


ORIGINAL RESEARCH

A remote and cost-optimized voting system using blockchain and smart contract

Mohammad Nabiluzzaman Nelay¹ | Md. Abdul Wahab¹ | Sheikh Wasif¹ |
 Abdulla All Noman¹ | Mustafizur Rahaman¹ | Tahmid Hasan Pranto¹ |
 A. K. M. Bahalul Haque² | Rashedur M. Rahman¹ 

¹Department of Electrical and Computer Engineering, North South University, Dhaka, Bangladesh

²Department of Software Engineering, LUT University, Lappeenranta, Finland

Correspondence

Rashedur M. Rahman, Department of Electrical and Computer Engineering, North South University, Dhaka, Bangladesh.
 Email: rashedur.rahman@northsouth.edu

Funding information

North South University, Faculty Research Grant, Grant/Award Number: CTRG-21-SEPS-19

Traditional voting procedures are non-remote, time-consuming, and less secure. While the voter believes their vote was submitted successfully, the authority does not provide evidence that the vote was counted and tallied. In most cases, the anonymity of a voter is also not sure, as the voter's details are included in the ballot papers. Many voters consider this voting system untrustworthy and manipulative, discouraging them from voting, and consequently, an election loses a significant number of participants. Although the inclusion of electronic voting systems (EVS) has increased efficiency; however, it has raised concerns over security, legitimacy, and transparency. To mitigate these problems, blockchain technology has been leveraged and smart contract facilities with a combination of artificial intelligence (AI) to propose a remote voting system that makes the overall voting procedure transparent, semi-decentralized, and secure. In addition, a system that aids in boosting the number of turnouts in an election through an incentivization policy for the voters have also developed. Through the proposed virtual campaigning feature, the authority can generate a decent amount of revenue, which downsizes the overall cost of an election. To reduce the associated cost of transactions using smart contracts, this system implements a hybrid storage system where only a few cardinal data are stored in the blockchain network.

1 | INTRODUCTION

The election system plays a crucial role in democracy, where general citizens can express their opinion by electing their leaders and their future way of living. Despite being evaluated over the years, today's voting systems restrain from improvement regarding voter turnout [1]. One of the probable reasons behind the low turnout could be the lack of a convenient and user-friendly remote voting system. Nowadays two most practiced voting systems are the traditional ballot system (submitting a ballot paper to the poll station) and digital voting through electronic voting system (EVS) [2]. In the USA (in some states), along with these two voting systems, "postal voting" (voting through the mail) is also accepted. During the vote-counting session, all the paper ballots are counted individually through a ballot-counting machine and later combined with the total

votes from the EVS, which is very time-consuming and cost-inefficient. Moreover, there is no assurance that every submitted ballot paper vote is counted and tallied. Additionally, during the submission of paper ballots or electronic ballots through EVS, the voter's details are also included with the ballot, violating one of the fundamental contexts of voter anonymity [3–6]. One of the most critical concerns in a voting system is to ensure security. Standard paper ballots can be manipulated within the related organization, or they might not be counted or tallied during the counting session due to some malfunction in the procedure. Electronic voting systems or EVSs are claimed to be secure to a great extent; nevertheless, these machines are entirely controlled by a central authority that has the power to monitor the voting procedures and also has allegations to be influenced by political parties to support their victory [7]. Moreover, EVSs can be hacked, and the votes can be tampered with as they are

This is an open access article under the terms of the [Creative Commons Attribution-NonCommercial-NoDerivs](https://creativecommons.org/licenses/by-nc-nd/4.0/) License, which permits use and distribution in any medium, provided the original work is properly cited, the use is non-commercial and no modifications or adaptations are made.

© 2023 The Authors. *IET Blockchain* published by John Wiley & Sons Ltd on behalf of The Institution of Engineering and Technology.

hardware devices [8]. The costs of a regular election depend on staff members, security forces, resources like ballots and ballot boxes, and polling stations [9] which are immensely high. Therefore, a remote-accessible, transparent, reliable, tamper-proof, and cost-effective voting system is essential, and this could also significantly impact voter turnouts.

Many recent solutions based on online voting systems have leveraged blockchain technology to develop a remote, reliable, transparent, and trustworthy voting platform [2, 3, 9–11], where smart contracts simplify the development of decentralized applications. Farooq et al. [9] proposed a blockchain and smart-contract-based remote voting system to limit the number of votes that the voters can cast on a poll using a modifier named the “Voting Coin,” which acts like the ERC20 (Ethereum Request for Comment) token in the Ethereum blockchain. Vairam et al. [10] developed a voting system on the local network using the Ganache private blockchain, which was mainly focused on generating transparent votes and producing a tamper-proof environment for voters. Another voting system solution has been proposed by Fernandes et al. [2] that focuses on several aspects like eligibility and privacy of voters, transparency of the system, and the immutability of votes. Apart from the actual proposed systems, Jafar et al. [3] analysed the current voting systems and the benefits of using blockchain in a voting system. They [3] mentioned some of the possible requirements that a remote voting system must have, including eligibility of voters, abstaining from the reusability of a vote, the privacy of voters, and completeness of ballot count. Most of the works mentioned above [2, 9–11] have proposed decentralized applications using smart contracts and blockchain. However, they did not focus on some issues [2, 9–11], like the high cost of a heavy smart contract and the requisite gas (Ethereum) fee of the voters. Although EVSs does not require voters to pay any fees to submit their vote unless it is a postal-based remote ballot or absentee ballot, according to the United States election system, the abatement issue in turnouts is still under concern [3]. Moreover, the cost of the elections can be reduced to some extent by a remote system [9] where the authority can minimize the operational cost by generating some revenue out of the overall process, but no such solution has been observed. Most of the proposed implementations [3, 9–12] have been focused on a single smart contract which is hardly implementable in the real-world scenario because the larger a smart contract gets, the higher the transaction fees reach. Another limitation we noticed is a reusable single smart contract for multiple polls. If every poll requires a different smart contract, the entire deployment cost will be escalated, which we attempt to reduce in our proposed system. In general, many of the earlier works have focused on making a system secure enough to allow voters to vote only once; however, none have tried to make the system a less costly and more efficient one [3, 9–12].

In this paper, we have proposed a remote, secure, and transparent voting system using blockchain technology. In addition, a reusable smart contract mechanism has been utilized to reduce overall contract deployment costs. Furthermore, AI technology has been applied to authenticate all the system’s actors. We also introduced an incentivization policy for the voters to increase

voter turnout. Overall, the contribution of the research can be summarized as follows:

1. A remotely accessible, secured, a privacy-preserved voting platform using blockchain and smart contracts using web 3.0.
2. A multilayer verification system to authenticate the voter identity based on artificial intelligence.
3. A tamper-proof, transparent, and automated vote-counting mechanism without any human interruption.
4. A stimulus incentive mechanism to step up the election turnouts.
5. Optimization of smart contracts with only crucial attributes and functions to reduce the overall cost of smart contract deployment.
6. A generalized platform for all kinds of election activity, for example, campaigns and promotions to downsize the election budgets.

The rest of the paper is structured as follows: The background and the related work of the utilities used in the voting system are shown in Section 2, and the proposed system is in Section 3. Algorithms and other program-based details are provided in Section 4. Section 5 focuses on a detailed preview of our proposed solutions, followed by their implications. At last, a conclusion and some future works to be done are included in Sections 6.

2 | BACKGROUND

2.1 | Blockchain and ethereum

Starting from the root of Bitcoin in 2008, which was created by Satoshi Nakamoto [13] concerning the double spending in the digital payment system, blockchain has been a solution since then to be used as a secured form of transactions on the internet. Today, blockchain is mainly acknowledged for different cryptocurrencies like Bitcoin, Ethereum, Hyperledger Fabric and Solana etc. [14, 15]. However, the blockchain is not just a cryptocurrency, it is a distributed ledger that holds growing records, and each of these records is known as a block [16]. The first ever block of a blockchain is known as the genesis [17], and each block is connected to its previous block and contains some data of its own and the hash of the previous block to which it is connected [13, 18]. The hash of a specific block is generated by combining its own data and the hash of the previous data. A cryptographic hash has an irreversible property which means that the data can be turned into a cryptographic hash using a cryptographic algorithm, but there is no way a hash can be reversed into its original form of data [19]. Most importantly, a single change in data will change the generated hash completely. So, if the data of a particular block is changed, then the hash of all the forward blocks will require their hash to be changed, which is a very daunting task. Moreover, all the records of transactions or data are stored over every node connected to the network, which is called a peer-to-peer network [13]. The

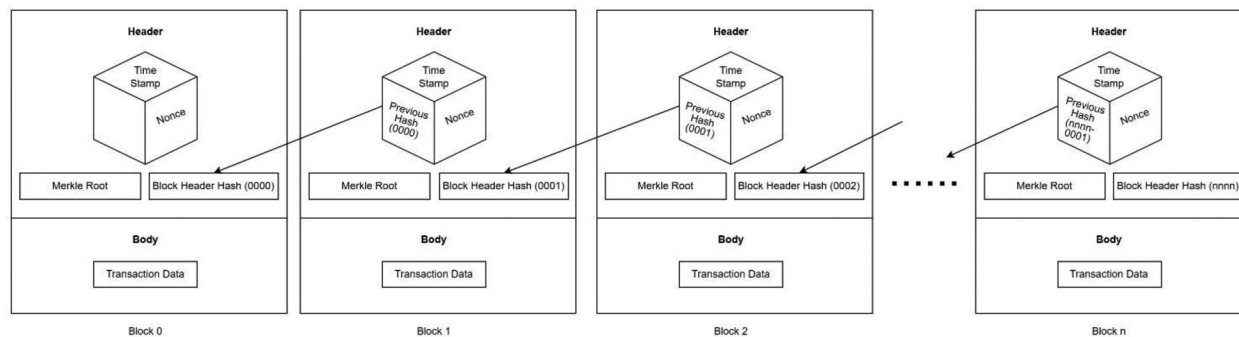


FIGURE 1 Blockchain architecture

data within a block in a blockchain is accessible by any participating node on the network, making the data more transparent and secure [17, 20]. On top of that, different distributed consensus mechanisms like proof of work (POW), proof of stake (POS), delegated proof of stake (DPOS), and proof of authority (POA) ensure that the maximum number of nodes on the network have approved the validity of a new transaction [9]. Figure 1 shows the architectural diagram of a blockchain.

Blockchain has a concept of public keys and private keys, which determine the completion of a transaction [21, 22]. Moreover, blockchain also follows a concept of anonymity and based on that context, anyone who wishes to make a transaction using cryptocurrencies is required to generate a random keypair and use it to control a wallet linked to a public key [15]. Public keys are addresses that are the cryptographic hashes representing an address to which someone's wallet is connected. Public keys are more like the account numbers in a bank, which are unique, and they are required to specify the individuals between the transactions. Alongside the public key, a private key is also required to fulfil a transaction. The public key is visible to everyone on the peer-to-peer network, and it is also vital to handle public keys without compromising security [23]. The private key is just like the unique signatures of an individual, and it is used as proof of authentication while completing a transaction. The private key is also a hash that is only valid for the public key for which it has been generated [24]. The private key is the master key of an individual public address, so whoever owns the private key of an address is the owner of the public address and will have the full privilege to perform any transaction. Every transaction in a blockchain requires both the public key and private key to make the transaction more authentic and secure [5].

Blockchain has already improved the finance sector with its highly secure and trustworthy mediums for many businesses and trade cases like insurance, investment deals, venture capital financing, and much more [25]. Today, the use of blockchain is spreading over different industries, and it is also being leveraged for the advancement of security in fields like travel [26], healthcare [27], agriculture [28, 29], and education [30, 31]. Blockchain-based mobile and web applications have also gained popularity recently over the internet, despite a few issues [32].

Ethereum is an open-source decentralized blockchain that supports smart contract functionality [33]. Smart contracts

facilitate the development of decentralized applications [12]. The native cryptocurrency used in this platform is Ether (ETH). It is rated as the second most used blockchain worldwide after Bitcoin [34]. Ethereum allows its users to create and exchange NFTs, and along with that, it also utilizes the ERC-20 token standard on top of the Ethereum blockchain [35]. The ERC-20 tokens can be generated within a smart contract and later be assigned to some of the specified unique public addresses to limit the transactions from the assigned public address over a smart contract. Previously, Ethereum used proof of work (POW) as their consensus mechanism for the mining procedure of their transactions over the network [16]. Recently, they have switched to the proof of stake (POS) consensus algorithm to make the mining procedure more trustworthy. Mining is an essential part of any blockchain as it is the procedure through which new blocks are generated and later included in the actual blockchain. Miners are the ones over a blockchain peer-to-peer network who are responsible to develop, validate, and add a block to the blockchain [36]. They are the ones who verify transactions. Miners are like any node on the network, except they can mine transactions and earn others in return.

To make sure that a transaction is being mined by someone who is trustworthy on the network, the concept of consensus algorithms is being used. In POW, nodes with the highest computational power are given the privilege to mine a transaction. On the other hand, in the case of POS, nodes that have placed a stake in their own cryptocurrency are given the ability to mine a transaction. POS is a safer consensus algorithm than POW because every voluntary miner has some stake to lose if any fault or illegal activity is found, however in POW, the miner has nothing to lose as it is all about his/her own computational power, which makes it less secure [37]. A node in a proof-of-stake (PoS) network is required to stake something it possesses, typically a cryptocurrency. The locked-up stakes get reduced, or payouts are withheld if a malicious node tries to alter the blockchain and the other nodes notice it [38]. Most of the nodes in a POW system must agree on an operation before it can be carried out, and the nodes are rewarded for doing so. Participants are not punished for carrying out a malicious operation, which is the exception in this situation. POW methods cannot prevent users from engaging in selfish mining [39] or a 51% attack. Newer generations of blockchains, such as Ethereum,

have begun to adopt proof-of-stake as the consensus mechanism to address this issue. Like a POW system, a POS system rewards members for engaging in non-harmful activity while holding them accountable and punishing them for any malicious activity [38–42].

2.2 | Smart contract

Smart contracts are programs or protocols that automate the control or execution of legally significant events and activities following the provisions of a contract or agreement. It is like any backend program which acts as an actual contract by removing the requirement of the middleman. Smart contracts are used to connect an application and a blockchain to perform specific tasks over the blockchain network. A smart contract is an automatic contract that is uploaded within a blockchain-managed computer program, and this program contains a set of rules that govern the communication and determination of the contract between the communicating parties [43–45]. Once the defined rules are fulfilled, the contract is automatically enforced to be active. Data interpretation is used to self-verify the terms of a smart contract. Each node connected on a peer-to-peer blockchain network confirms the proper execution of a smart contract, due to which the contract creator loses the ability to control the execution of the contract [43].

Smart contracts are self-executing programs based on agreements between two or more different parties [46]. The execution of any agreement or function within the smart contract requires a trigger, and the trigger can be provided by either the frontend or backend program of the main application. Without the smart contract, a blockchain is just like a normal database that cannot be changed or manipulated [43]. A blockchain only has the ability to store information based on some attributes which are fulfilled by the smart contract before storage. A smart contract itself can act as the backend of any application; however, many libraries on the internet assist the integration of smart contracts with many backend-based programs like AI or API development [47]. Smart contracts can store, retrieve, update, and verify information on a blockchain. Blockchain does not give us the ability to update any data within a block; nevertheless, through a smart contract, we can update the value of an attribute associated with it without changing the data within a block by adding new updated information into a new block [48]. Section focuses on the different ways a smart contract is built and the packages that can be used to integrate a smart contract with other technologies.

2.3 | Blockchain and smart contracts in voting systems

Many voting systems have been proposed and developed so far, and most of them have focused on making it more transparent and tamper-free using blockchain and smart contracts [49–51]. Vairam et al. [10] proposed an online voting system solution focusing on the effects that might occur due to physical damage

by a person on the EVS. They have also claimed that the EVSs have less security than an online voting system. Furthermore, according to [10], a blockchain-based online voting system has more potential to increase the turnouts of an election as it is going to be remote, and voters will have access to the system remotely.

Ethereum supports the use of tokens that limit the users from performing up to a specified number of transactions based on a smart contract. Farooq et al. [9] developed a solution that is based on smart contracts, blockchain, and the use of Voter Coin (VC). The Voter Coin is more like an ERC20 token in the Ethereum network, which is specially used to limit the number of transactions of an individual public address on a specific smart contract [16]. Generating a token or VC does not require any cost; however, while the token is being used, the voters have to pay some gas cost while submitting their vote using the Voter Coin or ERC20 token, which is a drawback from the cost reduction perspective [34, 35, 52]. According to their system [9], only one Voter Coin is provided to the voter after completing the registration process, which means the voter will get only one chance to vote. Nevertheless suppose a new poll is to be created, then the whole procedure of deploying the smart contract and registering every voter into the system needs to be done all over again so that the voters can gain another Voter Coin to submit a new vote into the new poll. This increases the cost of deploying the smart contract and also reduces the enthusiasm of the voters. During the registration process, the voters are required to submit any of their national government IDs so that they can be verified with the government database before they are given access to the Voter Coin and system. Blockchain does provides a certain level of trustworthiness; however, at some point, centralization is required because some information has to be kept secret [12].

Several works have focused on many aspects that have led the solutions to integrate blockchain into its system. A deep analysis of the voting system has been done by Jafar et al. [3] where the authors mentioned the necessary requirements that a voting system must have, which include the eligibility of a voter to take part in the voting process, restriction of voting twice, privacy for the voter to obtain information about his/her choice only, fairness of not obtaining intermediate voting results before any result publishing, soundness of detecting invalid ballots and taking those votes into account while tallying, completeness that all the valid ballots should be tallied and counted correctly, allow voters to have equal rights of participation, and organize a fair and healthy competition among the candidates [53]. In addition, the voters need to be assured that their vote has been successfully submitted and counted. According to them [3], even if blockchain is used in a voting system, it would be difficult to stop the coercers from standing behind the voter and controlling their actions. In the modern voting system where EVSs are used, the implementation of receipt-freeness relies on the trusted party, which means that even if the voter claims that their vote is recorded or tallied incorrectly, they possess no evidence to claim their injustice [3, 9]. Another factor that needs to be focused on while developing a voting system is to decrease organizational costs and increase voter

turnouts. Puneet et al. [11] provided a solution to make a voting system completely based on a smart contract that will only keep track of the number of votes of an individual poll. For every poll, a new smart contract needs to be deployed, and users need to login into the system as voters or admins to perform their respective tasks. One thing that has not yet been mentioned in this solution is the registration process through which a voter must be verified with the government database to access to the system [11].

3 | PROPOSED SYSTEM

The proposed system is concentrated on some of the main perspectives that a voting system should have, which include decentralization, secure voting, cost reduction in the organization, remote voting, voter validation, vote count surety, voter anonymity, fair competition among candidates, and increment in election turnouts. In our voting system, we verify the voter's and admin's validity to operate within the system using facial recognition and government database cross-match. Restricting the ability of a voter to cast a vote only once on a single poll without any transaction fees and optimizing the smart contract deployment to reduce the cost of deployment in a blockchain are among our primary concerns. We also look into giving the ability to create multiple polls within a state using a single smart contract. Creating a reward incentive for the voters to increase the poll turnouts, and developing a connected platform for election campaigns for different parties which will act as an influencing section for the voters to decide their candidate pick on the poll and also generate revenue for the authority or election administrator who will be paying all the transaction fees for all maximum of the processes executed by the smart contract, are some of the other features that we have integrated into our system. It is to be noted that the whole system will be specifically for a single state only. The smart contract is designed in such a way that a state can have only one single election administrator, and it is independent of other states. The system is built based on the U.S. state system.

3.1 | Actors and roles

The main actors of the proposed system are the voters, elections administrators, party ambassadors, and authority. The poll candidates have the role of being selected as a competitor in the polls; however, they do not have a separate panel to operate within the system. A brief of the actors and roles is given below.

- a. Authority: A government official or supervisor of a state from the IT sector who is responsible for deploying the smart contract.
- b. Election administrator: An election admin is an official who is responsible for creating polls and publish poll results at a specified time. An election admin is also required to verify himself with the government database before he/she is

registered into the system. A state can have only one election admin only.

- c. Voter: A voter is an individual who has the ability to cast a vote in a poll to their respective candidate once they are legally registered into the system.
- d. Party ambassador: A party ambassador is anyone who represents a party. Party ambassadors can create campaigns to influence voters to select their party's candidate as their favourite one.
- e. Poll candidates: A poll candidate is an individual who is participating as a candidate in an election. Poll candidates are to be included by the election admin when creating a new poll.

3.2 | Entities

There are four main entities involved in this system, which have been briefly discussed in Section 4. Some of the vital entities are described below:

- a. Poll: A poll is an election platform where voters can select their preferred candidate and cast their vote. Polls are meant to be created by an election admin only. A state can have multiple polls. Each poll can have two candidates only.
- b. Campaign: A campaign is a blog that consists of a heading, description, and image. Each campaign can be run for a week, and some fees are to be paid to post the campaign. The fees are to be paid in ethers.
- c. Wallet address or ID: A wallet address refers to the public address of an individual account generated by the Metamask wallet to perform transactions within the Ethereum network.
- d. State ID: It is the government identification of an individual citizen of a state. Each state has its own format of state ID, and it is required by the system to verify the users before registering them.

3.3 | System design

The main voting system is connected to a backend server, the primary controller of most functions. The backend server is based on Django, and it is further connected with the smart contract, blockchain, Metamask services, system database, and government database. A detailed view of the interconnectivity between the systems is shown in Figure 2.

There are several phases through which each user on the system needs to pass. The three main phases common for all users include the initialization, registration, and login process. Each phase has been described below:

3.3.1 | Initialization phase

The initialization phase consists of deploying the smart contract and connecting it to the main system. After the smart contract is deployed into a blockchain network, the smart contract address needs to be updated in the Django server, which is the central

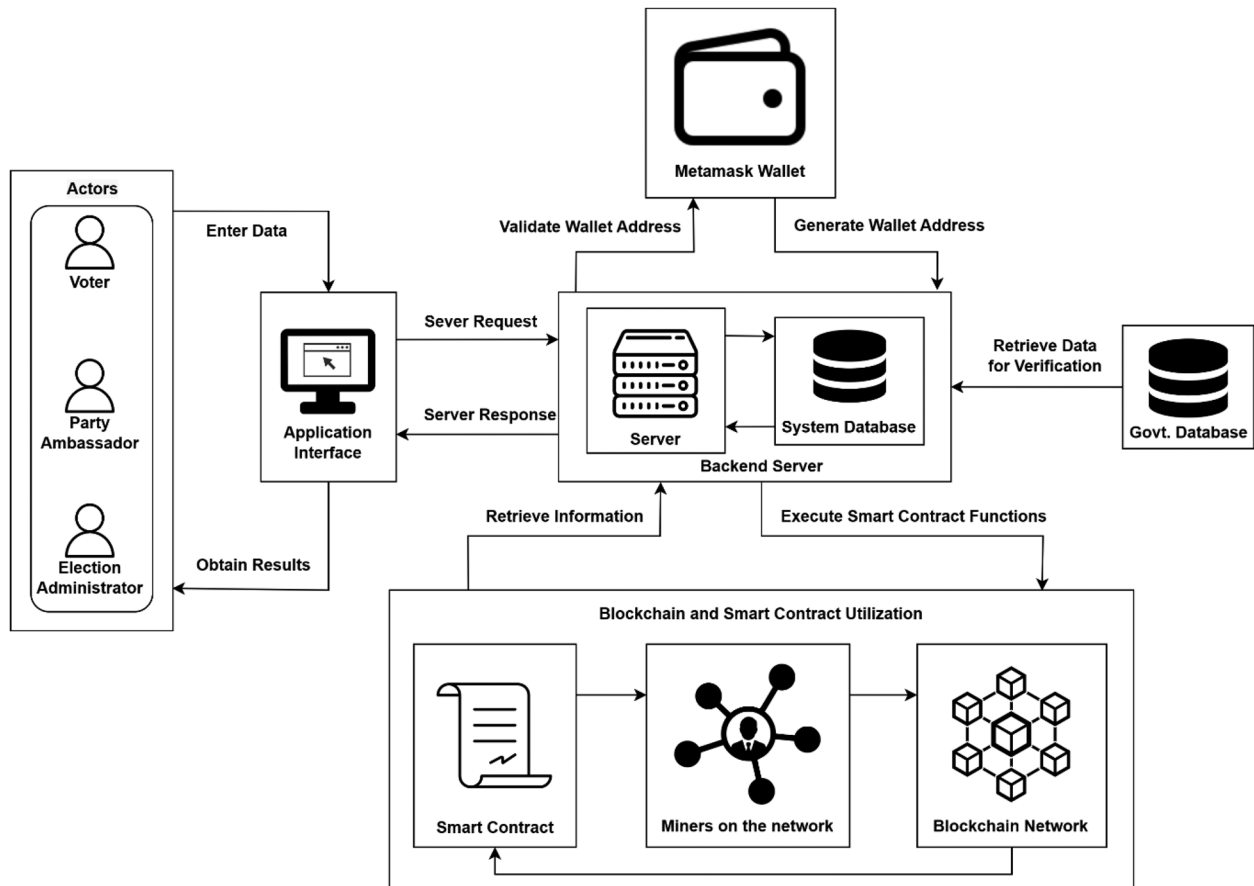


FIGURE 2 System architecture overview

system. In addition, to complete the connection between the backend server and the smart contract, the public and private keys of the account used to deploy the smart contract are required by the “Web3.py” package to be updated. In this case, the authority needs to update this information as they are the ones who are deploying the smart contract. Once this is done, the authority needs to deploy both the frontend and backend systems into different servers to make them available to the users. Figure 3 represents the initialization phase flow.

3.3.2 | Registration phase

The registration phase differs a little from user to user. The registration process begins with all the required details. The common details that every user needs to provide to register into the system include Full Name, Gender, Date of Birth, Email, Wallet Public Address or Wallet ID, State ID, and Password. The most important out of all are the State ID, the Wallet Address, and the Email. These three will be used to identify an individual on the smart contract and blockchain. The rest of the details can be stored in a Web 2.0 database like SQL, keeping the State ID and the transaction receipt as the primary key. The transaction receipt is a hexadecimal hash that will be generated when a transaction is successfully executed using the smart contract. In

a smart contract, a transaction can be referred as the execution of a function of the transfer of ethers from one wallet address to another. If less information is stored in a smart contract, then a better and cost-optimized smart contract can be obtained. The differences in the registration phase for the different users are mentioned below:

1. **Voter:** In the registration phase, the voters need to provide their face id, which will be used to verify the individual with the government database. Once the face recognition algorithm verifies the individual as a voter, he/she will be registered into the system. During the registration process, the voter's details are stored in the SQL database. Only the State ID, email and wallet address are stored into the blockchain database. A transaction receipt is produced when these details of the voter are stored in the blockchain. The State ID and the transaction receipt are used as a reference between the two databases (SQL and Blockchain) for retrieving information.
2. **Election administrator:** The election admins are required to provide their details along with an image of their face which is used to verify them with the government database to ensure that they are officially certified election admins. The only thing which needs to be included as an addition in the details is the private key of the wallet of the election admin.

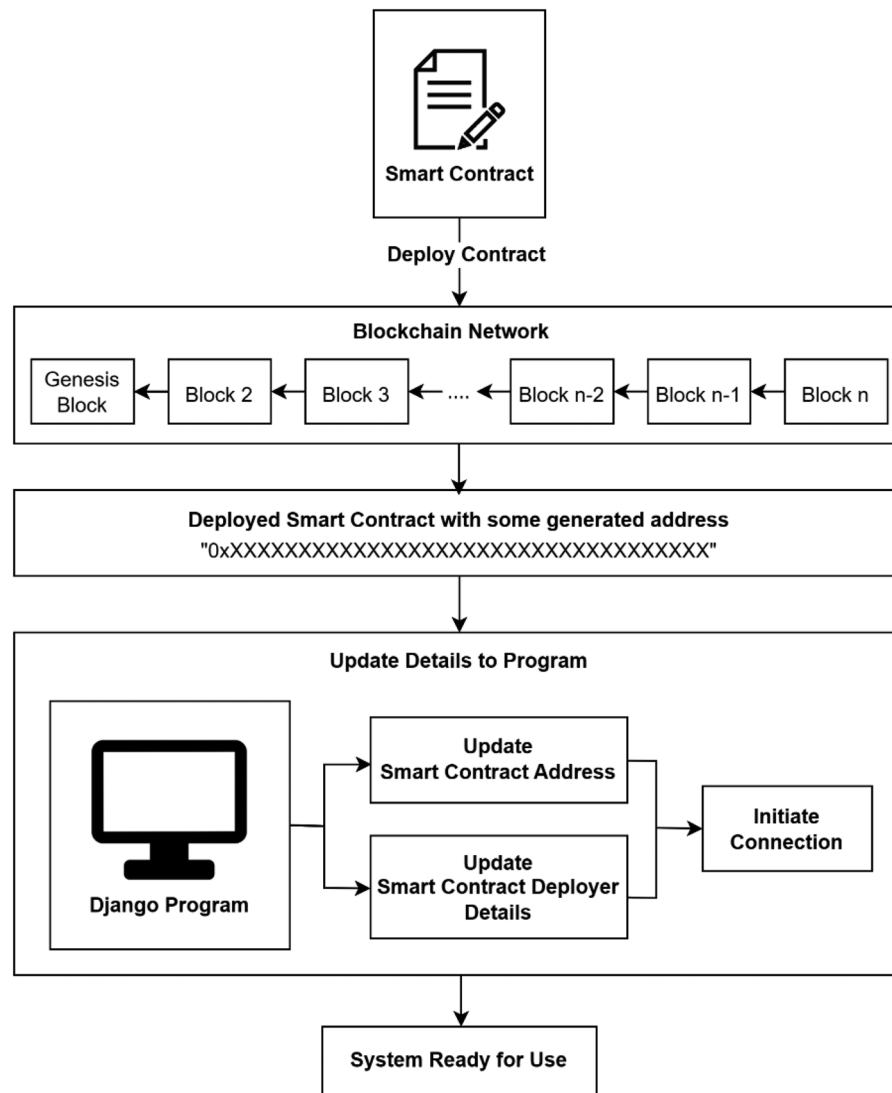


FIGURE 3 Initialization phase flow

Most of the functions in the smart contract can only be executed using the wallet address of the election admin, thus, to automate it in the backend APIs, the private key is required. The election admin is registered into the blockchain using the three essential details mentioned above. One thing to notice is that private keys cannot be stored within a smart contract or a blockchain, so they must be stored in a SQL database, however, in an encrypted way. There are encryption algorithms in python that can be used to encrypt the private key and later on decrypt it whenever required by the API. The python library used in our system is named “cryptocode” which assists in encrypting specific data using a keyword key. The encryption and decryption keys need to be the same; otherwise, the original data cannot be obtained after decryption. Furthermore, the private key of the administrator can be encrypted more than once before it is stored in the SQL database. Cryptocode allows using multiple keys for encrypting and decrypting data, adding extra layers

of protection. Our system uses multiple encryption keys to encrypt the private key for better protection. All the encryption keys will be within the python program and will not be accessible directly to anyone, thus making the private key of the administrator secure enough. Also, the control over the SQL database is going to be in the hands of the authority, so they do not improvise and cause some threat by changing any of the details of the election admin or voter. Even if they do, there will be no effect on the data stored in the blockchain as it is completely separated from the SQL database.

3. Party ambassador: Party ambassadors are not required to provide any image of their face; and, they will not be required to verify themselves with the government database.

An overview of the face recognition procedure during registration is shown in Figure 4. Once the verification with the government database is completed and approved, the image of

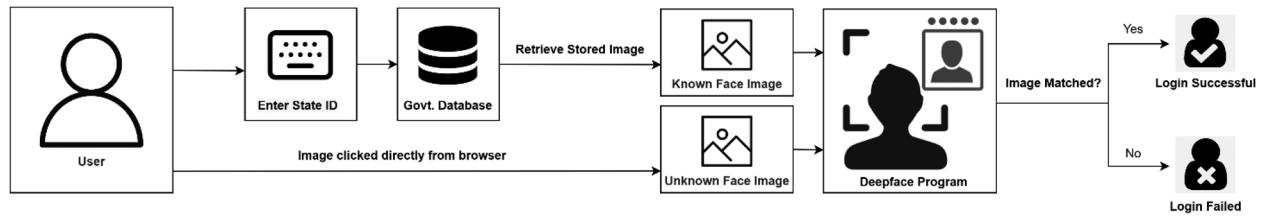


FIGURE 4 Face recognition procedure

the voter and the election admin is also stored in the system database for later use.

3.3.3 | Login phase

In the login phase, the users are verified using their email and password. The voters or election admins also need to use the face verification option to login into the system. The email verification of the user is directly done from the blockchain database and the Django API. Only face verification and password verification are to be done with the SQL database, as large files cannot be stored in a smart contract or blockchain.

3.3.4 | Poll creation phase for election admins

This phase is only accessible to the election admins as they have the ability to create polls. To generate a new poll, the admin needs to provide a unique poll name completely different from any previously generated poll. In addition, it also needs to include candidates for the poll. The details that are required for a candidate include “Full Name,” “Gender,” “Party Name,” and “State ID.” Once the data is validated, at first, the poll is registered, and then the details of the candidate are stored in the blockchain. A reference of the poll is attached with the candidate depending on which poll the candidate has been registered. Figure 5 shows the systematic flow of the poll creation procedure.

3.3.5 | Campaign creation phase for party ambassadors

The party ambassadors can create a campaign by providing a heading, description, and image for the blog regarding the campaign. They also need to enter the private key for the wallet address they provided during their registration phase. The private key is required as some fee has to be paid to publish the campaign. The fee is transferred to the admin wallet address automatically using the Django API and “Web3.py” package. Indirectly, the campaigns are acting as a revenue source for the authority. The fee is paid in ethers. Figure 6 shows the systematic flow of the campaign creation procedure.

3.3.6 | Voting phase for voters

In the voting phase, a voter can select a poll from available polls created by the election admin. Each poll has an expiry date after which voters cannot vote on that poll. Expired polls need to be visible to the voters. If the voter has not already voted on a particular poll, they can select the poll to cast their vote. Figure 7 shows the systematic flow of the voter panel.

Once the poll is selected, the voter can choose his/her preferred candidate and confirm his/her vote. As soon as the vote is confirmed by the voter, the “cast vote” API is executed. This API initially verifies whether the voter is valid by cross-checking the voter’s details with the details already registered in the blockchain. Then it checks whether the voter has voted on the particular poll. The votes of the voters are stored in the blockchain in an object array named “voter_tracks” which uses the string combination of the State ID and the name of the poll. The candidate’s State ID and State Code are stored at the given index. This completes the vote submission. Following the vote submission step, an object array “candidate_vote_tracks” is updated. For index identification, this array uses the string combination of the State ID of the Candidate and the name of the poll. At the given index, the number of votes for the candidate is incremented by one. Finally, the total vote counts of the poll are incremented by one. This completes the vote tally and confirms the vote count of the voter. A transaction receipt is generated for the execution of the “cast vote” function in the smart contract, which is provided to the voter for any future query. The receipt also confirms to the API that the voting process is complete, and the voter can be rewarded some cryptographic currency (ethers) for their vote submission. The reward is directly sent from the wallet address of the admin of the state. One thing to notice here is that the voter required no fee during his vote submission. The whole process execution gas fee is automatically paid directly from the admin wallet address from the “cast vote” API in Django.

4 | IMPLEMENTATION

To develop the system, we have used the Ganache, which is a private Ethereum blockchain used for testing newly created smart contracts. We have used the Metamask wallet to connect the Ganache server and utilize it in our Django APIs. A wallet is always required to perform a transaction on blockchain. We are using Metamask as the wallet by employing the free

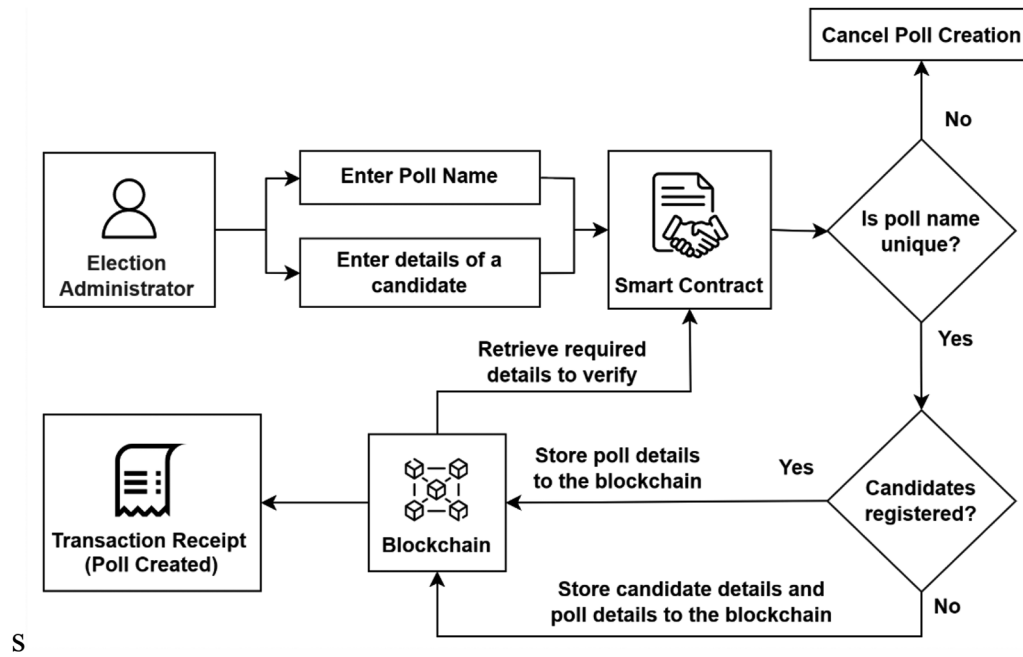


FIGURE 5 Poll creation process

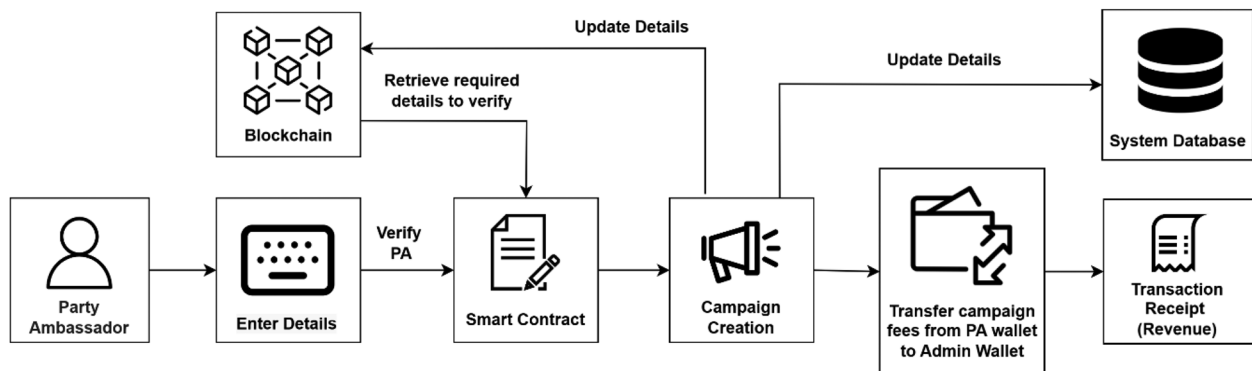


FIGURE 6 Campaign creation flow diagram

accounts provided by Ganache. Metamask has been used so that if in the future this system is uploaded into an actual Ethereum main-net, we can take through some analysis. Smart contracts are primarily developed using languages like Solidity, Rust, JavaScript, and Vyper. Solidity is one of the most popular languages that we have used to develop our smart contract for the Ethereum blockchain network. We used the Remix IDE to build the smart contract, and later when we integrated the smart contract into the Django APIs, we used the Web3.py package to trigger all the smart contract and blockchain-related tasks. Only the sensitive data have been added to the smart contract, and the rest are stored in the system database. We tried to develop the smart contract as efficiently as possible to reduce the cost of the contract deployment. The smart contract is mainly used to keep track of the vote counts and user validity. Using the smart contract, we are able to store the vote counts and the user data in the blockchain, which becomes immutable with every transac-

tion. The extra details that are obtained from the users are stored in the SQL database. Only the reference of the State ID, Email, and Wallet address from the blockchain is enough in the SQL database to identify a user. A modifier in a smart contract is used to limit the access of functions based on user privilege. Figures 8 and 9 shows the comparison of deployment cost between two different deployed smart contracts. Also, an extra layer of security has been included by the addition of biometric authentication. This will make the platform more reliable for the voters. Figure 10 demonstrates the working of the face-recognition feature. Table 1 shows the object attributes and the main functions in the smart contract, and Table 2 describes the attributes that are combined into a structure to form the object attributes.

As we are using a backend server for our system, we may face the problem of single-point-of-failure. However, these types of problems are faced when we design a network architecture for a whole system. As our system is focused on the software rather

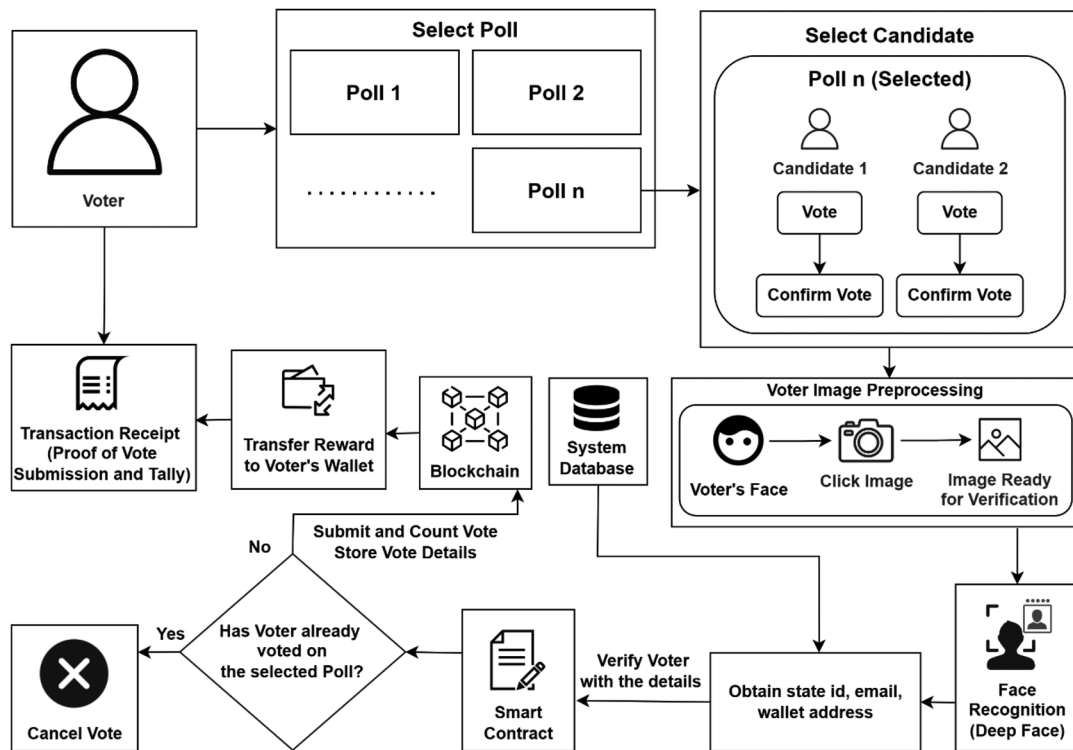


FIGURE 7 Vote casting procedure

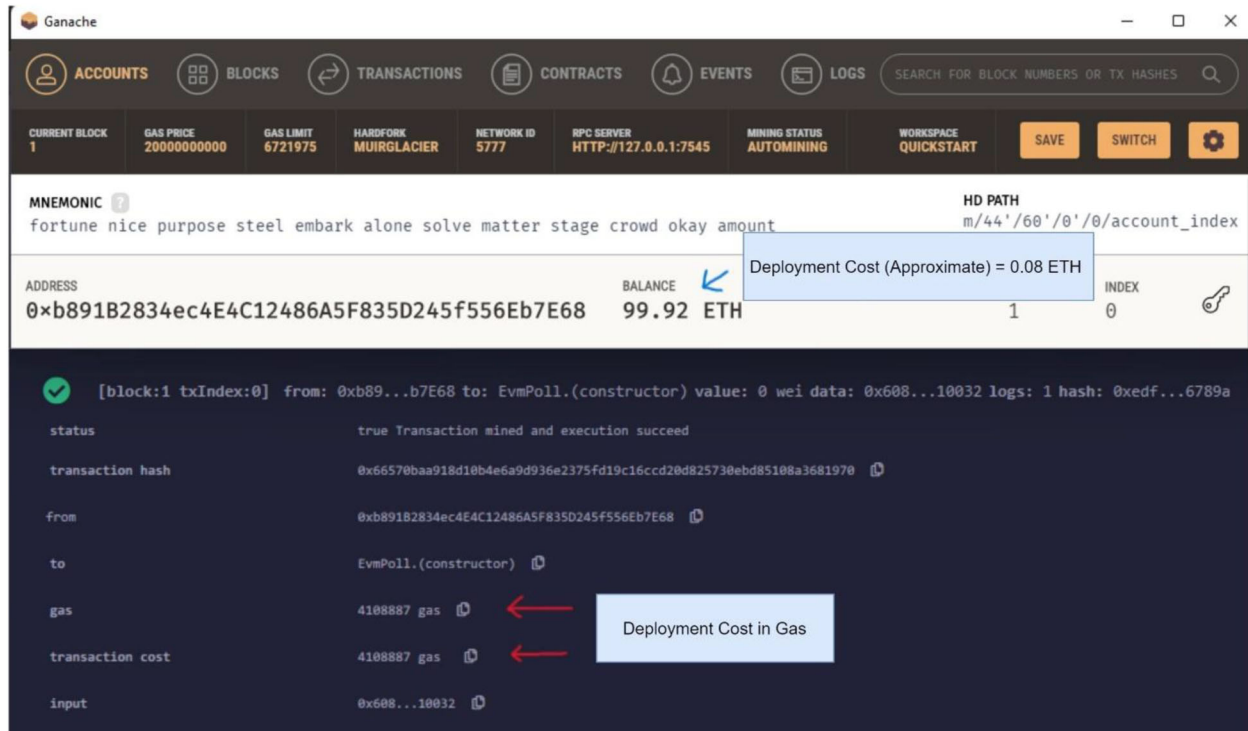


FIGURE 8 Comparison between smart contract deployment (high-cost smart contract)

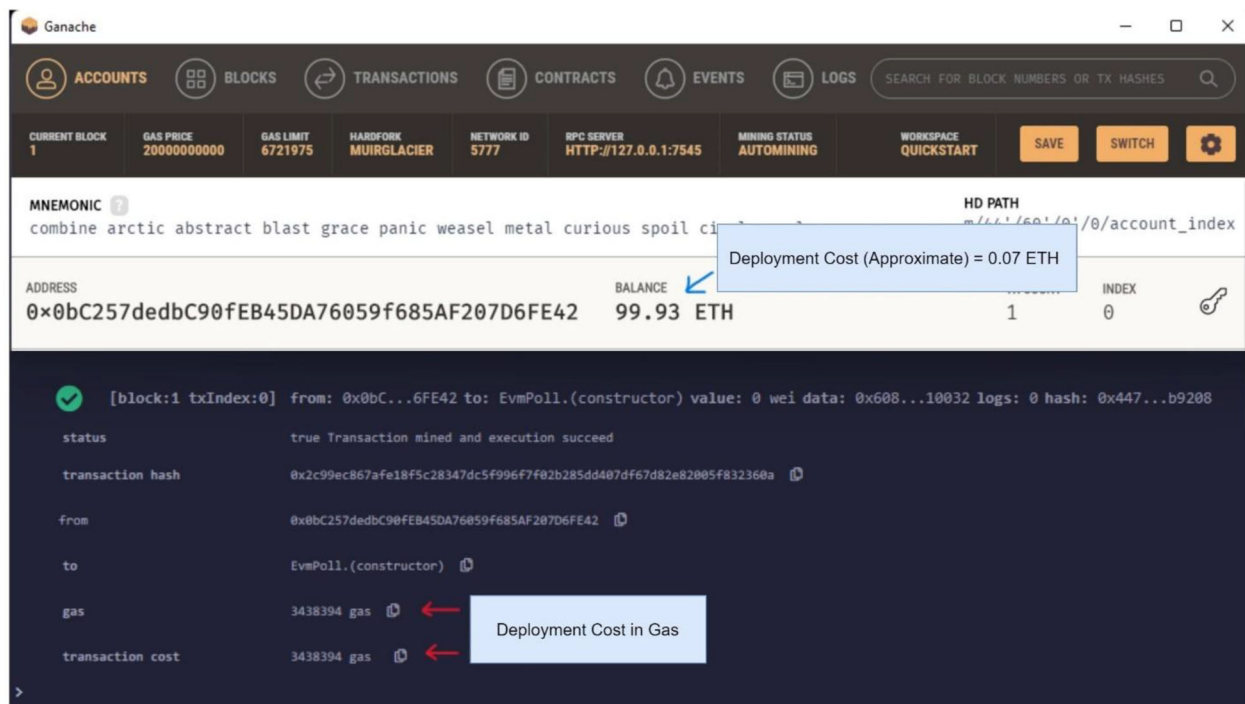


FIGURE 9 Comparison between smart contract deployment (low-cost smart contract)

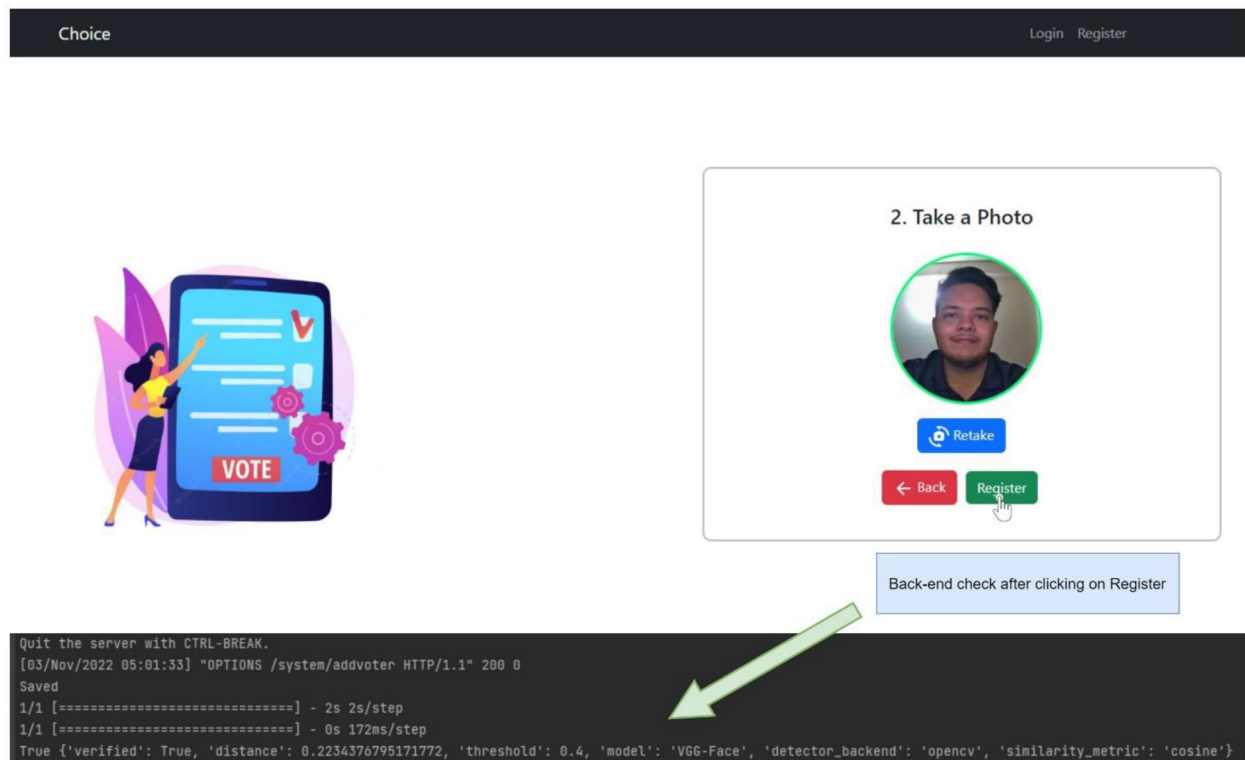


FIGURE 10 Face recognition during voter registration

TABLE 1 Smart contract entities and function

Object attributes	Modifier and functions
voters: Voter	adminPrivilege(): modifier
candi-	authorityPrivilege(): modifier
date_voter_tracks:	addAdmin(): authorityPrivilege
CandidateVoteTrack	createNewPoll(): adminPrivilege
polls: Poll	addVoter(): adminPrivilege
admins: Admin	castVote(): adminPrivilege
voter_tracks:	
VoterTrack	

TABLE 2 Smart contract structures and respective attributes

Structure	Attributes
Voter	state_id: string email: string wallet_id: address
Admin	state_id: string email: string wallet_id: address state: string
Poll	poll_name: string state: string total_votes: uint created_date: unit256 expire_date: unit256 admin_creator_wallet_id: address
CandidateVoteTrack	candidate_state_id: string poll_name: string state: string votes: uint

than the hardware component, and we have to obtain a server from server providers, to solve the issue of single-point-of-failure, we will have to obtain a server from server providers that give backup and redundant systems, have load balancers, include backup power to prevent the loss of power and always maintain an up-to-date data security infrastructure that mitigates the threat from cybersecurity attacks.

Most of the functions mentioned above are related to the smart contract and can be executed only if they are triggered automatically by the contract. The Web3.py package for python provides all the necessary components to trigger a function in the smart contract. We have integrated the smart contract with our Django APIs using this package. The APIs that we have developed using Django and their respective functions for different panels in the system are shown in Table 3.

Table 4 shows the gas cost comparison between the experimental smart contract functions and our prototype smart contract functions. A detailed calculation of the total gas cost in a poll for our system is demonstrated in Section 5.2).

Some crucial functions are represented as algorithms in the following part of the paper. The vote-casting procedure is one of the most essential features of the whole voting system, divided into two algorithms. One part of the algorithm is the actual API which takes the data entered by the voter from the

TABLE 3 Django APIs based on different user panels

User panel	API (function view)	API (sub-URL)
Admin	addAdmin(): status_response loginAdmin(): status_response profileAdmin(): json_data_response createPoll(): status_response pollsAdmin(): json_data_response pollPublish(): status_response	system/addadmin system/loginadmin system/profileadmin system/createpoll system/pollsadmin system/pollpublish
Voter	addVoter(): status_response loginVoter(): status_response profileVoter(): status_response pollsVoter(): json_data_response castVote(): status_response voterCam-paign(): json_data_response	system/addvoter system/loginvoter system/profilevoter system/pollsvoter system/castvote system/campaignsvoter
Party ambassador	addPartyAmbassador(): status_response loginPartyAmbassador(): status_response profilePartyAmbassador(): json_data_response createCampaign(): status_response paCam-paign(): json_data_response	system/addpartyambassador system/loginpartyambassador system/profilepartyambassador system/createcampaign system/campaignspa

TABLE 4 Gas cost comparison between development phase smart contracts

Smart contract functions	Experimental smart contract design (without SQL database) gas cost	Prototype smart contract design (with SQL database) gas cost
Deployment	4,108,887	3,441,862
Create new poll	151,413	149,449
Cast vote	233,734	203,734
Add new voter	110,854	100,686
Add new admin	88,805	86,535
Add new candidate	282,780	279,393

client and makes it ready for further use. The other part of the algorithm focuses on storing crucial information like the vote detail into the blockchain. Algorithm 1 and Algorithm 2 represent the algorithms used for the vote-casting purpose.

The face recognition feature is an addition to the authentication layer of the system, and it has been utilized to verify voters and election administrators before registering them into the application. The face recognition feature has been built using the Deep Face python package and OpenCV under TensorFlow. Usually, the Deep Face algorithm recognizes a match to be true if the distance between the images is less than "0.40". However, after running a number of tests, we decided that it would give

ALGORITHM 1 Proposed Vote Casting Algorithm.

```

01 Procedure InitializeContract(c)
02 Procedure BuildVoteCastingTransaction(V, w)
03   CheckVoterValidity (V)
04     X ← Voter Wallet address, candidate state id, poll name, state
05     return X
06 end Procedure
07 cv (Vote Casting Details) ← BuildVoteCastingTransaction(V, Wv)
08 st (Sign Transaction) ← Procedure SignTransaction(cv, p)
09 I (Initiate Transaction) ← Procedure SendTransaction(st, cv)
10   Procedure CastVote(cv)
11   R (Generate Transaction Receipt) ← Procedure GenerateTransactionReceipt(sd)
12   if transaction hash of R is not empty, then
13     Procedure TransferReward(from w to Wv)
14   else
15     DisplayError(message)
16   endif

```

ALGORITHM 2 Proposed Vote Casting Algorithm in Smart Contract.

```

01 if (Vvsi·w) is not empty then
02   if ((CVTcsi+p).p) is not empty then
03     if ((VTvsi+p).p) is empty then
04       VTvsi+p ← CreateNewVoterTrack(vsi, p, s, csi)
05       CVTcsi+p.v += 1
06       Ps+p.tv += 1
07   else
08     DisplayError(message)
09   endif

```

a better and more accurate result if we reduced the threshold to around “0.25”. Algorithm 3 shows the use of face recognition using the Deep Face package. IMG_{known} is the known image obtained from the system database, and IMG_{Uploaded} is the image clicked through the system. “R” is the details of the face match records between two different images and “d” is the distance of difference.

The total number of votes cast on a poll needs to be transparent. In addition, poll details stored in a blockchain can prohibit the creation of the same poll for a given state. Algorithm 4 shows the procedure of adding a new candidate based on a poll using the smart contract. Algorithms 5 and 6 focus on poll creation using the smart contract and the Django API, respectively.

The poll name is represented as “p,” and “s” is the state. CVT is the array that tracks the vote details for a candidate in a specific poll, “v” is the number of votes of a candidate, and “csi” is the candidate’s state id.

Polls are shown as P, the poll name is represented as “p,” the total number of votes in a poll as “tv,” “c” as the created date, and “e” as the expiry date. “s” is the state code.

ALGORITHM 3 Proposed Face Recognition Algorithm using Deep Face Package

```

01 IMGknown ← import image from database
02 IMGUploaded ← import recently clicked image
03 R ← DeepFace.verify(IMGknown, IMGUploaded)
04 if faces are found in IMGknown and IMGUploaded then
05   return false
06 else
07   check Rd
08   if Rd ≤ 0.25 then
09     Return true
10   else
11     return false
12   endif

```

ALGORITHM 4 New Candidate Adding Algorithm—Based on Poll Algorithm in Smart Contract.

```

01 if ((CVTcsi+p).p) is empty then
02   v ← 0
03   CVTcsi+p ← Procedure AddNewCandidateBasedOnPoll(p, s, v)
04 else
05   DisplayError(message)
06 endif

```

ALGORITHM 5 Proposed Poll Creation Algorithm in Smart Contract.

```

01 if ((Ps+p).c) is empty then
02   Ps+p ← CreateNewPoll(p, s, tv, c, e)
03 else
04   DisplayError(message)
05 endif

```

ALGORITHM 6 Proposed Poll Creation and Add Candidate Algorithm

```

01 poll_created ← CallCreatePoll (p, s)
02 if poll_created is true then
03   candidate_added ← CallAddCandidate(csi, p)
04   if candidate_added is true then
05     StoreCandidateDetailsInSystemDatabase()
06   else
07     DisplayError(message)
08   endif

```

ALGORITHM 7 Proposed Campaign Creation Algorithm

```

01 details_verified  $\leftarrow$  VerifyCampaignDetails(C)
02 if details_verified is true then
    | SaveCampaign(C)
    | PayFees(from paw using pawk, to adw)
03 else
    | DisplayError(message)
04 endif

```

The campaign creation feature is also an important one. Algorithm 7 shows the creation of a new campaign by a party ambassador and the revenue system for the authority. The campaigns do not have information that needs to be transparent and tamper-proof other than the campaign creation fees only.

5 | DISCUSSION

The characteristics and capabilities of our suggested system are illustrated in the previous two sections. Blockchain, smart contracts, and AI have all worked together to help create a platform for remote voting that is efficient, optimized, reliable, and trustworthy. The theoretical and practical implications of our study are specified below.

5.1 | Theoretical implications

We have increased the security of our system thanks to the use of blockchain technology and smart contracts. A variety of systems for medical, financial, and agricultural platforms can be developed using the features of making data unchangeable and visible. The intermediaries that must be eliminated from the perspective of report ownership are hospitals. Because patients should own their documents outright, using blockchain technology and smart contracts could help cut out hospitals as a middleman. Verified contracts provide the basis of real estate ownership. Even after a property transfer is complete, the real estate agent can still access the property information. In the future, fraud may be committed using this data. Without the use of a real estate agent, blockchain technology and smart contracts can aid in making this information secure and transparent. For a nation's agricultural industry to guarantee food safety, the traceability of the information about the food supply is essential. Plant species, seed quality, crop growth, and the distribution of food supplies are some of the details that need to be carefully considered and protected from shady business practices. Blockchain can be used to protect this data from unlawful and unethical activities. A summary of the theoretical implications of our study is given below:

TABLE 5 Gas cost comparison of proposed smart contract and other Ethereum based system

Smart contract functions	Prototype smart contract design (with SQL database) gas cost	Dagher et al., [65] Ethereum based system gas cost
Deployment	344,1862	3,817,723
Create new poll	149,449	–
Cast vote	203,734	120,1820
Add new voter	100686	473,640
Add new admin	86,535	–
Add new candidate	279,393	–

1. Eliminate the role of the hospital as a middleman and give patients full ownership of their reports by using the transparent property of blockchain [54, 55].
2. Removing the need for real estate agents and creating an automated system that uses a blockchain and smart contracts to perform land deals online [56–59].
3. Implement a secure blockchain database to protect the information about the food supply from unauthorized and immoral activity [60–62].

5.2 | Practical implications

Electronic voting systems (EVSs) began to take the place of traditional paper ballots, and today both systems are used in conjunction during elections in several nations, including the United States. However, vote casting time, confirmation of vote tally, vote manipulation, voter anonymity, and unfair competition are some aspects that require attention. Our proposed solution provides a remote platform where voters can vote quickly, safely, and more efficiently. Blockchain and Smart contracts [63] have granted the facility of protecting data by making it transparent and tamper-proof. Data protection is now possible thanks to blockchain technology [64] and smart contracts, which make data transparent and impenetrable. Although using blockchain to store data is advantageous, a hybrid system must be introduced to lower the cost of deploying smart contracts. By keeping the immutable data on the blockchain and the remaining information in the system database, our semi-decentralized approach reduces the deployment cost, which is advantageous for the authority. Additionally, it aids in demonstrating voter privacy. Our application concentrates on voting without paying any vote-casting fees, which is required in most of the previous solutions mentioned in Table 6. The integrated campaign system will lower the election cost and benefit the authority through its revenue-generation technique. The adoption of biometric authentication has added an additional degree of protection to all the users within the system. It ensures that only users with verified identities are using the system. Voters will have greater confidence in the platform as a result. Table 5 shows the gas cost comparison between our proposed smart contract and an Ethereum-based system by Dagher et al. [65].

TABLE 6 Comparison of proposed system and existing work

Paper Name	Working prototype	Cost-free vote casting	Face recognition biometric authorization	Reward incentive for voters	Ethereum based system	Optimization of smart contract deployment cost	Turnout improvement focused system	Transaction receipt as a proof of vote submission
Farooq et al., [9]	✓	✗	✗	✗	✗	✗	✓	✓
Vairam et al., [10]	✓	✗	✗	✗	✓	✗	✗	✓
Jafar et al., [3]	✗	✗	✗	✗	✓	✗	✓	✗
Puneet et al., [11]	✓	✗	✗	✗	✓	✗	✓	✓
Balaji et al., [66]	✓	✗	✗	✗	✗	✗	✓	✗
R et al., [67]	✓	✗	✗	✗	✓	✗	✗	✓
Our proposed system	✓	✓	✓	✓	✓	✓	✓	✓

Table 6 demonstrates that, to our knowledge, no other prior work has been published with a blockchain-based remote voting system that allows voters to cast ballots without paying any fees, has a biometric face recognition feature for user verification, and offers a reward incentive mechanism that serves as a turnout improvement process. Moreover, only a few studies have focused on developing a functioning prototype that provides transaction receipts as proof of vote submission and has used blockchain to increase transparency and tamper-proofness. The practical implications of our research can be summarized as follows:

1. Introduce a remote, reliable, trustworthy, and tamper-proof voting platform so voters can cast their ballots without being concerned about being hacked.
2. Establish fair competition between candidates by restricting the authority's ability to manipulate the vote, and limiting access to the system via biometric authentication to people whom the government has verified.
3. Motivate voters within the society to submit their ballot through a reward incentive mechanism used to improve election turnouts.
4. Give the parties a campaign platform so they can advertise their party throughout the election and make it more difficult for their opponents to win.
5. Use a system of income generation that is profitable for the authority to lower the overall election costs, and optimize the cost of deploying smart contracts through a semi-decentralized system.

6 | CONCLUSION

This paper introduces a solution to make a trustworthy, reliable, and optimized voting system leveraging blockchain technology and artificial intelligence. The voter anonymity issue has been solved using an intermediate server which is integrated with the main smart contract, thus making the system a semi-decentralized one. Voter verification has been improved with an extra layer of security, and the voters are confirmed about their vote submission with a tally through a transaction receipt in real-

time. The system provides a tamper-proof, non-manipulative, and fair platform for both the candidates and voters due to the use of blockchain and smart contracts. The cost of smart contract deployment and organization of the whole election event has been optimized by reducing the size of the smart contract and introducing a campaign system, and providing a remote platform for voting, respectively. Finally, voters do not have to pay any gas fees while submitting their vote; instead, the reward incentive mechanism assists in increasing the number of overall election turnouts, and the solution has been completely done without using any external token service like the ERC20 token. In the future, this system can be improved further to make it more reasonable by allowing for the addition of more than two candidates on a single poll, protecting by strengthening the authentication process with a multilayer of protection using voice recognition and fingerprint matching and optimizing by running an analysis on the system when the smart contract is uploaded on a main-net. A candidate panel can also be introduced where candidates will be able to perform campaign-based activities like party merchandise and video reels and, on top of that, organize party-based events. Moreover, the ability to specify the interval of a poll to the election administrator and to provide an option to the party ambassadors to create an interval-based campaign at different rates will make the system more feasible.

AUTHOR CONTRIBUTIONS

Mohammad Nabiluzzaman Nelay, Md. Abdul Wahab, Sheikh Wasif.: Conceptualization, methodology, software, formal analysis, validation, writing - original draft, writing - review and editing, visualization; Abdulla All Noman, Mustafizur Rahaman.: Methodology, formal analysis, software; Tahmid Hasan Pranto: Data curation, writing - original draft, visualization; A. K. M. Bahalul Haque: Investigation, resources, software; Rashedur M. Rahman: Investigation, validation, funding acquisition.

ACKNOWLEDGEMENTS

This work was supported by the Faculty Research Grant [CTRG-21-SEPS-19], North South University, Bashundhara, Dhaka 1229, Bangladesh.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

DATA AVAILABILITY STATEMENT

Data sharing does not apply to this article as no new data were created or analysed in this study.

ORCID

Rashedur M. Rahman  <https://orcid.org/0000-0002-4514-6279>

REFERENCES

- Park, S., Specter, M., Narula, N., Rivest, R.L.: Going from bad to worse: From internet voting to blockchain voting. *J. Cybersec* 7(1), tyaa025 (2021). <https://doi.org/10.1093/cybsec/tyaa025>
- Fernandes, A., Garg, K., Agrawal, A., Bhatia, A.: Decentralized online voting using blockchain and secret contracts. In: *International Conference on Information Networking*, pp. 582–587. IEEE, Piscataway, NJ (2021). <https://doi.org/10.1109/ICOIN50884.2021.9333966>
- Jafar, U., Aziz, M.J.A., Shukur, Z.: Blockchain for electronic voting system—review and open research challenges. *Sensors* 21(17), 5874 (2021). <https://doi.org/10.3390/s21175874>
- Park, H.D.: A decentralized E-voting system based on blockchain network. *Int. J. Innovative Technol. Explor. Eng.* 8(12), 3650–3652 (2019). <https://doi.org/10.35940/ijitee.L3815.1081219>
- Kurbatov, O., Kravchenko, P., Shapoval, O., Poluyanenko, N., Malchuk, M., Sakun, A., Kovtun, V.: Anonymous Decentralized E-Voting System. (pp. 12–22), *International Workshop on Conflict Management in Global Information Networks* (2020)
- Shah, S., Kanchwala, Q., Mi, H.: Block Chain Voting System, North-eastern University, Available: <https://www.economist.com/sites/default/files/northeastern.pdf>. Accessed: 25 Dec, 2022
- Blockchain based E-voting system for India using UIDAI's Aadhaar, In: Thakkar, H.K., Swarnkar, M., Bhadoria, R.S. (eds.) *Predictive Data Security using AI. Studies in Computational Intelligence*, vol. 1065. pp 89–103, Springer, Singapore (2019). <https://doi.org/10.5281/zenodo.3428327>
- Rashid, M.: EVM can be manipulated, says US academic. *The New Age Bangladesh* (2022). Available: <https://www.newagebd.net/article/181281/evm-can-be-manipulated-says-us-academic>
- Farooq, M.S., Iftikhar, U., Khelifi, A.: A framework to make voting system transparent using blockchain technology. *IEEE Access* 10, 59959–59969 (2022). <https://doi.org/10.1109/ACCESS.2022.3180168>
- Vairam, T., Sarathambekai, S., Balaji, R.: Blockchain based voting system in local network. In: *2021 7th International Conference on Advanced Computing and Communication Systems, ICACCS 2021*, pp. 363–366. IEEE, Piscataway, NJ (2021). <https://doi.org/10.1109/ICACCS51430.2021.9441912>
- Puneet, Chaudhary, A., Chauhan, N., Kumar, A.: Decentralized voting platform based on ethereum blockchain. In: *Proceedings of the 2021 1st International Conference on Advances in Electrical, Computing, Communications and Sustainable Technologies, ICAECT 2021*. IEEE, Piscataway, NJ, pp. 1–6 (2021). <https://doi.org/10.1109/ICAECT49130.2021.9392580>
- Sheer Hardwick, F., Gioulis, A., Naeem Akram, R., Markantonakis, K.: E-voting with blockchain: An E-voting protocol with decentralisation and voter privacy. In: *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pp. 1561–1567. IEEE, Piscataway, NJ (2018). <https://doi.org/10.1109/Cybermatics.2018.2018.00262>
- Nakamoto, S.: Bitcoin: A Peer-to-Peer Electronic Cash System (2008). <https://doi.org/10.2139/ssrn.3977007>
- Balaji, S., Vishagasudan, S., Ezhilarasi, K.: Online voting system using block chain technology. *Int. J. Health Sci. (Qassim)* 3978–3987 (2022). <https://doi.org/10.53730/ijhs.v6ns4.10425>
- Jafar, U., Aziz, M.J.A., Shukur, Z.: Blockchain for electronic voting system—review and open research challenges. *Sensors* 21(17), 5874 (2021). <https://doi.org/10.3390/s21175874>
- Buterin, V.: A next generation smart contract & decentralized application platform, Whitepaper, 2014. Available: <https://finpedia.vn/wp-content/uploads/2022/02/Ethereum-white-paper-a-next-generation-smart-contract-and-decentralized-application-platform-vitalik-buterin.pdf>
- Wang, H., Zheng, Z., Xie, S., Dai, H.N., Chen, X.: Blockchain challenges and opportunities: A survey. *Int. J. Web Grid Serv.* 14(4), 352 (2018). <https://doi.org/10.1504/ijwgs.2018.10016848>
- Li, Z., Gao, S., Peng, Z., Guo, S., Yang, Y., Xiao, B.: B-DNS: A secure and efficient DNS based on the blockchain technology. *IEEE Trans. Netw. Sci. Eng.* 8(2), 1674–1686 (2021). <https://doi.org/10.1109/TNSE.2021.3068788>
- Ben Ayed, A.: A conceptual secure blockchain based electronic voting system. *Int. J. Network Security Appl.* 9(3), 01–09 (2017). <https://doi.org/10.5121/ijnsa.2017.9301>
- Zhang, Y., Lin, X., Xu, C.: Blockchain-based secure data provenance for cloud storage. *Information and Communications Security, ICICS 2018, Lecture Notes in Computer Science*, vol. 11149, pp. 3–19. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01950-1_1
- Zheng, Z., Xie, S., Dai, H., Chen, X., Wang, H.: An overview of blockchain technology: Architecture, consensus, and future trends. In: *Proceedings - 2017 IEEE 6th International Congress on Big Data, BigData Congress 2017*, pp. 557–564. IEEE, Piscataway, NJ (2017). <https://doi.org/10.1109/BIGDATACONGRESS.2017.85>
- Olleros, F.X., Zhegu, M.: *Research Handbook on Digital Transformations*. Edward Elgar Publishing, Cheltenham, United Kingdom (2016)
- Wang, H., Xu, C., Zhang, C., Xu, J., Peng, Z., Pei, J.: vChain+: Optimizing verifiable blockchain boolean range queries. *Proc. Int. Conf. Data Eng.*, 1927–1940 (2022). <https://doi.org/10.1109/ICDE53745.2022.00190>
- Mccorrey, P., Shahandashti, S.F., Hao, F.: A Smart Contract for Boardroom Voting with Maximum Voter Privacy In: Kiyias, A. (eds) *Financial Cryptography and Data Security. FC 2017. Lecture Notes in Computer Science*, vol. 10322, pp. 357–375. Springer, Cham.
- Varma, J.R.: Blockchain in finance. *Vikalpa* 44(1), 1–11 (2019). <https://doi.org/10.1177/0256090919839897>
- Erceg, A., Sekuloska, J.D., Kelic, I.: Blockchain in the tourism industry - A review of the situation in Croatia and Macedonia. *Informatics* 7(1), 5 (2020). <https://doi.org/10.3390/informatics7010005>
- Haleem, A., Javaid, M., Singh, R.P., Suman, R., Rab, S.: Blockchain technology applications in healthcare: An overview. *Int. J. Intell. Networks* 2, 130–139 (2021). <https://doi.org/10.1016/j.ijin.2021.09.005>
- Sekhar Bhusal, C.: Blockchain technology in agriculture: A case study of blockchain start-up companies. *Int. J. Comput. Sci. Technology* 13(5), 31–48 (2021). <https://doi.org/10.5121/ijcsit.2021.13503>
- Pranto, T.H., Noman, A.A., Mahmud, A., Haque, A.B.: Blockchain and smart contract for IoT enabled smart agriculture. *PeerJ. Comput. Sci.* 7, 1–29 (2021). <https://doi.org/10.7717/PEERJ-CS.407/SUPP-1>
- Ocheja, P., Agbo, F.J., Oyelere, S.S., Flanagan, B., Ogata, H.: Blockchain in education: A systematic review and practical case studies. *IEEE Access* 10, 99525–99540 (2022). <https://doi.org/10.1109/access.2022.3206791>
- Bhaskar, P., Tiwari, C.K., Joshi, A.: Blockchain in education management: Present and future applications. *Interact. Technol. Smart Educ.* 18(1), 1–17 (2020). <https://doi.org/10.1108/ITSE-07-2020-0102/FULL/XML>
- Marijan, D., Lal, C.: Blockchain verification and validation: Techniques, challenges, and research directions. *Comput. Sci. Rev.* 45, 100492 (2022). <https://doi.org/10.1016/j.COSREV.2022.100492>
- Fernandes, A., Garg, K., Agrawal, A., Bhatia, A.: Decentralized online voting using blockchain and secret contracts. In: *International Conference on Information Networking*, pp. 582–587. IEEE, Piscataway, NJ (Jan. (2021)). <https://doi.org/10.1109/ICOIN50884.2021.9333966>
- Singh, A.: A new investment opportunity: Bitcoin & ethereum cryptocurrency. *Int. J. Sci. Res. Eng. Manage.* 06(10), (2022). <https://doi.org/10.55041/IJSREM16525>
- Victor, F., Lüders, B.K.: Measuring Ethereum-Based ERC20 Token Networks. In: Goldberg, I., Moore, T. (eds) *Financial Cryptography and Data Security. FC 2019. Lecture Notes in Computer Science*, vol. 11598 LNCS, pp. 113–129 (2019). Springer, Cham. https://doi.org/10.1007/978-3-030-32101-7_8

36. Fanning, K., Centers, D.P.: Blockchain and its coming impact on financial services. *J. Corporate Accounting Finance* 27(5), 53–57 (2016). <https://doi.org/10.1002/JCAF.22179>
37. Amanda Reaume: Proof of work vs. proof of stake: explained. *Seeking Alpha*. <https://seekingalpha.com/article/4468656-proof-of-work-vs-proof-of-stake>. Accessed: 16 Jan. 2022
38. Thin, W.Y.M.M., Dong, N., Bai, G., Dong, J.S.: Formal analysis of a proof-of-stake blockchain. In: *Proceedings of the IEEE International Conference on Engineering of Complex Computer Systems, ICECCS*, pp. 197–200. IEEE, Piscataway, NJ (2018). <https://doi.org/10.1109/ICECCS2018.2018.00031>
39. Eyal, I., Sirer, E.G.: Majority is not enough: Bitcoin mining is vulnerable, arXiv:1311.0243, (2013)
40. Transactions as Proof-of-Stake! by Daniel Larimer! dlarimer@invictus-innovations.com!
41. Saleh, F.: Blockchain without waste: Proof-of-stake. *Rev. Financ. Stud.* 34(3), 1156–1190 (2021). <https://doi.org/10.1093/RFS/HHAA075>
42. Siim, J.: Proof-of-stake, Research seminar in cryptography. pp. 2, <https://github.com/mit-dci/tangled-curl/blob/master/>. Accessed: 04 Dec. 2022
43. Khan, S., Arshad, A., Mushtaq, G., Khalique, A., Husein, T.: Implementation of decentralized blockchain E-voting. *EAI Endorsed Trans. Smart Cities* 4(10), 164859 (2020). <https://doi.org/10.4108/eai.13-7-2018.164859>
44. Kamilaris, A., Fonts, A., Prenafeta-Boldó, F.X.: The rise of blockchain technology in agriculture and food supply chains. *Trends Food Sci. Technol.* 91, 640–652 (2019). <https://doi.org/10.1016/J.TIFS.2019.07.034>
45. Abdellatif, T., Brousmiche, K.L.: Formal verification of smart contracts based on users and blockchain behaviors models. In: *2018 9th IFIP International Conference on New Technologies, Mobility and Security, NTMS 2018 - Proceedings*, pp. 1–5. IEEE, Piscataway, NJ (2018). <https://doi.org/10.1109/NTMS.2018.8328737>
46. Lauslahti, K., Mattila, J., Seppala, T.: Smart contracts – how will blockchain technology affect contractual practices? *SSRN Electr. J., ETLA Reports*, 68, pp. 1–32 (2017). <https://doi.org/10.2139/SSRN.3154043>
47. Wang, Q., Li, R., Wang, Q., Chen, S., Ryan, M., Hardjono, T.: Exploring web3 from the view of blockchain, arXiv:2206.08821, (2022). <http://arxiv.org/abs/2206.08821>
48. Panda, S.K., Satapathy, S.C.: An investigation into smart contract deployment on Ethereum platform using Web3.js and solidity using blockchain. In: Bhateja, V., Satapathy, S.C., Travieso-González, C.M., Aradhya, V.N.M. (eds.) *Data Engineering and Intelligent Computing. Advances in Intelligent Systems and Computing*, vol 1407 pp. 549–561. Springer, Singapore (2021). https://doi.org/10.1007/978-981-16-0171-2_52
49. Chan Zheng Wei, C., Chai Wen, C.: Blockchain-based electronic voting protocol, 2(4-2), pp. 336–341 (2018)
50. Decentralized application on voting system. *Int. J. Recent Technol. Eng.* 8(4), 12568–12571 (2019). <https://doi.org/10.35940/ijrte.d9757.118419>
51. Rathee, G., Iqbal, R., Waqar, O., Bashir, A.K.: On the design and implementation of a blockchain enabled E-voting application within IoT-oriented smart cities. *IEEE Access* 9, 34165–34176 (2021). <https://doi.org/10.1109/ACCESS.2021.3061411>
52. Gao, S., Peng, Z., Tan, F., Zheng, Y., Xiao, B.: SymmeProof: Compact zero-knowledge argument for blockchain confidential transactions. *IEEE Trans. Dependable Secure Comput.* 1 (2022). <https://doi.org/10.1109/TDSC.2022.3179913>
53. Mehboob Khan, K., Arshad, J., Khan, M.M.: Secure digital voting system based on blockchain technology, In: *Proceedings of IEEE Mysore Sub Section International Conference (MysuruCon)*, pp. 861–870. IEEE, Piscataway, NJ (2021)
54. Azaria, A., Ekblaw, A., Vieira, T., Lippman, A.: MedRec: Using blockchain for medical data access and permission management. In: *Proceedings - 2016 2nd International Conference on Open and Big Data, OBD 2016*, pp. 25–30. IEEE, Piscataway, NJ (2016). <https://doi.org/10.1109/OBD.2016.11>
55. Xie, Y., et al.: Applications of blockchain in the medical field: Narrative review. *J. Med. Internet Res.* 23(10), e28613 (2021). <https://doi.org/10.2196/28613>
56. Shuaib, M., Daud, S.M., Alam, S., Khan, W.Z.: Blockchain-based framework for secure and reliable land registry system. *TELKOMNIKA* 18(5), 2560–2571 (2020). <https://doi.org/10.12928/TELKOMNIKA.V18I5.15787>
57. Thakur, V., Doja, M.N., Dwivedi, Y.K., Ahmad, T., Khadanga, G.: Land records on blockchain for implementation of land titling in India. *Int. J. Inf. Manage.* 52, 101940 (2020). <https://doi.org/10.1016/J.IJINFOMGT.2019.04.013>
58. Ameyaw, P.D., de Vries, W.T.: Transparency of land administration and the role of blockchain technology, a four-dimensional framework analysis from the Ghanaian land perspective. *Land* 9(12), 491 (2020). <https://doi.org/10.3390/LAND9120491>
59. Shuaib, M., Alam, S., Daud, S.M.: Improving the authenticity of real estate land transaction data using blockchain-based security scheme. *Commun. Comput. Inf. Sci.* 1347, 3–10 (2021). https://doi.org/10.1007/978-981-33-6835-4_1/COVER
60. Shahid, A., Almogren, A., Javaid, N., Al-Zahrani, F.A., Zuair, M., Alam, M.: Blockchain-based agri-food supply chain: A complete solution. *IEEE Access* 8, 69230–69243 (2020). <https://doi.org/10.1109/ACCESS.2020.2986257>
61. Rana, R.L., Tricase, C., de Cesare, L.: Blockchain technology for a sustainable agri-food supply chain. *Br. Food J.* 123(11), 3471–3485 (2021). <https://doi.org/10.1108/BFJ-09-2020-0832/FULL/XML>
62. Caro, M.P., Ali, M.S., Vecchio, M., Giaffreda, R.: Blockchain-based traceability in agri-food supply chain management: A practical implementation. In: *2018 IoT Vertical and Topical Summit on Agriculture - Tuscany, IOT Tuscany 2018*, pp. 1–4. IEEE, Piscataway, NJ (2018). <https://doi.org/10.1109/IOT-TUSCANY.2018.8373021>
63. Haque, A.K.M.B., Bhushan, B.: Blockchain in a Nutshell: State-of-the-Art Applications and Future Research Directions, In S. Pani, S. Lau, X. Liu (eds.) *Blockchain and AI Technology in the Industrial Internet of Things*, pp. 124–143. IGI Global, Hershey, Pennsylvania (2021). <https://services.igi-global.com/resolvedoi/resolve.aspx?>
64. Haque, A.B., Muniat, A., Ullah, P.R., Mushsharat, S.: An automated approach towards smart healthcare with blockchain and smart contracts. In: *Proceedings - IEEE 2021 International Conference on Computing, Communication, and Intelligent Systems, ICCICIS 2021*, pp. 250–255. IEEE, Piscataway, NJ (2021). <https://doi.org/10.1109/ICCICIS51004.2021.9397158>
65. Dagher, G.G., Marella, P.B., Milojkovic, M., Mohler, J.: Bron covote: Secure voting system using ethereum's blockchain. In: *ICISSP 2018 - Proceedings of the 4th International Conference on Information Systems Security and Privacy*, pp. 96–107. SciTePress, Setúbal, Portugal (2018). <https://doi.org/10.5220/0006609700960107>
66. Balaji, S., Vishagasudan, S., Ezhilarasi, K.: Online voting system using block chain technology. *Int. J. Health Sci. (Qassim)* 6(S4), 3978–3987 (2022). <https://doi.org/10.53730/ijhs.v6ns4.10425>
67. Savitha, R., Ashwini, K.B., Prashanth, K.: Blockchain-based online voting system. *ECS Trans.* 107(1), 13195–13203 (2022). <https://doi.org/10.1149/10701.13195ecst>

How to cite this article: Nelay, M.N., Wahab, M.A., Wasif, S., All Noman, A., Rahaman, M., Pranto, T.H., Haque, A.K.M.B., Rahman, R.M.: A remote and cost-optimized voting system using blockchain and smart contract. *IET Blockchain* 1–17 (2023). <https://doi.org/10.1049/blc2.12021>