Blockchain Based E-Voting System Using Proof of Work Consensus Algorithm and Machine Learning

V.S. Deshmukh [1*], A. Dongre [2], A. Deshingkar [3], P. Barhanpurkar [4], Dr. R. Chopade [5]

[1]Frontend Developer, [2]ML Engineer, [3]System Engineer, [4]Full Stack Developer, [4]Head of Department, IT, MMCOE,

[1]Information Technology,

[1]Marathwada Mitra Mandal college of Engineering, Pune, India

Abstract - The introduction of electronic voting (e-voting) systems has ushered in a new era of efficiency and convenience in democratic processes. However, concerns regarding security vulnerabilities and authentication have impeded their widespread adoption. This research paper presents a novel e-voting system that harnesses the combined potential of blockchain technology and machine learning (ML) to bolster security and authentication mechanisms. Our specific focus lies in leveraging the proof of work consensus algorithm and a private blockchain to fortify the system's resilience and uphold privacy. Our proposal revolves around the development of a decentralized e-voting framework that relies on blockchain innovation. The implementation of our blockchain technology adheres to the principles of the proof of work consensus algorithm, which ensures decentralization and fosters a secure environment. In tandem, smart contracts are devised and deployed to facilitate transparent voting procedures. Additionally, ML algorithms are trained and integrated into the e-voting system, augmenting authentication precision and enabling the detection of fraudulent activities.

I. INTRODUCTION

The advent of blockchain technology has brought about significant advancements in various domains, with its potential to revolutionize traditional systems of record-keeping, transactions, and security. One area that stands to benefit from blockchain's inherent characteristics is the realm of electronic voting (e-voting) systems. By leveraging the distributed ledger and cryptographic features of blockchain, e-voting systems can enhance transparency, immutability, and trust in the electoral process [2][3]. This research paper explores the integration of blockchain technology, specifically the use of a private blockchain, in conjunction with the proof of work consensus algorithm, to develop a secure and reliable e-voting system.

Blockchain technology functions as a decentralized and immutable ledger that records transactions in a transparent and tamper-resistant manner. The proof of work consensus algorithm, commonly associated with public blockchains, provides an additional layer of security by requiring computational work from participants to validate and add transactions to the blockchain [4][7][9]. The advantages of utilizing the proof of work algorithm in e-voting systems include protection against double voting, prevention of unauthorized modifications, and resistance to censorship and tampering. By incorporating the proof of work consensus algorithm, the e-voting system can ensure the integrity of the voting process and uphold the principles of transparency and trust.

For the implementation of smart contracts, we have chosen Solidity as the programming language and the Ethereum blockchain platform. Solidity offers a comprehensive and secure development environment for writing smart contracts, which are self-executing contracts with the terms and conditions directly written into code. The Ethereum blockchain, renowned for its versatility and wide adoption, provides a robust and mature ecosystem for deploying and executing smart contracts in a decentralized manner. These technologies enable the e-voting system to utilize smart contracts to automate and enforce voting rules, facilitate transparent execution, and maintain the integrity of the electoral process [5][11][13].

In our proposed e-voting system, we advocate the use of a private blockchain. A private blockchain restricts access and participation to a specific group of authorized entities, offering enhanced privacy and control over the network. This approach addresses concerns related to scalability, confidentiality, and governance, which are crucial considerations for national-level e-voting systems [10][12]. By implementing a private blockchain, the e-voting system can strike a balance between transparency and confidentiality, ensuring that only authorized participants can engage in the voting process while maintaining the immutability and security provided by the blockchain technology.

In conclusion, the integration of blockchain technology, specifically a private blockchain, along with the use of the proof of work consensus algorithm, offers promising possibilities for improving the security, transparency, and trustworthiness of e-voting systems. By employing Solidity and the Ethereum blockchain platform, our proposed e-voting system leverages mature and robust technologies for the implementation of smart contracts. The utilization of a private blockchain enhances privacy and governance, addressing critical considerations in the design of national-level e-voting systems. Through this research, we aim to contribute to the growing body of knowledge in the field of blockchain-based e-voting systems and pave the way for more secure and reliable democratic processes.

II. BACKGROUND

(1)System Overview

Smart contracts are self-executing contracts with predefined rules and conditions written in code. In the context of a blockchain-based e-voting system, smart contracts automate and enforce various aspects of the voting process. They enable secure and transparent interactions between voters, candidates, and the voting system itself. Smart contracts handle tasks such as voter registration, candidate validation, vote casting, and tallying. By executing these tasks automatically and transparently, smart contracts ensure the integrity and immutability of the voting process [5][10].

(2)Consensus Algorithm

In the context of an e-voting system, proof-of-work serves as a mechanism to verify the authenticity and validity of each vote. Voters' transactions are subject to a computationally intensive process where they must provide proof of their work to validate their vote. This algorithm prevents malicious activities, such as double-spending or tampering with the voting records, as it requires significant computational power and effort to manipulate the blockchain. Proof-of-work enhances the security and trustworthiness of the e-voting system [3][11].

(3)Voter Side Interface

The voter side interface is the user interface that allows eligible voters to interact with the e-voting system. It provides an intuitive and user-friendly platform for voters to register, verify their eligibility, and cast their votes [5][13]. The voter side interface ensures a smooth and accessible voting experience for individuals, promoting their active participation in the electoral process. It may include features such as identity verification, candidate information, voting instructions, and real-time feedback to enhance transparency and trust.

(4)Admin Side Interface

The admin side interface is designed for election administrators who have the authority to manage and oversee the e-voting system. Administrators can access this interface to initiate and conclude elections, verify and validate candidates, and monitor the overall integrity of the system. The admin side interface empowers administrators to maintain transparency, fairness, and security throughout the electoral process. It provides features for managing voter registration, candidate verification, result tracking, and auditing of the voting system [5][13].

(5)Hashing

Hashing is a cryptographic technique used to convert data into fixed-length alphanumeric strings. In the context of an e-voting system, hashing is employed to secure and protect voter information, candidate details, and voting records [12]. Each piece of data is transformed into a unique hash value, which serves as a digital fingerprint. Hashing ensures the integrity and immutability of the data stored on the blockchain. Any changes to the data will result in a different hash value, making it easy to detect tampering or unauthorized modifications. Hashing plays a crucial role in maintaining the integrity and authenticity of the e-voting system.

III. RELATED WORK

Several research papers have contributed to the understanding and development of blockchain-based e-voting systems. In the paper titled "Blockchain-based E-Voting System" [1] by Pathak, Mrunal, Amol Suradkar, Ajinkya Kadam, Akansha Ghodeswar, and Prashant Parde, various approaches for implementing e-voting systems are explored, including user verification methods such as public key and private key cryptography and the utilization of Aadhar database.

Another paper by Al-Maaitah, Sarah, Mohammad Qatawneh, and Abdullah Quzmar titled "E-Voting System Based on Blockchain Technology: A Survey" [2] presents a new framework for using blockchain in an e-voting system. The framework utilizes Ethereum and smart contracts, implementing the Truffle framework for testing and checking smart contracts. The Meta-mask is used to connect with Ethereum nodes, and the framework is designed with three components: the creator, the register, and the voting process.

In the paper "A Framework to Make Voting System Transparent Using Blockchain Technology" [3] by Muhammad Shoaib Farooq, Usman Iftikhar, and Adel Khelifi, a framework is proposed for conducting voting activities digitally through blockchain without physical polling stations. The paper discusses the security aspects of the blockchain-based voting system, including the encryption of transactions using cryptographic hash and the prevention of a 51% attack on the blockchain.

Khan, Kashif Mehboob, Junaid Arshad, and Muhammad Mubashir Khan propose a secure digital voting system based on blockchain technology in their paper [4]. The paper describes an e-voting scheme that meets fundamental requirements and ensures end-to-end verifiability. The implementation of the proposed scheme using the Multichain platform is presented, emphasizing the advantages of using blockchain for e-voting purposes.

In the paper "Secure E-Voting System using Blockchain technology and authentication via Face recognition and Mobile OTP" [5], the authors emphasize the role of blockchain in protecting votes and ensuring the integrity of the electronic voting system. They propose a strategy that incorporates cryptographic techniques and the integration of Aadhar data to prevent vote duplication and modification. The use of biometric details and virtual identification numbers enhances the security and reliability of the voting process.

Finally, in paper [6], the author explores electronic voting frameworks based on blockchain technology. The paper provides a conceptual understanding of blockchain-based electronic voting applications and discusses the key challenges in existing systems. It highlights the potential of blockchain to address issues in voting systems, including privacy, security, and transaction speed. A sustainable blockchain-based electronic voting framework's security for remote participation and the requirement for transaction speed are also covered.

IV. PROPOSED SYSTEM

(1)System Overview

The proposed e-voting system consists of two primary entities: the User and the System Administrator. The System Administrator plays a crucial role in creating and managing the election process. They are responsible for setting the start and end time of the election, as well as registering voters and candidates for the election.

Upon creating the election, the System Administrator registers the eligible voters and candidates in the system. The registration details of voters, including their personal information and eligibility status, are stored securely in the system's database. These details serve as a means of authentication during the voting process, ensuring that only eligible voters can cast their votes.

When the election is active, voters access the system and cast their votes electronically. The system validates the authenticity of each voter based on their registered details. Once authenticated, the voter can submit their vote, which is then securely saved on the blockchain network. The blockchain ensures the immutability and transparency of the votes, making it resistant to tampering or alteration.

Throughout the election process, the system maintains a secure database to store the registration details and authentication information of voters. This database serves as a reference point for validating the eligibility of voters during the voting phase.

After the voting period concludes, the system generates the election results. These results are displayed on the system's frontend, providing an accessible and transparent representation of the outcome. The frontend interface may include graphical representations and statistical data to present the results comprehensively to users.

In summary, the proposed e-voting system allows the System Administrator to create and manage elections, register voters and candidates, and set the election timeframe. Voters can securely cast their votes, with the system verifying their eligibility through the stored registration details. The votes are stored on the blockchain, ensuring transparency and integrity. Finally, the system generates and presents the election results on the frontend interface, providing an accessible overview of the outcome.

(2)System Architecture

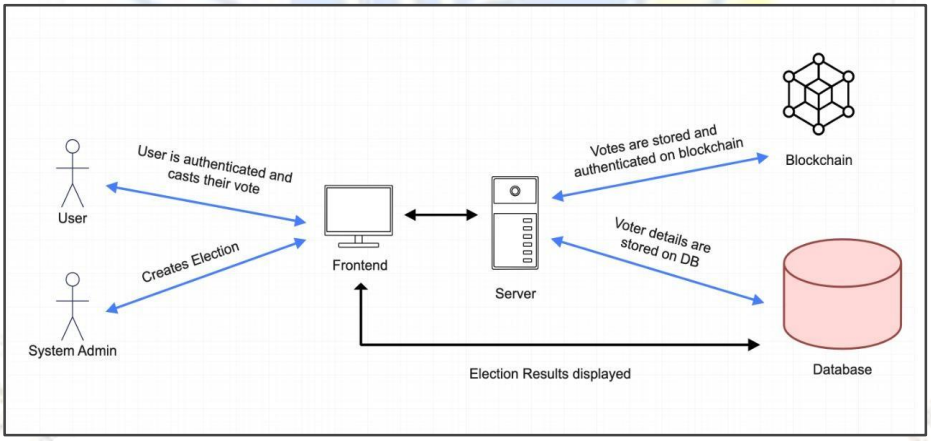

Fig.1 System Architecture

- The system architecture diagram depicts two primary entities: User and System Administrator.
- The System Administrator assumes the responsibility of creating the election and defining its start and end time.
- Additionally, the System Administrator undertakes the task of registering the voters and candidates for the election.
- Voters participate in the election by casting their votes through the system, which securely saves the votes on the blockchain.
- The registration details of voters, serving as an authentication mechanism, are stored in the database.
- These details validate the eligibility of voters during the voting process.
- The election results are displayed on the frontend, providing a transparent overview of the outcome.

(3)Blockchain Integration

The decentralized voting system can be implemented using the Ethereum blockchain and Ganache using Solidity. Ethereum is a decentralized open-source blockchain system that features smart contract functionality. Self-executing contracts known as "smart contracts" can be used to automate transactions and are kept on the blockchain. Ganache is a local blockchain development environment that can be used to test Ethereum applications.

To implement the decentralized voting system using Ethereum blockchain and Ganache using Solidity, the following steps have been followed:

- Created a smart contract that represents the voting system. The smart contract has the following functions:
- A function to add candidates to the election
- A function to verify voters' identities
- A function to allow voters to vote for candidates
- A function to count the votes and declare the winner of the election
- Compiled the smart contract and deployed it to the Ethereum blockchain using Ganache.
- Created a user interface that allows voters to interact with the smart contract. The user interface allows voters to view the list of candidates, verify their identities, and vote for candidates.
- Tested the voting system by running it on Ganache.

(4)Pseudocode of the Smart Contract

```
struct Candidate {
    uint256 candidateId;
    string header;
    string slogan;
    uint256 voteCount;
}
mapping(uint256 => Candidate) public candidateDetails;

function addCandidate(string memory _header, string memory _slogan)
    public
    onlyAdmin
{
    Candidate memory newCandidate =
        Candidate({
            candidateId: candidateCount,
            header: _header,
            slogan: _slogan,
            voteCount: 0
        });
    candidateDetails[candidateCount] = newCandidate;
    candidateCount += 1;
}
```

Fig.2 Candidate Contract Pseudocode

```
struct Voter {
    address voterAddress;
    string name;
    string phone;
    bool isVerified;
    bool hasVoted;
    bool isRegistered;
}
address[] public voters;
mapping(address => Voter) public voterDetails;

function registerAsVoter(string memory _name, string memory _phone) public {
    Voter memory newVoter =
        Voter({
            voterAddress: msg.sender,
            name: _name,
            phone: _phone,
            hasVoted: false,
            isVerified: false,
            isRegistered: true
        });
    voterDetails[msg.sender] = newVoter;
    voters.push(msg.sender);
    voterCount += 1;
}

function verifyVoter(bool _verifedStatus, address voterAddress)
    public
    onlyAdmin
{
    voterDetails[voterAddress].isVerified = _verifedStatus;
}
```

Fig.3 Voter Contract Pseudocode

(5)Use of ML for Authentication

In our proposed e-voting system, we are incorporating machine learning (ML) models for enhanced voter authentication. Specifically, we are utilizing ML algorithms for biometric and identity verification purposes. One of the key techniques we are employing is called "one-shot learning."

One-shot learning is a branch of machine learning that focuses on training models to recognize and classify objects or individuals based on a single example or a small number of examples. Traditional machine learning approaches often require a large dataset for training, which may not always be feasible in scenarios where unique individuals need to be authenticated, such as in the case of voter verification.

By leveraging one-shot learning techniques, our system can effectively authenticate voters based on a single biometric sample or a limited number of samples. This approach is particularly valuable for voter authentication, as it allows for accurate identification even with a minimal amount of training data.

To implement this, we would train the ML models using a diverse set of biometric data, such as facial images or fingerprint patterns, to capture the unique characteristics of individual voters. The models would be trained to recognize these characteristics and verify the identity of voters based on their biometric inputs.

By utilizing machine learning models for voter authentication, we aim to enhance the security and accuracy of the e-voting system. This approach ensures that only eligible voters can participate in the electoral process, minimizing the risk of fraudulent activities and maintaining the integrity of the voting system.

(6)Innovation

Innovations in our research paper encompass several key areas that enhance the security and privacy of the e-voting system. Firstly, we introduce a biometric authentication layer to strengthen the identity verification process. By incorporating biometric data, such as facial recognition or fingerprint patterns, we add an additional layer of security to ensure that only legitimate voters can participate in the electoral process. Biometric authentication provides a higher level of confidence in verifying the identity of voters, reducing the risk of impersonation and fraudulent activities.

Furthermore, we propose the utilization of the proof of work algorithm, a fundamental component of blockchain technology, to enhance the security and integrity of the e-voting system. The proof of work algorithm requires participants, known as miners, to solve complex computational puzzles in order to add new blocks to the blockchain. This consensus mechanism ensures that each transaction and vote recorded on the blockchain undergoes a rigorous validation process, making it extremely difficult for malicious actors to tamper with the voting data. By leveraging the proof of work algorithm, our e-voting system establishes a decentralized and secure environment, fostering trust and transparency in the electoral process.

Additionally, we emphasize the use of a private blockchain and the SHA-256 encryption standard to safeguard the personally identifiable information (PII) data of the voters. A private blockchain restricts access to authorized participants, ensuring that sensitive voter information remains confidential and protected from unauthorized access. Furthermore, the adoption of the SHA-256 encryption standard enhances data security by transforming PII data into a unique hash value that is virtually impossible to reverse-engineer. This encryption standard provides a robust layer of protection for voter data, mitigating the risks associated with data breaches and unauthorized data manipulation.

By combining the biometric authentication layer, proof of work algorithm, private blockchain, and SHA-256 encryption standard, our research paper introduces a comprehensive set of innovations that address the security and privacy challenges in e-voting systems. These advancements contribute to the establishment of a trusted and resilient platform, ensuring the integrity of the electoral process and fostering confidence among voters.

V. IMPLEMENTATION

(1)System Configuration

AddCandidate.js: This file is responsible for providing functionality to add candidates to the blockchain network. It likely includes functions to validate and store candidate information, such as their name, party affiliation, and other relevant details. The file interacts with the smart contract and uses appropriate methods to add candidate data to the blockchain, ensuring its immutability and transparency.

Verification.js: This file handles the verification process for voters before they are allowed to participate in the election. It may include functions to validate the voter's identity, such as checking their credentials, verifying their eligibility to vote, and ensuring they are registered in the system. The verification process might involve interacting with external identity verification systems or databases to authenticate the voter's information.

Registration.js: This file manages the registration process of voters. It provides functionality for capturing and storing voter information, such as their name, address, and other relevant details. It may include validation checks to ensure the accuracy and completeness of the provided information. Additionally, it could handle the generation and management of unique voter IDs or registration tokens to prevent duplicate registrations.

Result.js: This file is responsible for retrieving and displaying the election results once the voting process is completed. It interacts with the smart contract to fetch the vote count for each candidate and calculate the final results. It may also include functions to aggregate and present the results in a user-friendly format, such as generating charts, graphs, or reports.

Voting.js: The Voting.js file plays a crucial role in a decentralized voting system by allowing registered voters to cast their votes in an ongoing election. It provides a user interface where voters can select their preferred candidate(s) and submit their vote. The file interacts with the smart contract to record the votes securely on the blockchain. It may also include functions for handling any constraints or rules specific to the voting process, such as preventing multiple votes from the same voter or enforcing voting deadlines.

getWeb3.js: This file establishes the connection between the client-side application and the blockchain network. It utilizes the web3.js library, which provides a set of APIs for interacting with the Ethereum blockchain. The getWeb3.js file initializes and configures the web3 instance, allowing other components of the application to communicate with the blockchain. It may include functions for connecting to a local or remote Ethereum node, managing accounts, and handling network-related events.

App.js: This file acts as the central container for all other components of the decentralized voting system. It manages the overall flow and organization of the application. It may include routing logic, state management, and integration of various components to create a cohesive user experience. The App.js file is responsible for rendering and coordinating the different screens or views of the application, ensuring proper navigation and interaction between components.

Election.sol: The Election.sol file is a smart contract written in Solidity, the programming language for Ethereum smart contracts. It defines the data structures and functions required to create, manage, and close elections. The smart contract stores information about the candidates, voters, and the vote count for each candidate. It includes functions for candidate registration, voter registration, vote casting, and result calculation. The Election.sol contract ensures the integrity and transparency of the voting process by utilizing the blockchain's immutable and decentralized nature.

Migrations.sol: The Migrations.sol file is a contract used by the Truffle framework to keep track of the contract migrations to the blockchain network. It helps manage the deployment and upgrade process of smart contracts. The file contains a simple contract that stores the current state of each migration, ensuring that contracts are deployed in the correct order and allowing for seamless upgrades or changes to the smart contract system.

truffle-config.js: This file is a configuration file used by the Truffle framework. It provides the necessary settings and parameters for deploying smart contracts and interacting with the Ethereum blockchain. It specifies the network configurations, such as the network provider URL, account details, gas limits, and deployment destinations. Developers can customize this file to match their specific network setup and requirements.

Electronic voting systems have gained popularity in recent years due to the convenience and efficiency they offer. However, the security of these systems is always a concern. To address this issue, blockchain technology is being used to build secure and transparent e-voting systems. One of the most popular blockchain consensus algorithms is Proof of Work (PoW)

VI. PROOF OF WORK ALGORITHM

Blockchain networks employ the consensus mechanism known as Proof of Work to verify transactions and add new blocks. To add new blocks to the blockchain, participants—known as miners—must solve challenging mathematical puzzles. The block is added to the blockchain and the first miner to fix the issue is rewarded with newly created cryptocurrency coins. [8].

(1)Working

The PoW algorithm operates based on a cryptographic hash function, often represented as H, which receives an input and generates the hash value, a fixed-size output. In the case of PoW, the input consists of the block data and the nonce value. The goal is to find a nonce that, when combined with the block data, results in a hash value meeting certain criteria. Let's denote the block data as "data," the nonce as "n," and the target condition as "T." The formula for the Proof of Work algorithm can be expressed as $H(data, n) < T$. This inequality represents the condition that the resulting hash value must be less than the target condition to be considered a valid solution. The target condition is typically defined by requiring the hash value to have a specific number of leading zeros. Miners iterate through different nonce values, repeatedly hashing the block data with each nonce until they find a nonce value that satisfies the target condition [8]. This computational process ensures that miners expend a significant amount of computational resources to secure the blockchain network and deter malicious activities.

(2)Use in Voting Systems

When a voter submits their vote, it is first transformed into a hash value using a cryptographic hash function such as SHA-256. This hashed vote is then included in the transaction recorded on the blockchain. By storing the hashed vote instead of the actual vote data, the privacy and anonymity of voters are preserved, as the original votes cannot be traced back to specific individuals.The miners in the PoW consensus algorithm validate these transactions, including the hashed votes, by solving the mathematical problem. Once the transactions are validated, they are added to a new block. This block, containing the hashed votes and other validated transactions, is then added to the blockchain, creating a permanent and transparent record of the vote.

By combining the PoW consensus algorithm with hashing techniques for votes, e-voting systems can achieve a high level of security, immutability, and transparency [2][11]. The decentralized nature of blockchain ensures that no single entity can manipulate or tamper with the recorded.

VII. PROOF OF CONCEPT

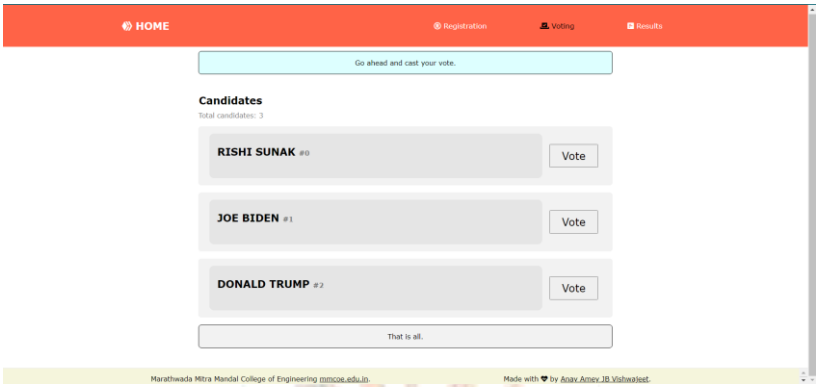

Fig.4 Voting Interface

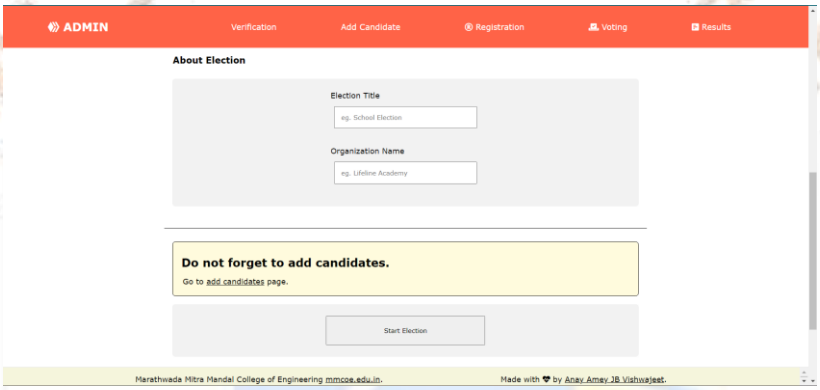

Fig.5 Admin Interface

VIII. CONCLUSION

In conclusion, this research paper presents a comprehensive e-voting system that integrates cutting-edge technologies to address the security, privacy, and authentication challenges associated with modern electoral processes. Through the incorporation of innovations such as biometric authentication, the use of the PoW algorithm and private blockchain, and the implementation of the SHA-256 encryption standard, we have achieved a robust and trustworthy e-voting framework.

By introducing a biometric authentication layer, our system enhances the verification process, ensuring that only legitimate voters can participate in elections. The utilization of biometric data adds an additional layer of security and reduces the risk of impersonation or fraudulent activities, thereby reinforcing the integrity of the electoral process.

The integration of the PoW algorithm and private blockchain technology strengthens the security and transparency of the e-voting system. The proof of work algorithm guarantees the validation of transactions and votes recorded on the blockchain, making it extremely difficult for malicious actors to tamper with the data. The use of a private blockchain restricts access to authorized participants, safeguarding the confidentiality of voter information and preventing unauthorized manipulation.

Furthermore, the adoption of the SHA-256 encryption standard ensures the protection of personally identifiable information by transforming it into irreversible hash values. This encryption standard adds an additional layer of data security, mitigating the risks associated with data breaches and unauthorized access.

By combining these innovations, our e-voting system provides a reliable and resilient platform for conducting secure and transparent elections. The implementation of advanced technologies not only addresses existing concerns but also instills trust and confidence among voters, fostering a stronger democratic process.

Overall, this research paper contributes to the advancement of e-voting systems, offering a viable solution that enhances security, privacy, and authentication, paving the way for more efficient and reliable electoral processes in the digital age.

IX. REFERENCES

[1] Pathak, Mrunal, Amol Suradkar, Ajinkya Kadam, Akansha Ghodeswar, and Prashant Parde. "Blockchain Based E-Voting System." (2021).

[2] Al-Maaitah, Sarah & Qatawneh, Mohammad & Quzmar, Abdullah. (2021). E-Voting System Based on Blockchain Technology: A Survey. 200-205. 10.1109/ICIT52682.2021.9491734.

[3] M. S. Farooq, U. Iftikhar and A. Khelifi, "A Framework to Make Voting System Transparent Using Blockchain Technology," in IEEE Access, vol. 10, pp. 59959-59969, 2022, doi: 10.1109/ACCESS.2022.3180168.

[4] Mehboob, Kashif & Arshad, Junaid & Khan, Muhammad. (2018). Secure Digital Voting System Based on Blockchain Technology. International Journal of Electronic Government Research. 14. 53-62. 10.4018/IJEGR.2018010103.

[5] A. Parmar, S. Gada, T. Loke, Y. Jain, S. Pathak and S. Patil, "Secure E-Voting System using Blockchain technology and authentication via Face recognition and Mobile OTP," 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kharagpur, India, 2021, pp. 1-5, doi: 10.1109/ICCCNT51525.2021.9580147.

[6] T M, Roopak and Ramakrishnan Sumathi. "Electronic Voting based on Virtual ID of Aadhar using Blockchain Technology." 2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA) (2020): 71-75.

[7] Yi, H. Securing e-voting based on blockchain in P2P network. J Wireless Com Network 2019, 137 (2019). https://doi.org/10.1186/s13638-019-1473-6

[8] Xiao, S., Wang, X.A., Wang, W., Wang, H. (2020). Survey on Blockchain-Based Electronic Voting. In: Barolli, L., Nishino, H., Miwa, H. (eds) Advances in Intelligent Networking and Collaborative Systems. INCoS 2019. Advances in Intelligent Systems and Computing, vol 1035. Springer, Cham. https://doi.org/10.1007/978-3-030-29035-1_54

[9] Tanwar, S., Gupta, N., Kumar, P. et al. Implementation of blockchain-based e-voting system. Multimed Tools Appl (2023). https://doi.org/10.1007/s11042-023-15401-1

[10] Hamilton M. Blockchain distributed ledger technology: An introduction and focus on smart contracts. Journal of Corporate Accounting & Finance. 2020 Apr;31(2):7-12.

[11] Taş R, Tanrıöver ÖÖ. A systematic review of challenges and opportunities of blockchain for E-voting. Symmetry. 2020 Aug 9;12(8):1328.

[12] Alvi ST, Uddin MN, Islam L. Digital voting: A blockchain-based e-voting system using biohash and smart contract. In2020 third international conference on smart systems and inventive technology (ICSSIT) 2020 Aug 20 (pp. 228-233). IEEE.

[13] Baudier P, Kondrateva G, Ammi C, Seulliet E. Peace engineering: The contribution of blockchain systems to the e-voting process. Technological Forecasting and Social Change. 2021 Jan 1;162:120397.

[14] Abuidris Y, Kumar R, Yang T, Onginjo J. Secure large-scale E-voting system based on blockchain contract using a hybrid consensus model combined with sharding. Etri Journal. 2021 Apr;43(2):357-70.

[15] Vo-Cao-Thuy L, Cao-Minh K, Dang-Le-Bao C, Nguyen TA. Votereum: An ethereum-based e-voting system. In2019 IEEE-RIVF International Conference on Computing and Communication Technologies (RIVF) 2019 Mar 20 (pp. 1-6). IEEE.