WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal Homepage: www.ijrpr.com ISSN 2582-7421

A Prototype Blockchain and Machine Learning Integrated Framework for Secure and Transparent Student Elections

Auwal Aminu Danhari[1], Dr. Umar Iliyasu[2], Rosemary Mrumun Dima[3]

[1] *University Library Complex, Federal University Dutsin-Ma, Katsina State, Nigeria, danharee@gmail.com*

[2] *Faculty of Computing, Federal University Dutsin-Ma, Katsina State, Nigeria uiliyasu@fudutsinma.edu.ng*

[3] *Department of Cybersecurity, Faculty of Computing, Federal University Dutsinma, Katsina State, Nigeria rdzer@fudutsinma.edu.ng*

ABSTRACT

Student elections in tertiary institutions require mechanisms that can support voter authentication, prevention of duplicate voting, vote integrity and detection of abnormal voting behaviour. This study presents a prototype blockchain and machine-learning-integrated framework for student elections. The prototype combines token-based authentication, a lightweight single-node blockchain for hash-chain tamper evidence, Random Forest-based anomaly detection, and client-side face verification as an additional identity-verification mechanism. Because a labelled institutional dataset containing legitimate and fraudulent voting transactions was not available, a synthetic dataset of 6,000 simulated voting transactions was generated using predefined behavioural and security-related rules and subsequently divided into training and held-out test partitions. The Random Forest classifier was tuned using Grid Search and evaluated on 1,800 held-out transactions using accuracy, precision, recall, F1-score and ROC-AUC. The resulting accuracy was 99.00%, with precision of 97.60%, recall of 93.98%, F1-score of 95.75% and ROC-AUC of 0.999. The blockchain prototype detected all 100 deliberate tampering attempts in the controlled integrity experiment, while the mean block-mining time was 1.678 ms. Face-verification benchmarking used synthetic descriptor comparisons because a real, ethically sourced multi-subject biometric dataset was not available; this experiment produced a 0% simulated false-acceptance rate and a 14.6% simulated false-rejection rate at the configured threshold. Fourteen system test cases were executed, with one genuine face-verification defect identified and corrected before retesting. Complete vote transactions averaged 35.1 ms in the controlled test environment. The findings demonstrate the feasibility of integrating the four mechanisms within a working student-election prototype, but they do not establish real-world election-security effectiveness, increased voter trust, or deployment readiness. The principal contribution is therefore positioned as a proof-of-concept framework requiring validation with real institutional data, real multi-subject biometric captures, repeated model evaluation and distributed blockchain testing.

Keywords: Blockchain; Machine Learning; Electronic Voting; Anomaly Detection; Random Forest; Biometric Authentication

1. Introduction

Elections within tertiary institutions provide an important mechanism through which students select representatives and participate in institutional governance. However, student-election processes may face challenges involving voter authentication, duplicate voting, ballot integrity, delayed result processing and limited auditability. Electronic voting has been proposed as a means of improving administrative efficiency and accessibility, but electronic systems may introduce additional concerns involving unauthorized access, database manipulation, authentication failure and dependence on centralized infrastructure.

Blockchain based voting has attracted attention because cryptographically linked records (Nakamoto, 2008; Crosby et al., 2016) can provide evidence of subsequent modification (Kshetri & Voas, 2018; Jafar et al., 2021). Nevertheless, blockchain recording alone does not determine whether a voting transaction was generated by an authorized voter or whether a sequence of transactions represents abnormal behaviour. Similarly, token based authentication can enforce authorization and one token one vote rules but does not independently analyse behavioural patterns occurring during voting.

Machine learning based anomaly detection provides a complementary analytical mechanism by examining behavioural and security related features associated with transactions. However, its usefulness depends strongly on the quality and representativeness of the data used for training and evaluation. Biometric verification can provide another identity layer, but biometric systems introduce their own concerns, including false acceptance, false rejection, presentation attacks, privacy, consent and data retention requirements.

The security properties of an electronic voting system should therefore be distinguished rather than treated as a single measure. Vote integrity concerns whether recorded votes can be modified without detection; authentication concerns whether only authorized voters can participate; double voting prevention concerns whether an authorized voting credential can be reused; anomaly detection concerns identification of unusual or potentially fraudulent behaviour; ballot privacy concerns protection of voter choices; end to end verifiability concerns the ability to verify electoral processing; coercion resistance concerns protection against compelled voting choices; and availability concerns continued access to the voting service.

Existing studies generally address subsets of these requirements. Blockchain based approaches primarily emphasize record integrity and auditability, while machine learning approaches focus on behavioural classification or anomaly detection. Token based mechanisms address authorization, whereas biometric mechanisms address identity verification. The literature reviewed for this study therefore indicated a gap in the integration of these mechanisms within a single student election prototype.

The objective of this study was not to claim that the resulting prototype provides complete protection against every possible electoral threat. Rather, the study implemented and evaluated a prototype combining token based authentication, a lightweight hash linked blockchain, machine learning based anomaly detection and client side face verification. The specific objectives were to: (i) generate and preprocess voting related data representing authentication, token usage and voting behaviour; (ii) implement an integrated student election prototype incorporating the four security mechanisms; (iii) evaluate the Random Forest anomaly detection component using standard classification metrics; and (iv) evaluate the prototype through functional, integrity, biometric and timing tests.

2. Related Work

2.1 Blockchain Based Electronic Voting

Blockchain based electronic voting has been widely studied because cryptographically linked records can provide evidence of post record modification and improve auditability (Kshetri & Voas, 2018). Blockchains have also been characterised as institutional technologies that can change how coordination and governance are organised (Davidson et al., 2016). Goyal and Kumar (2021) implemented a blockchain based voting system as part of a sustainable e-infrastructure proposal for the Indian context, and Hassan et al. (2023) developed a decentralised election application on Ethereum using the Truffle framework, Ganache and Remix; their results showed that transaction cost was not stable and increased with network load, with throughput reaching 14 transactions per second. Review studies consolidate the field: Jafar et al. (2021) examined the current status of blockchain based voting research and its associated difficulties, Benabdallah et al. (2022) reviewed blockchain solutions for remote e-voting against security, reliability and transparency requirements, and Jafar et al. (2022) identified scalability as a fundamental barrier in a systematic review of 76 articles. Evidence from another blockchain domain in developing countries points in the same direction, with unfavourable institutional environments, high costs and technological limitations shaping deployment (Kshetri, 2021).

However, blockchain does not automatically solve authentication, ballot privacy, availability, coercion resistance or malicious client problems. In particular, a lightweight blockchain operating on a single node, as in the present prototype, does not provide the same security properties as a distributed blockchain operating through independent consensus participants.

2.2 Machine Learning and Anomaly Detection

Machine learning based approaches have investigated classification and anomaly detection for identifying unusual transactions and suspicious behaviour. Alnuaimi and Albaldawi (2024) provide an overview of widely used supervised classification methods together with their advantages and disadvantages; Logistic Regression, Decision Trees and Random Forest (Breiman, 2001) are among the established approaches. Random Forest has been applied to intrusion detection outside voting, for example in the three branch random forest model of Zhang et al. (2022). Within e-voting, Jafar et al. (2024) proposed a framework combining a modified proof of stake blockchain, a Random Forest model for real time anomaly detection and MQTT based IoT data collection, reporting a 97% security breach detection rate and a latency of 2.3 seconds, and Jumagaliyeva et al. (2024) examined the integration of blockchain and artificial intelligence for e-voting security, pairing tamper resistant records with real time anomaly detection. Reported performance from different studies cannot always be compared directly, because datasets, feature definitions, class distributions, labelling procedures and evaluation protocols differ.

2.3 Token Based and Biometric Identity Verification

Token based authentication provides a mechanism for associating voting authorization with a unique credential. When combined with database enforcement, token reuse can be rejected, supporting the one student one vote requirement. Nevertheless, token possession alone does not necessarily establish that the person presenting the token is the legitimate voter. Biometric verification has therefore been combined with blockchain voting in several studies. Bismark et al. (2025) developed a blockchain based voting system that uses the Nigerian national identification number for biometric verification and tested it with Ganache, MetaMask and Solidity, reporting a 100% valid login success rate; Osayomore et al. (2025) designed a blockchain based voting system enhanced with fingerprint and facial recognition to address impersonation in the Nigerian context; and Kalyani et al. (2025) proposed a blockchain based decentralised voting system using the SHA-256 algorithm and facial recognition.

Face verification nevertheless requires consideration of false acceptance, false rejection, presentation attacks, liveness detection, privacy, consent and retention. Results obtained from synthetic descriptor pairs should therefore not be interpreted as equivalent to performance measured on a representative real world biometric population.

A further consideration is deployment complexity. Blockchain voting systems may introduce scalability, latency, consensus and infrastructure requirements (Hassan et al., 2023; Jafar et al., 2022). Privacy mechanisms may also conflict with transparent auditability if voter identity and ballot

information are not appropriately separated. Usability and adoption barriers are likewise reported as recurring obstacles in the wider blockchain voting literature (Jafar et al., 2021). These considerations motivate an integrated but carefully bounded prototype evaluation.

2.4 Summary of Related Studies

The table below summarises the related studies discussed above, based on the features stated in each study's abstract or summary; a dash means the feature is not reported there, not that it is necessarily absent from the full text.

Study	Focus	Main contribution reported	Relation to the present study
Goyal & Kumar (2021)	Blockchain e-voting for smart city e-infrastructure (India)	Implemented a blockchain based voting system enabling remote voting	Blockchain only; no behavioural anomaly detection or biometric verification reported
Hassan et al. (2023)	Ethereum e-voting application	Decentralised election application (Truffle, Ganache, Remix); 14 transactions/s; cost rose with network load	Focus on smart contract deployment and cost; no machine learning or biometrics reported
Jafar et al. (2021)	Review of blockchain e-voting	Reviewed the status of blockchain based voting research and open challenges	Review; no prototype
Jafar et al. (2022)	Systematic review: scalability	Systematic review of 76 articles on scalable blockchain e-voting	Review; scalability focus
Benabdallah et al. (2022)	Systematic review of remote e-voting	Reviewed blockchain solutions for remote e-voting	Review; no prototype
Jafar et al. (2024)	Blockchain + Random Forest + IoT	Proof of stake blockchain with Random Forest anomaly detection; 97% breach detection, 2.3 s latency	Combines blockchain and anomaly detection; no face verification or token based one vote enforcement reported
Jumagaliyeva et al. (2024)	Blockchain + AI for e-voting security	Immutable ledger with real time AI anomaly detection; reported gains in throughput, detection accuracy and scalability	No biometric verification reported
Idowu Bismark et al. (2025)	Blockchain + national ID biometric verification	Functional tests: 100% valid login success, 100% voting test accuracy	No machine learning anomaly detection reported
Osayomore et al. (2025)	Blockchain + fingerprint and facial recognition	Design addressing impersonation in the Nigerian context	No machine learning anomaly detection reported
Kalyani et al. (2025)	Blockchain (SHA-256) + facial recognition	Decentralised voting system with facial recognition	No machine learning anomaly detection reported
Zhang et al. (2022)	Random Forest intrusion detection (non-voting)	Three branch random forest intrusion detection model	Not voting specific; no blockchain

2.5 Research Gap

The reviewed literature shows that blockchain voting, machine learning based anomaly detection and biometric verification have each been investigated, and that some studies combine two of them. Blockchain voting studies (Goyal & Kumar, 2021; Hassan et al., 2023; Kshetri & Voas, 2018) and their reviews (Jafar et al., 2021, 2022; Benabdallah et al., 2022) emphasise record integrity, transparency and scalability. Blockchain has been combined with

machine learning anomaly detection (Jafar et al., 2024; Jumagaliyeva et al., 2024) and, separately, with biometric or facial verification (Idowu Bismark et al., 2025; Osayomore et al., 2025; Kalyani et al., 2025). Among the studies reviewed here, none reported blockchain based tamper evidence, machine learning based anomaly detection, token based one vote enforcement and face verification within a single student election prototype. The present study addresses this integration gap by implementing the four mechanisms within one student election prototype.

The contribution should not, however, be interpreted as evidence that the prototype resolves all electronic election security requirements. Rather, the contribution is the implementation and controlled evaluation of an integrated architecture that combines complementary security controls while explicitly identifying the limitations of synthetic data, synthetic biometric benchmarking and single node blockchain operation.

3. Methodology

3.1 System Requirements

This subsection specifies the functional requirements, hardware and software used to implement and evaluate the prototype.

3.1.1 Specific Requirements

Specific requirements describe the logical mechanisms the system must provide beyond generic hardware/software platform needs. For this prototype, the specific requirements were:

- SHA-256 block hashing with previous hash chain linking, providing tamper evidence across the vote transaction chain (3.6)
- Proof of work block mining (leading zero hash puzzle, configured difficulty = 2), providing a genuine "mining" step rather than a simple checksum
- Session based authentication with hashed credentials, enforcing token based one vote eligibility
- A Random Forest based anomaly scoring pipeline, applied to each voting transaction in real time (3.5)
- Client side face descriptor extraction and matching for identity verification (3.7)

3.1.2 Hardware Requirements

Component	**Specification**
Processor	Intel Core i5
RAM	4 GB
Hard disk	16 GB free space

These are the specifications of the machine used for development and local testing, as reported by the author; they were not independently benchmarked.

3.1.3 Software Requirements

Component	**Specification**
Operating system	Windows
Backend	Python 3.12, Flask 3.0.3
Database	MySQL (via XAMPP), accessed through PyMySQL 1.1.1
Machine learning libraries	scikit-learn 1.5.1, pandas 2.2.2, numpy 1.26.4, joblib
Biometric libraries	face-api.js, TensorFlow.js
Frontend	HTML5, CSS3, JavaScript

Component	Specification
Other requirements	HTML5/WebRTC-enabled browser, required for webcam-based face capture

3.2 Research Design

The study adopted a prototype oriented experimental design. The research involved system implementation followed by controlled evaluation of the machine learning, blockchain, biometric and voting components.

3.3 Synthetic Data Generation

A real institutional dataset containing labelled legitimate and fraudulent student voting transactions was not available. Consequently, a synthetic dataset containing 6,000 simulated voting transactions was generated programmatically.

The dataset contained behavioural and security related attributes including transaction identifiers, authentication token identifiers, timestamps, session duration, vote count per token, failed login attempts, synthetic device and IP hashes, time since the previous vote, geographic access indicators and smart contract validation status.

A rule based generation procedure was used to produce normal and anomalous transaction patterns. The generated records included overlapping and noisy behavioural patterns rather than a perfectly separable classification boundary. This was intended to provide a controlled experimental environment for evaluating the prototype's anomaly detection mechanism.

The labels were generated from predefined simulation rules rather than from verified real world fraudulent elections. Therefore, the resulting classification metrics describe performance on the simulated problem and should not be interpreted as estimates of real world fraud detection performance.

The simulation used fixed random seeds to support reproducibility. The resulting dataset contained 5,280 normal transactions and 720 anomalous transactions, corresponding to an anomalous proportion of 12%.

Transactions were generated under predefined normal and anomalous behavioural distributions, and the corresponding class labels were assigned during the simulation: each class drew its feature values from a distinct set of probability distributions, calibrated so that the classes overlap rather than being separated by a fixed rule. The table below describes the resulting tendencies of each class; because the underlying distributions overlap, these are typical patterns rather than absolute, deterministic conditions.

Behavioural / security indicator	Normal class tendency	Anomalous class tendency
Vote count per token	Predominantly one vote per token, with a small proportion (~6%) of noisy/borderline cases	Frequent or repeated voting attempts, including multiple votes associated with a reused token
Session duration	Predominantly a typical browsing/voting duration, with an occasional short/noisy case	Predominantly a short duration suggestive of automated behaviour, with a minority resembling normal timing
Failed login attempts	Usually low, occasionally one or two	Typically elevated, often two to five attempts
Time since previous vote	Typically a longer interval between votes	Predominantly a short interval, with a minority resembling normal timing
Geographic access indicator	Usually consistent with enrolment location (flagged in a small minority of cases)	Flagged as unusual/inconsistent in a majority of cases
Device / IP hash	Predominantly distinct per voter, with a small proportion of shared device/IP activity	Predominantly shared device/IP activity, with a minority of distinct cases

Approximate behavioural tendencies used to generate the simulated dataset's normal and anomalous classes; exact probabilities are defined in the data generation script.

3.4 Data Preprocessing and Feature Engineering

The generated data were examined for duplicates, missing values and inconsistent records. The raw device and IP hash strings were converted to numeric features using a deterministic MD5-based hash encoding function (modulo 100,000), producing device_hash_enc and ip_hash_enc, so that repeated device and repeated IP behaviour could be used directly by the classifiers. A vote rate indicator (vote_count_per_token divided by session_duration) was derived to capture rapid, repeated voting activity. The final nine feature model input comprised session_duration, vote_count_per_token, failed_login_attempts, time_since_last_vote, geo_location_flag, smart_contract_status, device_hash_enc, ip_hash_enc and vote_rate_indicator; the five continuous features (excluding the two flags and two hash encodings) were standardised using a StandardScaler fitted on the training partition only.

To reduce information leakage, model evaluation was performed on a held out test partition that was not used for the final performance calculation. The dataset was divided into 70% training data and 30% held out test data. The resulting test partition contained 1,800 transactions.

3.5 Random Forest Anomaly Detection

Random Forest (Breiman, 2001) was selected as the primary anomaly detection classifier because of its ability to combine multiple decision trees and model nonlinear relationships among behavioural features.

The final configuration obtained through Grid Search was:

Parameter	**Value**
Number of estimators	100
Maximum depth	None
Minimum samples split	5
Minimum samples leaf	2
Class weight	balanced
Max features per split	sqrt
Random state	42

The classifier was evaluated using accuracy, precision, recall, F1-score and ROC-AUC on the held-out test partition. In addition, Logistic Regression and Decision Tree baseline models were evaluated using the same stratified 70:30 split, and all three models were subsequently assessed using 5-fold stratified cross validation repeated five times (4.1.1). Independent hold out evaluation using freshly generated datasets was not performed and remains a limitation for future work (5.1).

Accuracy, precision, recall and F1-score were computed from the confusion matrix counts of true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN), where the positive class is an anomalous (fraudulent) transaction:

$$Accuracy = (TP + TN) / (TP + TN + FP + FN) \quad (2)$$

$$Precision = TP / (TP + FP) \quad (3)$$

$$Recall = TP / (TP + FN) \quad (4)$$

$$F1 = 2 \times (Precision \times Recall) / (Precision + Recall) \quad (5)$$

3.6 Blockchain Implementation

A lightweight blockchain was implemented in Python. Unlike a distributed public or permissioned blockchain network, the prototype operated as a single node hash linked chain.

Each block contained an index, timestamp, transaction information, previous block hash, nonce and current block hash. SHA-256 hashing was used to calculate block hashes, computed over the JSON serialised block contents as:

$$H_n = SHA256(I_n \| T_n \| Tx_n \| H_{n-1} \| N_n) \quad (6)$$

where H_n is the hash of block n, I_n is the block index, T_n is the timestamp, Tx_n is the vote transaction payload, H_{n-1} is the previous block's hash, N_n is the proof of work nonce, and ‖ denotes concatenation within the serialised block object.

The proof of work configuration required the block hash to begin with two leading hexadecimal zero characters. This configuration was used in the reported mining benchmark.

The integrity verification procedure recomputed block hashes and checked the previous hash relationship between consecutive blocks. The experiment therefore evaluated tamper evidence within a single node hash chain, rather than distributed consensus security.

3.7 Face Verification

Face verification was implemented using face-api.js and TensorFlow.js in the client browser. A face was represented by a 128-dimensional descriptor. The photograph itself was not transmitted as the stored biometric representation; the system stored the descriptor associated with the enrolled account.

Euclidean distance was used to compare a newly generated descriptor with the enrolled descriptor, computed over the 128 elements of each descriptor vector x and y as:

$$d(x, y) = \sqrt{\left(\Sigma_{i=1}^{128} (x_i - y_i)^2 \right)} \quad (1)$$

A threshold of 0.6 was adopted as the matching criterion.

Because a real, ethically sourced multi subject biometric dataset was not available, quantitative biometric benchmarking used simulated descriptor pairs. Genuine comparisons were generated by taking a base 128-dimensional descriptor and adding small random noise to produce a second capture of the same simulated identity; impostor comparisons were generated by independently sampling two unrelated simulated identities. Five hundred genuine comparisons and 500 impostor comparisons were generated for the statistical experiment using this procedure. This simulation is distinct from, and does not replace, the qualitative real face testing reported in 3.8, in which the researcher's own face was enrolled and verified directly. These results characterise the descriptor comparison algorithm's separation behaviour under the simulation conditions and should not be interpreted as population level biometric accuracy.

3.8 System Testing

Fourteen test cases were executed across functional, integration and security related components. The tests included blockchain tampering, machine learning scoring, face matching and rejection, token reuse, fraud review and account lockout.

Thirteen tests passed during the initial execution. One test involving a stale face descriptor exposed an implementation defect: the verification process was redesigned so that the face descriptor was captured at the point of verification, and the test was repeated successfully. The final test set therefore recorded 14 successful outcomes after correction and retesting.

3.9 Development Environment

The prototype was implemented using Python 3.12, Flask, MySQL through XAMPP, PyMySQL, scikit-learn, Pandas, NumPy, Matplotlib, face-api.js, TensorFlow.js, HTML5, CSS3 and JavaScript. Model development was performed in Google Colab, while Visual Studio Code was used as the development environment on the author reported hardware specified in 3.1.2.

3.10 Ethical Considerations and Data Protection

The voting dataset used in the machine learning experiments was synthetically generated and did not contain real student voting records or personally identifiable voting information. No institutional student voting database was used for model training or classification.

Quantitative biometric benchmarking (3.7, 4.3) was performed using synthetically generated face descriptors rather than a representative multi subject biometric dataset. For functional system testing (3.8), the researcher's own face was used to verify the operation of the face verification module; no student biometric dataset was collected or used for the reported quantitative biometric experiment.

Because face descriptors constitute biometric information, any future deployment involving students would require appropriate informed consent, purpose limitation, secure storage, access control, retention and deletion policies, and compliance with applicable institutional and data protection requirements.

This study did not conduct a human subject biometric study involving a representative sample of students, and no institutional ethics approval, consent procedure or committee reference number applies, since no student biometric or voter data were collected from human participants. Consequently, no claim is made that the simulated biometric experiment represents the biometric performance of the student population; validation using an ethically sourced, representative multi subject biometric dataset would be required before deployment level conclusions could be drawn.

3.11 Threat Model

To make explicit what the prototype was and was not evaluated against, the table below separates the threats exercised during system testing (3.8) from threats that remain out of scope for this study.

Category	**Threats**
Tested / within scope	Duplicate token use; abnormal/rapid voting behaviour; stale descriptor face verification impersonation; tampering with a stored blockchain transaction; repeated failed authentication and account lockout.
Out of scope / not evaluated	Malicious or colluding administrators; Sybil attacks; denial of service; cryptographic key compromise; voter coercion; presentation attacks / liveness bypass against the face verification module; distributed consensus attacks; large scale privacy leakage.

Threats tested versus threats outside the scope of this prototype evaluation.

4. Results

4.1 Random Forest Performance

The Random Forest classifier was evaluated on the held out test set containing 1,800 simulated transactions.

Metric	**Result**
Accuracy	99.00%
Precision	97.60%
Recall	93.98%
F1-score	95.75%
ROC-AUC	0.999

The confusion matrix contained 1,579 true negatives, 5 false positives, 13 false negatives and 203 true positives, consistent with the reported accuracy, precision, recall and F1-score. The two most influential features were vote count per token and time since the previous vote, followed by the derived vote rate indicator and failed login count.

4.1.1 Baseline Model Comparison and Repeated Cross Validation

To establish whether a simpler classifier could achieve comparable performance, Random Forest was compared against Logistic Regression and a Decision Tree baseline, and the single split evaluation reported in 4.1 was extended with repeated cross validation. Both are addressed here directly on the original dataset, generation script and trained model artefacts (simulated_voting_dataset.csv, 01_generate_dataset.py, random_forest_model.pkl and scaler.pkl).

As a first sanity check, the actual saved Random Forest model and scaler were applied to the full 6,000 record dataset using the exact feature engineering pipeline defined in ml_utils.py (MD5 based hash encoding of device_hash/ip_hash, and the vote_rate_indicator formula). This returned 99.60% accuracy, 98.60% precision, 98.06% recall, 98.33% F1-score and a 0.9997 ROC-AUC. Because the original train/test split file was not part of the project export, this full dataset figure is optimistic (it likely includes rows the model was trained on) and is reported only to confirm that the feature engineering pipeline reconstructed here matches the one actually used to produce the saved model it is not offered as a held out result.

For a fair, fully held out comparison, the same stratified 70:30 split (random_state = 42) used to report the original result was reproduced on the real dataset, and Logistic Regression and a Decision Tree (both with class_weight="balanced" for consistency with the Random Forest) were trained alongside a freshly fitted Random Forest using the exact hyperparameters in 3.5. The freshly fitted Random Forest reproduced the originally reported figures almost exactly (99.00% accuracy, 97.60% precision, 93.98% recall, 95.75% F1, 0.999 ROC-AUC, with the same 1,579/5/13/203 confusion matrix), confirming that this split and pipeline match the original study's protocol:

Model	**Accuracy**	**Precision**	**Recall**	**F1-score**	**ROC-AUC**
Logistic Regression	98.11%	88.24%	97.22%	92.51%	0.999
Decision Tree	98.33%	95.59%	90.28%	92.86%	0.949
Random Forest	99.00%	97.60%	93.98%	95.75%	0.999

The same three models were then evaluated using 5-fold stratified cross validation repeated 5 times (25 folds in total) on the full real dataset:

Model	**Accuracy (mean ± SD)**	**F1-score (mean ± SD)**	**ROC-AUC (mean ± SD)**
Logistic Regression	98.50% ± 0.29%	94.00% ± 1.15%	0.998 ± 0.001
Decision Tree	98.56% ± 0.33%	93.94% ± 1.41%	0.961 ± 0.010
Random Forest	99.21% ± 0.22%	96.67% ± 0.94%	0.999 ± 0.001

On the author's original synthetic dataset, Random Forest achieved the highest accuracy, precision and F1-score in the held out evaluation, while Logistic Regression achieved higher recall and the two models obtained the same ROC-AUC of 0.999. In repeated cross validation, Random Forest achieved the highest mean accuracy and F1-score, while its ROC-AUC was comparable to Logistic Regression and showed low variability across folds. These results provide empirical support for the selected Random Forest configuration for this simulated feature set, although the differences should not be interpreted as evidence of superiority on real world election data.

4.2 Blockchain Integrity Results

The blockchain benchmark used 200 sequential vote transaction blocks and a 201-block chain including the genesis block.

Metric	**Result**
Proof of work difficulty	Two leading hexadecimal zeros
Mean block mining time	1.678 ms
Median mining time	1.139 ms
Maximum observed mining time	7.210 ms
Mean nonce iterations	290.9
Full chain validation	1.139 ms
Tampering detected	100/100
False positives on untampered chains	0/100

The results demonstrate that modifications to the tested hash linked chain were detectable under the experimental conditions. They do not demonstrate resistance to attacks against distributed consensus, malicious administrators, compromised credentials, Sybil attacks, denial of service, key compromise, coercion or privacy attacks.

4.3 Face Verification Results

The controlled descriptor experiment produced a simulated false acceptance rate of 0% and a simulated false rejection rate of 14.6% at the 0.6 threshold. The genuine descriptor mean distance was 0.562 (SD 0.036; range 0.466–0.662) and the impostor descriptor mean distance was 2.389 (SD 0.153; range 1.792–2.878); the two distributions did not overlap at all.

The 14.6% simulated false rejection rate is operationally relevant because legitimate users may potentially be denied access under unfavourable capture conditions. The result reinforces the need for validation using real, ethically sourced multi subject biometric captures before any deployment claim can be made.

A sensitivity analysis across candidate threshold values, using the same simulated descriptor data, is shown below:

Threshold	False Acceptance Rate	False Rejection Rate
0.50	0.00%	95.00%
0.60 (configured)	0.00%	14.60%
0.65	0.00%	0.80%
0.70	0.00%	0.00%
0.80	0.00%	0.00%

Because the simulated genuine and impostor distributions were fully separated (genuine maximum 0.662 versus impostor minimum 1.792), the false rejections observed at 0.6 arose from the threshold sitting close to the upper tail of the genuine distribution rather than from confusability between the two simulated populations; raising the threshold to approximately 0.7 would eliminate simulated false rejections while keeping a 0% false acceptance rate in this simulation. This is recorded as a candidate refinement for future work, to be validated against real face captures before being adopted in the live system.

During functional testing, the system also identified a stale descriptor impersonation defect. The defect was corrected through a verify at submit redesign and successfully passed retesting.

4.4 Overall System Performance

The complete vote casting request, including feature processing, machine learning scoring, blockchain sealing and database operations, averaged 35.1 ms, with a median of 34.3 ms and an observed range of 8.9–89.9 ms in the controlled environment. The login request averaged 184.4 ms, with the majority of the time attributable to the password hashing operation.

These measurements describe the observed prototype environment. They do not establish that users would perceive the system as instantaneous, since no user perception or usability study was conducted.

4.5 Comparative Analysis

The table below, “Architectural Comparison of Selected Related Studies and the Proposed Prototype,” compares which mechanisms each study reports, based on the abstract or summary of each study. A dash means the mechanism is not reported there, not that it is necessarily absent from the full text. The final column lists each study's headline reported result only to show that studies report different kinds of measure; it is not a like for like performance ranking, because datasets, objectives and evaluation protocols differ and one entry is a non voting study.

Study	Blockchain	ML anomaly detection	Face / biometric verification	Headline result reported
Goyal & Kumar (2021)	Yes	—	—	Implementation; no metrics in abstract
Hassan et al. (2023)	Yes (Ethereum)	—	—	14 transactions/s throughput

Study	Blockchain	ML anomaly detection	Face / biometric verification	Headline result reported
Jafar et al. (2024)	Yes (proof of stake)	Yes (Random Forest)	—	97% breach detection; 2.3 s latency
Jumagaliyeva et al. (2024)	Yes	Yes (AI based)	—	Reported gains in throughput, detection accuracy, scalability
Idowu Bismark et al. (2025)	Yes	—	Yes (national ID biometric)	100% login success; 100% voting test accuracy
Osayomore et al. (2025)	Yes	—	Yes (fingerprint, face)	SWOT/SPSS based evaluation
Kalyani et al. (2025)	Yes (SHA-256)	—	Yes (face)	Not summarised here
Zhang et al. (2022), non-voting	—	Yes (Random Forest)	—	Not comparable (non-voting)
This prototype	Yes (single node)	Yes (Random Forest)	Yes (face)	99.00% accuracy on synthetic held out data

The table should not be interpreted as demonstrating that the proposed prototype is numerically more accurate than the listed studies; its purpose is to show differences in the combination of mechanisms reported in the reviewed literature. Taken together, the results support the view that the four mechanisms operate as complementary, defence in depth controls rather than redundant ones: the token layer enforces eligibility, face verification binds the credential to a person, the machine learning layer detects behavioural anomalies that neither of the first two can observe, and the blockchain makes any subsequent alteration of a recorded vote detectable.

5. Discussion

The Random Forest result of 99.00% accuracy demonstrates that the classifier successfully separated the simulated normal and anomalous transaction classes under the experimental conditions. The recall of 93.98% indicates that 13 of the 216 anomalous test transactions were classified as normal, while the precision of 97.60% indicates that only five of the 208 transactions predicted as fraudulent were normal.

However, these results should be interpreted in the context of the synthetic data generation procedure. The labels were generated from predefined simulation rules rather than verified institutional fraud cases. Consequently, the results demonstrate the model's performance on the simulated classification problem rather than its expected performance against previously unseen real world election fraud.

The feature importance results indicate that transaction frequency and temporal relationships were influential to the classifier. This is consistent with the design of the simulation, where rapid repeated activity and token reuse were among the characteristics used to represent anomalous behaviour. This relationship also means that the evaluation should not be interpreted independently of the data generation assumptions.

The blockchain experiment demonstrated reliable detection of deliberate modifications to the tested hash linked chain, supporting the prototype's intended tamper evidence property. Nevertheless, the implementation was single node and did not test distributed consensus. The result should therefore not be generalised to decentralised blockchain security.

The biometric experiment demonstrated separation between simulated genuine and impostor descriptor distributions under the selected simulation procedure. The 14.6% simulated false rejection rate, however, indicates that a threshold suitable for one controlled simulation cannot automatically be assumed to provide acceptable operational performance across a diverse population. Real world lighting, pose, camera quality, facial variation and presentation attacks could affect performance.

The functional testing also demonstrates the value of adversarial testing during prototype development. The stale descriptor test initially failed, exposing a genuine identity verification weakness; the defect was corrected and successfully retested. Reporting this failure and correction provides a more transparent account of the prototype evaluation than reporting only successful test cases.

The measured vote processing time indicates that the integrated prototype can execute the tested sequence within a relatively short processing interval in the development environment. However, no large scale load test or user perception study was performed; the result should therefore be reported as prototype latency rather than evidence of institutional scalability or instantaneous user experience.

Random Forest was also compared against Logistic Regression and a Decision Tree, and the evaluation was extended with repeated cross validation (4.1.1). Across both a held out 70:30 split and 25-fold repeated cross validation, Random Forest achieved the highest F1-score and showed low ROC-AUC variability across folds, comparable to Logistic Regression, though its advantage in raw accuracy over Logistic Regression was modest (all three models exceeded 98%). This supports, without overstating, a preference for Random Forest for this specific feature set and labelling scheme, while a comparison against a simple rule based detector remains outstanding.

5.1 Limitations

Several limitations should be considered when interpreting the findings.

1. The machine learning dataset was synthetic. Although the simulation incorporated behavioural variation and noise, its labels were derived from predefined rules rather than verified fraudulent election records. The reported ML metrics therefore describe performance under controlled simulation conditions.

2. The evaluation is now supported by repeated stratified cross validation (5-fold, 5 repeats) alongside the original 70:30 split (4.1.1), which reduces but does not eliminate the value of further independent repeated hold out experiments on additional freshly generated datasets in future work.

3. Baseline comparison against Logistic Regression and a Decision Tree is now reported in 4.1.1 on the author's real dataset: Random Forest outperforms both on F1-score and cross validated stability, though the margin over Logistic Regression on raw accuracy is modest. Comparison against a simple rule based detector was not performed and remains a gap.

4. The biometric quantitative evaluation used synthetic descriptor pairs rather than a representative real multi subject biometric dataset. The reported false acceptance and false rejection values should be regarded as simulation results.

5. The blockchain implementation was a lightweight single node Python hash chain. It demonstrated tamper evidence but did not evaluate distributed consensus, multi node failures or attacks against a decentralised network.

6. No student trust survey, usability study, turnout analysis or live institutional election was conducted. Claims concerning increased trust, adoption or institutional election effectiveness are therefore outside the evidence provided by this study.

7. The prototype does not claim comprehensive protection against malicious administrators, Sybil attacks, denial of service, key compromise, coercion, privacy leakage, presentation attacks or other threats that were not experimentally evaluated.

6. Conclusion

This study implemented a prototype framework integrating token based authentication, a lightweight blockchain hash chain, machine learning based anomaly detection and client side face verification for student elections.

The Random Forest classifier achieved 99.00% accuracy, 97.60% precision, 93.98% recall, 95.75% F1-score and 0.999 ROC-AUC on the held out synthetic test set. The blockchain experiment detected 100 of 100 deliberate modifications under the tested single node hash chain configuration. The biometric simulation produced a 0% simulated false acceptance rate and a 14.6% simulated false rejection rate, while functional testing identified and corrected a stale descriptor verification defect. Complete vote casting requests averaged 35.1 ms in the controlled experimental environment.

Taken together, these results demonstrate the technical feasibility of integrating the four mechanisms in a student election prototype. They do not, however, establish real world election security effectiveness, institutional scalability, increased student trust, or readiness for deployment in a live election.

The principal contribution of the study is therefore a proof of concept integrated framework and its controlled prototype evaluation. Stronger evidence would require real, ethically sourced voting data, repeated and independent machine learning evaluation, baseline model comparisons, representative biometric testing, liveness detection, distributed blockchain deployment and field based evaluation involving actual users.

7. Future Work

Future research should focus on five areas. First, real institutional voting data should be collected under appropriate ethical and data protection procedures to evaluate whether the simulated anomaly patterns correspond to real voting behaviour. Second, having now compared Random Forest against Logistic Regression and a Decision Tree (4.1.1), the remaining comparison is against a simple rule based detector, together with independent hold out evaluation on freshly generated or real datasets to check that the cross validated advantage reported here generalises.

Third, biometric evaluation should use a representative, ethically sourced multi subject dataset and include false acceptance, false rejection and liveness/presentation attack testing. Fourth, the blockchain component should be extended to a multi node architecture with explicit consensus and failure

testing. Finally, field studies should evaluate usability, student acceptance, trust, accessibility, scalability and electoral outcomes before any institutional deployment is considered.

References

Alnuaimi, K., & Albaldawi, N. (2024). An overview of machine learning classification techniques. BIO Web of Conferences, 97, 00133. https://doi.org/10.1051/bioconf/20249700133

Benabdallah, A., Audras, A., Coudert, L., El Madhoun, N., & Badra, M. (2022). Analysis of blockchain solutions for e-voting: A systematic literature review. IEEE Access, 10, 70746–70759. https://doi.org/10.1109/ACCESS.2022.3187688

Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5–32. https://doi.org/10.1023/A:1010933404324

Crosby, M., Nachiappan, Pattanayak, P., Verma, S., & Kalyanaraman, V. (2016). Blockchain technology: Beyond bitcoin. Applied Innovation Review, 2, 6–19.

Davidson, S., De Filippi, P., & Potts, J. (2016). Disrupting governance: The new institutional economics of distributed ledger technology. SSRN. https://doi.org/10.2139/ssrn.2811995

Goyal, M., & Kumar, A. (2021). Sustainable e-infrastructure for blockchain-based voting system. In Digital cities roadmap: IoT-based architecture and sustainable buildings (chap. 7). Wiley. https://doi.org/10.1002/9781119792079.ch7

Hassan, H. S., Hassan, R., & Gbashi, E. K. (2023). E-voting system based on Ethereum blockchain technology using Ganache and Remix environments. Engineering and Technology Journal, 41(4), 562–577. https://doi.org/10.30684/etj.2023.135464.1273

Idowu-Bismark, O., Oshin, O., & Adetiba, E. (2025). Development of a blockchain-based electronic voting system utilizing national identification number. International Journal of Reconfigurable and Embedded Systems, 14(3), 810–820. https://doi.org/10.11591/ijres.v14.i3.pp810-820

Jafar, U., Ab Aziz, M. J. A., & Shukur, Z. (2021). Blockchain for electronic voting system—review and open research challenges. Sensors, 21(17), 5874. https://doi.org/10.3390/s21175874

Jafar, U., Ab Aziz, M. J. A., Shukur, Z., & Hussain, H. A. (2022). A systematic literature review and meta-analysis on scalable blockchain-based electronic voting systems. Sensors, 22(19), 7585. https://doi.org/10.3390/s22197585

Jafar, U., Ab Aziz, M. J. A., Shukur, Z., & Hussain, H. A. (2024). Secure-Tech Triad: Enhancing electronic voting system security through integrated blockchain, AI, and IoT technologies. ITM Web of Conferences, 63, 01011. https://doi.org/10.1051/itmconf/20246301011

Jumagaliyeva, A., Muratova, G., Tulegulov, A., Rystygulova, V., Serimbetov, B., Yersultanova, Z., & Shegetayeva, A. (2024). The impact of blockchain and artificial intelligence technologies in network security for e-voting. International Journal of Electrical and Computer Engineering, 14(6), 6723–6733. https://doi.org/10.11591/ijece.v14i6.pp6723-6733

Kalyani, B. J. D., Modadugu, J. K., & Neelima, S. (2025). Blockchain-based decentralized voting system with SHA-256 algorithm and facial recognition. International Journal of Reconfigurable and Embedded Systems, 14(2), 481–489. https://doi.org/10.11591/ijres.v14.i2.pp481-489

Kshetri, N. (2021). Blockchain and sustainable supply chain management in developing countries. International Journal of Information Management, 60, 102376. https://doi.org/10.1016/j.ijinfomgt.2021.102376

Kshetri, N., & Voas, J. (2018). Blockchain-enabled e-voting. IEEE Software, 35(4), 95–99. https://doi.org/10.1109/MS.2018.2801546

Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. https://bitcoin.org/bitcoin.pdf

Osayomore, E. G., Ejenarhome, O. P., & Oise, G. P. (2025). Evaluation of electronic based voting system and design of blockchain-based electronic voting system enhanced with fingerprint and facial recognition technologies to address impersonation. FUDMA Journal of Sciences, 9(7), 13–25. https://fjs.fudutsinma.edu.ng/index.php/fjs/article/view/3701

Zhang, C., Wang, W., Liu, L., Ren, J., & Wang, L. (2022). Three-branch random forest intrusion detection model. Mathematics, 10(23), 4460. https://doi.org/10.3390/math10234460