Consensus Protocol	✗	0.33	High	Probabilistic	✗
Number of Transactions	2015	Lightning Network (Khan et al., 2019)	Multi-signature	✗	0.55	High	Deterministic	✗
	2016	Elastico (Luu et al., 2016)	Byzantine Consensus Protocol	✗	0.25	Low	Deterministic	✓
	2019	vDLT (Yu & He, 2019)	PBFT, DPoS	✗	0.33	High	Deterministic	✓
	2017	SegWit (Mechkaroska et al., 2018)	PoW	✗	0.51	High	Probabilistic	✓
	2017	OmniLedger (Kokoris-Kogias et al., 2018)	Byzantine Consensus Protocol	✗	0.33	Fair	Deterministic	✓
	2015	Duplex Micropayment Channels (Khan et al., 2019)	Multi-signature	✗	0.45	High	Deterministic	✗
Data Storage	2016	BigchainDB (McConaghy et al., 2016)	Bigchaindb Consensus Algorithm	✗	0.33	High	Deterministic	✓
	2017	Filecoin (Vimal & Srivatsa, 2019)	Proof of Storage	✗	0.51	High	Probabilistic	✓
	2015	Blockchain + DHT (Hassanzadeh-Nazarabadi, Küpçü, & Özkasap, 2019)	PoS	✗	0.51	High	Deterministic	✗
	2017	Desema (Klems et al., 2017)	PoW/PoS	✗	0.51	High	Probabilistic	✓
Block Size, Block Frequency, Ordering Service, Client Size, Organization Size, Transport Layer Security, Transaction Size, Block Interval Time	2021	Proposed Framework	PBFT/RAFT	✓	0.33	High	Deterministic	✓

the blockchain network. Moreover, the blockchain adaptor also sends the generated transaction to the client through which the workload is processed. The client's responsibility is to submit the transaction and receive acknowledgment of the response result. The predefined read performance statistic (e.g., successful transaction, latency, transaction per second (TPS), etc.) is analyzed by the transaction traffic measurement, and the response benchmark results are recorded in the benchmark database. In optimization, constraints and benchmark

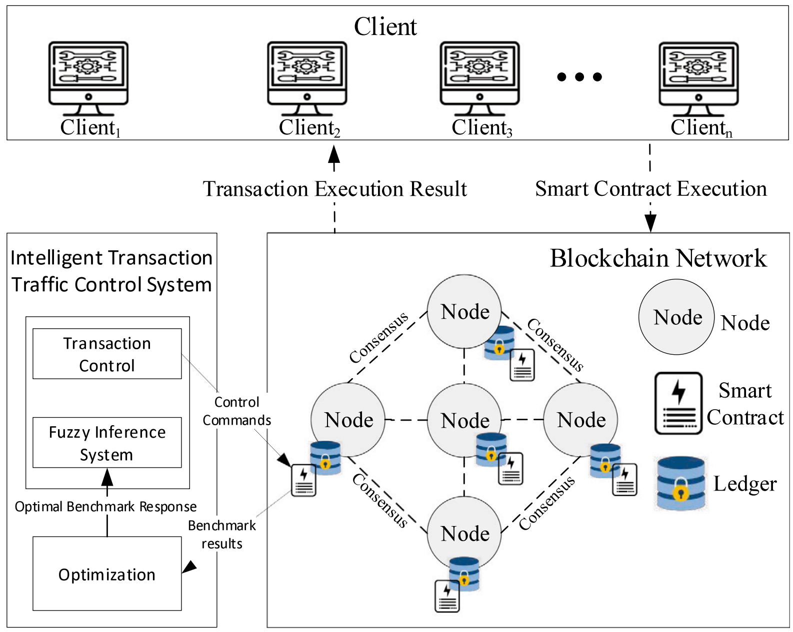


Fig. 2. System overview of transaction traffic control based on optimization and fuzzy logic control.

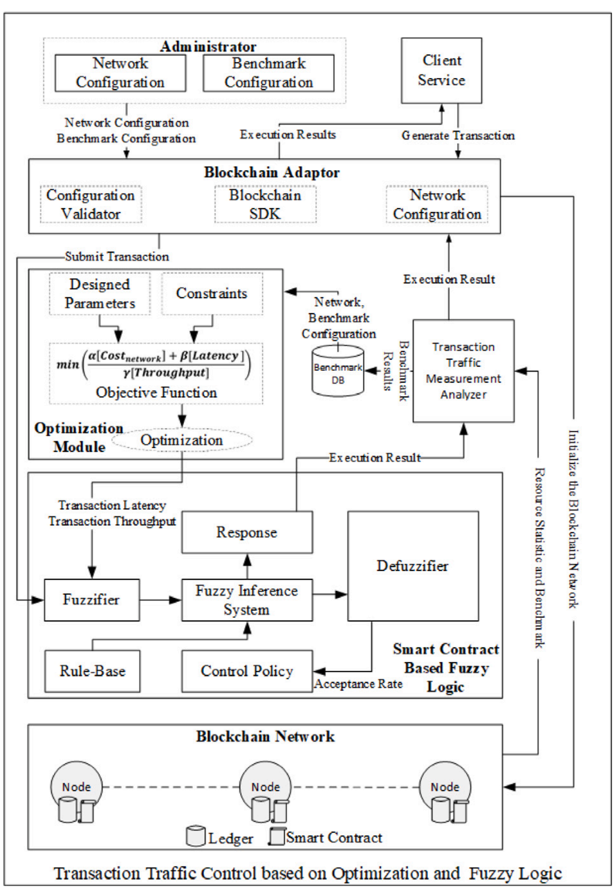


Fig. 3. Hybrid architecture of the transaction traffic control based on Optimization and fuzzy logic.

results data are supplied to the optimizer. The particle swan Optimizer (PSO) yields the velocity and particles data to the optimizer. The benchmark data (i.e., successful transaction, latency, and TPS) is optimized based on PSO constraint and configuration. The optimized benchmark result is passed to the smart contract-enabled fuzzy controller whose responsibility is to fine-tune the transaction acceptance rate by analyzing the acceptance rate with parameters, such as transaction latency and throughput. The fuzzifier takes transaction latency and throughput as input in a fuzzy controller. The inference engine is used to evaluate the rules. The output value, i.e., the acceptance rate, is converted into non-fuzzy values using the defuzzifier . The transaction control module provides and adjusts the output value, i.e., acceptance rate. Finally, the entire process is iterated, and the suitable throughput is achieved dynamically in the blockchain network.

Fig. 4, presented the detailed structure overview of blockchain adopters. The administrator is responsible for updating the network configuration and creating crypto-certificates for every network entity. Furthermore, the administrator also defined the blockchain network topology. The blockchain adopter comprises network configuration, client factory, config validator, blockchain SDK, and client worker. The configuration validator is responsible for validating every network configuration object. The blockchain SDK is used to connect with the network through an interface. Similarly, the information related to network connection can be accessed using connection profile configuration through the network configuration module. Finally, the blockchain adopter can also initialize the network configuration, such as peer and channel and smart contract installation in the network.

Fig. 5, presented the detailed structure of the transaction traffic measurement analyzer. The workload consists of local clients and rate control and behaves like the core functionality of the transaction traffic measurement analyzer. When the analyzer schedules the transactions, the workload module's responsibility is to set up the content of the transactions and submit it to the adopter. Based on the benchmark configuration, numerous local clients are developed, and every individual is linked with the rate control module. The rate control module monitors the transaction rate in two ways, e.g., fixed-rate or pursuing a peculiar profile. Similarly, the resource monitor is responsible for gathering statistics in terms of resource utilization during benchmarking. In contrast, the performance analyzer in the traffic measurement analyzer calculates the benchmark results based on performance statistics. The computed benchmark results are listed in the test report using the report generator and stored in the benchmark database. The configuration component provides run-time configuration and configuration-util to fetch or apply the configuration store.

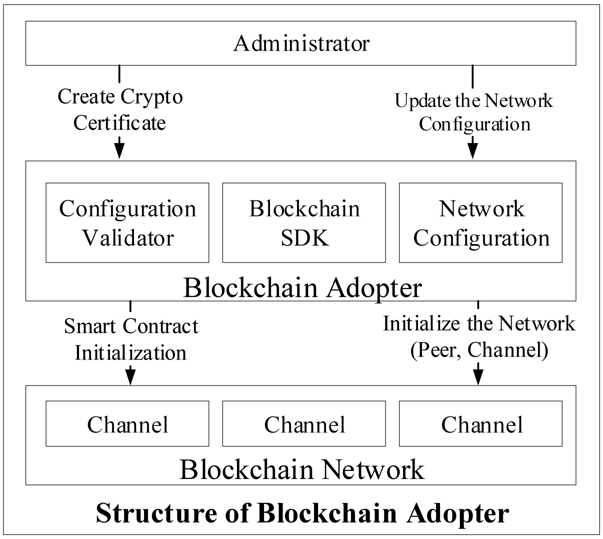


Fig. 4. Detailed structure overview of blockchain adopter.

3.3. Operational model of hybrid transaction traffic control

In this section, we explain the detailed operational model of hybrid transaction traffic control based on optimization and fuzzy control to improve the performance of the blockchain network as shown in Fig. 6. The real-time blockchain network feedback is collected and passed to the optimization module, which computes the optimized performance parameter, such as send rate, throughput, and latency. Initially, the client generated the transaction and passed it to the blockchain adopter along with the network configuration and benchmark file. The blockchain adopter initializes the blockchain network. The blockchain network computes the resource and benchmark statistics and passes them to the transaction traffic measurement analyzer. Afterward, the benchmark statistics, such as latency, throughput, and send-rate, are stored in the benchmark DB. The data is passed to the PSO-based optimization module for an optimal selection of benchmark statistics, where the optimal parameter is selected and passed to the fuzzy controller. The smart contract-enabled fuzzy controller is used to fine-tune the acceptance rate by measuring, analyzing, and matching the benchmark parameters with the acceptance rate. The input to the fuzzifier is the transaction send rate, throughput, and latency. The execution result in the form of acceptance rate is passed to the transaction traffic measurement analyzer and then notified to the client. The process is iterated, and the throughput transaction acceptance rate is maintained at an apt level.

3.3.1. Optimization based on transaction traffic control

In the proposed system, the optimization module aims to find the optimal value of the send-rate, TPS, and latency The mathematical model for the hybrid optimization-based transaction traffic control mechanism is presented in Box I.

3.3.2. Smart contract enabled fuzzy controller based on transaction traffic control

The smart contract-enabled fuzzy controller is used for decision making and consider a main component of the proposed transaction traffic control. In the proposed system, we have used the Mamdani fuzzy inference system, which is used to build a control system using the linguistic rules to maintain the transaction traffic control. The output of every rule is a fuzzy set. The fuzzy logic in the proposed system is comprised of fuzzification, fuzzy control rules, fuzzy inference engine, and defuzzification. Initially, we have defined some rules and membership functions for the proposed hybrid transaction traffic control. Afterward, the rules are constructed for each attribute. The computed outcome from the fuzzy control rules and membership are combined and defuzzified to get the transaction acceptance rate. Table 3 summarized the fuzzy variables along with fuzzy sets and linguistic terms.

Table 3
Fuzzy sets in proposed smart contract-enabled fuzzy controller.

Input	Terms	Fuzzy sets
Transaction throughput	VL	0,0,0.2,0.6
	L	0.2,0.6,1
	A	0.6,1,1.4
	H	1,1.4,1.8
	VH	1.4,4.8,2,2
Network latency	VL	0,0,0.0015,0.0045
	L	0.0015,0.0045,0.0075
	A	0.0015,0.0075,0.105
	H	0.0075,0.0105,0.103
	VH	0.0105,0.0103,0.0105,0.0105
Acceptance rate	VL	0,0,0.1,0.3
	L	0.1,0.3,0.5
	M	0.3,0.5,0.7
	H	0.5,0.7,0.9
	VH	0.7,0.9,1,1

Table 4
Linguistic-Terms defined in the proposed smart contract-enabled fuzzy controller.

Linguistic-terms	Terms
Very Low	VL
Low	L
Acceptable	A
High	H
Very High	VH
Medium	M

The main aim of the proposed smart-contract-enabled fuzzy controller is to maintain the optimal acceptance rate of the transaction. For instance, if the transaction throughput is high and the transaction latency is low, the acceptance rate will be medium. In contrast, the smart contract enabled fuzzy controller to maintain the transaction traffic in terms of network latency and throughput. Table 4 presents the linguistic terms used for defining the membership function and rules for the proposed smart-contract enabled fuzzy controller.

The proposed smart contract-enabled fuzzy controller is based on trapezoidal fuzzy membership functions, which indicate the fuzzy variables inside the fuzzy system. The fuzzy set of trapezoidal X produces $\mu X(y)$, which refers to four attributes, such as a,b,c, and d. The mathematical formulation of the proposed smart contract enabled fuzzy controller is given in Eq. (8).

$$\mu_A(y) = \begin{cases} 0, & y < a \\ \frac{y-a}{b-a}, & a < y < b \\ 1, & b < y < c \qquad (a < b \le c \le d) \\ \frac{d-y}{d-c}, & c < y < d \\ 0, & y > d \end{cases} \tag{8}$$

Similarly, Eq. (9) represents the operations of fuzzy intersection between A and B fuzzy sets.

$$\mu_{A\cap B}(Y) = min\mu_A(Y), \mu_B(Y), \quad \forall Y \in U \tag{9}$$

Likewise, the union of fuzzy operation between set A and B is defined in Eq. (9)

$$\mu_{A\cup B}(Y) = max\mu_A(Y), \mu_B(Y), \quad \forall Y \in U \tag{10}$$

Where U contains the set A and B. X represents the factor within space U.

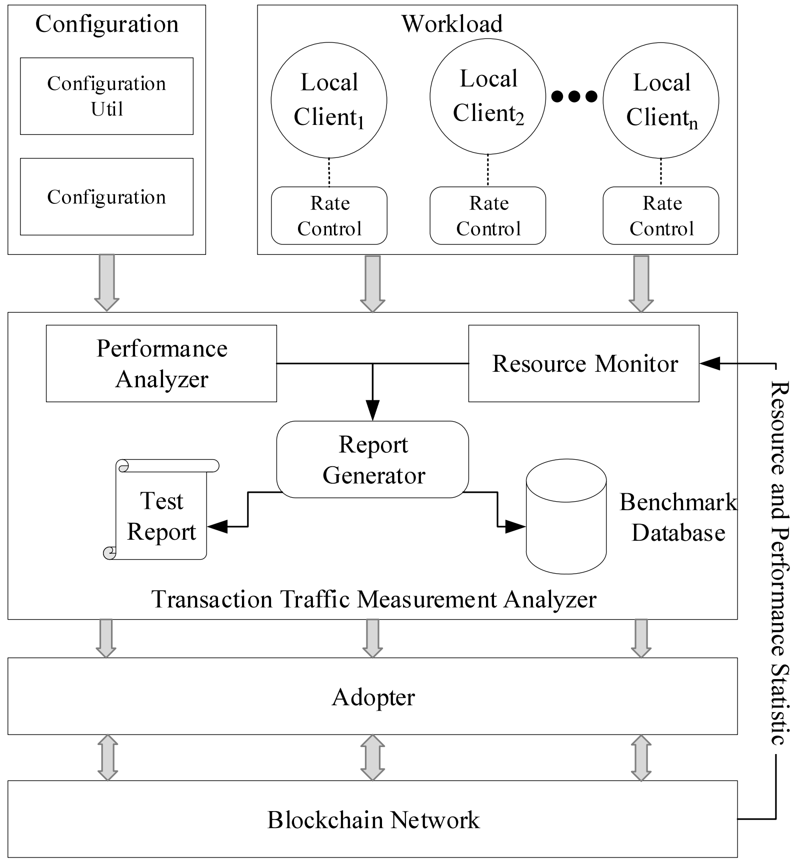


Fig. 5. Detailed Overview of the transaction traffic measurement analyzer.

$$min(\frac{Cost_{total} + Latency}{Throughput}) \tag{1}$$

$$Throughput = (\frac{TotalSuccessfulTransaction}{Throughput}) \tag{2}$$

$$Latency = ConfirmationTime - SubmissionTime \tag{3}$$

$$Cost_{Network_{Total}} \propto CPU_{utilization} + Memory_{utilization} + Traffic_{Network_{utilization}} \tag{4}$$

$$Latency_{Network} \propto (\frac{Client_{Total} + EndorsementPolicey_{Total} + Organization_{Total} + OrderingService}{BlockSize_{Total} + BlockFrequency_{Total}}) \tag{5}$$

$$Throughput_{Network} \propto (\frac{1}{Latency_{Network}}) \tag{6}$$

$$Latency_{Network} = \begin{cases} BlockSize = 30, 50, 100 \in \mathbb{R} \\ EndorsementPolicy = P \\ OrderingService = Solo, Raft, SoloRaft \\ TLS = Yes, No \\ Client_{Total} = 1, 5, 10 \in \mathbb{R} \\ BlockFrequency_{Total} = 30, 50, 100 \in \mathbb{R} \end{cases} \tag{7}$$

Box I.

In the designed smart contract-enabled fuzzy controller, we have considered two inputs: transaction throughput and latency, whereas the output will be the transaction acceptance rate. Furthermore, the input parameter is used by the fuzzy operator in order to define fuzzy rules. A fuzzy inference system evaluates every rule through a membership function. Eq. (11) defines the computation of fuzzy sets along with the

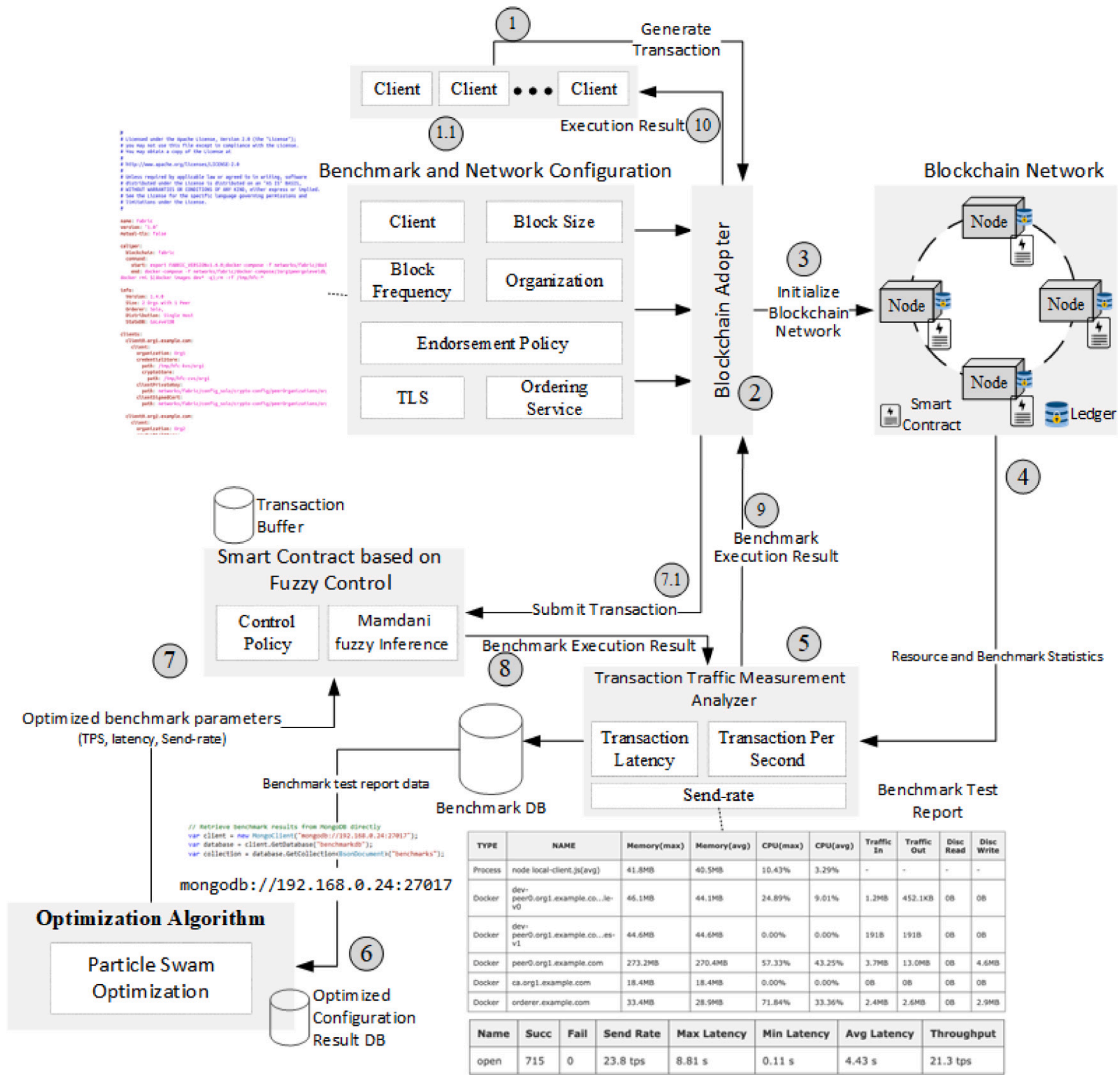


Fig. 6. Configuration of hybrid transaction traffic control based on optimization and fuzzy logic.

membership function.

$$mCoA = \frac{\int f(y).y dy}{\int f(y) dy} \tag{11}$$

where mCoA is denoted as modified center of area is a defuzzification method used to define the range of membership function input and output in terms of scaling. The aggregated membership function is represented by $f(y)$ and y depicts the output value.

3.4. Execution process of transaction traffic control

The execution process of transaction traffic control is presented in Fig. 7. Initially, the network administrator fulfills the test scenario by configuring the benchmark profile and blockchain network. The benchmark file contains the network configurations, such as transaction send rate, the total number of rounds, and test network settings used for the network evaluation test. Likewise, the client number, nodes configuration, and smart contract are defined in the network configuration file used to construct network topology. when the network is set up; the network administrator executes the script to start the benchmark test. Afterward, the transaction generated by the clients to the adopter is submitted to the fabric network. Concurrently, the transaction traffic measurement analyzer analyzed and gathered the benchmark results to compute the statistics further stored in benchmark DB. The benchmark results, such as latency, TPS, and successful transaction, are computed using a transaction traffic measurement analyzer are passed to the optimization module. Initially, each module is initialized and collected from the transaction traffic measurement analyzer. The configuration and constraints, along with the benchmark data, are input to the optimizer. The optimizer gets the velocity and particle information from the PSO in order to optimize the benchmark results based on the PSO configuration and constraints. The optimal benchmark results are passed to the smart contract-enabled fuzzy logic controller for further processing. As discussed earlier, the main aim of the optimization module is to get the optimal value of TPS and latency from the transaction traffic measurement analyzer. Similarly, the smart-contract-enabled fuzzy controller fetches the optimal latency, TPS, and successful transaction and passes to the fuzzy inference system. The input parameters are evaluated based on the rules defined in the inference engine. The defuzzifier computes the output in terms of acceptance rate, which is passed to the transaction control module as an input. The transaction module considers the acceptance rate to perform transaction traffic control operations. Finally, the client receives the response of the execution of the transaction.

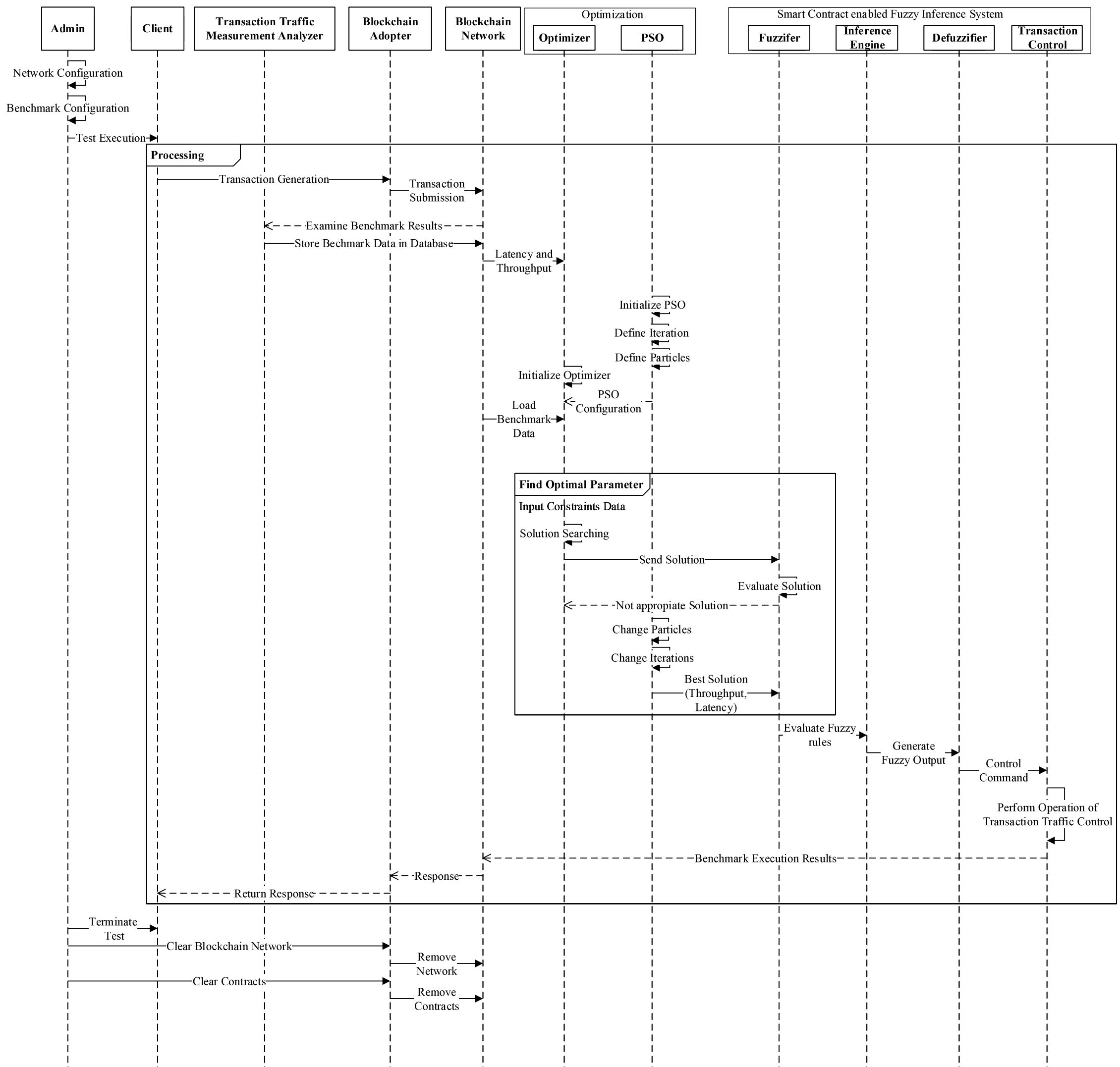


Fig. 7. Sequence diagram of the hybrid optimization based transaction traffic control.

4. Implementation of the hybrid transaction traffic control

4.1. Development environment

The tools and technologies used for the proposed hybrid model are presented in Table 5. The blockchain framework used in the proposed system is Hyperledger Fabric version 1.4.1, which is installed on a Linux operating system. We have used a virtual machine where each Hyperledger Fabric element are embedded in a docker container in the form of docker images. The Representational State Transfer (REST) API is developed in order to establish interaction between the fabric network and the client application. The REST API is used to commit and query transactions to and from the distributed ledger. Furthermore, we have used the Hyperledger Caliper version (version 2.0.0) to access the blockchain performance in latency and throughput. Particle Swarm Optimization algorithm is used to compute the real-time optimal performance measure parameters. In the case of the smart contract-enabled Fuzzy control module, we have used FuzzyIS; an open-source JavaScript library used to develop a fuzzy inference system using Node.js. The database used in the proposed model is MongoDB, a non-SQL database platform that is integrated to store the Benchmark result in the form of JSON. The front-end application Express.js server is implemented for interacting with MongoDB via REST APIs to consume services.

Table 5
Tools and technologies for the proposed hybrid model.

Element	Specification
CPU	Intel Core i-5-3.00 GHz
Memory	8 Giga Byte
Operating System	Ubuntu Linux (Version 18.4 LTS)
Docker-Engine	Version 19.3.8
Docker-Composer	Version 1.24
Node SDK	Node.js (Version 8.17.0)
Blockchain Platform	Hyperledger Fabric (Version 1.4.1)
Transaction Traffic Measurement Tool	Hyperledger Caliper (Version 2.0)
Fuzzy Inference System	Fuzzy IS (Mamdani Fuzzy Inference System)
Optimization Algorithm	Particle Swarm Optimization
Database Management System	Mongo DB
Web-server	Express JS
Development Toolkit	Visual Studio
Language	JavaScript

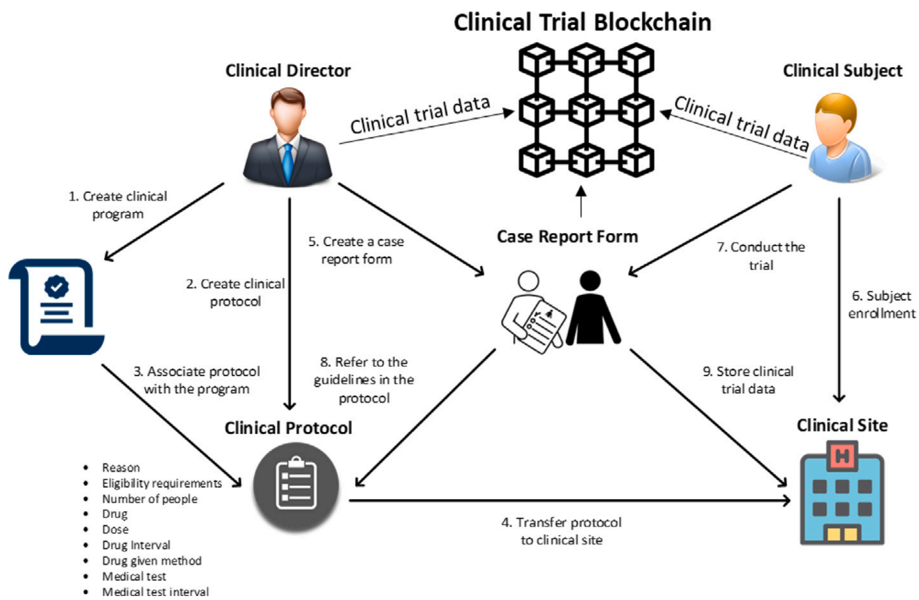


Fig. 8. Scenario of the proposed clinical trial service platform.

4.2. Use case implementation and deployment

In the proposed model, we have considered clinical trial service as a case study scenario. The proposed clinical trial service platform consists of five participants, such as healthcare devices, subject(patient), principal investigator (PI), research coordinator (RC), research associate(RA), and clinical director as shown in Fig. 8. The clinical director acts as the administrator responsible for creating a clinical trial program defining medicine, medical instruments, and clinical protocols. Similarly, the RC and RA carry out the clinical trial process at the clinical site. The RC also manages the data related to the clinical subject, whereas the PI manages the clinical trial's overall activity. The PI's job is to verify the execution of defined protocol and tasks of involved members in the clinical trial process. The clinical subject is enrolled in order to start the clinical process by the RC. The most important component of the clinical trial program is acquiring clinical data from the subject involved in the clinical trial. The bio-medical data is collected using different bio-medical devices from the subject. Each transaction involved in the clinical trial process is stored in the blockchain. Each service of the clinical trial process is defined in the form of REST APIs. The user of the clinical trial can consume the back-end service by invoking the specific REST API through the client application. Each step of the clinical trial is defined in the smart contract used to automate the clinical trial process. The DL records each individual step of the clinical trial with the timestamp, and any change in the ledger state is also reflected in the client application through REST API. Each record is stored in a key value representing the current ledger state.

Fig. 9 presents the proposed clinical trial platform's detailed configuration. The user of the clinical trial accesses the system through a clinical trial client application to consume the services defined in the smart contract. The smart contract is defined to automate the clinical trial process and used to define the core business logic of the entire clinical trial. The implemented clinical trial blockchain network contains a distributed ledger stored in a key–value format. The key–value database holds the current state of the ledger and updates the database after executing every transaction. The clinical trials network comprises channels where each channel has several identities, clinical organization, and rules for clinical trials operations. The proposed system is scalable to multiple clinical trials with a separate private channel where only the data is shared among the member of the organizations. In the designed clinical trial case study, every channel has four organizations(refer to an entity, such as a hospital, medical management company, research organization, etc.) that retain the ledger state to maintain database consistency.

4.2.1. Smart contract modeling

The smart contracts in the developed case study are comprised of assets, participants, and transactions. The participant is the users (e.g., PI, RC, RA, Clinical director, clinical subject) of the clinical trial blockchain who enroll in the designed business network. Similarly, the assets are the resources or entities (e.g., pillbox, eCRF bgm data). The designed application provides CRUD(Create, Read, Update, Delete) functionality on clinical trial resources in the form of a transaction initiated by the system user. Once the designed smart contract is deployed, all transactions in the smart contracts are visible to the client application as services. Furthermore, several access control rules have been defined to provide authorization to the participant on specific resources. For instance, the PI can only read and create the Electronic Case Report Form (eCRF). Moreover, we have also defined the Representational state transfer (REST) application programming interface (API) used to expose the back-end clinical services to the front-end client application. Every API has Uniform Resource Identifier (URI) and verbs. The verb is the actions performed on the specific resources deployed on a particular URI.

5. Performance evaluation

This section explained the evaluation setup used to obtain results of the developed blockchain-enabled clinical trial service platform. For performance evaluation, we have used Hyperledger Caliper (Sukhwani, Wang, Trivedi, & Rindos, 2018), an open-source benchmarking tool and framework used to evaluate the performance of the proposed clinical trial service framework in terms of transaction throughput and latency. In the proposed system, we have considered eight configurable blockchain network parameters that are used to improve the performance of the proposed model, as shown in Table 6.

The latency and throughput are the two parameters that are used to analyze the performance of the designed clinical trial service platform. The throughput, also known as transaction per second, is segregated into two sub-groups: transaction throughput and read throughput. The transaction throughput is a valid transaction executed during a defined time period, also known as transaction per second (tps) as shown in Eq. (12).

$$Transaction\ per\ second = \frac{Successful\ Transaction}{Time(Sec)} \tag{12}$$

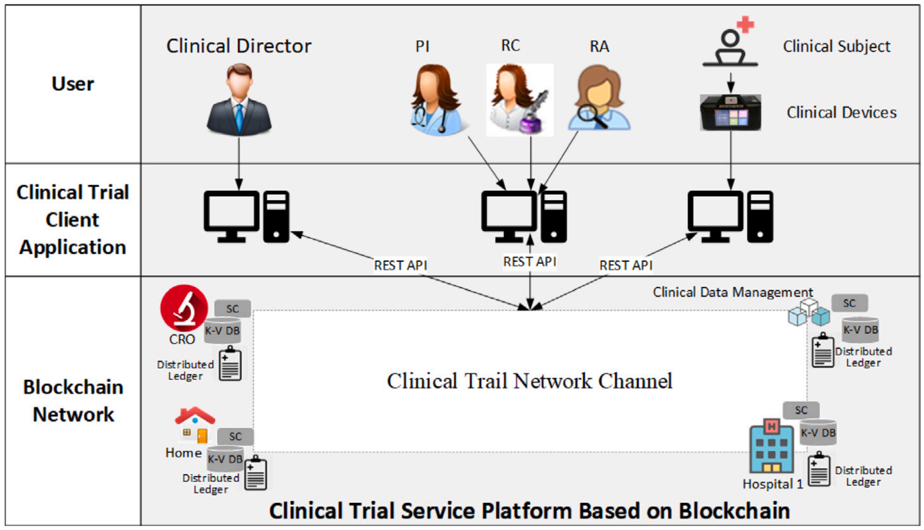


Fig. 9. Configuration of proposed clinical trial service platform.

Table 6
Configurable blockchain network parameters for performance evaluation.

Parameters	Values	Description
Organizations Count	1,2,3	The organization count represents the members who are able to join the blockchain network.
Number of Endorser Peers	2,4,6	Endorser peers are used to endorse transaction once it is proposed.
Size of Block	100 (transaction per block)	The block size represents the maximum transactions per block
Database	LevelDB	The ledger database is used to store the data into the state database.
Ordering Service	Solo, Solo Raft, Raft	The ordering service is used to order the transactions
Transport Layer Security	Yes, No	Transport Layer Security is used to provide additional security to encrypt data among peers.
Number of Clients	1,5,10	The clients are the end-users
Frequency of Block(ms) (maximum timeout to create a block)	50,100, 250	The block frequency determines the maximum timeout to create a block.
Endorsement Policy	Policy1, Policy2, Policy 3	The endorsement policy defines which peers need to agree on the results of a transaction before it can be appended to the ledger.

Furthermore, the tps is measured using all nodes across the entire blockchain. Similarly, the read throughput is calculated as a total count of read operations in a specifically defined time known as read per second (rps) as shown in Eq. (13).

$$Read\ Per\ Second = \frac{Read\ Transaction}{Time(Sec)} \tag{13}$$

The proposed clinical trial service framework is evaluated in terms of latency, such as read and transaction latency. The read latency is the total round trip time between the send request and receives a response as computed in Eq. (14).

$$Read\ Latency = Request\ response\ time - Request\ time \tag{14}$$

Similarly, the transaction latency is the total time to authenticate a transaction, including the processing of the consensus algorithm. We also defined a network threshold used to define the time to commit the transaction in the developed system. This paper uses Practical Byzantine Fault Tolerance (PBFT); therefore, the network threshold is set to a hundred. The mathematical formulation of transaction latency is a calculation using Eq. (15).

$$Transaction\ Latency = Transaction\ commit\ time \times network\ threshold - Request\ time \tag{15}$$

5.1. Experimental results

Fig. 10 investigated the throughput and latency evaluation results with predefined configurable blockchain network parameters, such as one organization, one client, and two peers. The baseline latency of the network is compared with the hybrid approach based on optimization and fuzzy logic techniques with varying send rate range from 25 tps to 200 tps. It is investigated from the graph that the optimized latency of the proposed clinical trial service framework in the best case is recorded as 75 ms at 200 send rate as compared to baseline latency of the network, which is recorded as 80 ms. Similarly, the optimized throughput of the proposed framework is 180 tps compared to the baseline and fuzzy-based approach recorded as 160 and 166, respectively, over the send rate of 200 tps. It is observed that the throughput of the designed system is increasing linearly until send rate of 150 tps.

Fig. 11, shows the graph of impact of network parameters with one organization, two peers, and five clients on transaction and latency. Increasing the number of clients improves the throughput. In the case of

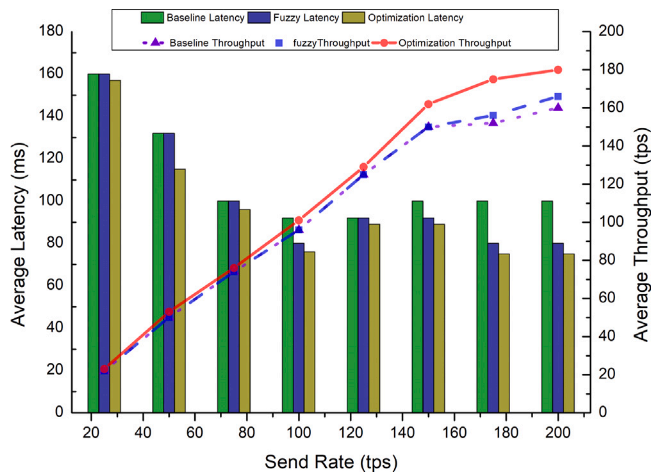


Fig. 10. Impact of network parameters with 1 organization, 2peers, and 1 client on transaction and latency.

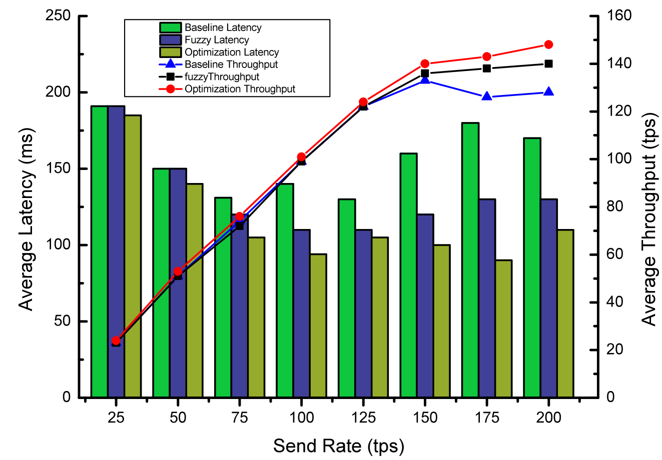


Fig. 12. Impact of network parameters with 2 organization, 2peers, and 1 client on transaction and latency.

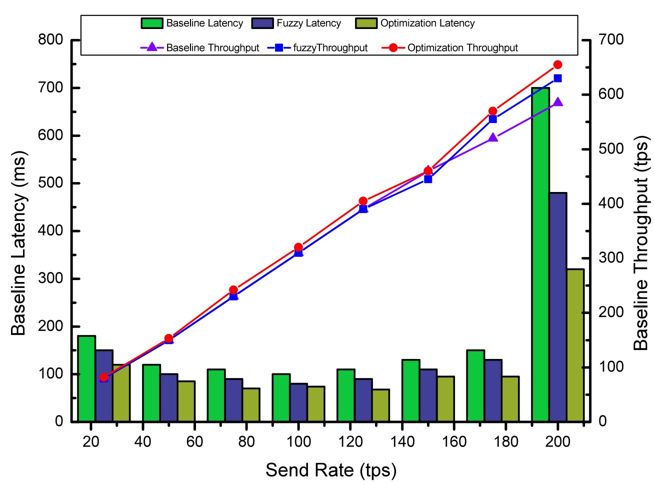


Fig. 11. Impact of network parameters with 1 organization, 2peers, and 5 client on transaction and latency.

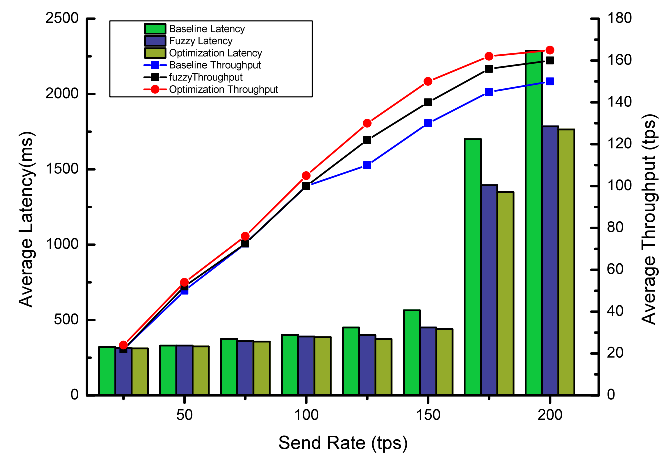


Fig. 13. Impact of network parameters with 2 organization, 2peers, and 5 client on transaction and latency.

5 clients, the optimized throughput in the best case is 655 transactions per second compared to the baseline throughput of the blockchain network, which is recorded as 585 tps at send rate of 200 tps. Similarly, 700 ms latency is recorded as a worst-case of the baseline network at send rate of 200 tps. However, the latency of the proposed model is recorded as 480 ms and 320 ms in the case of the fuzzy-based approach and PSO-based optimization approach at the send rate of 320 tps. It is observed from the graph that the number of clients does have a significant effect on performance. Increasing the number of clients can improve the throughput, but increasing it too much can significantly increase the latency due to increased network traffic volume.

Fig. 12, shows the impact of network parameters with configurable network parameters, such as two organizations, two peers, and one client, on transaction and latency. The baseline latency of the proposed clinical trial framework in worst and best cases is recorded as 700 ms and 100 ms at send rates of 100 and 200 tps. Similarly, as compared to baseline latency, the proposed fuzzy-based approach and optimized technique in the best and worst case is computed as 80 ms, 70 ms, and 480 ms,320 ms, respectively. Similarly, the throughput is increased linearly until the optimal send rate of 160 tps. However, the optimized throughput using the optimization approach performed better than the baseline and fuzzy-based approaches. The baseline throughput decreases with the rate of 3% after the increment of every 25 tps send rate.

The impact of network with two peers, two organizations, and five clients on transaction and latency of the proposed clinical trial service framework is shown in Fig. 13. It is observed from the graph that the latency is increased linearly after the optimal send rate of 150 tps. The worst latency recorded in the Similarly, the throughput of the optimized network increases with the rate of 8.1% compared to the baseline network. Likewise, the proposed fuzzy-based approach improves network throughput with a rate of 7.3% compared to the baseline network. It is observed from the graph that increasing the number of clients improves the network throughput.

6. Significance and comparison

In this paper, we implement a real-time case study of a blockchain-based clinical-trial service framework designed as part of experimental results carried out to evaluate the effectiveness of the proposed intelligent transaction traffic control using a hybrid approach based on optimization and fuzzy logic. The developed system is built on the top of hyperledger fabric, a permissioned blockchain network that maximizes throughput and minimizes latency. The proposed model is compared with one of the existing studies to demonstrate the proposed system's practicability. The Nextledger Accelerator (Lee et al.,

Table 7
Comparative analysis of proposed model with existing studies.

Approach	Total number of clients	Send-Rate (tps)	Throughput (tps)	Latency (ms)
Accelerator	1	200	165	420
	5	200	246	1880
	10	200	327	3340
Baseline-Network	1	200	110.5	635
	5	200	147.6	2325
	10	200	207.4	2742
Accelerator integrated with proposed model	1	200	190.4	290
	5	200	284.3	1180
	10	200	391.6	1675

2019) is an engine used to process the transaction of the blockchain-based applications. The accelerator act as an intermediate between the blockchain network and client application which provides the intelligent processing of transaction using scheduling and transaction processing algorithms. Furthermore, the accelerator classifies every incoming transaction in the form of a batch to avoid congestion. Table 7 presents the comparison of the proposed framework with the accelerator in terms of latency and throughput.

This study proposed intelligent transaction traffic control using a hybrid approach based on PSO-based optimization and fuzzy logic-based inference to control and enhance the performance of the designed blockchain network. The proposed study is scalable and capable of controlling numerous IoT devices in the domain of healthcare, supply chain, and mission-critical blockchain-based IoT systems. Many existing studies have a shortcoming in terms of time complexity, such as latency and throughput. However, the designed intelligent transaction traffic control enhances outperformed the other state-of-the-art approach in terms of latency and throughput while considering send rate(tps).

7. Conclusions and future work

The performance metrics, such as latency and throughput, have been considered as the primary factor for measuring the effectiveness of information systems. This article presents intelligent traffic control using a hybrid model based on the PSO-based optimization algorithm and fuzzy logic to improve blockchain performance. The real-time network feedback model is designed and used as an input to control the transaction traffic across the entire network in a robust way without human intervention. In order to evaluate the effectiveness of the designed model, a clinical trial service framework as a test network is implemented on top of Hyperledger Fabric. The case study is further compared with baseline network, network with fuzzy approach, and network with optimized parameter. The experiments show that the proposed model enhanced the network by maximizing the network throughput and minimizing the network latency. Furthermore, we have enhanced the open-source performance analyzer tool known as an accelerator which includes the function of the proposed model. In order to evaluate the effectiveness of the designed model, a clinical trial service framework as a test network is implemented on top of Hyperledger Fabric. The outcome of this study indicated a remarkable increase in transaction throughput (i.e., 38.5%) and a decrease in transaction latency of 40.5%. Moreover, the proposed model can easily be integrated with other existing blockchain-based performance-enhancing tools. The results indicate that the improved version of the accelerator outperformed the baseline model in terms of throughput and latency.

CRediT authorship contribution statement

Faisal Jamil: Conceptualization, Methodology, Software, Writing – original draft, Formal analysis, Software, Visualization. **Shabir Ahmad:** Data curation, Writing – original draft, Visualization. **Taeg Keun Whangbo:** Project administration, Visualization, Investigation, Writing – original draft, Supervision. **Ammar Muthanna:** Formal analysis, Software, Resources, Writing – review & editing. **Do-Hyeun Kim:** Supervision, Project administration, Funding acquisition, Software, Validation.

Acknowledgments

This research was supported by in parts by Energy Cloud R&D Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Science, ICT (2019M3F2A1073387), and by Institute of Information & communications Technology Planning & Evaluation (IITP), South Korea grant funded by the Korea government (MSIT) (2021-0-00188, Open source development and standardization for AI-enabled IoT platforms and interworking), and by GRRC, South Korea program of Gyeonggi province [GRRC-Gachon2020(B04), Development of AI-based Healthcare Devices], and by the National Research Foundation of Korea (NRF) funded by the Basic Education (2021R1I1A1A01045177), and also been supported by the RUDN University Strategic Academic Leadership Program (recipient Ammar Muthanna)

References

Abraham, I., Malkhi, D., Nayak, K., Ren, L., & Spiegelman, A. (2016). Solida: A blockchain protocol based on reconfigurable byzantine consensus. arXiv preprint arXiv:1612.02916.

Abraham, I., Malkhi, D., et al. (2017). The blockchain consensus layer and BFT. *Bulletin of EATCS*, *3*(123).

Ahmad, S., Ullah, I., Jamil, F., & Kim, D. (2020). Toward the optimal operation of hybrid renewable energy resources in microgrids. *Energies*, *13*(20), 5482.

Aziz, M. S., Khan, M. A., Jamil, H., Jamil, F., Chursin, A., & Kim, D.-H. (2021). Design and analysis of in-pipe hydro-turbine for an optimized nearly zero energy building. *Sensors*, *21*(23), 8154.

Bhardwaj, A., Shah, S. B. H., Shankar, A., Alazab, M., Kumar, M., & Gadekallu, T. R. (2020). Penetration testing framework for smart contract blockchain. *Peer-To-Peer Networking and Applications*, 1–16.

Cachin, C., et al. (2016). Architecture of the hyperledger blockchain fabric. In *Workshop on distributed cryptocurrencies and consensus ledgers, vol. 310*. (4), Chicago, IL.

Conley, J. P., et al. (2019). *Blockchain as a decentralized mechanism for financial inclusion and economic mobility*. Vanderbilt University, Department of Economics.

Danish, S. M., Zhang, K., & Jacobsen, H.-A. (2020). Blockam: An adaptive middleware for intelligent data storage selection for internet of things. In *2020 IEEE International conference on decentralized applications and infrastructures*.

Das, D. (2021). Toward next generation of blockchain using improvized bitcoin-NG. *IEEE Transactions on Computational Social Systems*, *8*(2), 512–521.

Duong, T., Chepurnoy, A., Fan, L., & Zhou, H.-S. (2018). Twinscoin: A cryptocurrency via proof-of-work and proof-of-stake. In *Proceedings of the 2nd ACM Workshop on blockchains, cryptocurrencies, and contracts* (pp. 1–13).

Eyal, I., Gencer, A. E., Sirer, E. G., & Van Renesse, R. (2016). Bitcoin-ng: A scalable blockchain protocol. In *13th {USENIX} Symposium on networked systems design and implementation ({NSDI} 16)* (pp. 45–59).

Hameed, K., Barika, M., Garg, S., Amin, M. B., & Kang, B. (2021). A taxonomy study on securing blockchain-based industrial applications: An overview, application perspectives, requirements, attacks, countermeasures, and open issues. arXiv preprint arXiv:2105.11665.

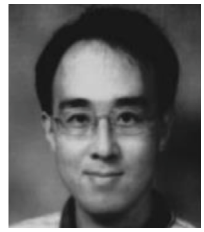
Hasan, H. R., & Salah, K. (2018). Proof of delivery of digital assets using blockchain and smart contracts. *IEEE Access*, *6*, 65439–65448.
Hassanzadeh-Nazarabadi, Y., Küpçü, A., & Özkasap, O. (2019). Lightchain: A dht-based blockchain for resource constrained environments. arXiv preprint arXiv:1904.00375.
Ibrahim, M., Jamil, F., Lee, Y., & Kim, D. (2022). Blockchain based secured load balanced task scheduling approach for fitness service. *CMC-COMPUTERS MATERIALS & CONTINUA*, *71*(2), 2599–2616.
Jamil, F., Ahmad, S., Iqbal, N., & Kim, D.-H. (2020). Towards a remote monitoring of patient vital signs based on IoT-based blockchain integrity management platforms in smart hospitals. *Sensors*, *20*(8), 2195.
Jamil, F., Hang, L., Kim, K., & Kim, D. (2019). A novel medical blockchain model for drug supply chain integrity management in a smart hospital. *Electronics*, *8*(5), 505.
Jamil, F., Ibrahim, M., Ullah, I., Kim, S., Kahng, H. K., & Kim, D.-H. (2022). Optimal smart contract for autonomous greenhouse environment based on IoT blockchain network in agriculture. *Computers and Electronics in Agriculture*, *192*, Article 106573.
Jamil, F., Iqbal, N., Ahmad, S., Kim, D., et al. (2021). Peer-to-peer energy trading mechanism based on blockchain and machine learning for sustainable electrical power supply in smart grid. *IEEE Access*, *9*, 39193–39217.
Jamil, F., Iqbal, M. A., Amin, R., & Kim, D. (2019). Adaptive thermal-aware routing protocol for wireless body area network. *Electronics*, *8*(1), 47.
Jamil, F., Kahng, H. K., Kim, S., & Kim, D.-H. (2021). Towards secure fitness framework based on IoT-enabled blockchain network integrated with machine learning algorithms. *Sensors*, *21*(5), 1640.
Jamil, F., Qayyum, F., Alhelaly, S., Javed, F., & Muthanna, A. (2021). Intelligent microservice based on blockchain for healthcare applications. *CMC Computer, Materials and Continua*, *69*, 2513–2530.
Jamil, H., Qayyum, F., Jamil, F., & Kim, D.-H. (2021). Enhanced pdr-ble compensation mechanism based on hmm and awcla for improving indoor localization. *Sensors*, *21*(21), 6972.
Khan, N., et al. (2019). Lightning network: A comparative review of transaction fees and data analysis. In *International congress on blockchain and applications* (pp. 11–18). Springer.
Kim, S., Kwon, Y., & Cho, S. (2018). A survey of scalability solutions on blockchain. In *2018 International conference on information and communication technology convergence*.
Klems, M., Eberhardt, J., Tai, S., Härtlein, S., Buchholz, S., & Tidjani, A. (2017). Trustless intermediation in blockchain-based decentralized service marketplaces. In *International conference on service-oriented computing* (pp. 731–739). Springer.
Kogias, E. K., Jovanovic, P., Gailly, N., Khoffi, I., Gasser, L., & Ford, B. (2016). Enhancing bitcoin security and performance with strong consistency via collective signing. In *25th {Usenix} security symposium ({Usenix} security 16)* (pp. 279–296).
Kokoris-Kogias, E., Jovanovic, P., Gasser, L., Gailly, N., Syta, E., & Ford, B. (2018). Omniledger: A secure, scale-out, decentralized ledger via sharding. In *2018 IEEE Symposium on security and privacy*.
Latif, R. M. A., Hussain, K., Jhanjhi, N., Nayyar, A., & Rizwan, O. (2020). A remix IDE: smart contract-based framework for the healthcare sector by using blockchain technology. *Multimedia Tools and Applications*, 1–24.
Lee, H., Yoon, C., Bae, S., Lee, S., Lee, K., Kang, S., et al. (2019). Multi-batch scheduling for improving performance of hyperledger fabric based IoT applications. In *2019 IEEE Global communications conference*.
Luu, L., Narayanan, V., Zheng, C., Baweja, K., Gilbert, S., & Saxena, P. (2016). A secure sharding protocol for open blockchains. In *Proceedings of the 2016 ACM SIGSAC Conference on computer and communications security* (pp. 17–30).
Maull, R., Godsiff, P., Mulligan, C., Brown, A., & Kewell, B. (2017). Distributed ledger technology: Applications and implications. *Strategic Change*, *26*(5), 481–489.
McConaghy, T., Marques, R., Muller, A., De Jonghe, D., McConaghy, T., McMullen, G., et al. (2016). Bigchaindb: a scalable blockchain database. *White Paper, BigChainDB*.
Mechkaroska, D., Dimitrova, V., & Popovska-Mitrovikj, A. (2018). Analysis of the possibilities for improvement of blockchain technology. In *2018 26th Telecommunications forum*.
Mišić, J., Mišić, V. B., & Chang, X. (2020). Performance of bitcoin network with synchronizing nodes and a mix of regular and compact blocks. *IEEE Transactions on Network Science and Engineering*, *7*(4), 3135–3147.
Mistry, I., Tanwar, S., Tyagi, S., & Kumar, N. (2020). Blockchain for 5G-enabled IoT for industrial automation: A systematic review, solutions, and challenges. *Mechanical Systems and Signal Processing*, *135*, Article 106382.
Moin, S., Karim, A., Safdar, Z., Safdar, K., Ahmed, E., & Imran, M. (2019). Securing IoTs in distributed blockchain: Analysis, requirements and open issues. *Future Generation Computer Systems*, *100*, 325–343.
Mougayar, W. (2016). *The business blockchain: promise, practice, and application of the next internet technology*. John Wiley & Sons.
Muratov, F., Lebedev, A., Iushkevich, N., Nasrulin, B., & Takemiya, M. (2018). YAC: BFT consensus algorithm for blockchain. arXiv preprint arXiv:1809.00554.
Seijas, P. L., Nemish, A., Smith, D., & Thompson, S. (2020). Marlowe: implementing and analysing financial contracts on blockchain. In *International conference on financial cryptography and data security* (pp. 496–511). Springer.
Shahbazi, Z., & Byun, Y.-C. (2020). Towards a secure thermal-energy aware routing protocol in Wireless Body Area network based on blockchain technology. *Sensors*, *20*(12), 3604.
Shahbazi, Z., & Byun, Y.-C. (2021a). A framework of vehicular security and demand service prediction based on data analysis integrated with blockchain approach. *Sensors*, *21*(10), 3314.
Shahbazi, Z., & Byun, Y.-C. (2021b). Improving transactional data system based on an edge computing–blockchain–machine learning integrated framework. *Processes*, *9*(1), 92.
Shahbazi, Z., & Byun, Y.-C. (2021c). Integration of blockchain, IoT and machine learning for multistage quality control and enhancing security in smart manufacturing. *Sensors*, *21*(4), 1467.
Shahbazi, Z., & Byun, Y.-C. (2021d). A procedure for tracing supply chains for perishable food based on blockchain, machine learning and fuzzy logic. *Electronics*, *10*(1), 41.
Shahbazi, Z., & Byun, Y.-C. (2021e). Smart manufacturing real-time analysis based on blockchain and machine learning approaches. *Applied Sciences*, *11*(8), 3535.
Sousa, J., Bessani, A., & Vukolic, M. (2018). A byzantine fault-tolerant ordering service for the hyperledger fabric blockchain platform. In *2018 48th Annual IEEE/IFIP International conference on dependable systems and networks*.
Sukhwani, H., Wang, N., Trivedi, K. S., & Rindos, A. (2018). Performance modeling of hyperledger fabric (permissioned blockchain network). In *2018 IEEE 17th International symposium on network computing and applications*.
Vimal, S., & Srivatsa, S. (2019). A new cluster P2P file sharing system based on IPFS and blockchain technology. *Journal of Ambient Intelligence and Humanized Computing*, 1–7.
Xie, J., Yu, F. R., Huang, T., Xie, R., Liu, J., & Liu, Y. (2019). A survey on the scalability of blockchain systems. *IEEE Network*, *33*(5), 166–173.
Yu, F. R., & He, Y. (2019). A service-oriented blockchain system with virtualization. *Trans. Blockchain Technol. Appl.*, *1*(1), 1–10.
Zaabar, B., Cheikhrouhou, O., Jamil, F., Ammi, M., & Abid, M. (2021). HealthBlock: A secure blockchain-based healthcare data management system. *Computer Networks*, *200*, Article 108500.

Faisal Jamil is currently serving as a Postdoctoral Researcher in the Department of IT Convergence Engineering, Gachon University, Korea. He has earned a Ph.D. degree in Computer Engineering and a Masters's degree in Computer Science from Jeju National University, Korea, and University of Engineering and Technology, Taxila, Pakistan, respectively. He did his BS in Computer Science from the Capital University of Science, Islamabad, Pakistan. He has been serving as an Editor of several reputed journals. His research work mainly focused on Internet of Things applications, blockchain, smart healthcare, cyber–physical systems and Intelligent systems.

Shabir Ahmad is currently serving as a Research Professor in the Department of IT Convergence Engineering, Gachon University, Korea. He has earned a Ph.D. degree in Computer Engineering and Masters's degree in Software Engineering from Jeju National University, Korea, and National University of Science and Technology, Islamabad, Pakistan, respectively. He did his Bachelor of Science in Computer Systems Engineering from the University of Engineering and Technology, Peshawar, Pakistan. He is a member of IEEE and has been serving as an Editor of several reputed journals. His research work mainly focused on Internet of Things applications, cyber–physical systems, and Intelligent systems.

Taeg Keun Whangbo received his M.S. degree from the City University of New York in 1988 in Computer Sciences and his Ph.D. degree from Stevens Institute of Technology in 1995 in Computer Sciences. Currently, he is a professor in the Department of Computer Science, Gachon University, Korea. Before he joined Gachon University, he was a software developer in Q-Systems which is located in New Jersey from 1988 to 1993. He also served Samsung Electronics as a researcher from 2005 to 2007. From 2006 to 2008, he was the president of the Association of Korea Cultural Technology. His research interests include Computer Graphics, HCI and VR/AR.

Ammar Muthanna received the B.Sc., M.Sc., and Ph.D. degrees from St. Petersburg State University of Telecommunications, Russia, in 2009, 2011, and 2016, respectively. In 2012 and 2013, he took part in the Erasmus Student Program with the Faculty of Electrical Engineering, University of Ljubljana. He is currently an Associate Professor with the Department of Telecommunication Networks, and also the Head of the Laboratory, St. Petersburg State University of Telecommunications. He has published more than 60 scientific articles. He acted as a reviewer for many international and high-ranked journals, he is editor in the editorial boards of several international scientific journals. His main areas of research include IoT, SDN and MEC.



DoHyeun Kim received the B.S. degree in electronics engineering from the Kyungpook National University, Korea, in 1988, and the M.S. and Ph.D. degrees in information telecommunication the Kyungpook National University, Korea, in 1990 and 2000, respectively. He joined the Agency of Defense Development (ADD), from Match 1990 to April 1995. Since 2004, he has been with the Jeju National University, Korea, where he is currently a Professor of Department of Computer Engineering. From 2008 to 2009, he has been at the Queensland University of Technology, Australia, as a visiting researcher. His research interests include sensor networks, M2M/IOT, energy optimization and prediction, intelligent service, and mobile computing.