Taylor & Francis
Taylor & Francis Group

Automatika
Journal for Control, Measurement, Electronics, Computing and Communications

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/taut20

Smarter and resilient smart contracts applications for smart cities environment using blockchain technology

M. Kavitha Margret & E. Golden Julie

To cite this article: M. Kavitha Margret & E. Golden Julie (2024) Smarter and resilient smart contracts applications for smart cities environment using blockchain technology, Automatika, 65:2, 572-583, DOI: 10.1080/00051144.2024.2307228

To link to this article: https://doi.org/10.1080/00051144.2024.2307228

AUTOMATIKA
2024, VOL. 65, NO. 2, 572–583
https://doi.org/10.1080/00051144.2024.2307228

Smarter and resilient smart contracts applications for smart cities environment using blockchain technology

M. Kavitha Margret[a] and E. Golden Julie[b]

[a]Department of Computer Science and Engineering, Sri Krishna College of Technology, Coimbatore, Tamil Nadu, India; [b]Department of Computer Science and Engineering, Anna University Regional Campus, Tirunelveli, India

ABSTRACT
Our daily activities revolve around various technologies, the smart city technologies and the services they offer have all influenced modern living conditions. The purpose of a smart city is to enhance people's quality of life and provide possibilities to address social and environmental issues. As a developing technology, blockchain is beneficial for enhancing smart city services including food tracking, supply-demand matching, the security of connected cars, and regulatory compliance. We propose a new framework for describing how blockchain technology is used in smart contracts to improve security, dependability, and many other positive outcomes in a smart city environment. We propose smarter and resilient smart contracts using blockchain technology to manage real estate information. We propose a framework with tamperproof functionality to store the data and retrieve the data. Smart contract mathematical computations like overhead rate, execution time, mean computational cost, standard deviation, throughput, and resource utilization are evaluated and the results are compared. This paper focuses on real-world rental file management scenarios to demonstrate the benefits of blockchain technology and how it is used to address the issues that currently exist in developing smart contracts and services while exchanging real land and other properties in a smart city environment.

ARTICLE HISTORY
Received 22 November 2023
Accepted 2 January 2024

KEYWORDS
Smart city; blockchain; smart contract; land registry system

1. Introduction

We all are now experiencing an atmosphere of pure technology in which everything around us has emerged as future application using Industry4.0 started to change and increasingly transform into a technological trend [1]. Urbanization, public administration and other sectors of life are based on technological environments with different characteristics and applications. Due to urbanization and modernization people are accepting new technologies that have gone through different stages in our life from using remote controls to control various smart home appliances to handle all events. Recent technologies like Internet of Things and Virtual Reality and the have gained thrust over the last decade and are now being added to blockchain technology to provide security over the network [2].

Smart Contracts [3] are used for verifying, digitally facilitating or negotiating legal contracts to avoid intermediates over a decentralized platform, smart contracts is a computerized protocol. Blockchain uses powerful smart contracts to negotiate different people over the decentralized platforms this technology can be used for more safe financial transactions. Advantages of smart contracts are immutable and distributed, with this feature no one able to modify the fixed contract and the contract is added to the public ledger, all the participating party are able to validate the contracts, participating party cannot claim later that the contract has not been activated and not verified.

A smart contract has the time restrictions that could impose deadlines on the contract and it has blockchain security coding. It maintains the data and permissions expressed in code which requires to occur in a specific order to result in the agreement of the terms specified therein. Smart contracts are embedded in blockchain [4], each smart contract on the blockchain has a unique specification as the smart contract in question uses basic criteria such as verifying that the transferred value actually exists in the sender's account. A shared agreement between two or more parties is known as a smart contract. Information is stored, processed, and outputs are written to the function which is predefined one. Smart contract uses constructor to enable the contract, constructor function is enabled to host a new contract, self-destructor function is also defined to destruct the smart contract, the user who own a contract can only

CONTACT M. Kavitha Margret kavithamargret@outlook.com Department of Computer Science and Engineering, Sri Krishna College of Technology, Coimbatore, Tamil Nadu, India

allowed to destruct the function. A Smart contract its similar to class consist of instance variable to refer state of the related fields through the events and operations to be performed. Methods in smart contracts will change the state variables, there are two different types of constant and writable states like the state functions are two types read only and write functions [5].

Several platforms provide exceptional features for creating smart contracts like contract code execution. These have allowed the creation of smart contracts through particular programming languages. The Bitcoin is used to lever transactions involving cryptocurrencies with a relatively limited capacity for computation. Bitcoin [6] platforms employ a bytecode scripting language that uses stacks. The capacity for Utilizing Bitcoin to design a smart contract with high complex logic, Major alterations would be finished to the mining processes with the incentive programmes to propose intelligent proper contracts on the blockchain. The open-source blockchain platform NXT [7] uses a proof-of-stake consensus mechanism. These platforms support templates for creating smart contracts and doesn't allow new contracts to deploy. The main objectives of the paper are

- ✠ The decentralized procedure is used to reduce the overheads and minimizing the expensive intermediaries through the proposed Blockchain-based land register system.
- ✠ The proposed framework has provided the distributed environment of land record which are stored in the network while the updated data has been connected with the Blockchain technology.
- ✠ The smarter and resilient smart contracts for land registry record in smart cities environment through the Blockchain technique has provided the solution for a critical problem of land registry.

2. Literature survey

The smart contracts which are subjected to the current traceability rules and processes which is intended for agro-food supply chains which has the blockchain technology for managing the F2F framework that has been utilized by the European Union [8]. The smart contracts with the multi-agent system to model Supply chain in blockchain which manages to track and coordinate the food supply chain model [9]. The study delves into the enablers of blockchain implementation, considering perspectives from technology, organization, supply chain, and the external environment. The smart containers [10] in a blockchain for efficient supply chain management, the proposed smart contracts to manage payment automation, recipient verification, verifying user and refund initiation for violations of predefined terms [11]. QuarkChain framework [12] has been utilized the blockchain-based interoperability model with reputation-enabled Proof-of Authority framework to address the issues related with the supply chain management. A blockchain-enabled traceability framework [13] has been enabled with highly clear, and tamper-proof traceability information as well as automatic regulatory compliance verification and adaptation for traceability situations involving imported products. CAIPY [14] is a smart contract-enabled bionetwork that enables considerable cost savings for insurance claims while maintaining the simplicity and transparency of current processes. DAPP [15] is a decentralized daily information sharing app based on smart contracts, users can register and be involvement without revealing of personal data. The Data trading approach [16] which uses smart contract provides solution using Machine Learning enabled Blockchain framework. The Mobile payment structure has been framed using smart contract based on sturdy certificate-less signatures [17]. Ethereum [18] is a public platform of blockchain which manages the users for deploying the smart contracts while the computational resource utilization of every execution has been evaluated through the Ether crypto currency. Hyperledger [19] a product of Linux foundation and it's an open source blockchain for managing the crypto currency. NEM [20] enables asset definition which maps realistic industrial instruments. It's an enterprise blockchain platform. R3 Corda [21] designed for the industrial integration, it has privacy features and it is a scalable blockchain platform. Stellar [22] is a platform for multi-currency payments that is effective in terms of how much processing power it uses. Waves [23] enables the users of bitcoin wallets to top up their balances in US dollars, a platform for dollar payments has been created. Fork of Ethereum called Ethereum Classic [24] allows for cheaper transactions. It is an enhanced form of Ethereum. Tezos [25] is a self-updating crypto ledger that uses two different sorts of accounts – implicit and originated – as well as native cryptocurrency. The originated accounts are linked to the smart contracts. NEO [26] is used for digitize the assets intended to create smart economy. Plutus is a new programming language used in Cardano [27] to develop new smart contract. EOS [28] was built primarily with the goal of creating decentralized apps that are both horizontally and vertically scalable. EOS has addressed a number of issues, such as governance, parallel processing, and enhanced usability. The decentralized data availability in a common position could be managed for providing the users to avail the specific services. The Blockchain data of the land registry system will make the workflow framework of the registration system as the verification purpose for the people, it will help the property chain right from the people. While implementing the selling and buying properties, the Blockchain enabled smart contracts provide the events having tracked in a sequential way and an online procedure has been followed [29].

3. Proposed work

The decentralized Blockchain-based land register system is proposed in this paper as the architecture enables the overall network security through the decentralization process to reduce the overheads and minimizing the expensive intermediaries. By assuring suitable authentication and validating transactions, the privacy of the stored information is guaranteed. Blockchain has the data immutability as it is very hard to modify data while it has been connected with the blockchain framework. The proposed framework suggested tamperproof functionally to store the data. The blockchain-based land registry methodology has away from the distributed databases of land records that are shared within the network. We propose Smarter and resilient smart contracts for land registry record in smart cities environment using blockchain technology, which is effectively solves a critical problem with the previous system, important three units of land registers are Objects, right and subject Objects is the property, right is associated with the property, subject is the person or property holder. When the property gets changes from one hand to another all-legal documents should be changes from one to another, legal records can always change from one to another and it can be tampered or altered at any time to avoid this we propose smart contract for land registration. In today's scenario of real estate, if a person wants to sell or buy a property he will contact the real estate agent, they will check with land registry records and confirms that the person is the real owner or not. Figure 1 illustrates classical land registration method used nowadays, we propose 3 smart contracts for land registration as follows, using blockchain buyer and seller get connected directly through blockchain platform, property owner can check and verify ownership whether they are eligible to sell property, the bank can check the status of the ownership over the blockchain. The process will remove the third party like brokers involvement, when the seller transforms the ownership to buyer payment automatically transforms from buyers account to sellers account, buyer, seller, bank can verify the contract over the block chain smart contract platform.

The smart contract for Land Registration has been designed with the following parameters:

1. "smart contracts": It holds contracts of varying levels of complexity, indicated by a number prefix in the filename.
2. "scripts": scripts used to deploy a contract.
3. "tests": file name called "Ballot" contains contract to test contract with unit tests in Solidity.

The "scripts" contains scripts for deploying the "Hold" contract. Any number of contracts can be added by passing Name, arguments in contract arguments, program can be implemented using web3.js or ethers.js libraries or any recent blockchain network, contracts can be developed through Solidity file, solidity files should be compiled first after that Script can to run by file explorer by right click on top of the file.

The keccak256 standards are followed to complete Registration task in the XDC network. Every transaction is referred by the unique number and this will be used completely different transactions, which has been illustrated in Figure 2.

Contract 1: Property:

The property contract has the values of Admin, Status, and Role to indicate that the land registration maintained the Approval and the buyer details as follows:

```
pragma solidity > = 0.7.0 < 0.9.0;
/// @title Property
/// @dev Store & retrieve value of a property
contract Property {
        address public Admin;
        enum Status {Not available, Awaiting,
          Approved, Excluded}
        enum Role {Visitor, owner, Admin, Buyer
}
```

Contract 2: Transfer property:

The transfer property has the list of completed transfer details as the property owner can deliver the sender message and the Access control is maintained while the updated completion list is developed through the access control. The Add address functionality with the help of access control has the total list of completed address of transfer property with authentication.

```
pragma solidity > = 0.7.0 < 0.9.0;
contract TRansPro
{
Address Public propertyOwner;
Uint public list_completedtransfer;
Public Constructor()
{
propertyOwner = sendermessage;
}
Modifier Accesscontrol()
{
If(sendermessage = = owner)_;
}
Function set(uint addcompleted)public Accesscontrol{
list_completedtransfer = addcompleted;
}
Function addaddress(address newone)public Access-
control
{TransPro addaddress = Transpro(new_address);
Add.address(Lizt_complatetransfer);
}
}
```

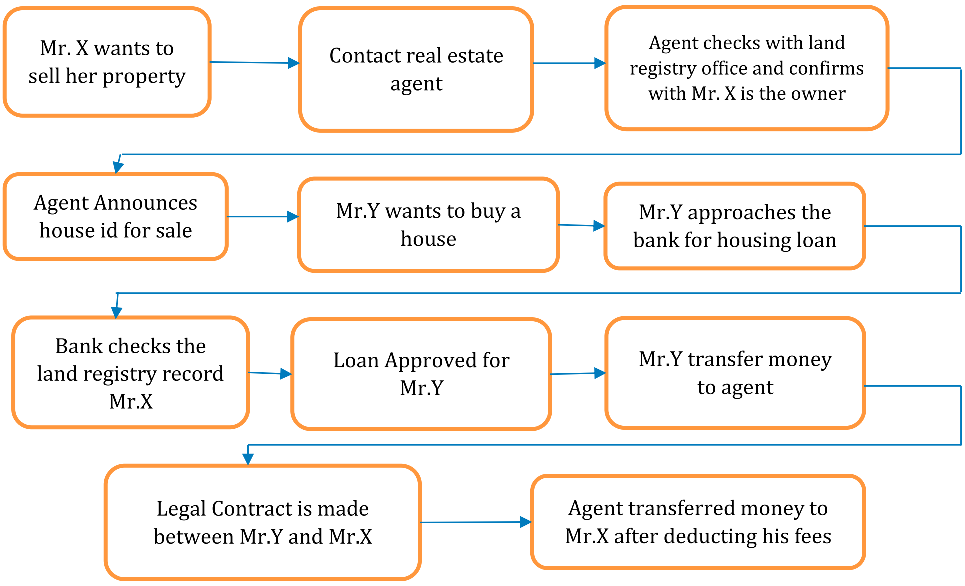

Figure 1. Land registration method.

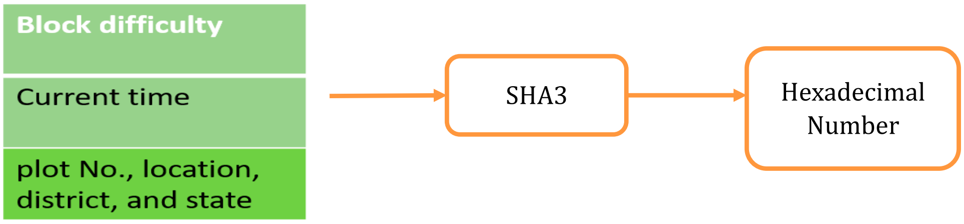

Figure 2. Unique number generation for node.

The method set () has the parameter "uint." "public" is a key that decides who can process this function. The variable "sendermsg" which is invoked when the transaction is initiated by the owner and stores the information about the address, amount of the person of ether XDC tokens. The modifier function is used to modify the values, changing state value of the transaction, view or pure function is used to view or not to change the state of the transaction property related details are stored in Structure here Struct is used to store the details.

```
struct LandDetail {
        Status Statenumber;
        uint256 propertyvalue;
        address Owner;
            }
```

The block node can be used to store PropertyId, ownerID, userrole, userid after user verification so that the named can be mapped to land details, below listed functionalities used to track the user details

// Stores all properties propertyId
(Mapping to Landdetils) public properties;
// Keeps track of property owner
(Mapping to address) public propertyOwnerChange;

// Keeps track of user and their privileges
Mapping to user Role) public userpriviledge;

// Keeps track of user verification (true / false)
(Mapping to Boolean value) public verifiedUsers;

Contract 3: Register Contract

Land registration contract is generated by the owner using DApp, the Land Registration smart contract handles this process very excellently, it demonstrates the pseudocode representation and functional execution of the register () function, which is used to register Lands in DAPP. It takes input arguments such as current block difficulty, current time, lot number, city, country, state, etc. The current block difficulty is calculated based on the time it takes to process the actual transaction, select or detach a block, or delete a block, it is an integer value. SHA3 algorithm taken all details as inputs, and it produce a spitted large size hexadecimal number. Entire process of generation of hexadecimal number is represented as the unique hexadecimal number is further used to register land and considered as unique id for registered land, it gives the details of pseudocode implementation of the computer Node() function to that implements the contract to add node on the blockchain.

Input: address of the owner (sender message pnumber(plotNo),l(location),district(district),
state(state),lmark(landmark),price(sellingprice),
AccessList(AL))

1. AL is the XDC address of all authorized user access list in the contract

2. If sender message eAL then
3. If pnumber(plotNo),l(location) Not exists
4. Register plotNo,sendermessage,l(location),district, state,landmark,sellingprice to the blockchain
5. Else
6. Revert contract and show error
7. End
8. Else
9. Revert contract and show error
10. end

Algorithm: RegisterPropertyToBlockchain

Input: – SenderMessage – PlotNo – Location – District – State – Landmark – SellingPrice – AccessList

1. If SenderMessage is in AccessList then
2. If PlotNo, Location combination does not exist in the blockchain then
3. Register the following information to the blockchain:
4. – PlotNo, SenderMessage, – LocationDistrict,State,Landmark, SellingPrice
5. Else
6. Revert contract and show an error (PlotNo, Location combination already exists)
7. End If
8. Else
9. Revert contract and show an error (SenderMessage not authorized)

The land registration contract has involved the authentication process as the node list in the specific address of the authorized nodes in the contract while the delivered details is in the list of nodes then calculates the hash value with keccak-256 and saves it into the blockchain otherwise the error message is displayed.

ComputeNode()

Input: siteNumber(Pno),location(l), Listofnode(AL),state(s),district(d)

1. AL in the XDC address of authorized nodes in the contract
2. If (sendermessage ∈ AL) then
3. Compute hashvalue using Keccak-256 and store it to the blockchain;
4. else
5. Revert contract state and display error message
6. End

The above contract details the process of computing and adding valid node on the block chainSmart contracts reside in XDC blockchain, after compilation Solidity generates bytecode that will be executed by EVM. “Gas-Cost.”Costs required to perform mathematical calculations for smart contracts, each smart contract is processed with.sol.file extension, Atom, VSCode are Editors that have the high level language. The bytecode and ABI are two objects were generated after the compilation of smart contracts, the smart contract is deployed to the XDC network, this address will consider as the smart contract address. ABI is vital meanwhile it is very difficult to understand the functionality of a smart contract based on bytecode alone. Each account can communicate with the organized smart contract requires this address. The specific account which needs to process a smart contract involves the ABI to hash function and generate EVM bytecode for it. The Sequence diagram for the proposed methodology is demonstrated in Figure 3.

The proposed model is best suitable to buy or sell land details with land registry DApp, it is an alternate to the current land register method. The landowner enters the specifics of the property and the asking price for it in the DApp. The registration process is only validated by the official identified as the validator. Although a government official is involved in the registration process, the complete procedure is tamper-proof and boosts system trust by offering demonstrable audit trails.

```
// Modifier for property owner access
    modifier Owner(uint256 _propId) {
    require(properties[_propId].currOwner  = =
      sendermessage);
    _;
  } //modifier for verified user access
  // a specific property
  modifier User(address _user) {
    require(Users[_user]);
    _;
  } // Modifier for validator access
  modifier validator() {
    require(
      userRoles[sendermessage] > = Role.validator
&& Users[sendermessage]
    ); _;
  } // Contract initialization
  constructor() public {
    creatorvalidator =  sender message;
    userRoles[creatorvalidator] = Role.SuperAdmin;
    Users[creatorAdmin] = true;
  } /// Function to create property
  /// @parameter property id
  /// @parameter Property Pricevalue
  /// @parameter Ownwer address
  function createProperty(
    get propertyid, Pricevalue, address _owneraddress
  )
  } /// Approve property
  /// @parameter _propertyId Identifier for property
  function approveProperty(uint256 _propId)
     external
    verifiedSupervalidator returns boolen value true /
false
    returns (bool)
  { the status is updated from the result receives from
validator
```

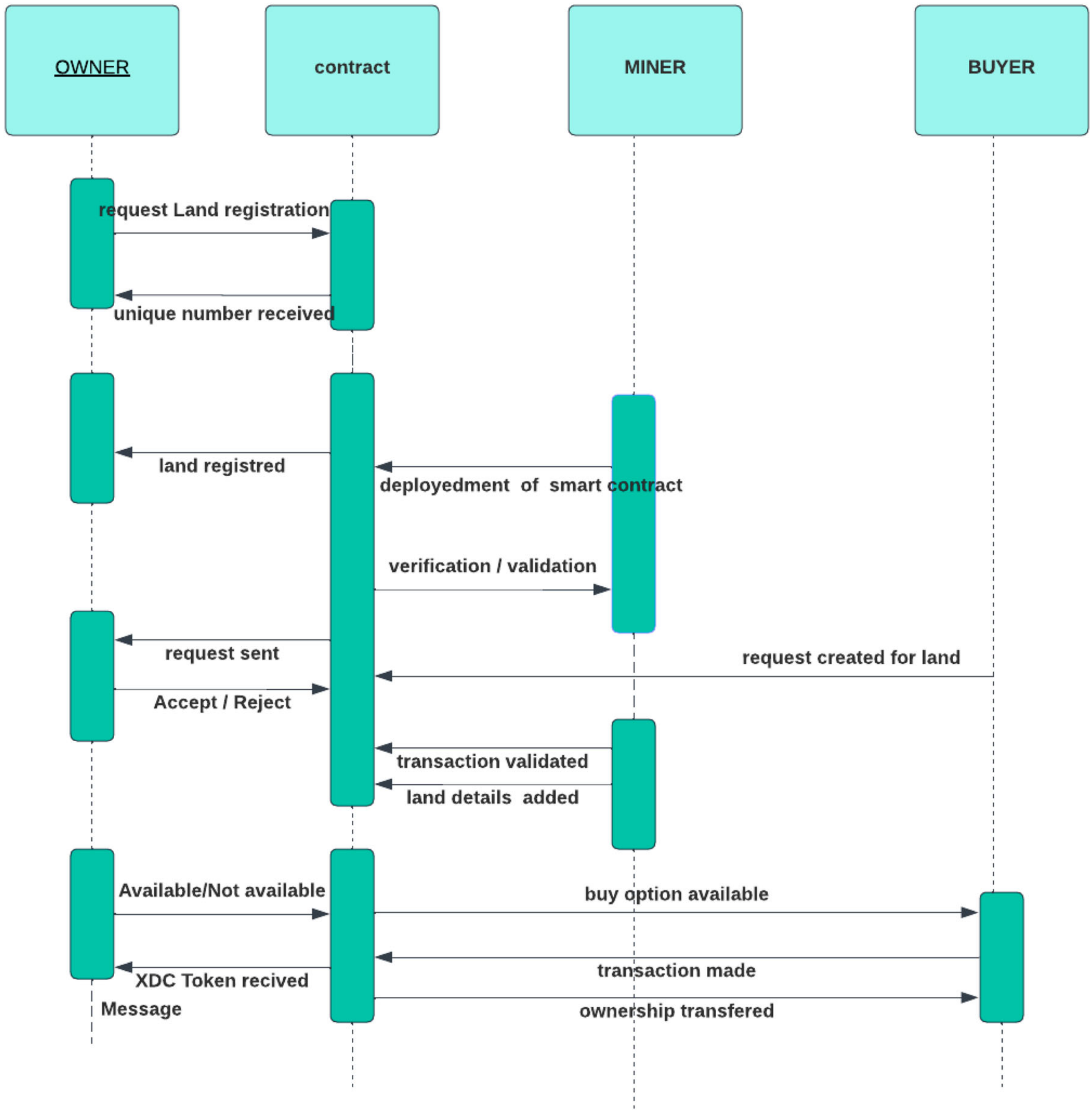

Figure 3. Sequence diagram for land registration.

```
  }
/// Function to reject property
  /// @param _propertyId Identifier for property from
validator
Get the property id verification status from
    verifiedSuperAdmin
    returns (input status as bool)
  { require(properties[_propId].currOwner! = sen-
dermessage);
    properties[_propertyId].status = Status.Rejected;
    return true;
  }
  /// Function to change property ownership
  /// @param _propertyId Identifier for property
  /// @param _newOwner new Owner address for
property
This function receives two input arguments are proper-
tyid, new ownerdetails
  function changeOwnership(property Id, newOwner)
  external
  onlyOwner(_propertyId)
  verifiedUser(_newOwner)
  returns (bool)
{
  require(currOwner! = _newOwner);
  require(OwnerChange[_propertyId] = =
    address(0));
  OwnerChange[_propertyId] = _newOwner;
  return true;
}
/// Function to approve change of ownership
/// @param _propertyId Identifier for property
This function takes input argument property id
function approveChangeOwnership(propertyId)
  external
  verified validator returns (bool)
{
  require(OwnerChange[_propertyId]! =
```

Table 1. Smart contract mathematical computations.

S.NO	Value	Operation	Gas used	Level of usage	Removed from stack	Added to stack	Usage of method
1	0XX0000	STOP()	0	NIL	0	0	Halt the program
2	0XX0001	ADD()	3	VERY LOW	3	1	Addition
3	0XX0002	SUB()	3	VERYLOW	3	1	Subtraction
4	0XX0003	MUL()	4	LOW	3	1	multiplication
5	0XX0004	DIV()	5	LOW	3	1	division
6	0XX0005	MOD()	5	LOW	3	1	Modulus operation

```
address());
properties[_property.currOwner = propOwner
  Change[propertyID];
//current owner will be removed the ownership
changes to new owner once he verified by validator
propOwnerChange[propertyId] = address();
return true;
}
/// Function to add a new user
/// @param _newUser new user address
function addNewUser(address _newUser) external
verifiedAdmin
returns (bool)
{
User role.update [newUser] == Role.Visitor;
verifiedUsers[_newUser] == false);
user Role updated [_newUser] = newUser;
return true;
}
/// Function to add a new admin
/// @parameter new user address
function addNewAdmin(address of newAdmin)
verified by SuperAdmin
returns (bool)
{
Userrole.update[_newAdmin] == Role.Visitor;
verifiedUse[_newAdmin] == false);
userRoleUpdated[_newAdmin] = newAdmin;
return true;
}
/// Function to add a new admin
/// @param _newSuperAdmin new super admin user
address
function addNewSuperAdmin(SuperAdminAddress)
external
after successful execution of consensus mechanism
new super admin is elected
returns (bool)
{
Add newSuperAdmin == Role.Visitor;
require(verifiedUsers[_newSuperAdmin] =
  false);
userRoles[_newSuperAdmin] = Role.SuperAdmin;
return true;
}
```

the smart contract with above functionality has been set up such that it can carry out all of the functions of the network's nodes without causing any problems. The client app is developed for delivering and capturing data from the smart contract as the third-party framework is utilized for delivering the request while the server node processes the client requests through the database which contains the delegation policy, the blockchain authorizes the users for broadcasting the applications. For producing the authentication, the blockchain checks the user requests through the delegation policy which have been saved into the blockchain framework.

4. Results and discussion

Design and implementation of entire network is done by smart contracts in the XDCtest Ethereum network, Node.js modules were used to deploy smart contract on XDC network, the same has been used for Compilingthe main file in node.js which further produced byte code and ABI. web3 library is used to interact with contracts in blockchain as everyfilehas to be compiled properly to utilize the class functionality. XDC test framework has utilizing web3 library and D'CENT wallet which is XDC client on the system were used to enterprise and deploy of smart contract in the proposed model. The web3 library are used to communicate to the XDC network from an application. Noxt.js is used forserver-siderouting while the server-side and the client-side render by JavaScript. The styling of the website is done with Semantic-ui-React. The "pages" and "components" folders in ReactJS contain all the files needed to create a website. The component files found in the components folder are needed by the pages folder's files, for server and routing process needs the files server.js and routes.js. Smart contracts are written using the Solidity programming which is translated into bytecode for further execution."Gas-Cost" is computed for each and every functionality in the execution of smart contract mathematical computations and the specific result is illustrated in Table 1.

Comparison of Existing model with Proposed model is demonstrated in Table 2 as the parameter considered for comparison has the required parameters of the Security, Scalability, decentralization, intermediary cost, ownership transfer and accountability.

The owner of smart contract should agree the gas limit while the validators can agree to process it within the smart contract. Unique address is generated for every smart contract for delivering the XDC tokens.

Table 2. Comparison with required parameters.

Feature	CAIPY	DAPP	Proposed
Security	Low	Low	High
Scalability	Low	High	High
Gascost	High	High	Low
Decentralization	No	No	Yes

Every person is identified uniquely by referring the address of smart contract call made by the processor and execute the authenticated access to contract. The proposed technique has the interaction with the smart contract while the system delivers the transaction with the evaluated value. The throughput and RTT within the several nodes in the system as the link nodes have the link within the sender and the receiver nodes. The higher throughput within the nodes is nearer to 7.0 Mbps, the smallest RTT is about 0.50 ms and the comparison result is demonstrated in Figure 4.

The comparison for RTT for the proposed technique within the value of 0–1 ms for the links 1–10 is demonstrated in Figure 5 as it holds the highest values for Link 1 and Link 10.

The proposed framework has the topology and dynamic routing of the entire network, the issue within the relay node of the sender to forward the data. The sender requires to identify the connection point into the adjacent node for managing the latest node for connection. The recovery time is computed in Equation (1).

$$Reco_T = Lost_T + AN_T + BC_T \quad (1)$$

where $Reco_T$ is the recovery time, $Lost_T$ is the lost time, AN_T denotes the time for communicating the adjacent node, BC_T demonstrates the time for the Blockchain. The experimental results proved that the blockchain

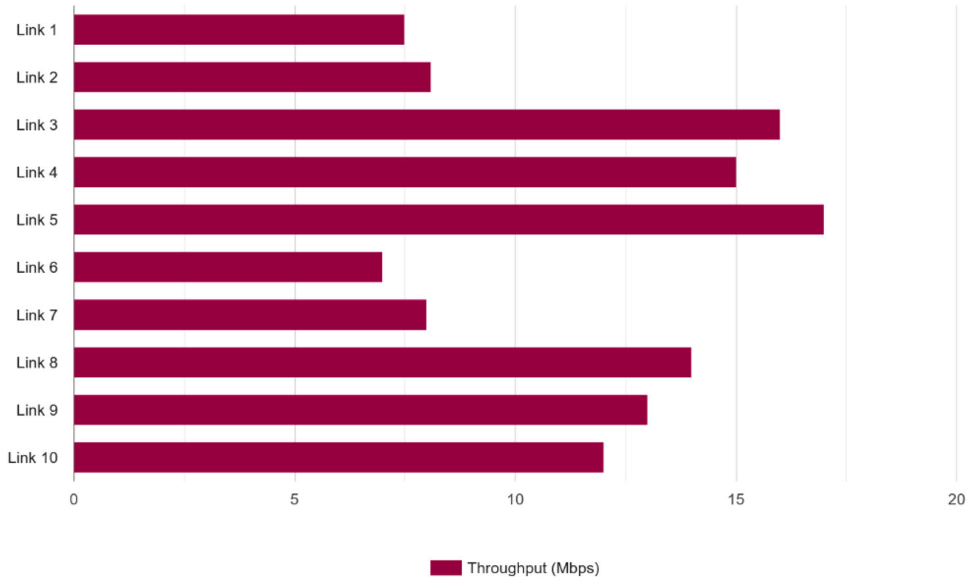

Figure 4. Throughput.

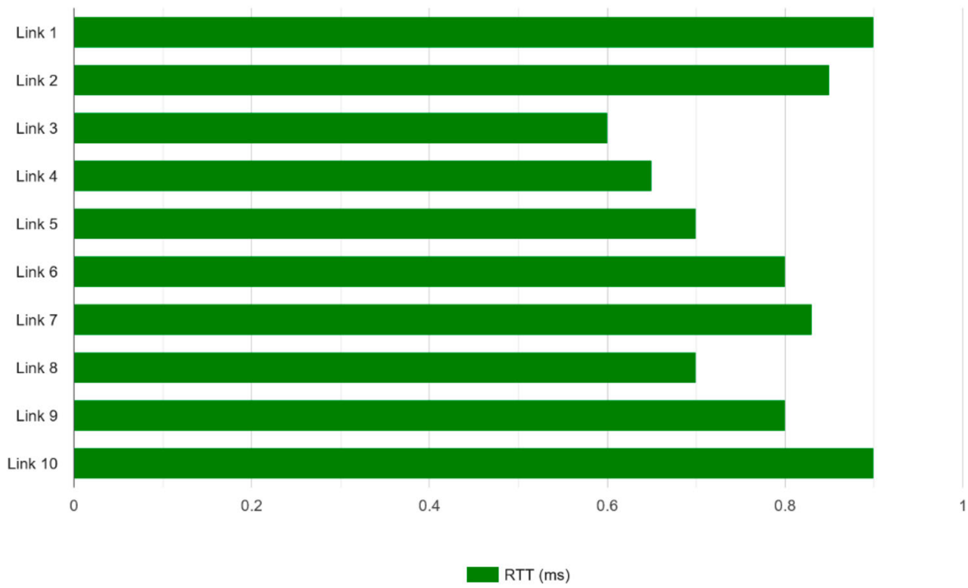

Figure 5. RTT.

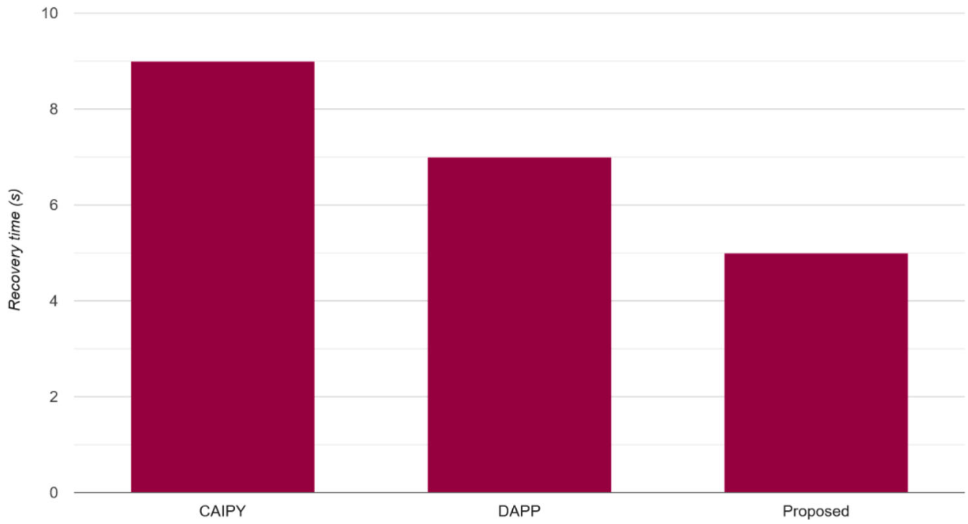

Figure 6. Recovery time.

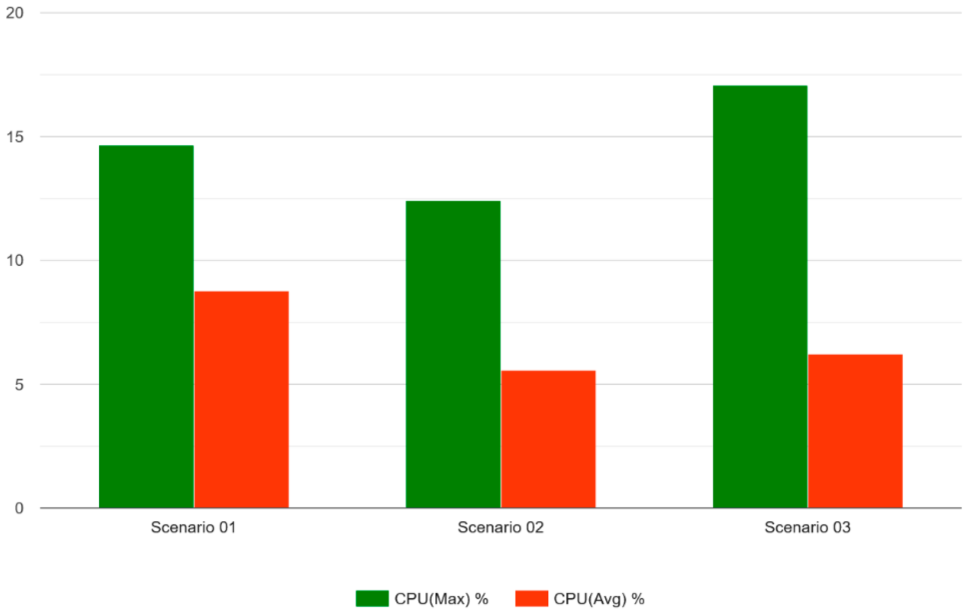

Figure 7. Resource utilization.

connection has the sender which broadcasts the packets to its adjacent node. The Recovery time comparison for the proposed technique with the existing techniques is illustrated in Figure 6.

The resource utilization of the proposed technique has the classification based on the CPU and memory which is performed in different level of iterations. The mean utilization of the resources is computed through the system in terms of performances of the CPU and memory and it is illustrated in Figure 7.

The proposed technique has various security related functionalities which is utilized to produce the block that makes 0.89 ms. The proposed technique has managed the cryptographic Blockchain transactions that utilizes the smart contracts with the security and it can be analysed through the standard deviation of 0.04 ms which is demonstrated in Figure 8, the standard deviation comparison for the proposed technique with the relevant techniques is illustrated in Figure 9.

The proposed methodology has been measured the execution time of every stage according to the current situation. In the performance evaluation, the virtual clients parallelly delivered the access request into the main smart contract while the total clients are in constant. Additionally, the smart contracts have the delegation policy to the specific resources and save it into the Blockchain, where it enhances the total delegation policy through the evaluation procedure of every stage into the restricted scalability and the mean authentication period which has some values while the policy revocation period is determined through the smart contracts and it is illustrated in Figure 10.

The overhead rate for the proposed methodology has the subsequent amount of client requests and the delegation policy. The experimental results proved that the overhead rate is minimum for the proposed technique compared to the relevant techniques. Hence, the proposed technique has produced the better scalability in

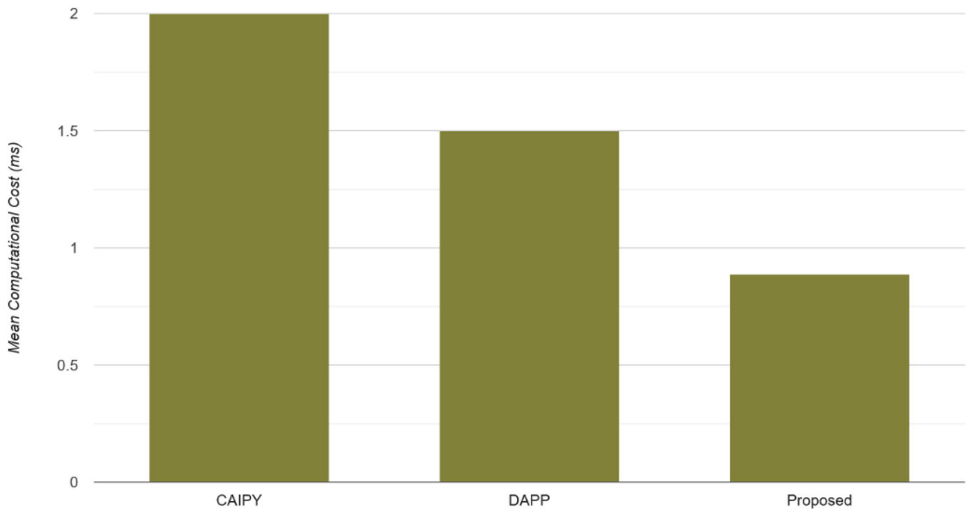

Figure 8. Mean Computational Cost.

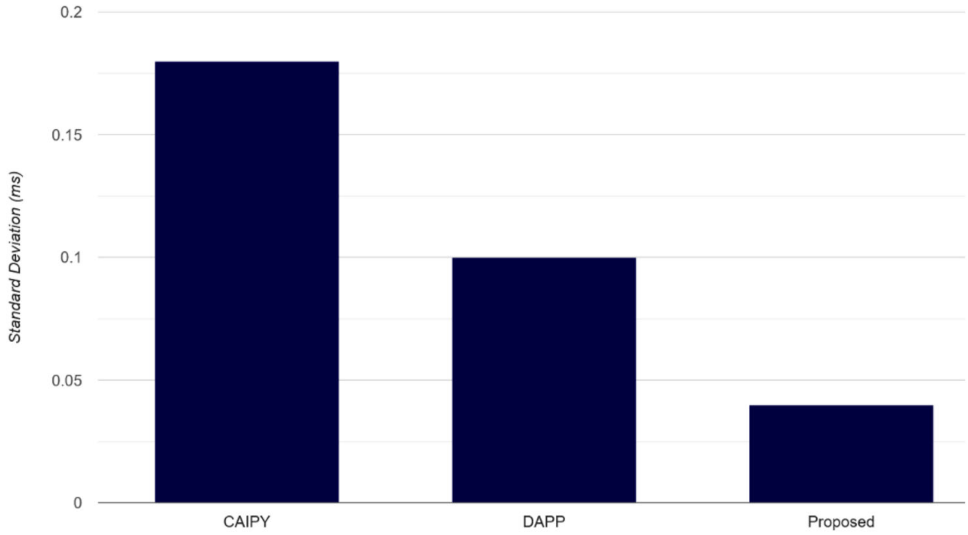

Figure 9. Standard Deviation.

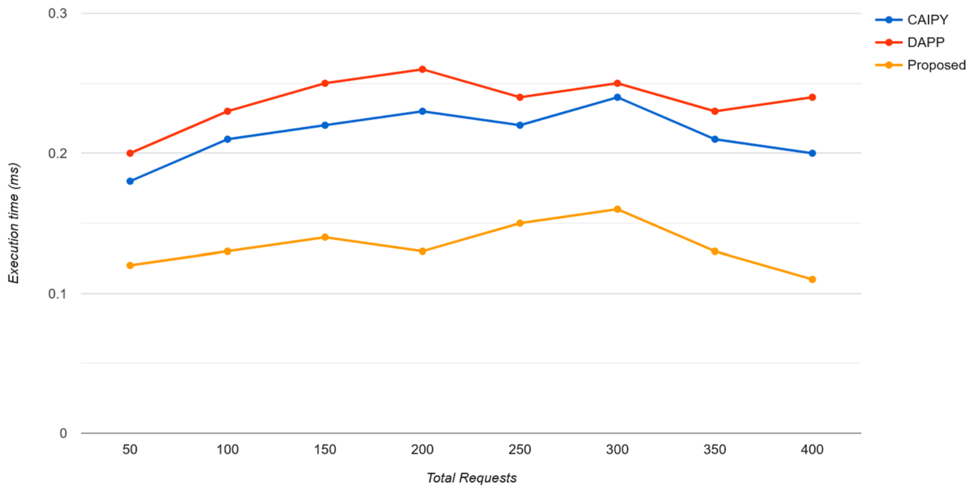

Figure 10. Execution time.

terms of delegation policy and the client requests, the overhead rate is shown in Figure 11.

The total cost of the proposed method is $O(|X| + |Y|)$, where $|X|\&|Y|$ are the parameter values for cost computation.

5. Conclusion

The predominant purpose of our proposed work is to analyse the blockchain platform and the way it is able to be implemented to resolve the troubles with

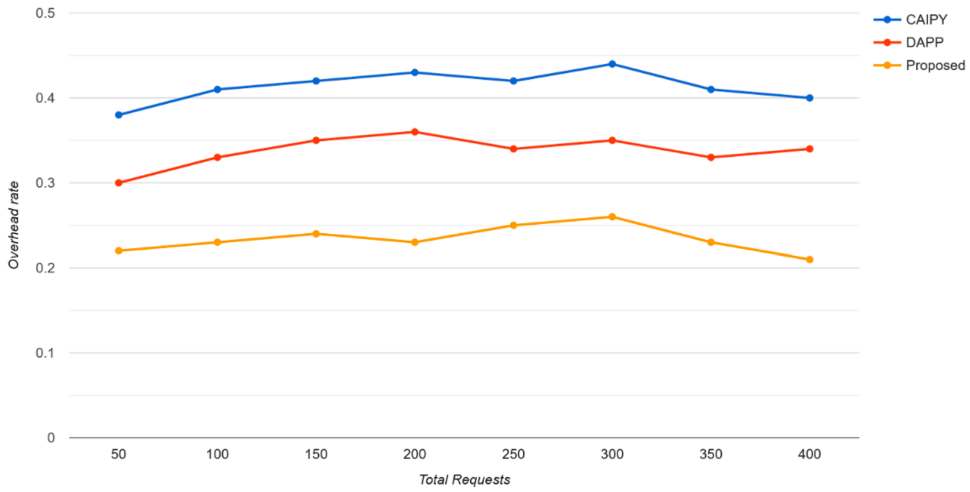

Figure 11. Overhead rate.

inside the current land registry structure. This research work provides a brief overview about the blockchain and its usage for land registration and the solution have been given clearly. Further, itsemphases on the secure land registry system based on blockchain-based model and emulates on the XDC platform for the design and development of DApp. The paper then effectively addresses the challenge of integrating the XDC token into the XDC platform and deliberates the whole process of method creation, implementation, and interaction. These blockchain implementations can be used in a variety of industries like healthcare, supply chain management, trading, e-commerce, and others, to address current problems and find a technical solution to security vulnerabilities in the current blockchain model. The research focuses on security, trustworthiness and the privacy of the network is ensured and results were discussed. The future enhancement of the blockchain framework is to provide the complete opportunities in many research aspects to observe and incorporate with Artificial Intelligence which may resolve the most common troubles in economic system and trade in extra powerful manner.

Disclosure statement

No potential conflict of interest was reported by the author(s).

References

[1] Javaid M, Haleem A, Singh RP, et al. Blockchain technology applications for industry 4.0: a literature-based review. Blockchain: Res Appl. 2021;2(4):100027. doi:10.1016/j.bcra.2021.100027
[2] Adhikari N, Ramkumar M. Iot and blockchain integration: applications, opportunities, and challenges. Network. 2023;3(1):115–141. doi:10.3390/network3010006
[3] Wang S, Yuan Y, Wang X, et al. An overview of smart contract: architecture, applications, and future trends. 2018 Ieee Intelligent Vehicles Symposium (IV); 2018; Changshu, People's Republic of China, p. 108–113.
[4] Taherdoost H. Smart contracts in blockchain technology: a critical review. Information. 2023;14(2):117. doi:10.3390/info14020117
[5] Zheng Z, Xie S, Dai H-N, et al. An overview on smart contracts: challenges, advances and platforms. Future Gener Comput Syst. 2020;105:475–491. doi:10.1016/j.future.2019.12.019
[6] Yu C, Yang W, Xie F, et al. Technology and security analysis of cryptocurrency based on blockchain. Complexity. 2022;2022: article id 5835457. doi:10.1155/2022/5835457
[7] Pervez H, Muneeb M, Irfan MU, et al. A comparative analysis of DAG-based blockchain architectures. 2018 12th International Conference on Open Source Systems and Technologies (ICOSST); 2018; Lahore, Pakistan, p. 27–34
[8] Tandon A, Dhir A, Najmul Islam AKM, et al. Blockchain in healthcare: a systematic literature review, synthesizing framework and future research agenda. Comput Ind. 2020;122:103290. doi:10.1016/j.compind.2020.103290
[9] Agi MAN, Jha AK. Blockchain technology in the supply chain: an integrated theoretical perspective of organizational adoption. Int J Prod Econ. 2022;247:108458. doi:10.1016/j.ijpe.2022.108458
[10] Shinde D, Padekar S, Raut S, et al. Land registry using blockchain – a survey of existing systems and proposing a feasible solution. 2019 5th International Conference on Computing, Communication, Control And Automation (ICCUBEA); 2019; Pune, India, pp. 1–6.
[11] Manzoor R, Sahay BS, Singh SK. Blockchain technology in supply chain management: an organizational theoretic overview and research agenda. Ann Oper Res. 2022. doi:10.1007/s10479-022-05069-5
[12] Aldakheel JS, AlAhmad MA, Al-Foudery A. Comparison between bitcoin and quarkchain. J Comput Theor Nanosci. 2019;16(3):818–822. doi:10.1166/jctn.2019.7960
[13] Kumar Agrawal T, Kumar V, Pal R, et al. Blockchain-based framework for supply chain traceability: a case example of textile and clothing industry. Comput Ind Eng. 2021;154:107130. doi:10.1016/j.cie.2021.107130
[14] Bader L, Bürger JC, Matzutt R, et al. Smart contract-based car insurance policies. 2018 IEEE Globecom Workshops (GC Wkshps); 2018; Abu Dhabi, United Arab Emirates, pp. 1–7.

[15] Mishra RA, Kalla A, Singh NA, et al. Implementation and analysis of blockchain based DApp for secure sharing of students' credentials. 2020 IEEE 17th Annual Consumer Communications & Networking Conference (CCNC); 2020; Las Vegas, NV, USA, pp. 1-2.
[16] Chen C, Wu J, Lin H, et al. A secure and efficient blockchain-based data trading approach for internet of vehicles. IEEE Trans Vehicular Technol. 2019;68(9):9110–9121. doi:10.1109/TVT.2019.2927533
[17] Obaid M, Aqel M, Obaid M. Mobile payment using blockchain security. J Appl Sci Eng. 2021;24(4):687–692.
[18] Alrubei S, Rigelsford J, Willis C, et al. Ethereum blockchain for securing the internet of things: practical implementation and performance evaluation. 2019 International Conference on Cyber Security and Protection of Digital Services (Cyber Security); 2019; Oxford, UK, p. 1–5.
[19] Honar Pajooh H, Rashid M, Alam F Hyperledger fabric blockchain for securing the edge internet of things. Sensors. 2021;21(2):359–388. doi:10.3390/s21020359
[20] Subramanian G, Thampy AS, Ugwuoke NV, et al. Crypto pharmacy – digital medicine: a mobile application integrated With hybrid blockchain to tackle the issues in pharma supply chain. IEEE Open J Comput Soc. 2021;2:26–37. doi:10.1109/OJCS.2021.3049330
[21] Mohanty D. R3 corda for architects and developers: with case studies in finance, insurance, healthcare, travel, telecom, and agriculture. Springer; June 2019, pp. 1–212. doi:10.1007/978-1-4842-4529-3
[22] Khan N, Ahmad T, State R. Feasibility of Stellar as a blockchain-based micropayment system, The 2nd International Conference on Smart BlockChain, SmartBlock 2019.
[23] Nærland K, Mueller-Bloch C, Beck R, et al. Blockchain to rule the waves – nascent design principles for reducing risk and uncertainty in decentralized environments. International Conference on Information Systems (ICIS); December 2017.
[24] https://www.kaggle.com/datasets/bigquery/crypto-ethereum-classic.
[25] Goodman L. "Tezos—a self-amending crypto-ledger white paper," URL:https://www.tezos.com/static/papers/white_paper.pdf, 2014.
[26] Elrom E. Neo blockchain and smart contracts. In: The blockchain developer: a practical guide for designing, implementing, publishing, testing, and securing distributed blockchain-based projects. Berkeley, CA: Apress; 2019. p. 257–298. doi:10.1007/978-1-4842-4847-8_7
[27] Kondratiuk D, Seijas PL, Nemish A, et al. Standardized crypto-loans on the cardano blockchain. In: Bernhard M, editor. International conference on financial cryptography and data security. Berlin, Heidelberg: Springer Berlin Heidelberg; 2021. p. 579–594.
[28] Song W, Zhang W, Zhai L, et al. EOS.IO blockchain data analysis. J Supercomput. 2022;78:5974–6005. doi:10.1007/s11227-021-04090-y
[29] Alharby M, Aldweesh A, Moorsel Av. Blockchain-based smart contracts: a systematic mapping study of academic research (2018). 2018 International Conference on Cloud Computing, Big Data and Blockchain (ICCBB); 2018; Fuzhou, People's Republic of China, p. 1–6.