Received: 22 March 2024 | Revised: 25 June 2024 | Accepted: 26 July 2024
DOI: 10.1111/exsy.13694

ORIGINAL ARTICLE

Expert Systems WILEY

DemocracyGuard: Blockchain-based secure voting framework for digital democracy

Mritunjay Shall Peelam | Gaurav Kumar | Kunjan Shah | Vinay Chamola

Department of Electrical and Electronics Engineering, BITS-Pilani, Pilani, India

Correspondence
Mritunjay Shall Peelam, Department of Electrical and Electronics Engineering, BITS-Pilani, Pilani 333031, India.
Email: mritunjay.peelam@pilani.bits-pilani.ac.in

Abstract

Online voting is gaining traction in contemporary society to reduce costs and boost voter turnout, allowing individuals to cast their ballots from anywhere with an internet connection. This innovation is cautiously met due to the inherent security risks, where a single vulnerability can lead to widespread vote manipulation. Blockchain technology has emerged as a promising solution to address these concerns and create a trustworthy electoral process. Blockchain offers a decentralized network of nodes that enhances transparency, security, and verifiability. Its distributed ledger and non-repudiation features make it a compelling alternative to traditional electronic voting systems, ensuring the integrity of elections. To further bolster the security of online voting, we propose *DemocracyGuard* platform on the Ethereum blockchain, which incorporates facial recognition technology to authenticate voters. By leveraging these advancements, *DemocracyGuard* aims to provide a secure and resilient platform for online voting, paving the way for its broader adoption and revolutionizing the electoral landscape.

KEYWORDS
blockchain, decentralized, digital democracy, elections, electronic voting, Ethereum

1 | INTRODUCTION

Conventional voting systems have been designed to uphold the essential tenets of democratic elections and referendums. These principles encompass safeguarding the right to vote, ensuring ballot secrecy, preserving the integrity of voters' intentions, and preventing intimidation or coercion during the voting process. Conversely, e-voting refers to electronic systems for casting and tallying votes in electoral processes (AboSamra et al., 2017). Voting is a widespread practice ingrained in diverse societies in various forms. Nevertheless, peer voting stands apart from conventional voting procedures, such as presidential elections. Peer voting primarily occurs in an online environment as opposed to traditional physical ballot casting, which consequently presents unique challenges (Zhang et al., 2018). People gradually recognize the electoral system's significance as more votes are cast in real-life elections. Currently, most voting systems are centralized, encompassing mixnet-based voting, blind signatures, and homomorphic encryption technology. These systems involve the central agency recording, managing, calculating, and verifying the votes. Nevertheless, it is essential to assume the existence of a trustworthy bulletin board and corresponding credible vote-counting authorities. The reliance on single central institutions and the handling of extensive data pose vulnerabilities to the security of electronic voting (Wang et al., 2018). Voting in India has been a contentious issue for many years, from the initial implementation of the Balloting System during the 1951-52 General Elections to the more recent adoption of 'Electronic Voting Machines' in 1998. Under the balloting system shown in Figure 1, voters cast their votes using pre-printed ballot papers under the supervision of a voting official. These physical ballots were then collected and transported to a centralized vote-counting centre. This method had its shortcomings, which were subsequently addressed by

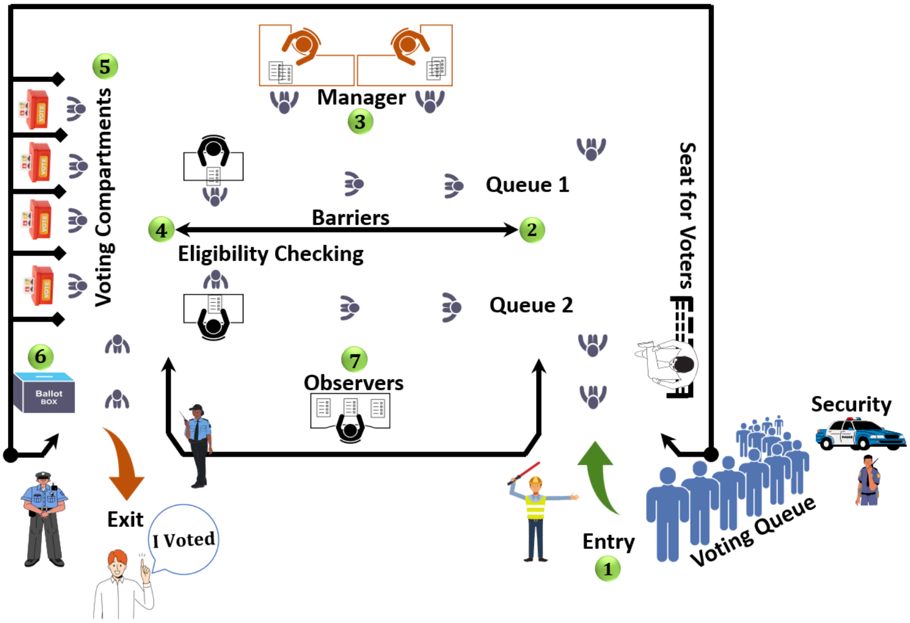


FIGURE 1 Ballot box memories: The way voting was conducted in the past in India.

transitioning to an electronic voting system. This updated approach records votes on electronic balloting devices, transfers them to a central location, and tabulates them using a control unit. While electronic voting machines (EVMs) are touted as tamper-proof, concerns persisted regarding the need for oversight during the voting process and allegations of political party interference in their favour, highlighting two prominent challenges associated with the system. These issues underscore the persistent problems of the system's reliance on an authority to monitor the voting process and accusations of political party influence to support their cause. (Sharma et al., 2022). Even in the world's most prominent democracies, such as India and the United States, the electoral systems still grapple with imperfections. Notable concerns within the current voting process include vote tampering, EVM hacking, election manipulation, and the capture of polling booths (Benny, 2020).

1.1 | EVM-based voting in democracy

A simple electronic device, the EVM, has replaced paper ballots and voting boxes in modern elections. Because they are not easily duplicated, stolen, or shared, biometric identifiers are considered more trustworthy for individual identification than conventional tokens or knowledge-based techniques (Kumar & Begum, 2013; Xuemin et al., 2024). In 1977, the Chief Election Commissioner advocated using EVMs. The Election Commission of India worked with two primary businesses, Bharat Electronics Limited (BEL) of Bangalore and Electronics Corporation of India Limited (ECIL) of Hyderabad, on the creation and design of EVMs. Three main parts make up an EVM, as shown in Figure 2.

1. Balloting Unit (BU)
2. Control Unit (CU)
3. Voter Verifiable Paper Audit Trail (VVPAT)

A cable of five metres joins these two parts together. The Balloting Unit is kept safely within the voting compartment, while the Presiding Officer or a Polling Officer holds the Control Unit (Prasad et al., 2016). In many countries, like India, where they are widely utilized, VVPATs are stored in the voting compartment. Using a paper copy of their electronic vote cast on an EVM, voters may validate their vote using the VVPAT method. After casting their ballot, voters are given a few seconds to look at the printed results to ensure accuracy (VVPAT, n.d.; Ma & Hu, 2022).

1.2 | Blockchain-based voting in democracy

Integrate blockchain technology with Cryptographic Hash Functions and Digital Signatures to establish a decentralized electronic voting system that fulfils all the voting process requirements without relying on a trusted third-party. This e-voting protocol explores blockchain as transparent

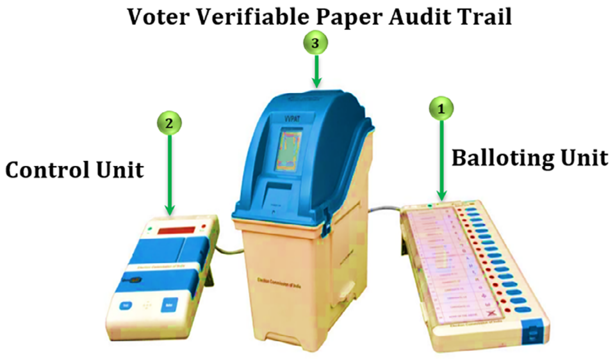


FIGURE 2 A close-up of an electronic voting machine with its Control Unit (CU), VVPAT, and Ballot Unit (BU).

ballot boxes connected through cryptographic methods. It is implemented as a smart contract running on the Ethereum network and utilizes Node.js to create nodes for each user. These nodes store encrypted vote details in individual blocks, ensuring a transparent and resilient system suitable for medium-sized elections (Akshay & Arun, 2019). Blockchain technology is applied in information sharing (Liu, Han, et al., 2024; Yang et al., 2023) and has recently emerged as a transformative solution, augmenting the efficiency of systems across various domains. Initially conceived for tracking cryptocurrency transactions, its applications have expanded considerably in recent years. Notably, blockchain-based e-voting systems have become a robust solution to address challenges inherent in electronic voting. These systems are poised to revolutionize modern electronic voting by exploring the blockchain's immutable nature to establish a decentralized, distributed ballot box. By incorporating sustainability information into voting systems, blockchain encourages governments to embrace intelligent and sustainable voting practices, ensuring that all participants access dependable data on sustainable assets. Acknowledging that several challenges persist despite the increasing adoption of blockchain for electronic voting security enhancement is crucial (Taş & Tanrıöver, 2020).

Numerous online voting systems have been developed utilizing blockchain technology to prevent ballot tampering. These existing systems can be broadly classified into two categories. The first category involves systems with a tallying authority. Despite leveraging the tamper-resistant nature of blockchain to record votes, these schemes still depend on a centralized authority, like a tallying authority, to decrypt the encrypted ballots and calculate the election results. Consequently, other entities cannot verify the accuracy of the voting results as the authority's secret key remains confidential. In contrast, the second category comprises self-tallying systems, which treat the tallying algorithm as a transparent process. This allows all entities, including voters and candidates, to verify all ballots and obtain the final election results (Yin et al., n.d.; Yang et al., 2021). Blockchain technology provides a decentralized online voting and electronic balloting framework. Distributed ledger technologies, like blockchain, have been increasingly leveraged to develop electronic voting systems, primarily due to their end-to-end verification capabilities. Blockchain presents an attractive alternative to traditional electronic voting systems, boasting decentralization, non-repudiation, and robust security measures. It finds utility in corporate boardroom decisions and public voting processes (Jafar et al., 2021; Sun et al., 2018). In the age of digital advancements, the traditional methods of conducting elections face numerous challenges that threaten the integrity and fairness of the democratic process, as shown in Table 1. The need for a secure and transparent voting framework has become increasingly urgent. With the proliferation of technology and the growing concern over election interference and fraud, it has become imperative to develop a robust and trustworthy voting system that can safeguard the principles of democracy. This problem statement lays the foundation for developing *DemocracyGuard*. A robust blockchain-based voting system must prioritize security, decentralization, and transparency. Utilizing strong cryptographic algorithms, a distributed ledger, and a reliable consensus mechanism (Liu, Zhao, et al., 2024; Qi et al., 2024) ensures the integrity and immutability of the voting process. Secure identity verification methods and maintaining voter anonymity are essential for building trust. The system should be user-friendly, accessible, and scalable to accommodate many transactions. Auditability through transparent processes, timestamping, and open-source code enhances accountability. The integration of smart contracts automates key aspects of the voting process. The inclusion of *DemocracyGuard*, as detailed in Table 2, further secures the system, adding an extra layer of security, privacy, and adherence to legal and regulatory requirements, ensuring a resilient, fair, and democratic electoral experience.

1.3 | Research motivation and novelty

DemocracyGuard is driven by the increasing recognition of the essential role that electoral systems play in democracies and the persistent challenges they face, such as vote tampering, EVM hacking, and election manipulation. The development of *DemocracyGuard* is motivated by the desire to address these challenges by implementing a blockchain-based system that enhances security, transparency, and verifiability in the voting

TABLE 1 Comparison of centralized and blockchain-based voting systems (*DemocracyGuard*).

Disadvantages of centralized voting	Improvements with *DemocracyGuard*
Lack of accessibility (Weiss et al., 2022)	Enhanced accessibility for all
Limited voting hours (Jafar et al., 2021)	Extended voting timeframes
Long lines at polling stations (Cooley et al., 2018)	Reduced wait times
Voter suppression (Weiss et al., 2022)	Reduced risk of suppression
Difficulty for disabled voters (Jafar et al., 2021; Kho et al., 2022; Weiss et al., 2022)	Improved accessibility for disabled
Limited voting locations (Akshay & Arun, 2019; Alvi et al., 2022)	Increased voting venues
Potential for voter intimidation (AboSamra et al., 2017; Vladucu et al., 2023)	Enhanced voter privacy
Inconsistent ballot design (Bhadoria et al., 2022; Wahab et al., 2022)	Standardized ballot format
Paper ballot errors (Herrnson et al., 2012)	Reduced human errors
Possibility of lost ballots (Leemann & Bochsler, 2014; Oppliger, 2002)	Immutable ballot records
Voter misidentification (Shi et al., 2015)	Enhanced voter verification
Potential for ballot tampering (Wallach, 2020)	Secure and transparent system
Lack of transparency (Riera & Brown, 2003)	Enhanced transparency
Inefficient voter registration (Kasdan, 2013)	Streamlined registration process
Difficulty for out-of-state voters (Roberts, 2016)	Simplified out-of-state voting
Lack of verifiable results (Hao & Ryan, 2016)	Enhanced results verification
Miscounted votes (He & Su, 1998)	Reduced risk of vote miscount

TABLE 2 Blockchain-based voting system requirements with *DemocracyGuard* integration.

Requirement	Description
Blockchain security	
• Immutable ledger	Ensure the integrity of the voting records through an immutable blockchain ledger
• Cryptographic hashing	Implement cryptographic hashing for secure and tamper-evident data
• Decentralization	Utilize decentralized blockchain technology to prevent a single point of failure
• Smart contract integration	Incorporate smart contracts for automated and transparent execution of voting rules
Voter privacy	
• Anonymous transactions	Enable anonymous transactions to protect voter privacy
• Confidentiality	Ensure confidential handling of voter data through encryption techniques
Accessibility and usability	
• User-friendly interface	Design an intuitive and user-friendly interface for all voters
• Inclusive design	Ensure the system is accessible to voters with disabilities
• Multilingual support	Provide support for multiple languages
Transparent verification	
• Public verification	Allow public verification of the voting results on the blockchain
• Voter verification	Enable voters to verify that their votes are correctly recorded on the blockchain
Resilience and redundancy	
• Distributed storage	Use distributed storage for redundancy and resilience against data loss
• Fault tolerance	Implement fault-tolerant mechanisms to ensure continuous operation
Scalability	
• Scalable architecture	Design the system with scalability to handle a growing number of voters
• Efficient consensus mechanism	Employ an efficient consensus mechanism to handle a large number of transactions
Regulatory compliance	
• Legal framework	Adhere to legal and regulatory frameworks governing elections
• Standards compliance	Ensure compliance with industry standards for blockchain technology
Cybersecurity measures	
• DDoS protection	Implement measures to mitigate the risk of DDoS attacks
• Encryption	Use advanced encryption techniques to secure communication and data

process. The integration of facial recognition technology for voter authentication further aims to support the security of online voting, addressing inherent security risks and promoting the adoption of this technology for a more secure and resilient electoral process.

The Novelty of this research lies in the combination of blockchain technology with facial recognition to authenticate voters, a feature that sets DemocracyGuard apart from existing voting systems. Blockchain ensures a decentralized, tamper-proof system where votes are recorded as immutable transactions, enhancing trust in the electoral process. Incorporating facial recognition technology for voter authentication represents a significant advancement in ensuring the integrity of the voting process. This innovative approach to combining blockchain with biometric verification aims to revolutionize online voting, making it more secure, accessible, and efficient. There are several contributions listed below.

1. *Blockchain and Facial Recognition Integration*: The novel integration of blockchain technology with facial recognition for voter authentication distinguishes DemocracyGuard from existing solutions.
2. *Decentralized, Tamper-proof System*: Utilizing a decentralized network of nodes, the framework provides a tamper-proof system where votes are recorded as immutable transactions, enhancing the integrity of elections.
3. *Enhanced Voter Verification*: The innovative use of facial recognition technology offers a more reliable method of voter authentication compared to traditional tokens or knowledge-based techniques.
4. *Transparent Electoral Process*: DemocracyGuard introduces a transparent electoral process, with every transaction being verifiable and recorded on the blockchain, ensuring that all participants have access to reliable data.
5. *Accessibility and Inclusivity*: The framework's design focuses on making voting more accessible and inclusive, catering to a wide range of voters with different needs and capabilities.
6. *Advancement Towards Digital Democracy*: By addressing key challenges of traditional and electronic voting systems, DemocracyGuard represents a significant advancement towards realizing a secure, efficient, and transparent digital democracy.

1.4 | Salient contribution

DemocracyGuard contributes to the advancement of digital democracy, making electoral processes more transparent, verifiable, and resilient against fraud. The following are the contributions of DemocracyGuard.

1. This paper presents a novel approach to combine blockchain technology with facial recognition to authenticate voters, which is both secure and efficient.
2. Distributed ledger technology inherent to blockchains, the system ensures that records cannot be altered after they have been logged, promoting a tamper-proof electoral environment.
3. The system introduces an enhanced method for voter verification that surpasses traditional means, providing a more reliable verification process.
4. It offers transparency in the electoral process, allowing all participants to verify the procedures and outcomes, thus enhancing trust in the system.
5. DemocracyGuard is designed to be accessible to all eligible voters, regardless of location or mobility, promoting inclusivity in the electoral process.
6. DemocracyGuard represents a significant step forward in the use of digital technologies to conduct democratic processes, moving towards a more modernized and efficient model of governance.

1.5 | Organization of *DemocracyGuard*

Figure 3 shows the organization of DemocracyGuard, which outlines the use of EVMs-Based Voting in Democracy and introduces a blockchain-based voting system. Section 1 introduces the concept and covers the research motivation, novelty, and salient contributions, including integrating blockchain and facial recognition for a decentralized, tamper-proof system. Section 2 systematically analyses and synthesizes the existing research and discussions on electronic voting systems. Section 3 presents the 'Proposed System Model', detailing the voter's registration process using Azure Face API and admin-controlled Know Your Customer (KYC) for registration security. Section 4 describes the 'Methodology' employed, including transparent voter authentication and candidate list verification, as well as a secure vote verification algorithm. The 'Results' in Section 5 showcase case studies and implementation data. Section 6, 'Conclusion and Future Work', offers an analysis of data and a comparison with traditional voting systems. Table 3 compares frequently utilized standard terms and their respective acronyms within the DemocracyGuard.

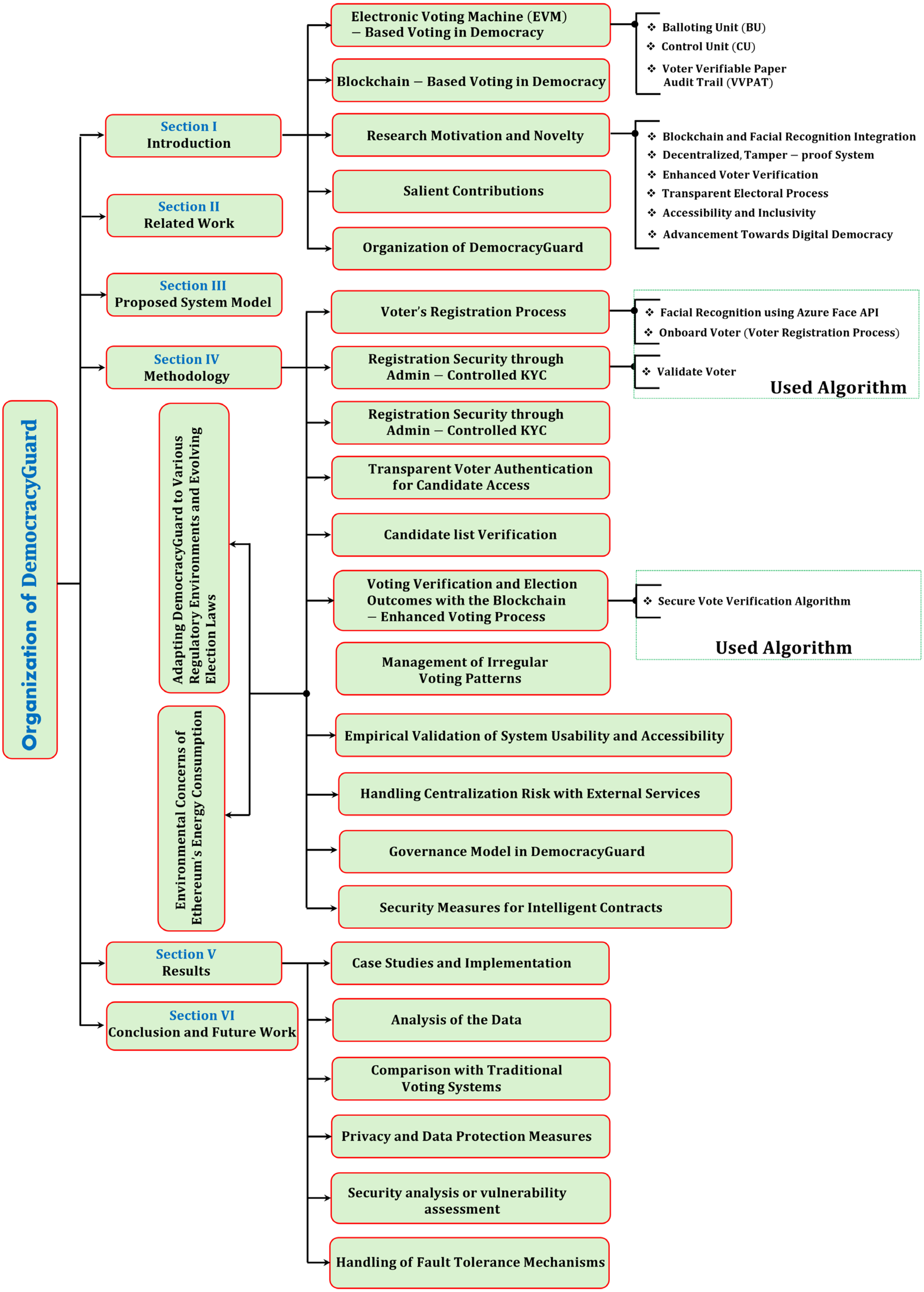

Organization of DemocracyGuard
Section I
Introduction
Section II
Related Work
Section III
Proposed System Model
Section IV
Methodology
Section V
Results
Section VI
Conclusion and Future Work
Electronic Voting Machine (EVM) – Based Voting in Democracy
Blockchain – Based Voting in Democracy
Research Motivation and Novelty
Salient Contributions
Organization of DemocracyGuard
❖ Balloting Unit (BU)
❖ Control Unit (CU)
❖ Voter Verifiable Paper Audit Trail (VVPAT)
❖ Blockchain and Facial Recognition Integration
❖ Decentralized, Tamper – proof System
❖ Enhanced Voter Verification
❖ Transparent Electoral Process
❖ Accessibility and Inclusivity
❖ Advancement Towards Digital Democracy
Adapting DemocracyGuard to Various Regulatory Environments and Evolving Election Laws
Environmental Concerns of Ethereum's Energy Consumption
Voter's Registration Process
Registration Security through Admin – Controlled KYC
Registration Security through Admin – Controlled KYC
Transparent Voter Authentication for Candidate Access
Candidate list Verification
Voting Verification and Election Outcomes with the Blockchain – Enhanced Voting Process
Management of Irregular Voting Patterns
❖ Facial Recognition using Azure Face API
❖ Onboard Voter (Voter Registration Process)
❖ Validate Voter
Used Algorithm
❖ Secure Vote Verification Algorithm
Used Algorithm
Empirical Validation of System Usability and Accessibility
Handling Centralization Risk with External Services
Governance Model in DemocracyGuard
Security Measures for Intelligent Contracts
Case Studies and Implementation
Analysis of the Data
Comparison with Traditional Voting Systems
Privacy and Data Protection Measures
Security analysis or vulnerability assessment
Handling of Fault Tolerance Mechanisms

FIGURE 3 Comprehensive layout of *DemocracyGuard* outlining the structure and flow of content.

TABLE 3 Comparison of abbreviations and terms used in DemocracyGuard.

Abbreviation	Term used in DemocracyGuard
EVMs	Electronic Voting Machines
BEL	Bharat Electronics Limited
ECIL	Electronics Corporation of India Limited
BU	Balloting Unit
CU	Control Unit
VVPAT	Voter Verifiable Paper Audit Trail
KYC	Know Your Customer
SUS	System Usability Scale
PoW	Proof of Work
PoS	Proof of Stake
PBFT	Practical Byzantine Fault Tolerance
PoET	Proof of Elapsed Time
LPoS	Liquid Proof-of-Stake
DPoS	Delegated Proof-of-Stake
FBA	Federated Byzantine Agreement
dBFT	Delegated Byzantine Fault Tolerance
PPoS	Pure Proof-of-Stake
NPoS	Nominated Proof-of-Stake
OPOS	Ouroboros Proof-of-Stake
PoSA	Proof of Stake Authority
PoH	Proof of History
IBFT	Istanbul Byzantine Fault Tolerance
TBD	To Be Determined
AI	Artificial Intelligence
EVS	Electronic Voting Systems
UI	User Interface
SC	Smart Contract
ERC	Ethereum Requests for Comments
SLA	Service Level Agreements
GDPR	General Data Protection Regulation
CCPA	Central Consumer Protection Authority
RBAC	Role-Based Access Control
SOC	System and Organization Controls
BFT	Byzantine Fault Tolerance
DDOS	Distributed Denial of Service

2 | LITERATURE REVIEW

In the realm of voting systems, blockchain technology, developed over the past decade, has emerged as a game-changer. It offers a secure, transparent, and tamper-proof method for casting and counting votes. Blockchain ensures the integrity of the process by recording votes as unchangeable transactions, increasing participation, and enhancing trust in elections. It is a significant step in modernizing and securing the democratic process. Table 4 represents a comparative analysis of blockchain frameworks crucial for protecting digital democracy. It highlights their consensus mechanisms, generation times, accessibility, transaction rates, scalability, and transaction costs. *DemocracyGuard* can use this information to select the most suitable blockchain framework to enhance the security, efficiency, and inclusivity of digital voting and decision-making processes. Ashok Kumar et al. compares three fingerprint-matching methods using EVMs for election accuracy and time efficiency. (Kumar & Begum, 2013). Election integrity depends on fair procedures, but fraud can occur in various ways. Leemann et al. propose a method for fraud detection, addressing multiple forms and instances. Using a Swiss referendum case, we apply statistical tests revealing irregularities in some municipalities that lost

TABLE 4 Comparative analysis of blockchain-based frameworks for digital democracy.

Blockchain framework	Consensus mechanism	Generation time	Accessibility	Transaction rate	Scalability	Transaction cost (USD)
Ethereum (Vo-Cao-Thuy et al., 2019)	Proof of Work (PoW) transitioning to Proof of Stake (PoS)	15 s	Public	30 TPS	Moderate to high	$0.50
Hyperledger Fabric (Mukherjee et al., 2020)	Practical Byzantine Fault Tolerance (PBFT)	3–5 s	Permissioned	1000+ TPS	Moderate to high	Free
Hyperledger Sawtooth (Vivek et al., 2020)	Proof of Elapsed Time (PoET)	5–10 s	Permissioned	Scalable	High	$0.80
Corda (Benji & Sindhu, 2019; Jani, 2020)	Notary (Permissioned, no global consensus)	10–30 s	Permissioned	Customizable	High	$1.00
Tezos (Cortier et al., 2021)	Liquid Proof-of-Stake (LPoS)	1 min	Public	40 TPS	To be determined (TBD)	$0.30
EOS (Amoah & Oh, 2021)	Delegated Proof-of-Stake (DPoS)	0.5 s	Public	4000+ TPS	High	$0.20
Stellar (Barański et al., 2020)	Federated Byzantine Agreement (FBA)	2–5 s	Public	1000+ TPS	High	$0.40
NEO (Coelho et al., 2019)	Delegated Byzantine Fault Tolerance (dBFT)	15 s	Public	1000+ TPS	High	$0.60
TRON (Yadav et al., 2021)	Delegated Proof-of-Stake (DPoS)	3 s	Public	2000+ TPS	High	$0.25
Algorand (Esposito & Choi, 2023)	Pure Proof-of-Stake (PPoS)	4.5 s	Public	1000 TPS	High	$0.70
Avalanche (Sapák, n.d.)	Avalanche Consensus	1–2 s	Public	4500+ TPS	High	$0.90
Polkadot (Mutuku, 2023)	Nominated Proof-of-Stake (NPoS)	6 s	Public	1000 TPS (per parachain)	High	$1.10
Cardano (Lamela Seijas et al., 2020)	Ouroboros Proof-of-Stake	20 s	Public	1000+ TPS (ongoing optimization)	High	$0.75
Binance Smart Chain (BSC) (Duguleanǎ & Gîrbacia, 2021)	Proof of Stake Authority (PoSA)	3 s	Public	100 TPS	Moderate to high	$0.35
Solana (Yakovenko, 2018)	Proof of History (PoH) + Proof of Stake (PoS)	400 ms	Public	65,000+ TPS	Very high	$0.05
Flow (Hentschel et al., 2019)	Flow Consensus (Proof of Stake)	1 s	Public	1000+ TPS (initially)	Scalable	$0.60
Bitcoin (Vranken, 2017)	Proof of Work (PoW)	10 min	Public	7 TPS	Limited (by design)	$2.00
Exonum (Yanovich et al., 2018)	Proof of Stake (Exonum Consensus)	3–5 s	Permissioned	2000+ TPS	High	$1.30
Quorum (Baliga et al., 2018)	Istanbul Byzantine Fault Tolerance (IBFT)	15 s	Permissioned	100–200+ TPS	Moderate to high	$1.20
ZCash (Akcora et al., 2022)	Proof of Work (Equihash)	2.5 min	Public	20–30 TPS (approx.)	Moderate to high	$0.95

ballots. Managing multiple tests presents challenges, and we discuss two strategies with their strengths and weaknesses (Leemann & Bochsler, 2014). In 2018, Hjalmarsson et al. explored using blockchain for distributed electronic voting. It introduces a novel blockchain-based voting system that overcomes limitations in current systems. Various blockchain frameworks are evaluated to construct this system, focusing on distributed ledger technologies. A case study outlines the election process, showing how a blockchain application enhances security and reduces nationwide election costs (Hjálmarsson et al., 2018). In 2019, Yi et al. discussed the application of blockchain in a peer-to-peer network to enhance the security of electronic voting (e-voting). They introduce models for voting records, user credentials, and vote withdrawal to create a practical and secure e-voting system that addresses forgery concerns, utilizing distributed ledger technology and elliptic curve cryptography (Yi, 2019). In 2021, Kamil et al. addressed the rising concerns related to the COVID-19 pandemic and its impact on public safety and elections. The author proposed a solution in the form of a blockchain-based E-voting system, allowing remote voting through electronic devices. This innovative approach enhances security, minimizes data fraud, and provides real-time access to decentralized voting results. Kamil's research, using the System Usability Scale (SUS), yielded a high score of 90, indicating the system's acceptability and positive impact on effectiveness and efficiency during the pandemic (Kamil et al., 2021). In 2021, Yang et al. examined the significance of elections in democracies and the cryptographic challenges of E-voting. Their research introduced PriScore, a blockchain-based self-tallying election system ensuring privacy in score voting. The system employs a dual zero-knowledge proof technique to satisfy range and sum constraints, delivering fairness, dispute resolution, and robust security (Yang et al., 2021). In 2021, Jafar et al. explored the growing trend of online voting in modern society. They acknowledged its potential to reduce costs and boost voter participation. However, security concerns led them to investigate blockchain technology as a solution. Their research provided an overview of blockchain-based electronic voting systems, highlighting the need for improved privacy and transaction speed to ensure the sustainability of such systems (Jafar et al., 2021). In 2022, Farooq et al. addressed the widespread mistrust in traditional voting systems, acknowledging the violations of fundamental rights and the lack of transparency in existing digital voting systems. They identified the vulnerability of these systems to exploitation and aimed to rectify these issues. Their research proposed a blockchain-based platform to ensure election fairness, fostering trust between voters and election authorities. This framework enables digital voting without physical polling stations, supported by scalable blockchain and robust security measures, including the Chain Security Algorithm and smart contracts (Farooq et al., 2022). In 2022, Bhadoria et al. addressed the paramount significance of democratic elections and governmental efforts to enhance their competitiveness and equity. Their paper explored the adoption of blockchain technology in election processes, utilizing a distributed digital ledger to record transactions securely. This technology ensures transparency and confidentiality by employing encryption algorithms and tamper-proof data storage (Bhadoria et al., 2022). In 2022, Alvi et al. explored the significance of voting in democratic societies and the limitations of paper balloting, which is prone to errors and abuse. Their research introduced a blockchain-based voting system, ensuring anonymity, privacy, and integrity. Implemented on Ethereum 2.0, the system employs smart contracts to enhance security and reduce infrastructure costs (Alvi et al., 2022). In 2023, Vladucu et al. conducted a study emphasizing the increasing global adoption of electronic voting systems for public office elections. These systems offer benefits such as remote voting and expedited tallying while enhancing privacy and reducing voting bias. Blockchain technology fortifies the process by ensuring immutable vote storage, thwarting tampering, and safeguarding the legitimacy of elections. Countries like Germany, Russia, Estonia, and Switzerland have integrated blockchain into their e-voting systems (Vladucu et al., 2023). In 2023, Neloy et al. conducted a study highlighting the limitations of traditional voting methods, which lack remote access, are time-consuming, and suffer from security issues. Electronic voting systems (EVSs), while improving efficiency, raise concerns regarding security, legitimacy, and transparency. To address these challenges, the researchers utilized blockchain technology, incorporating smart contracts and artificial intelligence (AI) to develop a remote voting system that enhances transparency, decentralization, and security (Neloy et al., 2023).

3 | PROPOSED SYSTEM MODEL

The proposed system model, shown in Figure 4 begins with voter registration, where eligible voters are registered. Each registered voter is assigned a unique identifier (a voter ID). Subsequently, the voter image capture process captures biometric data for enhanced security. Voter QR Code Generation then creates a QR code linked to the voter's information. For voter image verification, the system utilizes Azure's facial recognition API, ensuring the voter's identity through biometric matching. In the Voting Party List phase, the voter selects their preferred candidate or party, represented as Voting Party List P_1,P_2...P_n. The selection triggers the execution of the *Voting.sol* smart contract on the Ethereum Blockchain, securely recording the voter's choice, initiating the transaction, and writing to the Ethereum Blockchain B_1,B_2...B_n, guaranteeing transparency and immutability. The system offers a result panel to display the election outcomes for the parties P_1,P_2...P_n, assuring a fair and secure electoral process. We have selected the Ethereum blockchain over other options, such as Hyperledger, IOTA, and so forth, for *DemocracyGuard*. Our decision is based on a combination of factors that align closely with the requirements and goals of *DemocracyGuard*. We have given the key reasons for selecting Ethereum:

1. *Maturity and Robustness*: Ethereum is one of the most mature and widely used blockchain platforms, with a large and active developer community (Metcalfe et al., 2020). This maturity ensures a stable and robust environment, critical for a system like *DemocracyGuard*, which requires high reliability and security.
2. *Smart Contract Functionality*: Ethereum's Turing-complete smart contract capability is unparalleled in the blockchain ecosystem (Buterin, 2022). This feature allows for complex logic and automation, essential for implementing the various functionalities of *DemocracyGuard*, such as voting, verification, and governance.
3. *Decentralization and Security*: Ethereum operates on a decentralized network with a strong focus on security, backed by its proof-of-work (PoW) consensus mechanism [soon to be proof-of-stake (PoS) with Ethereum 2.0] (Asif & Hassan, 2023). This decentralization ensures that no single entity can control the network, aligning with the democratic principles of DemocracyGuard.
4. *Interoperability and Standards*: Ethereum support several standards (such as ERC-20 and ERC-721) that facilitate interoperability with a wide range of decentralized applications (dApps) and services (Di Angelo & Salzer, 2023). This flexibility is advantageous for future integration and expansions of the *DemocracyGuard* platform.

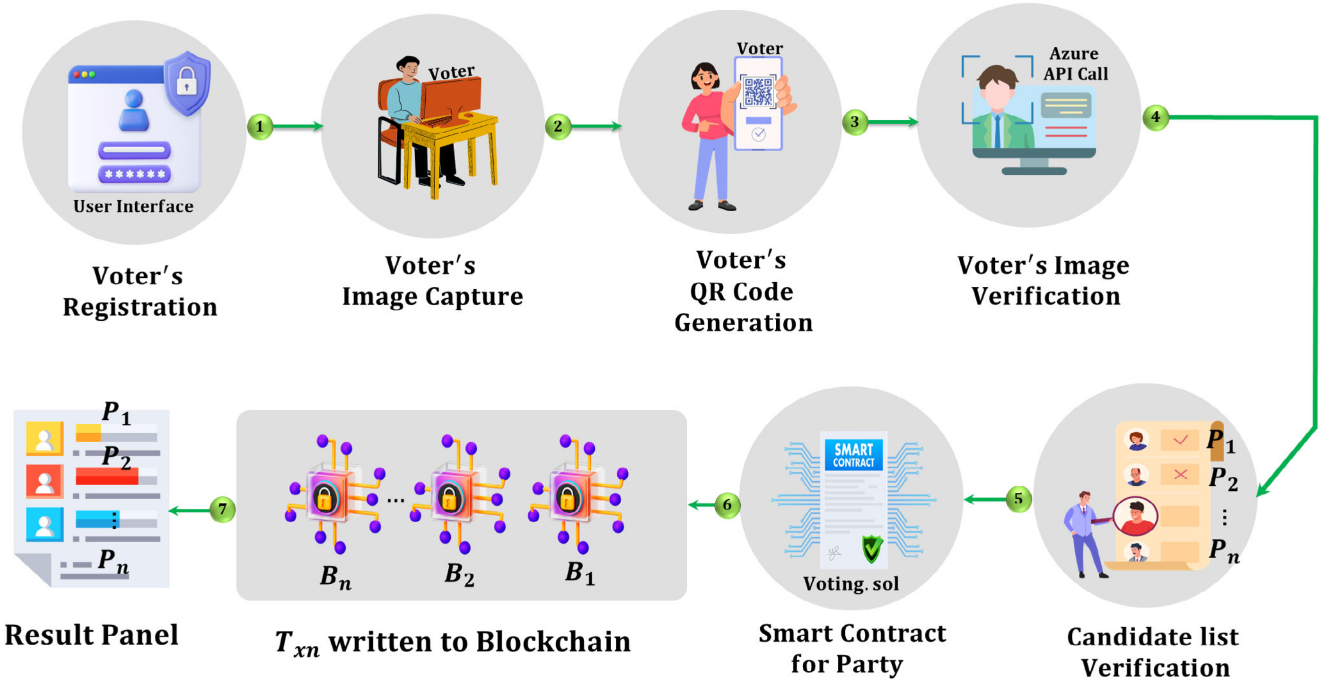


FIGURE 4 DemocracyGuard's blockchain smart contract-based architecture: ensuring trust from voter registration to transparent election results.

5. *Active Ecosystem and Tools*: Ethereum ecosystem includes numerous tools, libraries, and frameworks that accelerate development and deployment (Mukhopadhyay, 2018). This rich ecosystem allows for rapid prototyping and testing, which is beneficial for research and iterative improvements.

4 | METHODOLOGY

4.1 | Voter's registration process

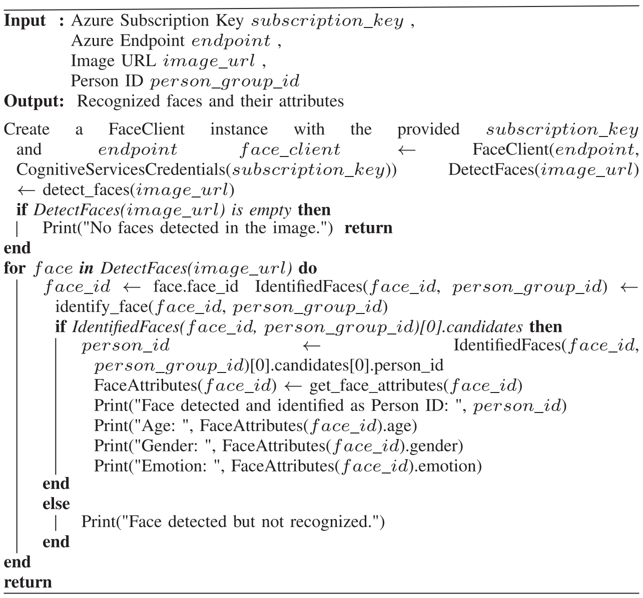
The registration phase commences with the Voter User Interface (UI) page, where voters input their details, such as their name (*N*), Aadhaar number (*A*), and constituency (*C*). Subsequently, the voter is prompted to capture a photograph (*P*). This photograph is then transmitted to the Azure Face API, which, through advanced facial recognition algorithms as shown in Algorithm 1, uniquely identifies the individual in the photo, resulting in a photo identifier string, *U*. Facial recognition has been chosen over other biometric and non-biometric methods due to its high accuracy, ease of use, and non-intrusive nature (Rodwell et al., 2007), as shown in Table 5. The collected data is encoded into a QR code, denoted as $QR(N,A,C,U)$. Two essential Boolean variables, *isValid* and *hasVoted*, are initialized as follows:

$$isValid = \begin{cases} \text{True} & \text{if thevoter}'\text{sdetailsarevalidated,} \\ \text{False} & \text{otherwise.} \end{cases} \tag{1}$$

$$hasVoted = \begin{cases} \text{True} & \text{if the voter has participated} \\ & \text{in the electoral process,} \\ \text{False} & \text{otherwise.} \end{cases} \tag{2}$$

These variables are crucial for tracking the voter's eligibility and participation in the electoral process, and during the registration phase, these variables are *False*. The logic ensures that voting status is accurately recorded. The complete voter registration process is shown in Figure 5, and Algorithm 2 shows the complete registration process of voters.

Algorithm 1 **Facial Recognition using Azure Face API**



```
Input  : Azure Subscription Key $subscription\_key$ ,
         Azure Endpoint $endpoint$ ,
         Image URL $image\_url$ ,
         Person ID $person\_group\_id$
Output: Recognized faces and their attributes
Create a FaceClient instance with the provided $subscription\_key$
 and $endpoint$ $face\_client \leftarrow$ FaceClient($endpoint$,
 CognitiveServicesCredentials($subscription\_key$)) DetectFaces($image\_url$)
 $\leftarrow$ detect_faces($image\_url$)
 if DetectFaces($image\_url$) is empty then
    Print("No faces detected in the image.") return
end
for $face$ in DetectFaces($image\_url$) do
    $face\_id \leftarrow$ face.face_id IdentifiedFaces($face\_id$, $person\_group\_id$) $\leftarrow$
     identify_face($face\_id$, $person\_group\_id$)
     if IdentifiedFaces($face\_id$, $person\_group\_id$)[0].candidates then
        $person\_id \leftarrow$ IdentifiedFaces($face\_id$,
          $person\_group\_id$)[0].candidates[0].person_id
          FaceAttributes($face\_id$) $\leftarrow$ get_face_attributes($face\_id$)
          Print("Face detected and identified as Person ID: ", $person\_id$)
          Print("Age: ", FaceAttributes($face\_id$).age)
          Print("Gender: ", FaceAttributes($face\_id$).gender)
          Print("Emotion: ", FaceAttributes($face\_id$).emotion)
    end
    else
        Print("Face detected but not recognized.")
    end
end
return
```

4.2 | Registration security through admin-controlled KYC procedures

Registration security through admin-controlled KYC procedures establishes a robust and trustworthy voter onboarding mechanism. Upon completion of the registration process, voters receive a QR code, denoted as QR_{code}, initially tagged as $isValid = \text{False}$, signifying that it has yet to undergo verification by the admin through KYC protocols. To bolster security, voters must seek manual KYC verification from the administration, as illustrated in Figure 6. Within the admin panel, a comprehensive scrutiny of the voter's information takes place, with an unwavering commitment to upholding the integrity of voter data. Let $Voter_{\text{info}}$ represent the voter's information. KYC verification can be expressed as follows:

$$KYC\_verification = \begin{cases} \text{True} & \text{if} Admin\_ \\ & verification(Voter_{\text{info}}) \\ \text{False} & \text{otherwise} \end{cases} \tag{3}$$

Elaborate logs, denoted as *Logs*, are meticulously maintained throughout this verification process. Once the administrator successfully validates the voter's information, the *isValid* status is elevated to True, denoted as $isValid = True$ as shown in Algorithm 3, thereby granting the voter access to the subsequent phase, guaranteeing a highly secure and verified voter constituency.

4.3 | Transparent voter authentication for candidate access

Upon completing the KYC verification process, voters are prompted to upload their QR code and undergo a subsequent image capture. Let *V* represent the voter, *QR* represent the QR code, and *I* represent the captured image, and let *A* be the Azure API call, and *D* denote the stored image in the Azure database during the voter's registration process. The image authentication process can be represented as follows:

TABLE 5 Comparative analysis of voter verification methods.

Feature/ criteria	Facial recognition	Fingerprint recognition	Iris recognition	Voter ID cards	PIN/password
Accuracy and reliability	High accuracy with advanced systems	High accuracy but affected by skin issues	Extremely high accuracy	Moderate accuracy, prone to forgery	Moderate accuracy
Ease of use	Simple, non-intrusive	Requires physical contact	Requires close proximity to scanner	Simple, well understood	Simple, familiar
Speed of verification	Fast, instant recognition	Relatively fast	Fast but requires precise alignment	Fast	Fast
Equipment required	Camera	Fingerprint scanner	Specialized iris scanner	Voter ID card readers	No special equipment
Non-contact method	Yes	No	Yes	No	Yes
Integration with systems	Easy integration with Azure Face API	Requires specialized hardware integration	Requires specialized hardware integration	Easy integration with existing systems	Easy integration with existing systems
Privacy and data protection	Secure with Azure compliance, non-storage	Secure but requires data storage	Secure but perceived as invasive	Secure with proper handling, prone to loss	Secure but prone to theft or hacking
User consent	Required, ensures transparency	Required	Required	Implied through possession	Required
Data storage	Temporary, not stored long-term	Stored in database	Stored in database	Physical possession	Stored in database
Susceptibility to fraud	Low security	Low-to-moderate, possible with forged prints	Low	High, possible with fake IDs	High, can be shared or stolen
Health concerns	Minimal, non-contact	High, physical contact	Minimal, non-contact	Minimal, physical handling	None

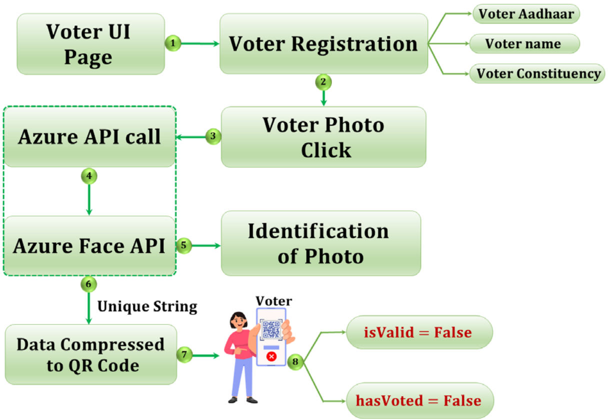


FIGURE 5 Architecture of voter registration process.

$$A(V,QR,I,D)=\begin{cases}\text{True} & \text{if}\, I=D\, \text{and} \\ & V\, \text{is authorized by}\, QR, \\ \text{False} & \text{otherwise}\end{cases} \tag{4}$$

If $A(V,QR,I,D)=True$, it indicates that the voter's captured image matches the stored image, and the QR code is valid, ensuring a secure authentication. Once the image authentication is successfully validated, voters access a comprehensive list of constituent candidates, as shown in Figure 7. This rigorous authentication ensures a transparent and secure electoral experience, empowering voters to make well-informed decisions during the voting process. The combination of KYC verification, QR code validation, and image authentication forms a robust system (S) for maintaining the integrity and security of the electoral process:

Algorithm 2 **Onboard Voter (Voter Registration Process)**

```
Input: name, aadhaar_card, constituency, photo
Output: QR code
  Step 1: Enter name, aadhar_card, constituency
  Step 2: Click face photo
  Step 3: photo_identifier_string ← AZURE_FACE_API(photo)
  Step 4: Store <name, aadhar_card, constituency, photo_identifier_string, $isValid$ = $False$, $hasVoted$ = $False$> to election DB
  Step 5: if $isValid$ == $False$ then
      user_qr_code ← generate_qr_code(name, aadhar_card, constituency, photo_identifier_string)
      return Download user_qr_code to user phone
else
    return QR code generation skipped (already onboarded)
```

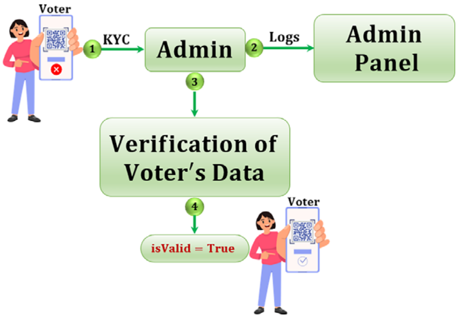


FIGURE 6 Admin-controlled KYC verification process.

$$S(V,QR,I,D)=\begin{cases} True & \text{if} A(V,QR,I,D)=1 \\ False & \text{otherwise} \end{cases} \tag{5}$$

This system S is designed to ensure the transparency and security of voter authentication for accessing the electoral candidate lists, promoting a reliable electoral experience.

4.4 | Candidate list verification

The validation of a candidate, as depicted in Algorithm 4, involves checking the authenticity and eligibility of a candidate, typically by comparing their provided information or identifier against a predefined set of valid candidates.

If the candidate is found in the list of valid candidates, the validation process returns *True*, indicating that the candidate is legitimate. If not, it returns *False*, signifying that the candidate is not authorized or eligible for the process. This validation step is essential for maintaining the security and integrity of systems and ensuring that only valid participants are allowed to proceed, enhancing the reliability and trustworthiness of the system. Let C be the set of valid candidates, C_i be the candidate identifier for the ith candidate in C, and c_{check} be the candidate identifier to be validated. The validation process can be expressed as:

$$c_{\text{check}} \in C \Rightarrow \text{Validation} = \text{True} \tag{6}$$

$$c_{\text{check}} \notin C \Rightarrow \text{Validation} = \text{False} \tag{7}$$

where $c_{\text{check}} \in C$ signifies that the candidate identifier c_{check} belongs to the set of valid candidates C, leading to a 'True' validation outcome, while $c_{\text{check}} \notin C$ indicates that the candidate identifier c_{check} is not within C, resulting in a 'False' validation.

Algorithm 3 **Validate Voter**

```
Input: Aadhaar Card
Output: Validation Status
  Step 1: Admin logs in using admin_username and admin_password
  Step 2: Voter physically goes for KYC and submits Aadhar card
  Step 3: Admin authenticates voter using Aadhar card
if voter details exist in election DB then
    if isValid == False then
        if Aadhar card details are correctly verified and photo matches then
            go to Step 4
        else
            return Authentication failure
    else
        Ask voter to Onboard first
        return Authentication failure
Step 4: Admin authorizes voter
  if $current\_year - birth\_year \geq 18$ then
      isValid = True
      Update voter info in election DB
else
    return Authorization failure
```

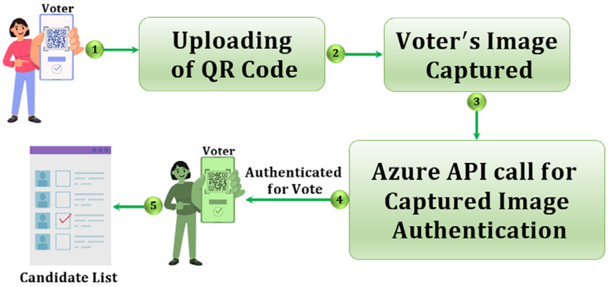


FIGURE 7 Accessing electoral candidate lists after secure authentication.

4.5 | Voting verification and election outcomes with the blockchain-enhanced voting process

In the voting process, a voter receives a list of candidates (*C*) and selects their preferred choice. This initiates a transaction (*T*), where the voter's choice is formally recorded. The voter's selection can be represented as:

$$preferred\_candidate \in C \tag{8}$$

The smart contract (*SC*) processes the transaction and employs a secure vote validation using Algorithm 5 to verify the legitimacy of the chosen candidate. The validation algorithm ensures that the selected candidate is a valid member of the candidate set *C*.

$$\text{Validation}(preferred\_candidate, C) = \{\text{True}\} \tag{9}$$

Upon successful validation, the transaction is permanently recorded on a blockchain (*B*), resulting in a ledger of transactions. As a crucial indicator of successful participation, the *hasVoted* condition variable is *true*, preventing multiple votes from the same voter and confirming their engagement in the election.

$$B = \{T_1, T_2, \ldots, T_n\} \tag{10}$$

Algorithm 4 Validate Candidate

```
Input: Candidate identifier (candidate_id $\in C$), candidate_list ($C$)
Output: Validation Result $\in \{\text{True}, \text{False}\}$
Initialize $i = 0$
while $i < |C|$ do
    if $candidate\_list[i] = candidate\_id$ then
        return Validation Result = True
    $i \leftarrow i + 1$
return Validation Result = False
```

Algorithm 5 Secure Vote Verification Algorithm

```
Input: QR_code, live_photo
Output: Validation Error, Vote For Candidate
    Step 1: On Voting day, submit QR_code on the application
    Step 2: voter_details ← scan(QR_code)
    Step 3: Query election DB using voter_details
    if hasVoted == False then
        if isValid == True then
            Step 4: Click a live_photo
            isCorrectPhoto ← AZURE_FACE_API(live_photo, voter_details.photo_identifier_string)
            if isCorrectPhoto == True then
                Step 5: Display list of candidates for voter_details.constituency
                candidate ← select single preferred candidate
                isValidCandidate ← ValidateCandidate(candidate)
                if isValidCandidate == True then
                    Step 6: UpdateVotesFor(candidate)
                else
                    return Error("Candidate not found")
            else
                return ValidationError("Photo mismatch")
        else
            return ValidationError("Ineligible to vote")
    else
        return ValidationError("Already Voted")
```

The election result is then determined through a mathematical function aggregating and counting the valid votes, providing a clear and verifiable outcome. This can be expressed as:

$$Result = \text{Count\_Valid\_Votes}(B, C) \tag{11}$$

The combination of mathematical verification, secure blockchain technology (*BC*), and transparent result computation enhances the overall integrity and accountability of the electoral process, ensuring that the election outcomes accurately reflect the will of the voters. The election result becomes accessible on the result panel, as shown in Figure 8, offering a comprehensive and easily interpretable display of the outcome. This intricate, technology-driven procedure merges the convenience of *DemocracyGuard* with the robust security and transparency of blockchain technology, enhancing the overall integrity and accountability of the electoral process.

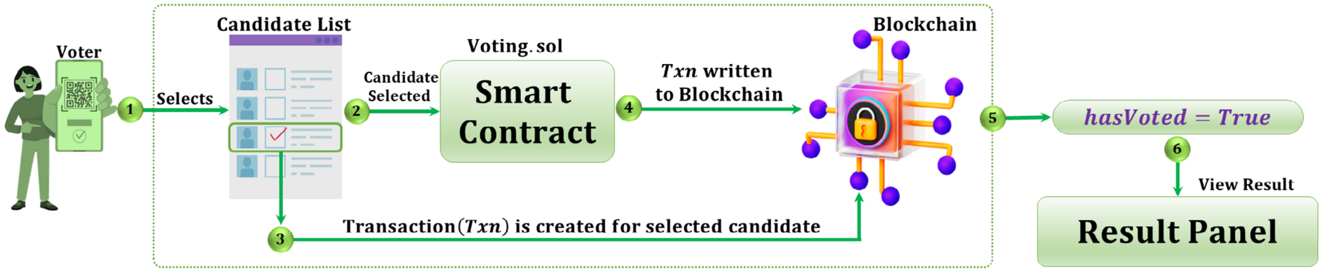


FIGURE 8 Blockchain-enhanced voting process unveils election outcomes.

4.6 | Management of irregular voting patterns

Recognizing the essential role of addressing irregular or anomalous voting patterns, *DemocracyGuard* integrates multiple mechanisms to manage such occurrences effectively, as given below.

1. *Anomaly Detection Algorithms*: We have implemented sophisticated anomaly detection algorithms that analyse voting patterns in real-time. This algorithm detects deviations from expected behaviour, such as sudden spikes in voter turnout or unusual voting patterns, which may indicate fraudulent activity or system errors.
2. *Threshold-based Triggers*: *DemocracyGuard* equipped with threshold-based triggers that are activated when predefined thresholds for irregularities are exceeded. For example, if the number of votes from a particular region surpasses a certain threshold compared to historical data, it triggers further investigation.
3. *Real-time Monitoring*: *DemocracyGuard* have real-time monitoring tools in place to continuously monitor the voting process. This enables us to detect and respond promptly to any irregularities or anomalies as they occur, minimizing the potential impact on the integrity of the voting system.
4. *Manual Verification Processes*: *DemocracyGuard* have established manual verification processes involving election officials or auditors. These individuals are trained to review flagged voting patterns manually and take appropriate action, such as conducting further investigations or initiating corrective measures (Hassija, Zeadally, et al., 2021).
5. *Error Handling Protocols*: *DemocracyGuard* is equipped with robust error handling protocols to address system errors or technical glitches that inadvertently affect the voting process. These protocols include mechanisms for vote reconciliation, data validation, and system recovery to ensure the integrity of the electoral outcome.

4.7 | Empirical validation of system usability and accessibility

We have recognized the need for empirical evidence to support empirical validation of system usability and accessibility. For this, we have conducted comprehensive usability studies involving user surveys, focus groups, and hands-on testing sessions with diverse participants. These studies utilized established usability metrics, such as the SUS and Heuristic Evaluations, to systematically assess the system's ease of use, efficiency, and overall user satisfaction. The results provided robust empirical evidence validating our claims and identifying areas for further improvement, ensuring the system effectively met user needs and expectations. The following shows the empirical validation of the system's usability and accessibility through comprehensive usability studies and their findings.

1. *Comprehensive Usability Testing*: We have incorporated comprehensive usability testing methodologies, including user surveys, focus groups, and hands-on testing sessions, to evaluate the system's user-friendliness and accessibility.
2. *Diverse Participant Involvement*: A diverse group of participants were involved in the usability studies to ensure representation across varying levels of technical expertise and individuals with disabilities.
3. *Use of We have Established Usability Metrics*: We have established usability metrics and frameworks, such as the SUS and heuristic evaluations, were utilized to systematically assess the system's performance in terms of ease of use, efficiency, and overall user satisfaction.
4. *Validation of Claims*: The results from the usability studies provided robust empirical evidence validating the claims regarding the system's user-friendliness and accessibility.

4.8 | Environmental concerns of Ethereum's energy consumption

The environmental concerns associated with Ethereum's high energy consumption were recognized and proactively addressed. By transitioning to Ethereum 2.0's PoS mechanism with lower environmental footprints, DemocracyGuard ensured sustainable and responsible technology use, significantly reducing overall energy usage.

1. *Transition to Ethereum 2.0*: To address the energy efficiency concerns in *DemocracyGuard*, we have transition to Ethereum 2.0, which employs a PoS consensus mechanism. PoS significantly reduces energy consumption by eliminating the need for intensive computational work to validate transactions. This transition is expected to decrease Ethereum's energy usage by over 99%, making it a more sustainable option.
2. *Energy Efficiency Measures*: Energy efficiency was enhanced by optimizing our smart contracts to perform efficiently, reducing unnecessary computational load and energy consumption (Hassija et al., 2019).

4.9 | Adapting *DemocracyGuard* to various regulatory environments and evolving election laws

DemocracyGuard is designed with a robust framework to adapt and handle diverse regulatory environments and changes in election laws. The system incorporates flexibility to accommodate variations in legal requirements across jurisdictions, ensuring compliance and effectiveness. Key features include configurable settings that allow customization based on local regulations, such as voter eligibility criteria and verification methods. Regular updates and consultations with legal experts enable *DemocracyGuard* to promptly integrate new legislative developments, enhancing its responsiveness and suitability for evolving electoral landscapes. *DemocracyGuard* maintains its integrity and usability across different regulatory frameworks by prioritizing adaptability and compliance, supporting fair and transparent elections worldwide.

4.10 | Handling centralization risk with external services

DemocracyGuard addresses the potential risk of centralization associated with external services like Azure Face API through a multifaceted approach. This approach includes utilizing services from providers such as Amazon Rekognition, Google Cloud Vision API, and Face++ to diversify and distribute reliance. Robust redundancy protocols ensure continuous service availability, supplemented by clear service level agreements (SLAs) defining performance and security standards. Strict data protection measures, including encryption and adherence to GDPR standards, are enforced to secure sensitive biometric data. Continuous feedback from users and stakeholders drives ongoing enhancements in system reliability, security protocols, and user satisfaction, reinforcing the resilience and operational integrity of *DemocracyGuard's* facial recognition system for voter verification.

4.11 | Governance model in *DemocracyGuard*

The governance model in *DemocracyGuard* has been designed to facilitate decentralized decision-making processes. By employing a PoS consensus mechanism, all nodes in the network have a voice in validating transactions and creating new blocks. This approach ensures that decisions regarding protocol updates, changes in network rules, and other critical actions are made collectively, preventing centralization of control and promoting fair participation across the network. The comprehensive overview of the governance model is as follows.

1. *Decentralized Decision-Making*: The governance model in *DemocracyGuard* is designed to facilitate decentralized decision-making processes. Network participants have made decisions collectively regarding protocol updates, changes in network rules, and other critical actions.
2. *Consensus Mechanism*: By employing a PoS consensus mechanism, all nodes in the network have a voice in validating transactions and creating new blocks. This approach ensures that decisions regarding protocol updates, changes in network rules, and other critical actions are made collectively, preventing centralization of control and promoting fair participation across the network.

4.12 | Security measures for intelligent contracts

DemocracyGuard has addressed concerns regarding the security of intelligent contracts by implementing specific measures to ensure their accuracy and security (Dang et al., 2023). Comprehensive measures have been taken to solve potential bugs or exploits. These include various testing protocols, code reviews, and implementing best practices in smart contract development. The following points show how to handle these Measures Implemented to Mitigate Potential Bugs or Exploits.

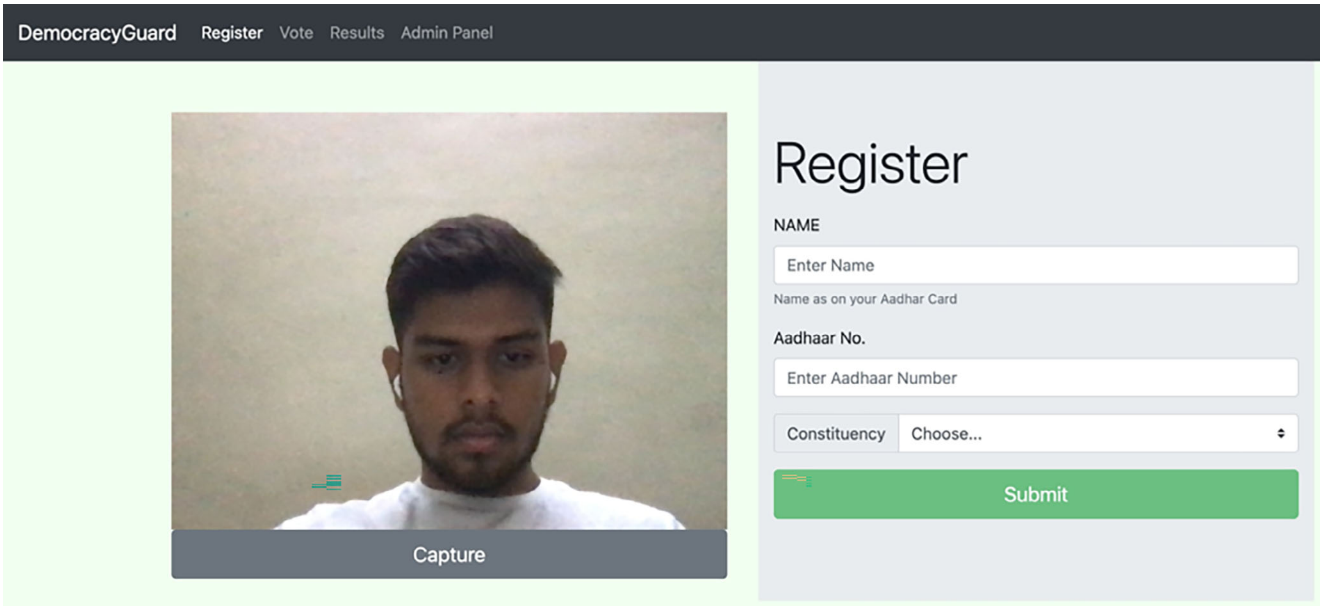
1. *Rigorous Testing and Development Protocols*: *DemocracyGuard* has prioritized the security of intelligent contracts by implementing rigorous testing protocols. Before deployment, all smart contracts undergo extensive testing to identify and rectify potential bugs or vulnerabilities (Hassija, Chamola, & Zeadally, 2020). This process ensures that the contracts function accurately and securely during the voting process.
2. *Vulnerability Management and Patching*: *DemocracyGuard* maintains a robust vulnerability management process. Any identified vulnerabilities are promptly patched, and updates are deployed to ensure that the intelligent contracts remain resilient against emerging threats.

5 | RESULTS AND FINDINGS

DemocracyGuard, the innovative blockchain-based voting framework designed for digital democracy, has undergone rigorous case studies and simulations to evaluate its efficacy and potential impact on modern democratic processes. These assessments provide valuable insights into the strengths and potential challenges associated with implementing *DemocracyGuard* compared to traditional voting systems.

5.1 | Case studies and implementation

The case studies involved simulated elections across diverse scenarios, considering factors such as voter turnout, system resilience to cyber threats, and overall user experience. In each instance, *DemocracyGuard* demonstrated robust performance, ensuring the integrity and security of the voting process. Simulations revealed that the decentralized nature of the blockchain infrastructure significantly reduced the risk of tampering or unauthorized access (Hassija, Bansal, et al., 2020). In Figure 9, DemocracyGuard, a blockchain-based voting framework for digital democracy, is depicted, capturing the essence of democracy through a snapshot from the voters' registration process. This image highlights the initial stages of civic engagement within the innovative and secure platform. Following successful registration, Figure 10 displays a welcome message within DemocracyGuard, greeting voters and emphasizing the user-centric approach of the digital democracy system. Figure 11 then reveals a greeting along with the effortless QR code submission procedure, showcasing the cutting-edge technology integrated into DemocracyGuard for casting votes efficiently and securely. Figure 12 provides a comprehensive visualization during the Cast Your Vote phase, presenting voter details and candidate choices to empower users in making informed decisions. In Figure 13, the administrative panel of DemocracyGuard is featured, unveiling voter and candidate details for rigorous verification, underlining the system's commitment to ensuring the integrity of electoral processes. Figure 14 showcases DemocracyGuard's transparency by revealing election poll results in the Section 5, solidifying its role as a pioneering blockchain-based voting framework for advancing digital democracy.



Register

NAME

Enter Name

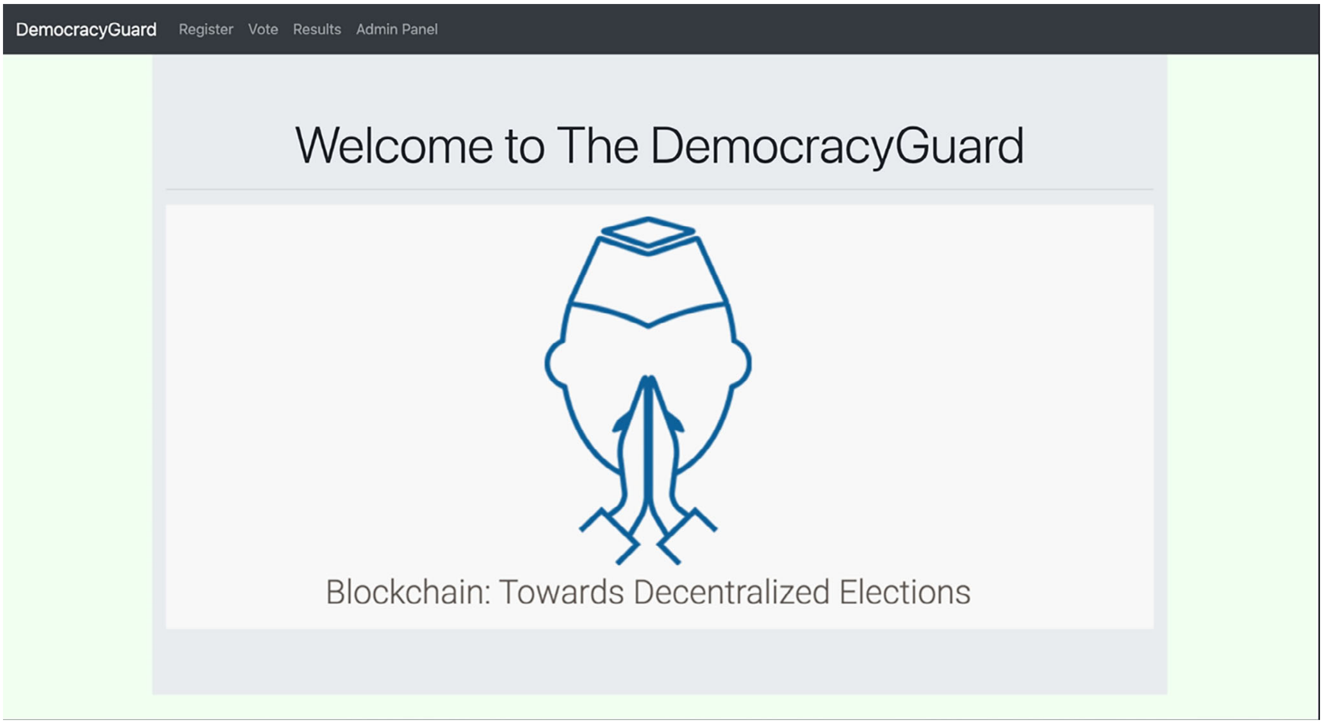
Name as on your Aadhar Card

Aadhaar No.

Enter Aadhaar Number

Constituency Choose...

FIGURE 9 Capturing the essence of democracy: a snapshot from the voters registration process in DemocracyGuard.



Welcome to The DemocracyGuard

Blockchain: Towards Decentralized Elections

FIGURE 10 Welcome message greets voters upon successful registration in DemocracyGuard.

5.2 | Analysis of the data

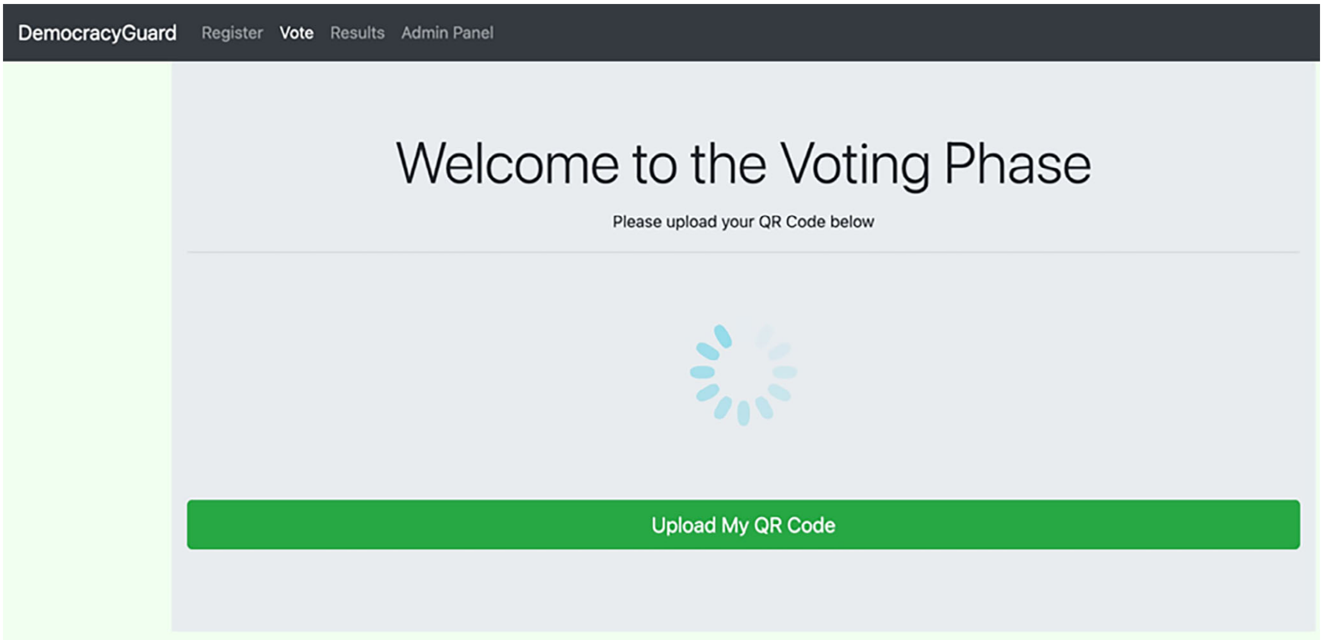
In-depth analysis of the data generated during the case studies highlighted several key findings. The transparency inherent in blockchain technology allowed for real-time tracking of votes, providing a verifiable and immutable record of the electoral process (Hassija, Saxena, & Chamola, 2021). Smart contracts in *DemocracyGuard* streamlined the voting process, minimizing errors and ensuring adherence to predefined rules. Figure 15 shows the Smart Contract Creation within *DemocracyGuard*. This pivotal snapshot encapsulates the intricate steps in transforming predefined rules and conditions into self-executing contracts on a blockchain. As the digital landscape evolves, this visual glimpse into the creation of smart contracts underscores the fusion of technology and governance, ushering in a new era of decentralized and automated decision-making within the *DemocracyGuard* platform. Figure 16 illustrates the cumulative operation time required for casting the vote, offering valuable insights into the efficiency and duration of the voting procedure.

5.3 | Comparison with traditional voting systems

A comparative analysis between *DemocracyGuard* and traditional voting systems revealed distinct advantages for the blockchain-based framework. Traditional systems often face challenges related to centralized vulnerabilities, susceptibility to manipulation, and logistical issues. *DemocracyGuard*, on the other hand, demonstrated superior resilience to tampering, increased accessibility, and a reduced likelihood of errors or disputes. Table 6 reveals key insights into the features of various blockchain-based voting frameworks, with a particular focus on *DemocracyGuard*. Among the evaluated frameworks, *DemocracyGuard* scores consistently high, receiving a positive mark (✓) in every analysed category. Specifically, *DemocracyGuard* excels in providing cost-free voting, biometric verification, a robust blockchain infrastructure, efficient smart contract implementation, enhanced voter turnout mechanisms, secure transaction verification, and an administrator-verified KYC process. This comprehensive approach signifies *DemocracyGuard's* commitment to addressing multiple facets of secure and transparent voting systems.

5.4 | Privacy and data protection measures

There are significant privacy concerns associated with facial recognition technology (Hassija, Batra, et al., 2021), particularly regarding handling and protecting sensitive biometric data. To handle these concerns, *DemocracyGuard* employs several robust privacy and data protection measures:



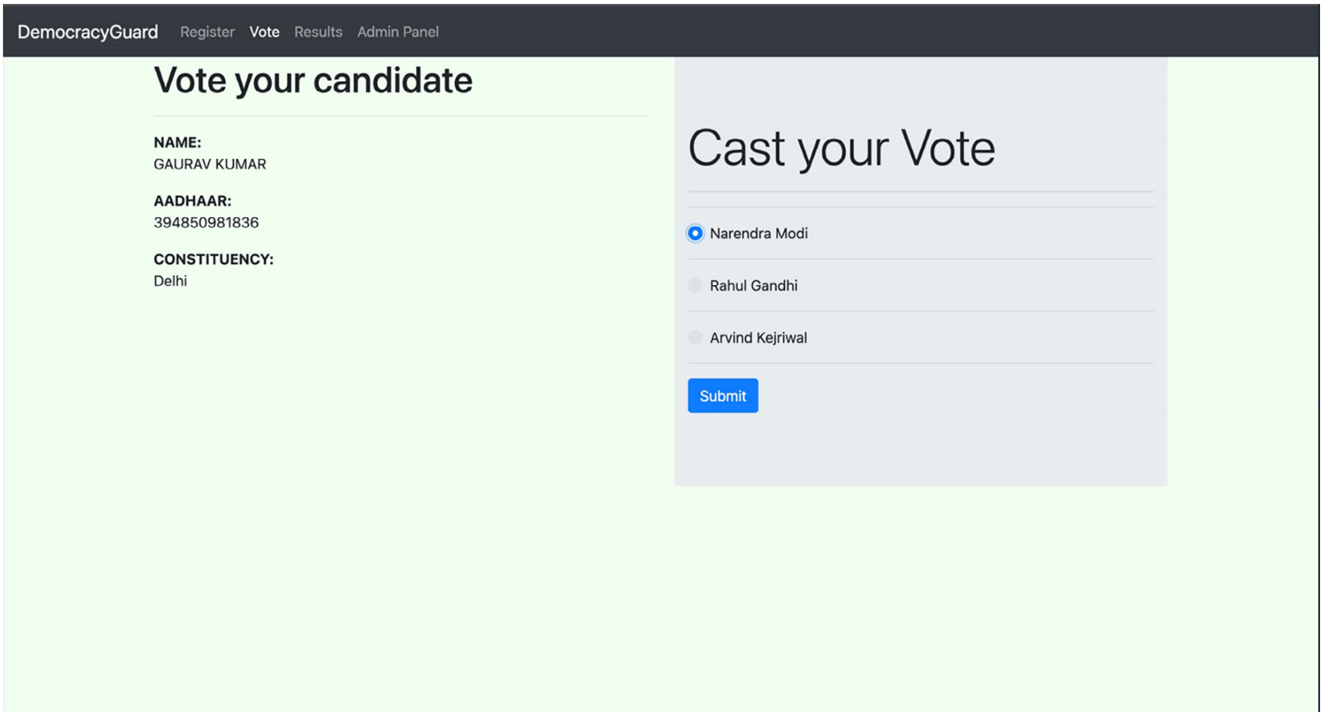
DemocracyGuard Register Vote Results Admin Panel

Welcome to the Voting Phase

Please upload your QR Code below

Upload My QR Code

FIGURE 11 Revealing a greeting and the effortless QR code submission procedure for casting votes in DemocracyGuard.



DemocracyGuard Register Vote Results Admin Panel

Vote your candidate

NAME:
GAURAV KUMAR

AADHAAR:
394850981836

CONSTITUENCY:
Delhi

Cast your Vote

● Narendra Modi

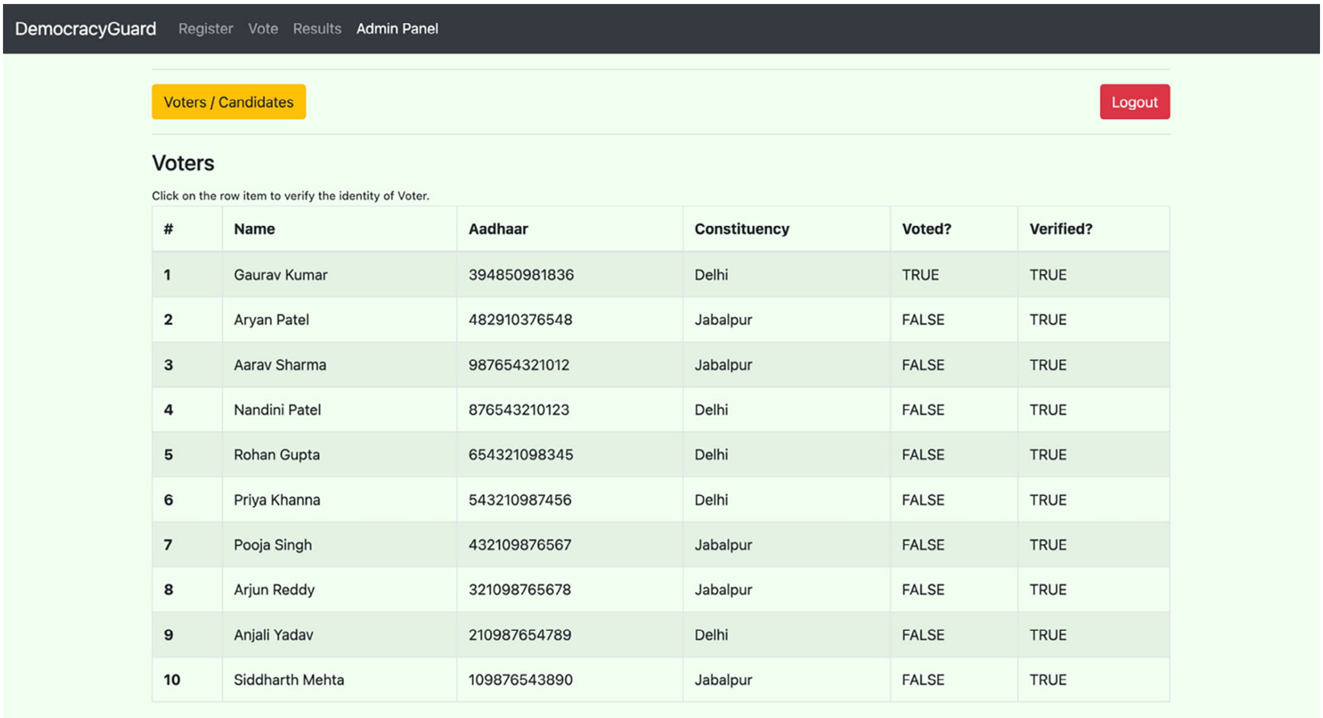
○ Rahul Gandhi

○ Arvind Kejriwal

Submit

FIGURE 12 A comprehensive visualization of voter details and candidate choices for informed decision-making during cast your vote phase.

1. *Third-Party API Use*: We have utilized the Azure Face API for facial recognition, exploring its advanced security features. Microsoft Azure adheres to stringent security and compliance standards, including GDPR and CCPA, ensuring robust biometric data protection (Galiveeti et al., 2021).
2. *No Blockchain Storage*: Biometric data is not stored on the Ethereum blockchain. The blockchain solely stores the final, anonymized vote records, ensuring the immutability and transparency of the voting results without exposing sensitive personal data.
3. *Encryption and Secure Transmission*: All biometric data transmitted to the Azure Face API is encrypted using industry-standard protocols (e.g., HTTPS/TLS) (Li et al., 2019). This ensures that the data is protected from interception during transmission.



Voters

Click on the row item to verify the identity of Voter.

#	Name	Aadhaar	Constituency	Voted?	Verified?
1	Gaurav Kumar	394850981836	Delhi	TRUE	TRUE
2	Aryan Patel	482910376548	Jabalpur	FALSE	TRUE
3	Aarav Sharma	987654321012	Jabalpur	FALSE	TRUE
4	Nandini Patel	876543210123	Delhi	FALSE	TRUE
5	Rohan Gupta	654321098345	Delhi	FALSE	TRUE
6	Priya Khanna	543210987456	Delhi	FALSE	TRUE
7	Pooja Singh	432109876567	Jabalpur	FALSE	TRUE
8	Arjun Reddy	321098765678	Jabalpur	FALSE	TRUE
9	Anjali Yadav	210987654789	Delhi	FALSE	TRUE
10	Siddharth Mehta	109876543890	Jabalpur	FALSE	TRUE

FIGURE 13 Admin panel, revealing voter and candidate details for rigorous verification in electoral processes.

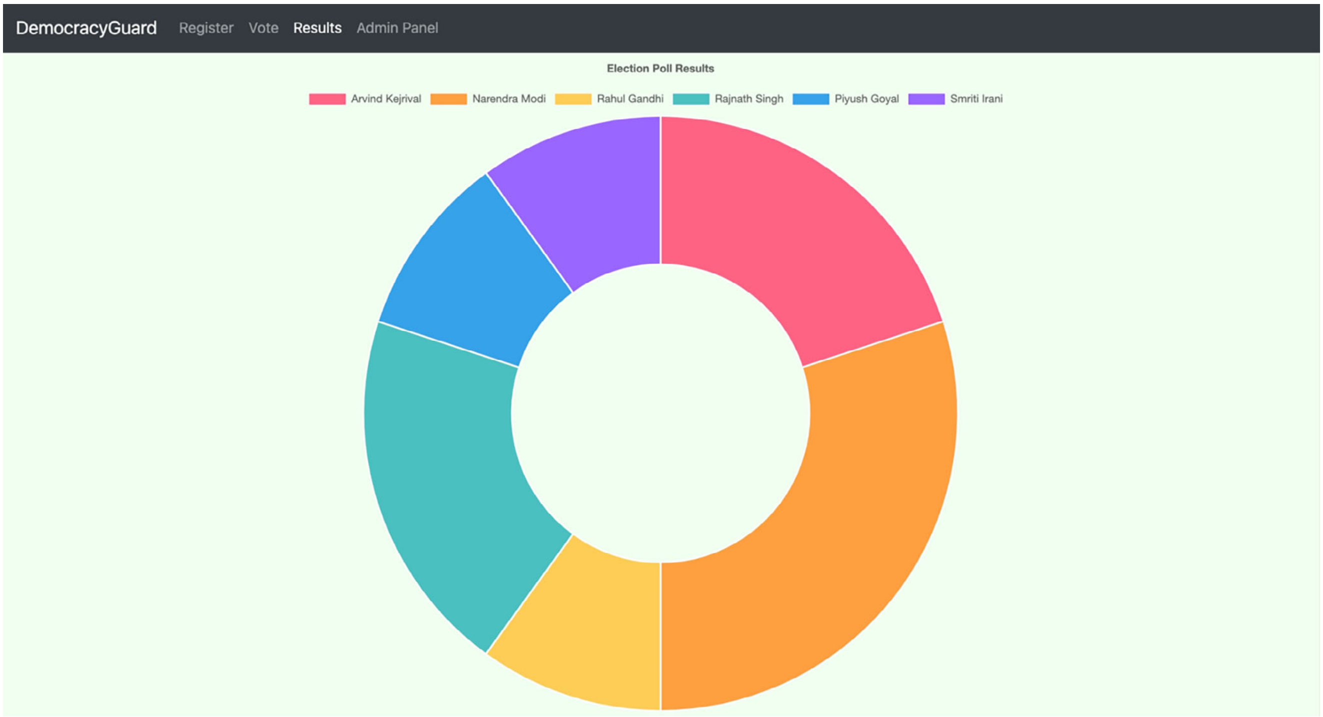
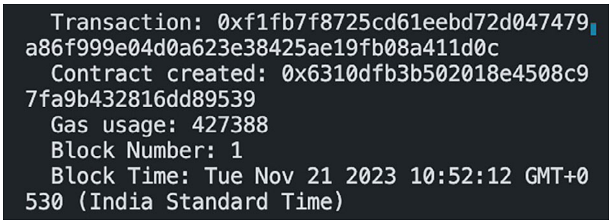


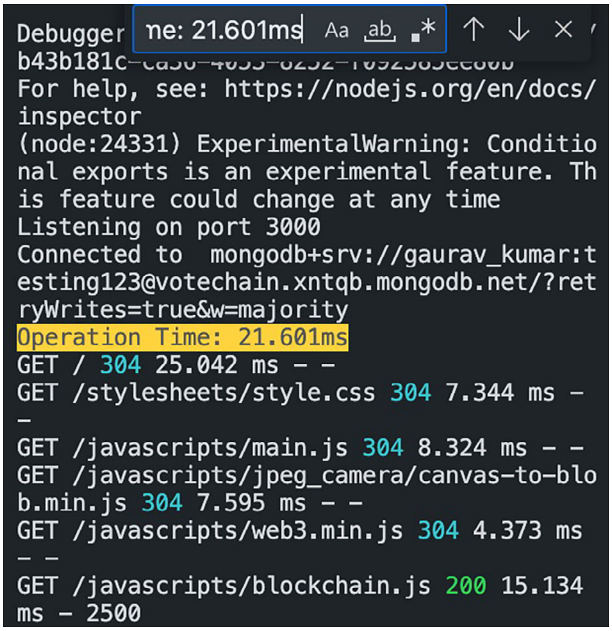
FIGURE 14 DemocracyGuard reveals election poll results in Section 5.

4. *Temporary Data Handling*: Biometric data is only processed temporarily for voter authentication (Wolf et al., 2017). Once the authentication is complete, *DemocracyGuard* will not store or retain the biometric data. This temporary handling minimizes the risk of data breaches and misuse.



```
  Transaction: 0xf1fb7f8725cd61eebd72d047479
a86f999e04d0a623e38425ae19fb08a411d0c
  Contract created: 0x6310dfb3b502018e4508c9
7fa9b432816dd89539
  Gas usage: 427388
  Block Number: 1
  Block Time: Tue Nov 21 2023 10:52:12 GMT+0
530 (India Standard Time)
```

FIGURE 15 Smart contract creation process in DemocracyGuard.



```
Debugger
b43b181c-ca50-4055-8252-f092585ee80b
For help, see: https://nodejs.org/en/docs/
inspector
(node:24331) ExperimentalWarning: Conditio
nal exports is an experimental feature. Th
is feature could change at any time
Listening on port 3000
Connected to  mongodb+srv://gaurav_kumar:t
esting123@votechain.xntqb.mongodb.net/?ret
ryWrites=true&w=majority
Operation Time: 21.601ms
GET / 304 25.042 ms - -
GET /stylesheets/style.css 304 7.344 ms -
-
GET /javascripts/main.js 304 8.324 ms - -
GET /javascripts/jpeg_camera/canvas-to-blo
b.min.js 304 7.595 ms - -
GET /javascripts/web3.min.js 304 4.373 ms
- -
GET /javascripts/blockchain.js 200 15.134
ms - 2500
```

FIGURE 16 Total operation time for casting the vote in DemocracyGuard.

5. *Azure Security Features*: Azure provides robust security features, including role-based access control (RBAC), advanced threat protection, and comprehensive monitoring (Rashid & Chawla, 2013). These features help ensure that biometric data is processed securely and access is restricted to authorized personnel only.
6. *User Consent and Transparency*: Voters are informed about the use of the Azure Face API for facial recognition. They must provide explicit consent before the biometric authentication process. Clear information is provided about how their data is used, ensuring transparency and building trust.
7. *Compliance with Regulations*: The use of the Azure Face API aligns with regulatory requirements for data protection and privacy. Microsoft Azure's compliance certifications include ISO/IEC 27001, SOC 1, SOC 2, and SOC 3, among others, ensuring that our platform meets high data security and privacy standards (Weil, 2018).
8. *Regular Security Audits*: We perform regular security audits and assessments of our integration with the Azure Face API. Independent third-party security experts conduct these audits to ensure our implementation meets the highest security standards.
9. *Data Minimization and Anonymization*: Only the minimum necessary biometric data is processed for authentication purposes. No biometric data is retained after the complete authentication process, and the voting process remains anonymous.
10. *Voter Control and Rights*: Voters have the right to withdraw their consent for facial recognition at any time. They can also request information about their data and its use, giving them control over their personal information.

5.5 | Security analysis or vulnerability assessment

Ethereum's well-established ecosystem, along with advanced security tools such as formal verification and regular audits, ensures the robustness and security of our smart contracts (Chamola et al., 2023). To substantiate our security claims, we have emphasized the importance of detailed security analysis, including internal code reviews, identification of common vulnerabilities, mitigation strategies, and testing, as given below.

TABLE 6 Comparative analysis of features in blockchain-based voting framework.

Paper title	Cost-free voting	Biometric verification	Blockchain infrastructure	Smart contract efficiency	Voter turnout enhancement	Transaction verification	Admin verified KYC	Azure face API
Wolchok et al. (2010)	✓	×	×	×	×	×	×	×
Hjálmarsson et al. (2018)	✓	×	✓	✓	✓	✓	×	×
Zhang et al. (2018)	✓	×	✓	✓	×	✓	×	×
Wang et al. (2018)	×	×	✓	×	×	✓	×	×
Pandey et al. (2019)	✓	×	✓	×	×	✓	×	×
Patil et al. (2019)	×	×	✓	✓	×	✓	×	×
Yi (2019)	✓	×	×	✓	×	×	×	×
Khan et al. (2018)	×	×	✓	×	×	✓	×	×
Kamil et al. (2021)	✓	×	✓	×	×	✓	×	×
Jafar et al. (2021)	✓	×	✓	×	✓	✓	×	×
Taş and Tanrıöver (2021)	✓	×	✓	✓	×	✓	×	×
Alvi et al. (2022)	✓	×	✓	✓	×	✓	×	×
Farooq et al. (2022)	×	×	✓	✓	×	✓	×	×
Bhadoria et al. (2022)	✓	×	✓	✓	×	✓	×	×
Wahab et al. (2022)	✓	×	✓	✓	×	✓	×	×
Neloy et al. (2023)	✓	✓	✓	✓	✓	✓	×	×
Vladucu et al. (2023)	✓	×	✓	✓	×	✓	×	×
DemocracyGuard	✓	✓	✓	✓	✓	✓	✓	✓

1. Basic security analysis

   Static Analysis Tools: We have used accessible static analysis tools like *MythX* and *Slither* to scan the smart contract code (Matulevicius & Cordeiro, 2021). These tools effectively identify common vulnerabilities such as *reentrancy attacks*, *integer overflows*, and *unchecked call returns*.

2. Internal code review

   Peer Review: We have conducted peer reviews to identify potential security issues and ensure adherence to best practices. Given the simplicity of the smart contract, this process will be thorough yet manageable.

3. Common vulnerabilities and mitigations

   Reentrancy Attacks: We have Implemented the Checks-Effects-Interactions pattern in our smart contracts to prevent reentrancy attacks.

4. Testing

   Unit Tests: We have Developed comprehensive unit tests to cover all smart contract functionalities, ensuring that votes are recorded accurately and securely.

   Integration Tests: We have Performed integration tests to confirm that the smart contract interacts correctly with the Ethereum network and other system components.

5.6 | Handling of fault tolerance mechanisms

DemocracyGuard employs robust fault tolerance mechanisms to ensure continuous and reliable operation despite unexpected disruptions. Various fault tolerance mechanisms have been implemented to ensure the system's resilience and reliability.

1. *Redundancy in Data Storage*: Multiple copies of critical data are stored across different nodes and locations. This redundancy ensures that even if some nodes fail or are compromised, the data remains accessible and intact, preventing data loss and ensuring continuous operation.
2. *Decentralized Consensus Algorithms*: *DemocracyGuard* uses decentralized consensus algorithms, such as byzantine fault tolerance (BFT) and PoS. These algorithms allow the system to agree on the state of the blockchain, even in the presence of faulty or malicious nodes. This decentralization prevents any single point of failure and enhances the security and robustness of the system.
3. *Resilient network architecture*: The network architecture is designed to be resilient against Distributed Denial of Service (DDoS) attacks and other network-based threats. Measures such as rate limiting, traffic filtering, and decentralized routing enhance the network's ability to withstand and recover from such attacks.

6 | CONCLUSION AND FUTURE WORK

Incorporating blockchain technology into online voting systems holds great potential for addressing the pressing security concerns associated with electronic voting. The decentralized architecture, transparency features, and non-repudiation capabilities inherent in blockchain offer a robust foundation for establishing a trustworthy and resilient electoral process. The proposed *DemocracyGuard* platform, built on the Ethereum blockchain and complemented by facial recognition technology, represents a significant stride in fortifying voter authentication and enhancing the overall security of online voting. Implementing blockchain in online voting systems requires ongoing attention to various challenges and considerations. Future work should include comprehensive security audits to identify and mitigate potential vulnerabilities, ensuring the platform's resistance to manipulation and unauthorized access. Efforts should be directed towards refining the user experience, making the platform more intuitive and accessible to a diverse range of voters. Scalability remains a critical aspect, and further research should be conducted to optimize the performance of blockchain-based online voting systems, especially as they handle increasing transactions during elections. The *DemocracyGuard* platform stands as a testament to the potential of blockchain in revolutionizing the electoral landscape. Still, a sustained commitment to improvement and adaptation will be crucial for its long-term success.

DATA AVAILABILITY STATEMENT

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

ORCID

Mritunjay Shall Peelam https://orcid.org/0000-0002-8022-3815
Gaurav Kumar https://orcid.org/0009-0002-4457-8333
Kunjan Shah https://orcid.org/0009-0001-5897-8605
Vinay Chamola https://orcid.org/0000-0002-6730-3060

REFERENCES

AboSamra, K. M., AbdelHafez, A. A., Assassa, G. M., & Mursi, M. F. (2017). A practical, secure, and auditable e-voting system. *Journal of Information Security and Applications*, *36*, 69–89.

Akcora, C. G., Gel, Y. R., & Kantarcioglu, M. (2022). Blockchain networks: Data structures of Bitcoin, Monero, Zcash, Ethereum, Ripple, and Iota. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *12*(1), e1436.

Akshay, S., & Arun, M. (2019). Decentralized E-voting system.

Alvi, S. T., Uddin, M. N., Islam, L., & Ahamed, S. (2022). Dvtchain: A blockchain-based decentralized mechanism to ensure the security of digital voting system voting system. *Journal of King Saud University-Computer and Information Sciences*, *34*(9), 6855–6871.

Amoah, E., & Oh, J.-Y. (2021). Blockchain adoption in project management. *Issues in Information Systems*, *22*(4), 143–156.

Asif, R., & Hassan, S. R. (2023). Shaping the future of Ethereum: Exploring energy consumption in proof-of-work and proof-of-stake consensus. *Frontiers in Blockchain*, *6*, 1151724.

Baliga, A., Subhod, I., Kamat, P., & Chatterjee, S. (2018). Performance evaluation of the quorum blockchain platform. *arXiv*, abs/1809.03421, 1–8.

Barański, S., Szymański, J., Sobecki, A., Gil, D., & Mora, H. (2020). Practical i-voting on stellar blockchain. *Applied Sciences*, *10*(21), 7606.

Benji, M., & Sindhu, M. (2019). A study on the corda and ripple blockchain platforms. In *Advances in Big Data and Cloud Computing: Proceedings of ICBDCC18* (pp. 179–187). Springer.

Benny, A. (2020). Blockchain based e-voting system.

Bhadoria, R. S., Das, A. P., Bashar, A., & Zikria, M. (2022). Implementing blockchain-based traceable certificates as sustainable technology in democratic elections. *Electronics*, *11*(20), 3359.

Buterin, V. (2022). *Proof of stake: The making of Ethereum and the philosophy of blockchains*. Seven Stories Press.

Chamola, V., Goyal, A., Sharma, P., Hassija, V., Binh, H. T. T., & Saxena, V. (2023). Artificial intelligence-assisted blockchain-based framework for smart and secure emr management. *Neural Computing and Applications*, *35*(31), 22959–22969.

Coelho, I., Coelho, V., Lin, P., & Zhang, E. (2019). Community yellow paper: A technical specification for neo blockchain. *NeoResearch*, *1*, 1–30.

Cooley, R., Wolf, S., & Borowczak, M. (2018). Blockchain-based election infrastructures. In *2018 IEEE International Smart Cities Conference (ISC2)* (pp. 1–4). IEEE.

Cortier, V., Gaudry, P., & Glondu, S. (2021). Possible evolutions of the voting system in tezos.

Dang, W., Cai, L., Liu, M., Li, X., Yin, Z., Liu, X., Yin, L., & Zheng, W. (2023). Increasing text filtering accuracy with improved LSTM. *Computing and Informatics*, *42*(6), 1491–1517.

Di Angelo, M., & Salzer, G. (2023). Identification of token contracts on Ethereum: standard compliance and beyond. *International Journal of Data Science and Analytics*, *16*(3), 333–352.

Duguleană, M., & Gîrbacia, F. (2021). Augmented reality meets non-fungible tokens: Insights towards preserving property rights. In *2021 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)* (pp. 359–361). IEEE.

Esposito, C., & Choi, C. (2023). Design and implementation of a blockchain-based e-voting system by using the algorand platform. *Proceedings of the 38th ACM/SIGAPP Symposium on Applied Computing*, *1*, 715–723.

Farooq, M. S., Iftikhar, U., & Khelifi, A. (2022). A framework to make voting system transparent using blockchain technology. *IEEE Access*, *10*, 59959–59969.

Galiveeti, S., Tawalbeh, L., Tawalbeh, M., & El-Latif, A. A. A. (2021). Cybersecurity analysis: Investigating the data integrity and privacy in aws and azure cloud platforms. In *Artificial Intelligence and Blockchain for Future Cybersecurity Applications* (pp. 329–360). Springer.

Hao, F., & Ryan, P. Y. (2016). *Real-world electronic voting: Design, analysis and deployment*. CRC Press.

Hassija, V., Bansal, G., Chamola, V., Kumar, N., & Guizani, M. (2020). Secure lending: Blockchain and prospect theory-based decentralized credit scoring model. *IEEE Transactions on Network Science and Engineering*, *7*(4), 2566–2575.

Hassija, V., Bansal, G., Chamola, V., Saxena, V., & Sikdar, B. (2019). Blockcom: A blockchain based commerce model for smart communities using auction mechanism. In *2019 IEEE International Conference on Communications Workshops (ICC Workshops)* (pp. 1–6). IEEE.

Hassija, V., Batra, S., Chamola, V., Anand, T., Goyal, P., Goyal, N., & Guizani, M. (2021). A blockchain and deep neural networks-based secure framework for enhanced crop protection. *Ad Hoc Networks*, *119*, 102537.

Hassija, V., Chamola, V., & Zeadally, S. (2020). Bitfund: A blockchain-based crowd funding platform for future smart and connected nation. *Sustainable Cities and Society*, *60*, 102145.

Hassija, V., Saxena, V., & Chamola, V. (2021). A mobile data offloading framework based on a combination of blockchain and virtual voting. *Software: Practice and Experience*, *51*(12), 2428–2445.

Hassija, V., Zeadally, S., Jain, I., Tahiliani, A., Chamola, V., & Gupta, S. (2021). Framework for determining the suitability of blockchain: Criteria and issues to consider. *Transactions on Emerging Telecommunications Technologies*, *32*(10), e4334.

He, O., & Su, Z. (1998). A new practical secure e-voting scheme.

Hentschel, A., Shirley, D., Lafrance, L., & Zamski, M. (2019). Flow: Separating Consensus and Compute–Execution Verification. arXiv preprint arXiv: 1909.05832.

Herrnson, P. S., Hanmer, M. J., & Niemi, R. G. (2012). The impact of ballot type on voter errors. *American Journal of Political Science*, *56*(3), 716–730.

Hjálmarsson, F., Hreiðarsson, G. K., Hamdaqa, M., & Hjálmtỳsson, G. (2018). Blockchain-based e-voting system. In *2018 IEEE 11th International Conference on Cloud Computing (CLOUD)* (pp. 983–986). IEEE.

Jafar, U., Aziz, M. J. A., & Shukur, Z. (2021). Blockchain for electronic voting system—Review and open research challenges. *Sensors*, *21*(17), 5874.

Jani, S. (2020). *Smart contracts: Building blocks for digital transformation*. Indira Gandhi National Open University.

Kamil, M., Bist, A. S., Rahardja, U., Santoso, N. P. L., & Iqbal, M. (2021). Covid-19: Implementation e-voting blockchain concept. *International Journal of Artificial Intelligence Research*, *5*(1), 25–34.

Kasdan, D. (2013). *State restrictions on voter registration drives*. Brennan Center for Justice.

Khan, K. M., Arshad, J., & Khan, M. M. (2018). Secure digital voting system based on blockchain technology. *International Journal of Electronic Government Research (IJEGR)*, *14*(1), 53–62.

Kho, Y.-X., Heng, S.-H., & Chin, J.-J. (2022). A review of cryptographic electronic voting. *Symmetry*, *14*(5), 858.

Kumar, D. A., & Begum, T. U. S. (2013). A comparative study on fingerprint matching algorithms for evm. *Journal of Computer Sciences and Applications*, *1*(4), 55–60.

Lamela Seijas, P., Nemish, A., Smith, D., & Thompson, S. (2020). Marlowe: implementing and analysing financial contracts on blockchain. In *Financial Cryptography and Data Security: FC 2020 International Workshops, AsiaUSEC, CoDeFi, VOTING, and WTSC* (Vol. 2020, pp. 496–511). Springer.

Leemann, L., & Bochsler, D. (2014). A systematic approach to study electoral fraud. *Electoral Studies*, *35*, 33–47.

Li, S., Liu, F., Liang, J., Cai, Z., & Liang, Z. (2019). Optimization of face recognition system based on azure iot edge. *Computers, Materials & Continua*, *61*(3), 1377–1389.

Liu, S., Han, W., Zhang, Z., & Chan, F. T. (2024). An analysis of performance, pricing, and coordination in a supply chain with cloud services: The impact of data security. *Computers & Industrial Engineering*, *192*, 110237.

Liu, Y., Zhao, B., Zhao, Z., Liu, J., Lin, X., Wu, Q., & Susilo, W. (2024). Ss-did: A secure and scalable web3 decentralized identity utilizing multi-layer sharding blockchain. *IEEE Internet of Things Journal*, *11*, 25694-25705.

Ma, J., & Hu, J. (2022). Safe consensus control of cooperative-competitive multi-agent systems via differential privacy. *Kybernetika*, *58*(3), 426–439.

Matulevicius, N., & Cordeiro, L. C. (2021). Verifying security vulnerabilities for blockchain-based smart contracts. In *2021 XI Brazilian Symposium on Computing Systems Engineering (SBESC)* (Vol. 2021, pp. 1–8). IEEE.

Metcalfe, W., et al. (2020). Ethereum, smart contracts, dapps. *Blockchain and Crypt Currency*, *77*, 77–93.

Mukherjee, P. P., Boshra, A. A., Ashraf, M. M., & Biswas, M. (2020). A hyper-ledger fabric framework as a service for improved quality e-voting system. In *2020 IEEE Region 10 Symposium (TENSYMP)* (Vol. 2020, pp. 394–397). IEEE.

Mukhopadhyay, M. (2018). *Ethereum smart contract development: Build blockchain-based decentralized applications using solidity*. Packt Publishing.

Mutuku, R. K. (2023). Modernizing the kenyan electoral system through polkadot blockchain network. *East African Journal of Information Technology*, *6*(1), 77–90.

Neloy, M. N., Wahab, M. A., Wasif, S., All Noman, A., Rahaman, M., Pranto, T. H., Haque, A. B., & Rahman, R. M. (2023). A remote and cost-optimized voting system using blockchain and smart contract. *IET Blockchain*, *3*(1), 1–17.

Oppliger, R. (2002). How to address the secure platform problem for remote internet voting, *SIS*, *2*, 153–173.

Pandey, A., Bhasi, M., & Chandrasekaran, K. (2019). Votechain: A blockchain based e-voting system. In *2019 Global Conference for Advancement in Technology (GCAT)* (Vol. 2019, pp. 1–4). IEEE.

Patil, H., Ladkat, P., Jituri, A., Desai, R., & Shinde, D. S. (2019). Blockchain based e-voting system. In *Proceedings of International Conference on Communication and Information Processing (ICCIP)*. Elsevier Inc.

Prasad, R. M., Bojja, P., & Nakirekanti, M. (2016). Aadhar based electronic voting machine using arduino. *International Journal of Computer Applications*, *145*(12), 39–42.

Qi, H., Zhou, Z., Irizarry, J., Lin, D., Zhang, H., Li, N., & Cui, J. (2024). Automatic identification of causal factors from fall-related accident investigation reports using machine learning and ensemble learning approaches. *Journal of Management in Engineering*, *40*(1), 04023050.

Rashid, M., & Chawla, E. R. (2013). Securing data storage by extending role-based access control. *International Journal of Cloud Applications and Computing (IJCAC)*, *3*(4), 28–37.

Riera, A., & Brown, P. (2003). Bringing confidence to electronic voting. *Electronic Journal of e-Government*, *1*(1), 14–21.

Roberts, T. S. (2016). Enhanced disclosure as a response to increasing out-of-state spending in state and local elections, *50*, 137.

Rodwell, P. M., Furnell, S., & Reynolds, P. L. (2007). A non-intrusive biometric authentication mechanism utilising physiological characteristics of the human head. *Computers & Security*, *26*(7-8), 468–478.

Sapák, F. Security and performance analysis of avalanche distributed consensus protocol.

Sharma, A., Chavikant, T., Singh, T., Aggarwal, T., Jain, D., Singh, P., et al. (2022). Blockchain based e-voting. In *2022 4th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N)* (pp. 2054–2058). IEEE.

Shi, J., Wang, J., & Fu, F. (2015). Fast and robust vanishing point detection for unstructured road following. *IEEE Transactions on Intelligent Transportation Systems*, *17*(4), 970–979.

Sun, G., Li, Y., Liao, D., & Chang, V. (2018). Service function chain orchestration across multiple domains: A full mesh aggregation approach. *IEEE Transactions on Network and Service Management*, *15*(3), 1175–1191.

Taş, R., & Tanrıöver, Ö. Ö. (2020). A systematic review of challenges and opportunities of blockchain for e-voting. *Symmetry*, *12*(8), 1328.

Taş, R., & Tanrıöver, Ö. Ö. (2021). A manipulation prevention model for blockchain-based e-voting systems. *Security and Communication Networks*, *2021*, 1–16.

Vivek, S., Yashank, R., Prashanth, Y., Yashas, N., & Namratha, M. (2020). E-voting system using hyperledger sawtooth. In *2020 International Conference on Advances in Computing, Communication & Materials (ICACCM)* (pp. 29–35). IEEE.

Vladucu, M.-V., Dong, Z., Medina, J., & Rojas-Cessa, R. (2023). E-voting meets blockchain: A survey. *IEEE Access*, *11*, 23293–23308.

Vo-Cao-Thuy, L., Cao-Minh, K., Dang-Le-Bao, C., & Nguyen, T. A. (2019). Votereum: An Ethereum-based e-voting system. In *2019 IEEE-RIVF International Conference on Computing and Communication Technologies (RIVF)* (pp. 1–6). IEEE.

Vranken, H. (2017). Sustainability of bitcoin and blockchains. *Current Opinion in Environmental Sustainability*, *28*, 1–9.

VVPAT Vvpat | district doda | India. https://doda.nic.in/vvpat/

Wahab, Y., Ghazi, A., Al-Dawoodi, A., Alisawi, M., Abdullah, S., Hammood, L., & Nawaf, A. (2022). A framework for blockchain based e-voting system for Iraq. *International Journal of Interactive Mobile Technologies*, *16*(10), 210–222.

Wallach, D. S. (2020). On the security of ballot marking devices. *arXiv*, *16*, 558.

Wang, B., Sun, J., He, Y., Pang, D., & Lu, N. (2018). Large-scale election based on blockchain. *Procedia Computer Science*, *129*, 234–237.

Weil, T. (2018). Taking compliance to the cloud—using iso standards (tools and techniques). *IT Professional*, *20*(6), 20–30.

Weiss, D., Wolmer, J., & Vatsa, A. (2022). Blockchain-based electronic voting system for modern democracy: A review. In *2022 IEEE Integrated STEM Education Conference (ISEC)* (pp. 162–166). IEEE.

Wolchok, S., Wustrow, E., Halderman, J. A., Prasad, H. K., Kankipati, A., Sakhamuri, S. K., Yagati, V., & Gonggrijp, R. (2010). Security analysis of India's electronic voting machines. In *Proceedings of the 17th ACM Conference on Computer and Communications Security* (pp. 1–14). Association for Computing Machinery.

Wolf, P., Alim, A., Kasaro, B., Namugera, P., Saneem, M., & Zorigt, T. (2017). *Introducing biometric technology in elections*. International Institute for Democracy and Electoral Assistance ….

Xuemin, Z., Haitao, D., Zenggang, X., Ying, R., Yanchao, L., Yuan, L., & Delin, H. (2024). Self-organizing key security management algorithm in socially aware networking. *Journal of Signal Processing Systems*, *96*, 1–15.

Yadav, J. S., Yadav, N. S., & Sharma, A. K. (2021). A qualitative and quantitative parametric estimation of the Ethereum and Tron blockchain networks. In *2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO)* (pp. 1–5). IEEE.

Yakovenko, A. (2018). *Solana: A new architecture for a high performance blockchain v0. 8.13*. Whitepaper.

Yang, J., Yang, K., Xiao, Z., Jiang, H., Xu, S., & Dustdar, S. (2023). Improving commute experience for private car users via blockchain-enabled multitask learning. *IEEE Internet of Things Journal*, *10*.

Yang, Y., Guan, Z., Wan, Z., Weng, J., Pang, H. H., & Deng, R. H. (2021). Priscore: blockchain-based self-tallying election system supporting score voting. *IEEE Transactions on Information Forensics and Security*, *16*, 4705–4720.

Yanovich, Y., Ivashchenko, I., Ostrovsky, A., Shevchenko, A., & Sidorov, A. (2018). Exonum: Byzantine fault tolerant protocol for blockchains. *Computer Science Engineering*, *1*, 1–36.

Yi, H. (2019). Securing e-voting based on blockchain in p2p network. *EURASIP Journal on Wireless Communications and Networking*, *2019*(1), 1–9.

Yin, L., Wang, L., Lu, S., Wang, R., Ren, H., AlSanad, A., AlQahtani, S. A., Yin, Z., Li, X., & Zheng, W. Afbnet: A lightweight adaptive feature fusion module for super-resolution algorithms.

Zhang, W., Yuan, Y., Hu, Y., Huang, S., Cao, S., Chopra, A., & Huang, S. (2018). A privacy-preserving voting protocol on blockchain. In *2018 IEEE 11th International Conference on Cloud Computing (CLOUD)* (pp. 401–408). IEEE.

AUTHOR BIOGRAPHIES

Mritunjay Shall Peelam is a Research Scholar at the Birla Institute of Technology and Science - Pilani (BITS Pilani), where he works under the guidance of Prof. Vinay Chamola. His research focuses on Quantum Computing, Blockchain, IoT, and Intelligent Transport Systems. Mr. Peelam earned his M.Tech in Computer Science and Engineering from the University School of Information, Communication, and Technology (USICT) at Guru Gobind Singh Indraprastha University, New Delhi, in 2021, where he was awarded the "Mr. Talented of the Year" accolade. He completed his B.Tech in Computer Science and Engineering at IETE, New Delhi, in 2012, achieving an All India 9th Rank. He has served as an Assistant Professor in the Department of Computer Science and Engineering at Pranveer Singh Institute of Technology (PSIT), Kanpur, India.

Gaurav Kumar holds a Bachelor's of Engineering degree in Computer Science from BITS Pilani, Pilani Campus. He currently works as a Machine Learning Engineer at Coinbase. With multiple years of experience in the field of Blockchain, Gaurav has established himself as a proficient and knowledgeable professional in this rapidly evolving domain. In addition, he has specialized experience in the field of zero-knowledge proofs.

Kunjan Shah has completed his Bachelor's of Engineering from Birla Institute of Technology and Science, Pilani, Rajasthan in 2023. Currently he is pursuing Master's degree in Computer Science at Birla Institute of Technology and Science, Pilani, Rajasthan. His research interests are Blockchain-enabled verification systems, ad-hoc vehicular networks, cryptography, and federated learning. He has several publications in Digital Communications and Networks journal, Expert Systems journal and International Conference on Information Networking (ICOIN).

Prof. Vinay Chamola obtained his B.E. in Electrical and Electronics Engineering and his M.Tech in Communication Engineering from the Birla Institute of Technology and Science (BITS-Pilani), India, in 2010 and 2013, respectively. He completed his Ph.D. in Electrical and Computer Engineering at the National University of Singapore in 2016. In 2015, he was a Visiting Researcher with the Autonomous Networks Research Group (ANRG) at the University of Southern California, Los Angeles, CA, USA. He also worked as a Post-Doctoral Research Fellow at the National University of Singapore. Prof. Chamola is currently an Associate Professor in the Department of Electrical and Electronics Engineering at BITS-Pilani, where he leads the Internet of Things Research Group/Lab. His research interests encompass IoT Security, Blockchain, UAVs, VANETs, 5G, and Healthcare. He serves as an Area Editor for the Ad Hoc Networks Journal, Elsevier, and the IEEE Internet of Things Magazine, and is an Associate Editor for the IEEE Transactions on Intelligent Transportation Systems, IEEE Networking Letters, IEEE Consumer Electronics Magazine, IET Quantum Communications, IET Networks, among others. Prof. Chamola has co-chaired several notable workshops, including IEEE Globecom Workshop 2021, IEEE INFOCOM 2022 Workshop, IEEE ANTS 2021, and IEEE ICIAfS 2021. He is recognized among the World's Top 2% Scientists by Stanford University. He is the co-founder and President of the healthcare startup Medsupervision Pvt. Ltd. He is a Senior Member of IEEE and a Fellow of the IET.

How to cite this article: Peelam, M. S., Kumar, G., Shah, K., & Chamola, V. (2024). DemocracyGuard: Blockchain-based secure voting framework for digital democracy. *Expert Systems*, e13694. https://doi.org/10.1111/exsy.13694