IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Blockvote: A Decentralized Electronic Voting System With Multi-Modal Biometric Verification Using Ethereum Smart Contracts

[1]Eshwank Kapoor, [2]Oshan Manhas, [3]Er. Sheetal Gandotra, [4]Dr. Bhawna Sharma

[1,2]Students, Dept. of Computer Science & Engineering Government College of Engineering and Technology, Jammu
J&K, India

[3]Associate Professor, Dept. of Computer Science & Engineering Government College of Engineering and Technology, Jammu
J&K, India

[4]Professor, Dept. of Computer Science & Engineering Government College of Engineering and Technology, Jammu
J&K, India

Abstract: Modern democratic processes require electoral frameworks that ensure ballot confidentiality, transparent result verification, and absolute voter authenticity. While conventional centralized e-voting architectures frequently compromise at least one of these critical security pillars, decentralized blockchain solutions often introduce severe usability hurdles, particularly regarding cryptocurrency wallet management. To bridge this gap, this paper introduces BlockVote, a comprehensive decentralized electronic voting system that merges Ethereum smart contracts with a robust multi-modal biometric authentication framework. Our proposed architecture utilizes a ResNet-34 facial recognition model to extract 128-dimensional face embeddings, alongside WebAuthn/FIDO2-compliant fingerprint verification. For initial voter onboarding, the system leverages Aadhaar-based national identity cross-referencing paired with an email-delivered one-time password (OTP). To strictly enforce a one-person, one-vote policy, we implemented an exhaustive Euclidean-distance comparison against all stored facial profiles during registration, utilizing a rigorous duplicate threshold ($\theta dup = 0.45$). Furthermore, the entire election lifecycle is governed by a three-phase state machine embedded within a Solidity smart contract, guaranteeing an immutable audit ledger. To eliminate cryptocurrency-related friction, a gasless voting relay abstracts transaction fees away from the end-user. Privacy is strictly maintained by processing biometric data locally within the browser and recording only the SHA-256 hash of the voter's email on the public blockchain. Comprehensive Truffle/Mocha testing validates the system's resilience against double-voting and unauthorized access. Finally, we discuss current limitations and future enhancements, including liveness detection and zero-knowledge proofs for enhanced ballot secrecy.

Index Terms — blockchain, e-voting, biometric authentication, face recognition, Ethereum, smart contracts, WebAuthn, FIDO2, Solidity, electoral security, decentralised systems, ResNet-34, Aadhaar, OTP, gasless relay.

I. INTRODUCTION

The integrity of democratic governance relies heavily on free and equitable electoral processes. With the ongoing digital transformation of societal infrastructure, the need for secure, remote, and easily accessible voting mechanisms has grown exponentially. However, despite extensive academic efforts, current electronic voting platforms consistently fail to balance the core pillars of an optimal election framework: robust privacy, uncompromised correctness, coercion resistance, strict eligibility validation, and transparent verifiability [10].

Conventional centralized electronic voting architectures are inherently flawed due to their reliance on single-authority databases. This centralization creates severe vulnerabilities, as insider threats or external cyberattacks can manipulate ballot records without leaving forensic evidence. Such vulnerabilities—highlighted by controversies surrounding standalone electronic voting machines (EVMs)—have severely undermined public trust in digital elections [13].

The advent of distributed ledger technology, specifically blockchain, offers a paradigm shift by providing a decentralized, tamper-proof, and transparent audit trail. Smart contracts on the Ethereum network enable the automated enforcement of complex electoral rules—such as phase management and duplicate prevention—without requiring a trusted intermediary [1]. However, purely blockchain-based systems introduce steep learning curves, requiring average users to navigate cryptocurrency wallets and gas fees, thereby creating a digital divide [11].

Concurrently, advancements in biometric verification—particularly facial and fingerprint recognition—have become widely accessible on commodity hardware. Leveraging multi-factor biometrics can drastically reduce identity spoofing and unauthorized account sharing, addressing vulnerabilities that standard password protocols cannot mitigate [2].

A. Problem Statement

Current electronic voting systems are plagued by critical limitations. Centralised platforms suffer from single points of failure, enabling silent vote tampering without an audit trail. Conversely, while blockchain mitigates this, existing deployments impose severe usability friction via crypto-wallets and fail to protect ballot secrecy. Furthermore, without integrated biometrics, both approaches remain highly vulnerable to identity fraud and double voting.

B. How Blockchain Helps in BlockVote

A blockchain functions as a sequential ledger where each block contains a cryptographic hash linking it to its predecessor, alongside timestamps and transaction data. This dependency ensures that any retrospective alteration of records immediately invalidates the hash chain, alerting the entire network. This inherent immutability guarantees that while data can be written, it cannot be covertly modified post-acceptance.

In the context of BlockVote, this decentralization eliminates the single point of failure inherent to traditional databases, ensuring no central administrator can manipulate tallies. Furthermore, the public nature of the ledger provides an append-only, post-election audit trail. BlockVote strategically utilizes Ethereum smart contracts to govern the election state machine while the biometric layer acts as the gatekeeper to determine eligibility [1, 8, 9, 19, 20].

C. Proposed Solution and Key Contributions

BlockVote bridges these gaps through a unified architecture that integrates: (1) a Convolutional Neural Network (CNN) for facial recognition featuring a dual-threshold mechanism to prevent duplicate profiles; (2) WebAuthn/FIDO2 hardware-bound fingerprint authentication; (3) primary identity verification via Aadhaar databases coupled with email OTP; (4) a strict, three-phase election state machine deployed via Solidity smart contracts; (5) a server-side gasless transaction relay to enhance voter usability; and (6) client-side biometric processing and SHA-256 data anonymization to ensure strict voter privacy.

The primary contributions of this paper include a holistic, decentralized e-voting framework, a novel one-face, one-vote validation algorithm utilizing 128-D embeddings, and a gasless relay mechanism that abstracts the complexities of Ethereum from the end-user while preserving immutable auditing capabilities.

D. Paper Organisation

The subsequent sections are structured as follows: Section II reviews existing literature. Section III outlines the proposed system architecture. Section IV details the mathematical models and algorithms. Section V provides technical implementation specifics. Section VI discusses system evaluation and security analysis. Section VII addresses limitations and future work, while Section VIII concludes the paper.

II. Related Work

A. Blockchain-Based Voting Systems

Hjálmarsson et al. [11] produced one of the first complete Ethereum-based e-voting architectures, demonstrating that smart contracts can replace trusted third-party tallying authorities. Their system, however, relies exclusively on wallet-based voter identity and does not integrate any biometric verification layer, leaving the system susceptible to account sharing and identity fraud.

Shahzad and Crowcroft [10] introduced blind signatures as a voter anonymisation mechanism, adapting the approach to an adjusted blockchain protocol. Their scheme improves ballot secrecy over naive on-chain storage but requires voters to manage multiple cryptographic keys, increasing the technical barrier to participation.

Ayed [12] provided a conceptual framework focusing on auditability and immutability of blockchain ledgers for electoral applications. The work establishes formal correctness properties but does not consider the integration of identity verification or biometric authentication. Khan et al. [13] designed a permissioned blockchain voting system using cryptographic commitments for ballot secrecy, targeting enterprise deployments (Hyperledger Fabric), whereas BlockVote targets general-purpose Ethereum deployments with open voter registration.

Recent surveys continue to frame scalability, privacy, and usability as the major open issues in blockchain e-voting, while newer prototypes increasingly combine face recognition, deepfake detection, and structured consensus design to narrow the gap between theory and deployment [19–23].

B. Biometric Authentication in E-Voting

The application of face recognition to voter identity verification has been explored in the context of polling-station automation. ResNet-34 [2]—a 34-layer residual neural network—has emerged as a dominant backbone for face embedding models, producing compact 128-dimensional descriptors with strong discriminative power across millions of identities. Face-api.js [5] provides a JavaScript implementation of this model, enabling browser-side inference without server round-trips for the raw image, which is a privacy advantage that BlockVote exploits.

C. WebAuthn and FIDO2 in Authentication

The W3C Web Authentication [6] standard (WebAuthn/FIDO2) provides phishing-resistant, password-less authentication using device-bound public-key cryptography. Unlike biometric templates stored on a server, WebAuthn credentials are bound to a specific device, meaning that even if the server database is compromised, the attacker cannot replay credentials from a different device. Despite its strong security properties, WebAuthn has not been extensively studied in the context of e-voting systems; BlockVote is among the first to integrate it as a second biometric factor in a blockchain voting pipeline.

D. Positioning of BlockVote

Table 1 provides a structured comparison. BlockVote is the only system in this comparison to simultaneously provide blockchain immutability, face-biometric verification, fingerprint authentication, a gasless relay, and OTP-based initial identity verification.

table 1: comparison of e-voting approaches

System	BC	Face	FP	Gasless	OTP/Biom.
Paper ballot	✗	✗	✗	–	✗
Central EVM	✗	✗	✗	✓	✗
MetaMask-only	✓	✗	✗	✗	✗
Hjal. [11]	✓	✗	✗	✗	✗
Shahzad [10]	✓	✗	✗	✗	✗
Khan [13]	✓	✗	✗	✗	✗
BlockVote	✓	✓	✓	✓	✓

BC=Blockchain, FP=Fingerprint/WebAuthn

III. System Architecture

A. Architecture Overview

BlockVote is built upon a robust, four-tier architectural model. The **Frontend Layer** operates entirely within the user's browser, housing the user interface (EJS + TailwindCSS), the SimpleWebAuthn library, and face-api.js neural network models for localized biometric inference, together with Web3.js for direct blockchain interactions. The **Backend Layer** consists of a Node.js/Express.js RESTful API handling vote relaying, biometric profile management, and identity verification. The **Data Layer** manages off-chain state through a PostgreSQL database on Supabase alongside a local Ganache Ethereum simulator. The **Blockchain Layer** contains the compiled Contest.sol smart contract, acting as the immutable ledger for election phases, voter registries, and candidate tallies. The complete multi-tier architecture is illustrated in Fig. 1.

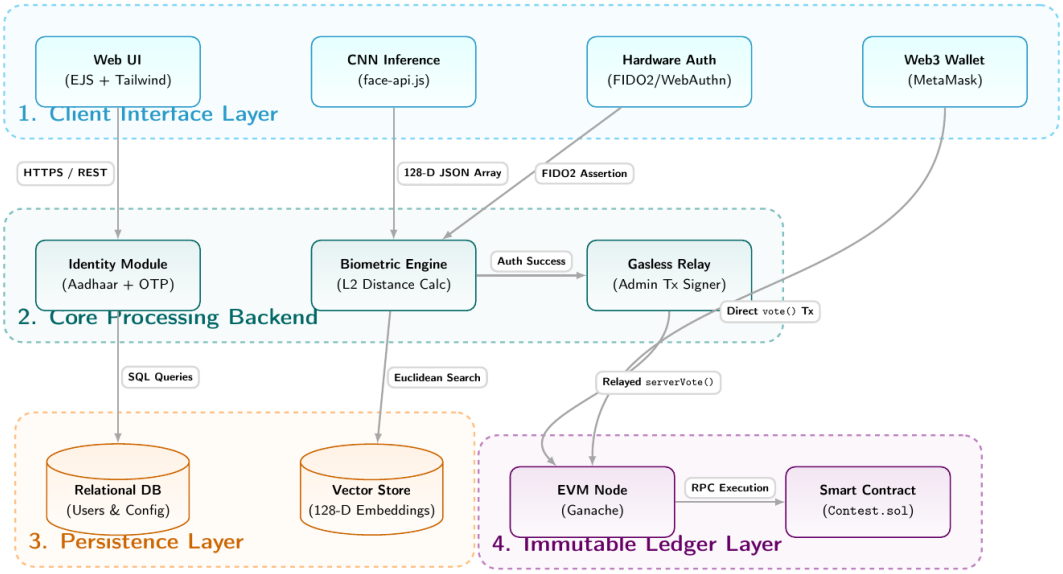

fig. 1: comprehensive multi-tier architecture of blockvote

B. Voter Journey and Data Flow

The electoral process follows a strictly enforced, four-step chronological sequence (Fig. 2): (1) **Account Creation**—email and bcrypt-hashed password; (2) **Identity Validation**—Aadhaar ID with age ≥ 18 check and 6-digit email OTP; (3) **Biometric Enrolment**—128-D facial embedding with deduplication (Algorithm 1) and optional WebAuthn token; (4) **Authenticated Vote Casting**—live facial re-authentication within θverify triggers the serverVote() blockchain transaction.

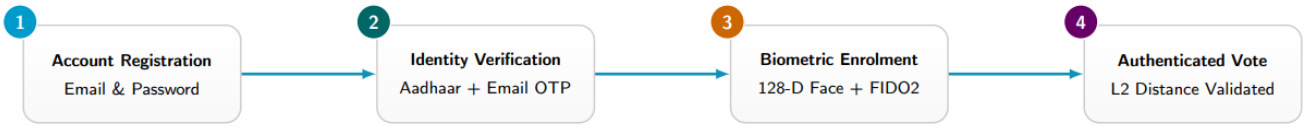

fig. 2: voter journey: four-phase sequential flow

C. Database Schema

Off-chain data is persisted across five distinct PostgreSQL tables hosted on Supabase. Biometric embeddings are stored as JSONB objects to facilitate efficient comparative queries, while SHA-256 cryptographic hashes are maintained to prevent unauthorized database modifications. WebAuthn records utilize monotonic counters to defend against credential replay attacks.

D. Smart Contract Architecture

The core electoral logic is encapsulated within Contest.sol (Solidity 0.5.16), deployed via Truffle to a Ganache network. The contract uses two primary data structures: Contestant (tracking candidate names and vote counts) and Voter (tracking ballot status), mapped against integer IDs and hashed voter identities (bytes32). Upon a successful ballot cast, the contract emits a votedEvent. Two voting methodologies are supported: vote(uint candidateId)—direct MetaMask-signed transaction; and serverVote(bytes32 voterId, uint candidateId)—gasless admin-relayed transaction following biometric clearance. Security is enforced via onlyAdmin and validState(PHASE x) modifiers.

IV. METHODOLOGY

A. Face Detection — TinyFaceDetector

Initial facial capture within the browser is executed using TinyFaceDetector [5], a lightweight CNN based on the MobileNetV1 architecture [3] (~193 KB). A valid detection requires a confidence score exceeding the base threshold: $s \geq \tau det$, $\tau det = 0.50$ (Eq. 1). For smooth real-time tracking, the feed is downsampled to 320 × 320 px; the final capture for embedding extraction is processed at 416 × 416 px.

$$s \geq \tau det, \quad \tau det = 0.50 \qquad (1)$$

B. Facial Landmark Detection — FaceLandmark68Net

Following bounding box detection, the isolated facial region undergoes geometric standardization via FaceLandmark68Net (~357 KB), mapping 68 distinct facial coordinates: $\mathscr{L} = \{(xk, yk)\}^{68}_{k=1}$ (Eq. 2). These reference points facilitate affine transformation into standardized frontal alignment, minimizing translation and rotation variation before feature extraction.

$$\mathscr{L} = \{(xk, yk)\}^{68}_{k=1} \qquad (2)$$

C. Face Embedding — ResNet-34 Recognition Model

The geometrically aligned facial image is processed by a ResNet-34 neural network [2] (~6.44 MB). Residual skip connections prevent gradient vanishing, producing a highly discriminative 128-D descriptor: $d = f\theta(Iface) \in \mathbb{R}^{128}$ (Eq. 3). Before storage, validity is checked ensuring all 128 values are finite real numbers; invalid captures trigger an immediate retry.

$$d = f\theta(Iface) \in \mathbb{R}^{128} \qquad (3)$$

D. Face Similarity — Euclidean Distance

Biometric similarity between descriptors $a, b \in \mathbb{R}^{128}$ is computed via the standard ℓ_2 (Euclidean) distance (Eq. 5). Matching faces typically yield distances below 0.6; non-matching faces score above 0.8 [2]. The verification decision rule: match(a,b) = Verified if $d(a,b) \leq \theta verify$ ($\theta verify = 0.65$); Rejected otherwise. The duplicate-detection rule: dup(a,b) = Duplicate if $d(a,b) \leq \theta dup$ ($\theta dup = 0.45$). The 0.20 margin safely absorbs legitimate lighting and angle variations while aggressively preventing Sybil attacks.

$$d(a, b) = \|a - b\|_2 = \sqrt{(\Sigma_{i=1}^{128} (a_i - b_i)^2)} \qquad (5)$$

The raw distance is inverted and scaled to a user-facing confidence score: C = max(0, min(100, round((1 − d(a,b)) × 100))) (Eq. 8).

E. Face Descriptor Integrity — SHA-256

A cryptographic digest stored alongside the descriptor safeguards against unauthorized database manipulation: hd = SHA-256(JSON.stringify(d)) (Eq. 9). Any discrepancy upon recomputation immediately flags the record as corrupted.

F. One-Face-One-Vote Duplicate Prevention

Algorithm 1 illustrates the O(n) backend deduplication executed during registration. The routine fetches all existing face profiles, computes Euclidean distance between the new descriptor and each stored profile, and rejects the submission if any distance falls at or below $\theta dup = 0.45$. Upon passing all comparisons, the descriptor and its SHA-256 hash are upserted into the database.

Algorithm 1 One-Face-One-Vote Enrolment Check

Require: New descriptor **d**new, enrolling user email e
1: $\mathcal{P} \leftarrow$ **FETCHALLFACEPROFILES**
2: **for** each profile P $\in \mathcal{P}$ where P.email $\neq$ e **do**
  3: **d**P $\leftarrow$ **NORMALIZE**(P.descriptor)
  4: **if d**P is invalid **then continue**
  5: **end if**
  6: $\delta \leftarrow d$(**d**new, **d**P) ▷ *Eq. 5*
  7: **if** $\delta \leq \theta$dup **then**
    8: **return REJECT** ▷ *"This face is already registered"*
  9: **end if**
10: **end for**
11: h $\leftarrow$ SHA256(JSON.stringify(**d**new))
12: **UPSERTFACEPROFILE**(e, **d**new, h)
13: **return ACCEPT**

The routine operates with O(n) time complexity, where n represents the total number of registered biometric profiles. While exhaustive linear scanning is highly effective for current prototype scopes, transitioning to approximate nearest-neighbour (ANN) indexing would be necessary for large-scale deployments.

G. WebAuthn/FIDO2 Fingerprint Authentication

The supplementary hardware-based biometric layer uses the W3C WebAuthn Level 2 protocol [6] via @simplewebauthn/server v13.3.0. During assertion, a monotonic counter is strictly enforced (Eq. 10): ctrnew > ctrstored. Any failure instantly denies the request, thwarting credential cloning and replay attacks.

$$ctrnew > ctrstored \quad (10)$$

H. Password Hashing — bcrypt

All user passwords are hashed using bcrypt [4] with cost factor r = 10: Hpw = bcrypt(p, r = 10) (Eq. 11), equating to 1,024 key-schedule iterations that render brute-force and rainbow-table attacks economically unfeasible.

I. Voter Anonymisation — SHA-256

The voter's email is hashed prior to blockchain transaction submission: voterID = SHA-256(evoter) (Eq. 12), ensuring a deterministic yet computationally irreversible on-chain identifier, preserving absolute ballot secrecy.

J. Smart Contract Election Lifecycle

The electoral timeline is governed by a finite state machine with three discrete phases: S = {reg = 0, voting = 1, done = 2} (Eq. 13). State progression is unidirectional and restricted exclusively to the contract administrator, guaranteeing votes cannot be cast during registration and the registry cannot be altered once voting commences.

K. OTP Generation and Biometric Session TTL

During Aadhaar verification, a cryptographically secure 6-digit OTP is generated in range [100000, 999999] with Totp = 10 minutes TTL (Eq. 16). Upon successful biometric authentication, a 10-minute biometric session window is enforced: $\Delta t = t_{now} - t_{verified} \leq T_{bio} = 600{,}000$ ms (Eq. 17). Vote attempts after TTL expiry are rejected, neutralizing risks from abandoned sessions.

V. IMPLEMENTATION

A. Technology Stack

Table 2 documents the complete suite of software technologies and frameworks utilized in BlockVote.

table 2: technology stack

Layer	Technology	Version/Notes
Frontend	EJS Templates + TailwindCSS	UI rendering
Frontend	face-api.js (ResNet-34)	v0.22.2, browser CNN inference
Frontend	SimpleWebAuthn	FIDO2/WebAuthn credentials
Frontend	Web3.js	Blockchain interactions
Backend	Node.js + Express.js	RESTful API server
Backend	bcrypt	Password hashing (r=10)
Backend	Helmet + express-rate-limit	Security hardening
Data	PostgreSQL on Supabase	Off-chain relational store
Blockchain	Solidity 0.5.16	Smart contract language
Blockchain	Truffle + Ganache	Dev/test Ethereum network
Testing	Truffle + Chai + Mocha	Contract unit testing

B. Backend REST API Design

The Express.js backend facilitates core electoral operations. Voter onboarding endpoints (POST /registerdata and POST /otpverify) validate Aadhaar IDs, enforce minimum age restrictions (≥ 18), and manage OTP dispatch and verification. Biometric endpoints (POST /api/face/register and POST /api/face/verify) execute Algorithm 1 and compute Euclidean distances for returning users. The POST /api/cast-vote endpoint confirms a valid biometric session flag, verifies TTL compliance, and confirms the user's voter registry membership before hashing the voter identity and broadcasting the serverVote transaction to Ganache.

C. Global Middleware and Security Hardening

HTTP security is enforced via Helmet middleware (strict CSP, X-Frame-Options: DENY, Strict-Transport-Security), neutralizing XSS, clickjacking, and MIME-sniffing vulnerabilities. API throttling limits all endpoints to 300 requests per 15-minute window, preventing credential-stuffing and DoS attempts. Sessions use httpOnly cookies with sameSite: 'lax' policies to defend against CSRF attacks.

D. Frontend Biometric Pipeline

The client-side facial recognition module executes a six-step workflow: (1) preload neural network weights; (2) acquire webcam via getUserMedia; (3) continuous tracking via requestAnimationFrame loop; (4) high-fidelity 416 × 416 px frame capture; (5) serialize and POST the 128-D vector; (6) render confidence metric on the interface. Critically, visual media never leaves the local machine; only mathematical representations traverse the network.

VI. Results and Evaluation

A. Smart Contract Unit Tests

The Truffle testing suite successfully verified all critical smart contract invariants (Table 3), confirming proper phase initialization, administrative access controls, and strict rejection of double-voting attempts.

table 3: smart contract unit test results

Test Case	Expected Outcome	Result
Correct phase initialization	Phase = REGISTRATION (0)	PASS
Admin-only phase transition	Non-admin rejected	PASS
Vote during REGISTRATION phase	Transaction reverted	PASS
Valid vote in VOTING phase	votedEvent emitted	PASS
Double-vote attempt	Transaction reverted	PASS
Vote after DONE phase	Transaction reverted	PASS
serverVote() non-admin call	Transaction reverted	PASS

B. Face Verification Threshold Analysis

The designated biometric parameters (Table 4) proved highly resilient. The 0.20 differential between θverify and θdup comfortably accommodated legitimate intra-user lighting and angle variations (scores 0.30–0.55) while the strict duplication threshold trapped identical biometric submissions (scores below 0.40), effectively eliminating duplicate account creation.

table 4: biometric system parameters

Parameter	Symbol	Value	Purpose
Duplicate threshold	θdup	0.45	Registration dedup
Verification threshold	θverify	0.65	Vote casting auth
Face detection threshold	τdet	0.50	TinyFaceDetector min confidence
OTP time-to-live	Totp	10 min	Aadhaar verification expiry
Biometric session TTL	Tbio	600,000 ms	Vote casting window
bcrypt cost factor	r	10	1,024 key-schedule iterations
Rate limit window	–	15 min / 300 req	DoS/brute-force mitigation

C. Security Feature Analysis

Table 5 maps the implemented security controls against their corresponding attack vectors, illustrating comprehensive architectural coverage against centralized tampering and identity spoofing.

table 5: security feature analysis

Attack Vector	Mitigation	Implementation
Vote tampering	Blockchain immutability	Ethereum smart contract
Double voting	On-chain voter registry	bytes32 voterId mapping
Identity fraud	Multi-modal biometrics	ResNet-34 + WebAuthn FIDO2
Sybil / duplicate face	θdup = 0.45 exhaustive scan	Algorithm 1 (O(n) check)
Replay attack	Monotonic WebAuthn counter	ctrnew > ctrstored enforced
Brute-force login	bcrypt + rate limiting	300 req/15 min window
XSS / Clickjacking	Helmet CSP + X-Frame-Options	DENY policy enforced
Session hijacking	httpOnly + sameSite:lax	express-session config
Biometric spoofing	Client-side only + hash integrity	SHA-256 descriptor hash
Gas-fee friction	Gasless serverVote() relay	Admin-funded transactions

VII. Limitations and Future Work

As a foundational prototype, BlockVote currently relies on trusted external infrastructure for Aadhaar and email validation. The O(n) biometric deduplication scan, while effective for small cohorts, must be upgraded to an approximate nearest-neighbour (ANN) architecture for national-scale concurrency. Future iterations should incorporate active liveness detection to prevent photographic spoofing, alongside zero-knowledge proofs (ZKPs) to mathematically guarantee absolute ballot privacy.

VIII. Conclusion

This paper presented BlockVote, a cohesive, decentralized voting architecture that bridges Ethereum smart contracts with multi-factor biometric authentication. By abstracting blockchain complexities through a gasless server-side relay and fortifying identity validation via localized facial and hardware fingerprint recognition, the system successfully addresses the most pressing vulnerabilities of modern e-voting. This implementation presents a viable, transparent, and highly secure alternative to traditional centralized voting platforms.

Acknowledgment

The authors would like to thank the Department of Computer Science & Engineering, GCET Jammu, for providing the necessary resources and support to carry out this research.

References

[1] V. Buterin, "Ethereum: A Secure Decentralised Generalised Transaction Ledger," 2014.
[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in Proc. CVPR, 2016.
[3] A. G. Howard et al., "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," arXiv:1704.04861, 2017.
[4] N. Provos and D. Mazières, "A Future-Adaptable Password Scheme," in Proc. USENIX Annual Technical Conference, 1999.
[5] V. Mühler, "face-api.js: JavaScript API for Face Recognition in the Browser," project documentation, 2019.
[6] W3C, "An API for Accessing Public Key Credentials - Level 2," W3C Recommendation, Apr. 2021.
[7] NIST, "Digital Identity Guidelines: Authentication and Authenticator Management," SP 800-63B-4, 2025.
[8] U. Jafar et al., "Blockchain for Electronic Voting System—Review and Open Research Directions," Computers & Security, 2021.

[9] M. Hajian Berenjestanaki et al., "Blockchain-Based E-Voting Systems: A Technology Review," Electronics, 2024.
[10] B. Shahzad and J. Crowcroft, "Trustworthy Electronic Voting Using Adjusted Blockchain Technology," IEEE Access, vol. 7, pp. 24477–24488, 2019.
[11] F. P. Hjalmarsson, G. K. Hreidarsdottir, M. Hamdaqa, and G. Hjalmtysson, "Blockchain-Based E-Voting System," 2018.
[12] M. Ayed, "A Conceptual Secure Blockchain-Based Electronic Voting System," 2017.
[13] K. Khan et al., "Secure Digital Voting System Based on Blockchain Technology," 2018.
[14] A. M. Al-Madani et al., "Decentralized E-Voting System Based on Smart Contract by Using Blockchain Technology," in Proc. ICSIDEMPC, 2020.
[15] S. T. Alvi et al., "A Privacy-Aware Digital Voting System Employing Blockchain and Smart Contracts," in Proc. IEEE Asia-Pacific Conf. on Computer Science and Data Engineering, 2021.
[16] J. Goyal, M. Ahmed, and D. Gopalani, "A Privacy Preserving E-Voting System with Two-Phase Verification based on Ethereum Blockchain," 2022.
[17] V. Neziri et al., "Assuring Anonymity and Privacy in Electronic Voting with Blockchain," Applied Sciences, 2022.
[18] W. A. B. W. Abdulah and S. F. S. Adnan, "Blockchain Based Electronic Voting System Design with Smart Contracts," in Proc. IEEE Symposium on Computers and Informatics, 2023.
[19] M. Sharp et al., "Blockchain-Based E-Voting Mechanisms: A Survey and a Comparative Analysis," Technologies, 2024.
[20] E. Daraghmi et al., "Secure and Transparent E-Voting Systems with Blockchain: A Comprehensive Study," Future Internet, 2024.
[21] S. A. Joni et al., "Hybrid-Blockchain-Based Electronic Voting Machine System Embedded with Deepface, Sharding, and Post-Quantum Techniques," Blockchains, 2024.
[22] S. Paudel, A. Poudel, and S. Paudel, "Enhancing Electoral Integrity and Accessibility: A Blockchain and Facial Recognition-Based Electronic Voting System," Information Dynamics and Applications, 2025.