Asian Journal of Electrical Sciences
ISSN: 2249-6297 (P); 2583-7923 (E)
Vol.13 No.2, 2024, pp.25-35

www.crijournals.org
DOI: https://doi.org/10.70112/ajes-2024.13.2.4250

Electoral System Optimization Through Voting Quorum, Voter Turnout, Candidate Viability, and Electoral Integrity Analysis

Mobolaji Olusola Olarinde*, Akeem Adekunle Abiona and Micheal Olalekan Ajinaja
Department of Computer Science, Federal Polytechnic Ile-Oluji, Nigeria
E-mail: akeabiona@fedpolel.edu.ng, micajinaja@fedpolel.edu.ng
*Corresponding Author: mobolarinde@fedpolel.edu.ng
(Received 15 August 2024; Revised 1 September 2024, Accepted 4 October 2024; Available online 12 October 2024)

***Abstract* - Electoral systems are pivotal in shaping democratic governance; however, they often face challenges related to voter turnout, candidate viability, and electoral integrity. These challenges necessitate innovative solutions to optimize electoral processes while ensuring accuracy and fairness. This study aims to optimize electoral systems by analyzing the impacts of voting quorum, voter turnout, candidate viability, and electoral integrity. The goal is to develop and test a software-based voting system that addresses these critical factors while maintaining robustness and efficiency. A digital voting platform was designed, allowing users to log in using their surname and matriculation number. Each voter was enabled to cast a single vote per post, ensuring accuracy and uniqueness. The system incorporated automated candidate and voter registration processes, distributing unique voting credentials via email to registered voters. A mathematical model leveraging linear and nonlinear programming techniques was formulated to optimize electoral outcomes, addressing constraints such as quorum requirements, voter turnout, candidate viability, and integrity safeguards to prevent duplicate and invalid votes. The integration of the proposed parameters in the voting system significantly improved its integrity and efficiency. Empirical testing demonstrated high accuracy in vote counting and robust protection against fraudulent activities. The model achieved optimized outcomes by balancing the constraints and promoting an equitable electoral process. The findings highlight the effectiveness of incorporating mathematical modeling and secure software solutions in optimizing electoral systems. This framework provides a scalable and reliable approach to addressing the challenges of modern electoral processes, offering practical implications for enhancing democratic practices in electronic voting systems.**
***Keywords:* Electoral Systems, Voter Turnout, Candidate Viability, Voting System, Mathematical Modeling**

I. INTRODUCTION

The integrity and effectiveness of electoral systems are fundamental to the democratic process, as they ensure that the collective will of the electorate is accurately and fairly represented. With the advancement of technology, electronic voting systems have been increasingly adopted to enhance the efficiency, security, and accessibility of elections [1]. However, the optimization of these systems remains a critical challenge, particularly in balancing factors such as voter turnout, candidate viability, and electoral integrity [2]. This paper explores these aspects within the context of a university student election system, aiming to develop a mathematical model that optimizes electoral outcomes by addressing key variables: voting quorum, voter turnout, candidate viability, and electoral integrity.

The voting process in the developed system allowed students to log in using their surname and matriculation number to cast a single vote per post, with all votes being automatically counted. On the administrative side, the system facilitated the registration of candidates and voters, ensured that voters could only vote once, and provided each registered voter with a unique set of voting credentials via email. This system design sought to address common issues in electronic voting, such as vote duplication, unauthorized access, and inaccurate vote counting [3]. A critical aspect of electoral optimization is ensuring sufficient voter turnout, which directly impacts the legitimacy and acceptance of election results [4]. The implementation of a voting quorum - requiring a minimum percentage of registered voters to participate - is one method to address this challenge. Previous studies have shown that setting a quorum threshold can mitigate the effects of low turnout and prevent decisions from being made by an unrepresentative minority [5]. However, achieving this balance requires careful consideration of the quorum level, as excessively high thresholds may invalidate legitimate elections, while a too-low threshold may compromise the credibility of the results [6].

Candidate viability is another important factor in the optimization of electoral systems. Viability, typically defined by a candidate's ability to garner a significant proportion of votes, is crucial for maintaining a competitive and fair election environment [7]. The mathematical model developed in this study incorporates a viability threshold to ensure that only candidates who achieve a minimum level of support are considered viable, thus reducing the number of fringe candidates and streamlining the decision-making process [8]. Electoral integrity encompasses the mechanisms and safeguards that ensure the election process is conducted fairly, transparently, and free from fraud or manipulation [9]. In the context of electronic voting, integrity is maintained through measures such as secure authentication, robust vote counting, and stringent access controls. The system developed in this study integrated these elements to create a reliable and secure voting environment that minimizes the risk of electoral malpractice.

This paper presents a comprehensive mathematical model that optimizes the electoral process by balancing voter turnout, candidate viability, and electoral integrity. By applying this model to a university student election, we demonstrate its effectiveness in enhancing the overall quality and reliability of the voting system. The findings offer valuable insights into the critical factors influencing electoral success in electronic voting systems and provide a framework for future improvements in electoral design and implementation.

The manuscript is organized into different sections to ensure clarity, progression, and a thorough examination of the proposed electoral system optimization. Section 2 offers a detailed review of existing literature, pinpointing research gaps and emphasizing the unique contributions of the study. This review sets up the innovative approach of the proposed work by comparing it with prior research. Section 3 explores the role of technology in online voting, focusing on the benefits of voting quorum, voter turnout, candidate viability, and electoral integrity analysis. It highlights how these technologies can enhance the security and integrity of the voting process. Section 4 presents the framework architecture of the proposed internet voting system, describing its modular components and overall design. It provides a foundational blueprint for implementing the system. Section 5 looks into the specifics of the system's implementation, including the voting algorithm and the process from voter registration to vote casting. It also builds on the results by addressing potential challenges and limitations and suggesting future research directions. It reflects critically on the findings, discussing broader implications and possible improvements. Finally, Section 7 summarizes the research journey, encapsulating key insights, findings, and implications for the realm of online voting. This concluding section synthesizes the contributions of the study, highlighting its impact on the optimization of electoral systems and its potential to advance the field of electronic voting through the integration of advanced technologies and robust mathematical modeling.

II. REVIEW OF LITERATURE

The optimization of electoral systems, particularly within the context of online voting, has been extensively researched in recent years. The central themes often explored include voter turnout, candidate viability, electoral integrity, and the implementation of secure and efficient online voting systems. One critical aspect of electoral systems is voter turnout. Studies have shown that the introduction of online voting systems significantly impacts voter turnout. For instance, research conducted by [10] indicated that online voting systems generally increased voter participation by offering convenience, which was particularly beneficial in settings such as universities where students were often geographically dispersed. Similarly, [11] discussed the relationship between online voting and voter engagement, noting that the ease of access and the elimination of physical barriers to voting led to higher turnout rates.

Another important aspect is the impact of online voting on candidate viability. [12] explored how online voting systems influenced voter behavior and candidate viability. The study suggested that the transparency of the process and the ease of obtaining information about candidates online led to more informed voting decisions, which affected the viability of candidates who relied on less informed voters. Additionally, [13] discussed how the design of the voting system, including features such as the display order of candidates, subtly influenced voter choices, potentially impacting candidate viability.

Electoral integrity is a critical issue in the design and implementation of online voting systems. The work of [14] emphasized the importance of security measures to prevent fraud and ensure the integrity of the electoral process. The study outlined several security protocols, including encryption, voter authentication, and audit trails, which are essential to maintaining the trustworthiness of online voting systems. Similarly, [15] highlighted the challenges of ensuring transparency and security in online elections, advocating for the use of advanced cryptographic methods to protect voter anonymity and prevent tampering.

The concept of a voting quorum was another area of interest in electoral system optimization. Research by [16] explored the implications of different quorum requirements on the legitimacy and acceptance of election outcomes. They argued that online voting systems could be designed to dynamically adjust quorum thresholds based on real-time voter turnout, thereby ensuring that election results are both representative and legitimate. In the context of academic institutions, where voting systems are often used for student government elections, specific considerations are necessary. For example, [17] focused on the design of online voting systems for universities, highlighting the importance of user-friendly interfaces and robust security measures to ensure widespread adoption and trust among student voters. Additionally, their work discussed the integration of SMS and email notifications to enhance voter engagement, which aligns with the system described in this paper.

The existing literature provides a solid foundation for understanding the complexities of optimizing electoral systems, particularly within the context of online voting. The research highlights the importance of voter turnout, candidate viability, electoral integrity, and quorum requirements, all of which are critical to the design of effective online voting systems. The studies cited offer valuable insights that informed the development and implementation of the system proposed in this paper.

A. The Role of Technology in Online Voting

Online voting has become increasingly prevalent, offering numerous benefits over traditional voting methods. Key to its success are technologies that enhance Voting Quorum, Voter Turnout, Candidate Viability, and Electoral Integrity. Each of

these aspects plays a crucial role in ensuring a secure and effective voting process.

Voting Quorum refers to the minimum number of votes required for an election to be valid. Technology facilitates the management of voting quorums by automating the process of tracking and counting votes. Online voting systems can instantly tally votes and determine if the quorum has been met, thereby streamlining the election process and reducing the chances of human error [18]. Furthermore, automated quorum checks ensure that elections proceed only when sufficient participation is achieved, which enhances the legitimacy of the results.

Voter Turnout is a critical factor in the democratic process, as higher turnout generally reflects a more representative electorate. Online voting systems can improve voter turnout by making voting more accessible and convenient. With technology, voters can cast their ballots from anywhere with internet access, eliminating barriers such as geographical distance and time constraints [19]. Additionally, the use of email notifications to provide voters with their login credentials ensures that they are reminded to participate, further increasing turnout [20].

Candidate Viability is assessed through the voting process, which technology can enhance by providing real-time data on voter preferences and trends. Online systems can analyze vote patterns to offer insights into candidate performance and public support. This capability allows for a more dynamic and responsive electoral process, where candidates can adjust their campaigns based on up-to-date information [21]. Such data-driven approaches ensure that the most viable candidates emerge from the process, reflecting the true preferences of the electorate.

Electoral Integrity is paramount to maintaining trust in the electoral process. Technology plays a significant role in ensuring the security and accuracy of online voting systems. Measures such as secure login procedures, encrypted data transmission, and automated vote counting help protect against tampering and fraud. Moreover, technology enables detailed auditing and verification processes, allowing for a thorough examination of the vote counts and ensuring that results are accurate and trustworthy [22]. Online voting systems also employ mechanisms to prevent duplicate voting and ensure that each voter casts only one ballot. This is crucial for maintaining the integrity of the electoral process and ensuring that results accurately reflect the votes cast by eligible voters [23]. The combination of these technologies contributes to a more secure, transparent, and reliable voting experience.

III. METHODOLOGY

The methodology section outlines the approaches and processes involved in developing, implementing, and analyzing the proposed electoral system. This section also covers the mathematical models used to optimize voter turnout, candidate viability, and electoral integrity. The system was developed using the Agile development methodology, which allows for iterative design and continuous feedback from stakeholders. The development process was broken down into the following phases:

1. *Requirement Analysis:* During the requirement analysis phase, the needs of both the voters and the administrators were gathered through interviews, surveys, and reviews of existing systems. This phase ensured that all necessary features, such as secure voter authentication, real-time vote counting, and administrative oversight, were accurately identified and documented.
2. *System Design:* Based on the requirements, a detailed system design was created, which included the system architecture, database schema, user interfaces, and security protocols. This design phase incorporated feedback from the requirement analysis and laid the foundation for the development process.
3. *Development and Integration:* The development phase involved coding the system components, including the front-end interfaces, back-end services, and database interactions. The system was developed using a combination of modern web technologies (HTML5, CSS3, JavaScript) for the front end and a robust server-side framework (Node.js/Django) for the back end. The integration of these components ensured seamless interaction between the user interfaces and the underlying data processes.
4. *Testing:* Comprehensive testing was conducted at various stages of development, including unit testing, integration testing, and system testing. Special attention was given to security testing to ensure that the system was resistant to common vulnerabilities, such as SQL injection, cross-site scripting (XSS), and unauthorized access.
5. *Deployment:* The final system was deployed on a cloud-based server, ensuring scalability and availability. Continuous monitoring was set up to detect and address any issues that might arise during the operational phase.

To ensure the integrity of the election, simple statistical models were used to detect potential electoral fraud and anomalies. The single-vote constraint is expressed as $\sum_{j=1}^{m} x_{ij} = 1$ for each voter i, where x_{ij} is a binary variable (1 if voter i votes for candidate j, 0 otherwise). The integrity factor (I) includes the absence of duplicate votes, invalid votes, and ensures the correct execution of the voting process.

A. Quorum Calculation

The system also incorporated a dynamic quorum calculation model to ensure that the election results were valid and representative. The quorum was calculated as a function of the total number of eligible voters and the voter turnout rate. The system dynamically adjusted the quorum requirement based on real-time turnout data to maintain the legitimacy of the election.

$$Q = T\ x\ \left(\frac{V_{actual}}{V_{expected}}\right) \quad (1)$$

Where Q is the quorum threshold, T is the total number of eligible voters, $V_{\_actual}$ is the actual voter turnout, and $V_{\_expected}$ is the expected voter turnout. The quorum threshold (q) is set as a percentage (e.g., 50%). The quorum is met if T≥q; if the quorum is not met, the election is considered invalid. The next phase is candidate viability. This dynamic adjustment ensures that the quorum requirement reflects the actual level of voter engagement, thereby enhancing the election's credibility. Let C={C1,C2,…,Cm} be the set of m candidates, and let the votes received by candidate i be denoted by v_i. Viability index for candidate *i* :

$$C_i = \frac{v_i}{V} \text{ (proportion of votes received)} \quad (2)$$

A Viability threshold is set is *c*. A candidate is viable if $C_i \geq c$. To maximize the electoral outcome based on voter turnout, candidate viability, and integrity:

$$\text{Maximize } O = f(T, C_i, I) \quad (3)$$

$$\text{A potential form could be } O = \alpha T + \beta \sum_{i=1}^{m} C_i + \gamma I \quad (4)$$

Where α, β, γ are weights to balance the importance of turnout, candidate viability, and integrity. Voter Turnout Constraint (V) is $\sum_{i=1}^{N} v_i$ (sum of all votes cast should equal the actual voter count), quorum Constraint (T ≥ q), Candidate Viability Constraint ($C_i \geq c$) for viable candidates, and Single Vote Constraint ($\sum_{j=1}^{m} x_{ij} = 1$ for each voter *i*) are represented. To prevent duplicate votes, each voter *i* can vote only once, ensuring $x_{ij} \in \{0,1\}$ and no votes should be counted if cast by unregistered voters or in a duplicate manner.

$$Maximize\ O = \alpha \left(\frac{V}{N}\right) + \beta \sum_{j=1}^{m} \left(\frac{\sum_{i=1}^{N} x_{ij}}{V}\right) + \gamma I \quad (5)$$

Subject to

1. Voter Turnout ($V = \sum_{i=1}^{N} \sum_{j=1}^{m} x_{ij}$),
2. quorum requirement ($\frac{V}{N} \geq$ q),
3. candidate viability ($\frac{\sum_{i=1}^{N} x_{ij}}{V} \geq$ c), and
4. single vote constraint ($\sum_{j=1}^{m} x_{ij} = 1,\ \forall_i$).

For integrity constraints, no voter is allowed to cast multiple votes, and only votes from registered voters are counted (handled by the variable x_{ij}, which is a binary and controlled variable). The objective function seeks to maximize the overall electoral outcome, which is a weighted sum of voter turnout, candidate viability, and electoral integrity. The constraints ensure that the voting process adheres to rules such as the quorum requirement, candidate viability, and the maintenance of electoral integrity.

B. Data Collection and Analysis

Data collection for testing and validating the system was carried out in several stages. A pilot test was conducted in a controlled environment using a sample group of voters. This test allowed for the assessment of system functionality, voter experience, and the effectiveness of the mathematical models in real-world scenarios. During the pilot, data on voter behavior, system performance, and voting patterns were collected. This data was used to fine-tune the predictive models and validate the accuracy of voter turnout predictions, candidate viability rankings, and fraud detection mechanisms.

The methodology employed in developing the electoral system was comprehensive, combining rigorous system development practices with advanced mathematical modeling. The integration of predictive models, multi-criteria decision analysis, and statistical fraud detection techniques ensured that the system was not only functional but also optimized for fairness, security, and accuracy. Through iterative testing and validation, the system was refined to meet the needs of both voters and administrators, ensuring a reliable and trustworthy electoral process.

C. System Architecture

The architecture diagram illustrates the overall structure of the electoral system, showing how different components interact to facilitate the voting process:

1. *User Interface:* This component is where voters interact with the system. It provides the front-end interface for voters to log in, select candidates, and cast their votes.
2. *Admin Interface:* This is the front-end interface for administrators, allowing them to manage the election process, including voter registration, candidate management, and monitoring of vote counts.
3. *Authentication Service:* This central component handles the authentication of both voters and administrators. It verifies login credentials and ensures that only authorized users can access the system.
4. *Voter Registration Module:* This module allows administrators to register voters by inputting their details into the system. It interacts with the authentication service to manage voter access.
5. *Candidate Management Module:* Administrators use this module to register and manage candidates for the election. It ensures that candidate information is accurately reflected in the voting interface.
6. *Voting Module:* This is where the actual voting takes place. It allows voters to select their preferred candidates and submit their votes, which are then securely recorded.
7. *Vote Counting and Compilation:* This module automatically counts the votes as they are cast and compiles the results. It provides real-time updates to the administrators.
8. *Database:* The database stores all essential data, including voter information, candidate details, and cast votes. It is central to the operation of the system and ensures data integrity.
9. *Security & Audit Trail:* This component ensures the security of the electoral process. It includes encryption, fraud detection mechanisms, and maintains a comprehensive audit trail to track all actions within the system.

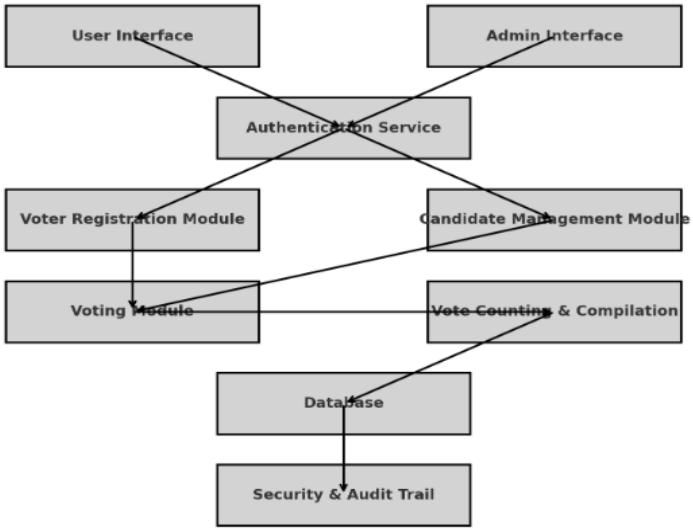

Fig. 1 Architecture diagram for the application

Arrows between components indicate the flow of information, illustrating how data is processed from user input to final vote compilation while ensuring security and integrity throughout the process.

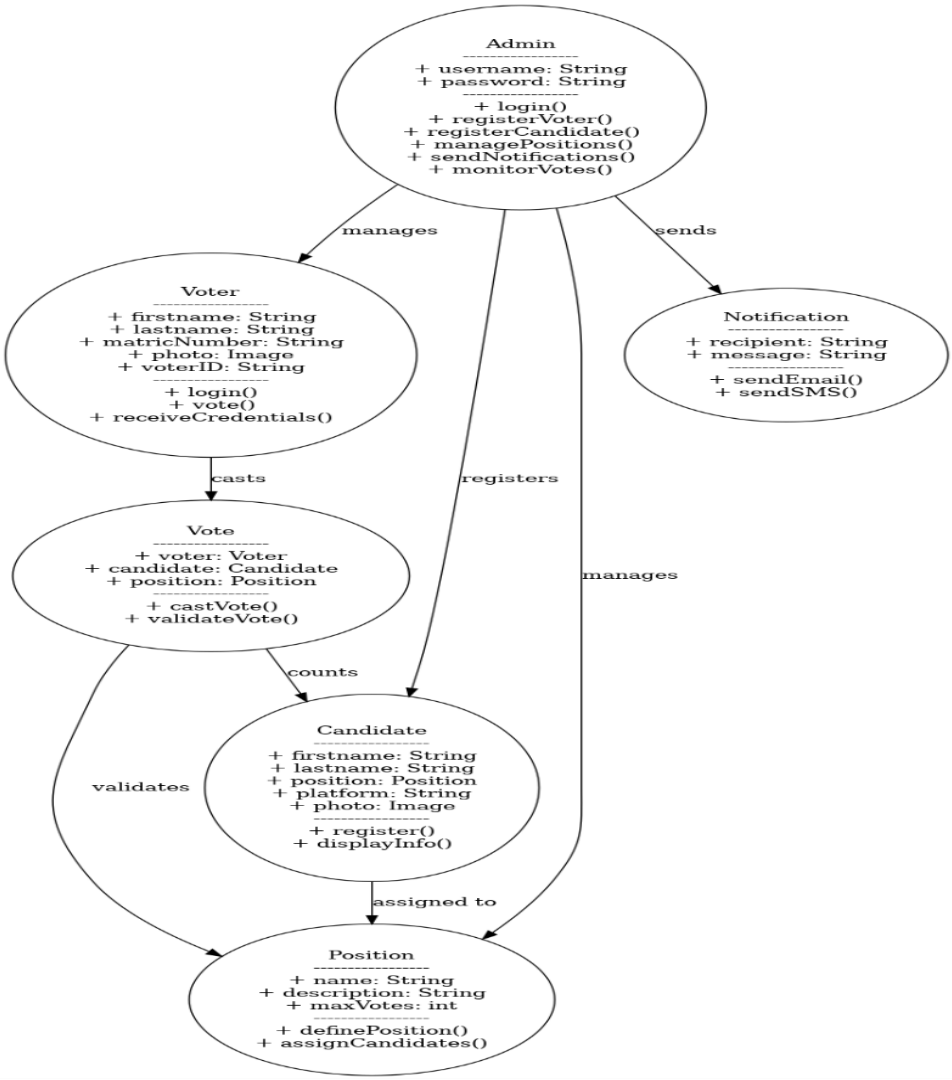

Fig. 2 Class diagram for the application

D. Use Case Diagram

The use case diagram outlines the primary actors - Admin and Voters - and their interactions with the system. For the Admin, use cases include registering candidates and voters, managing positions and candidate data, sending emails and SMS, and overseeing election integrity by monitoring votes and voter turnout. Voters interact with the system by logging in using their surname and matriculation number, receiving credentials via email, and casting their vote once per position.

The class diagram models the structure of the system by defining classes such as Admin, Voter, Candidate, Position, Vote, and Notification. Each class includes relevant attributes and methods; for example, the Voter class contains attributes like first name, last name, voter ID, and methods for validating a single vote per voter. The Admin class has methods for managing the election, including adding voters, candidates, and positions, as well as sending notifications, as shown in Fig. 2. The sequence diagram represents the flow of activities during the voting process. It begins with the Voter receiving an email containing login credentials, followed by logging into the system and casting their vote. The system validates the vote, updates the tally for the selected candidate, and confirms the vote. For the Admin, the sequence includes logging in, accessing the dashboard, uploading voter details, registering candidates, and monitoring vote counts and positions.

The activity diagram captures the process flows in the system, such as the voting process and admin management activities. For voting, the diagram shows the steps from voter login, presentation of the voting interface, vote submission, confirmation, and logout. Admin activities include managing voter and candidate registration, sending notifications, and overseeing the election process on the dashboard.

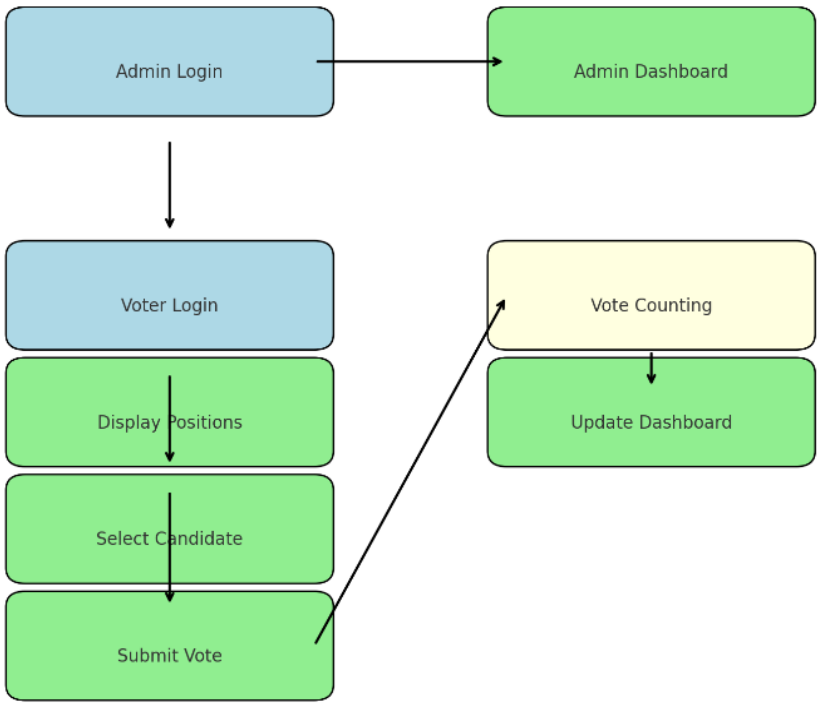

Fig. 3 Activity diagram for the application

The diagram details the various modules within the admin interface: Dashboard (summarizes election data), Votes Module (shows positions, candidates, and votes), Voters Module (manages voter data upload), Position Module (defines election positions), Candidates Module (displays candidate information), Send SMS Module (handles notifications), and Ballot Position Module (displays candidate positions).

The diagrams collectively provide a comprehensive visual representation of how the voting system operates, emphasizing the relationships between different system components and the flow of activities essential for ensuring the system's integrity, efficiency, and usability.

E. Entity Relational Diagram

The Entity-Relationship Diagram (ERD) for the electoral system encapsulates the interactions and data flow between various components of the system. At its core, the ERD integrates the Admin, Candidate, Voter, Position, Vote, SMS, and Ballot_Position entities, each with specific roles and relationships. The Admin is the central figure responsible for managing all aspects of the system. They handle the registration and oversight of Candidates, Voters, and Positions. This entity ensures that each Candidate is appropriately associated with a Position and that Voters are correctly set up to participate in the election. The Admin also has the ability to send SMS messages to Voters, containing critical voting credentials and instructions.

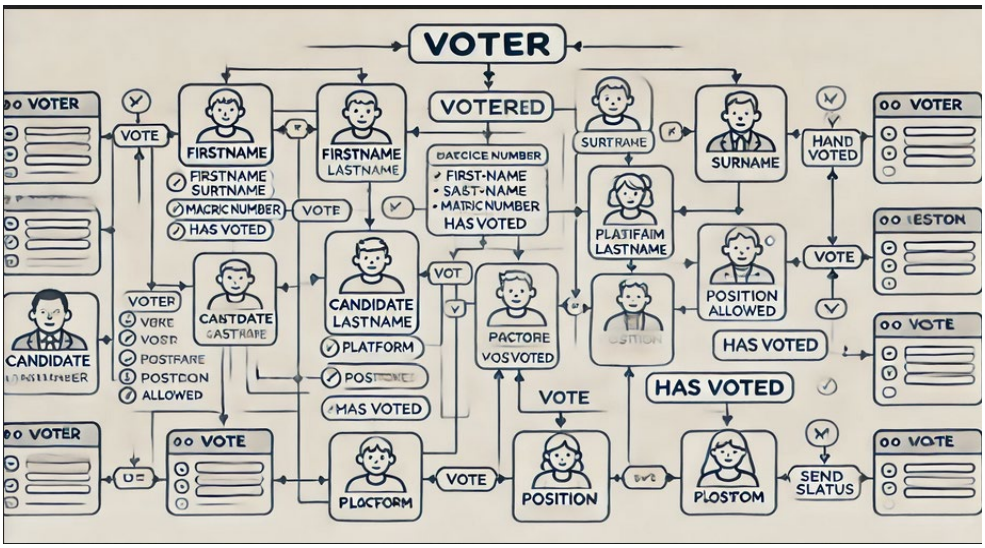

Fig. 4 Entity Relational Visualization diagram for the application

Candidates are linked to specific positions and are the recipients of votes cast by voters. Each candidate's profile includes personal details and a platform statement, reflecting their campaign. Voters, on the other hand, are uniquely identified and permitted to cast a single vote for a candidate in each position. Votes serve as the connection between voters, candidates, and positions. Each vote records which voter selected which candidate for which position, ensuring accurate and traceable vote counting.

The position entity defines the roles or offices up for election, including a description and the maximum number of votes that can be assigned. Each position can have multiple candidates vying for it and receives a tally of votes cast. SMS functionality plays a supportive role in the system, allowing admins to send messages to voters. These messages include essential voting information, enhancing the system's communication efficiency.

Finally, the ballot_position module provides an overview of the electoral standings, displaying the current position of candidates within their respective positions. This module aggregates and presents the results to facilitate transparency and integrity in the election process. Together, these components form a cohesive system where each entity's attributes and relationships contribute to a streamlined and secure electoral process.

IV. ARCHITECTURAL FRAMEWORK OF THE SYSTEM

The architectural framework of the electoral system was designed to support a secure, efficient, and scalable online voting process. It follows a layered architecture, with key components organized into distinct layers, each responsible for different aspects of the system's functionality:

1. *User Interface (UI):* This layer is where users (admins and voters) interact with the system. It consists of the login screens, dashboards, voting interface, and administrative panels. The UI is designed to be intuitive and responsive, ensuring ease of use across different devices.
2. *Admin Dashboard:* Provides a summary of key election data, including the number of voters, votes cast, and positions. Admins can manage voters, candidates, and view real-time election results.
3. *Business Logic:* This layer contains the core functionality of the system, including vote management, user authentication, and data processing. It enforces the rules, such as allowing only one vote per voter and ensuring the integrity of the voting process.
4. *Vote Counting Module:* Automatically tallies votes as they are submitted, ensuring accurate and real-time updates to the election results.
5. *Email/SMS Notification Module:* Sends login credentials and notifications to voters, ensuring they are informed about the voting process.
6. *Database Management:* This layer handles data storage and retrieval. It stores information about voters, candidates, votes, and election positions. The database is designed to be secure and efficient, with mechanisms in place to prevent unauthorized access and ensure data integrity.
7. *Data Import/Export:* Allows the admin to upload voter data via an Excel sheet and export election results for further analysis.
8. *Authentication & Authorization:* Ensures that only registered voters can access the voting system and that admins have the necessary credentials to manage the election process. Each voter is provided with a unique ID and password to prevent multiple votes.
9. *Data Encryption:* Protects sensitive data, such as voter credentials and votes, ensuring that all communications within the *system are secure.*
10. *API Services:* Facilitates integration with external services, such as email and SMS gateways for sending notifications. This layer also allows for potential future integrations with other electoral systems or data analytics tools.

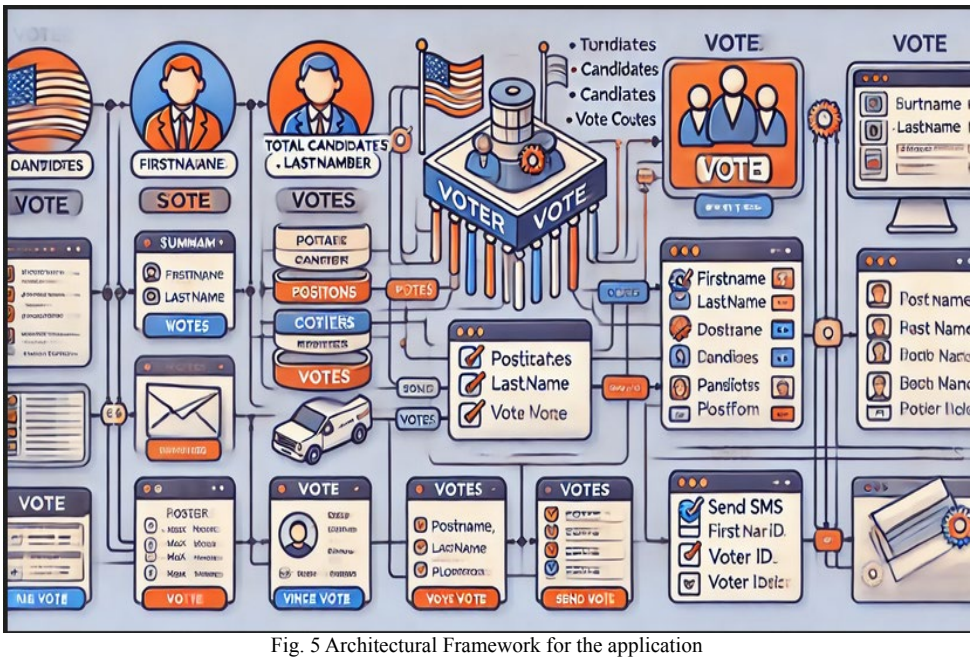

Fig. 5 Architectural Framework for the application

This architectural framework ensures that the system is robust, secure, and scalable, capable of handling a large number of voters while providing accurate and real-time election results.

V. SYSTEM IMPLEMENTATION

The voting system algorithm outlined here is designed to streamline and secure the electoral process within an educational or organizational setting. It encompasses multiple phases, each crucial for ensuring the integrity, efficiency, and accuracy of the voting process.

The process begins with Admin Registration, where the system administrator registers all candidates and voters. Each voter is assigned a unique ID and password, which are securely stored in the database. An email containing these credentials is sent to each voter, ensuring they can access the voting system. Candidates are also registered, with their details stored for later reference.

When a voter receives their login credentials, they log into the system. The system validates these credentials to ensure the voter is legitimate. It then checks whether the voter has already cast their vote. If the voter has not yet voted, they are allowed to proceed. The voter selects one candidate for each post, and their choices are recorded in the system. Importantly, the system ensures that each voter can only vote once per post, preventing duplicate votes.

After the voting period concludes, the system transitions to the Vote Counting phase. The system aggregates all votes for each post and counts them. This count is securely stored in the database. This automated process minimizes human error and ensures the results are accurate. The system generates and displays the results based on the vote counts stored in the database. Results are made available to the admin and, if applicable, to the public, ensuring transparency in the electoral process.

Throughout the process, the system prioritizes security and integrity. All voting data is encrypted and securely stored to prevent unauthorized access and tampering. Regular audits are conducted to detect and address potential issues, maintaining the system's reliability and trustworthiness.

Admin Registration
    FOR each voter:
        Generate unique ID and password
        Store voter details in database
        Send email with ID and password to voter
    FOR each candidate:
        Register candidate details in database
Voter Login:
    Voter enters ID and password
    IF credentials are valid:
        Check if voter has voted
        IF voter has not voted:
            Proceed to voting process
        ELSE:
            Display "You have already voted"
    ELSE:
        Display "Invalid credentials"
    Voter selects candidate for each post Record selected candidate in database
Vote Counting:
    FOR each post:
        Initialize vote count for each candidate
        FOR each vote:
            Increment the count for the selected candidate
        Store results in database

This algorithm provides a robust framework for efficiently and securely managing the electoral process, leveraging technology to enhance the accuracy and accessibility of voting.

VI. FINDINGS OF THE STUDY

The voting system was designed to streamline and manage the electoral process efficiently. It integrates multiple modules to facilitate various aspects of election management, from voter registration to vote counting and result dissemination. The system is secured with an admin login interface featuring two fields: username and password, as shown in Fig. 3. Admins must enter their credentials to access the system. Upon successful authentication, the admin is redirected to the main dashboard, as shown in Fig. 4.

The dashboard provides a high-level summary of essential electoral metrics, including:

1. *Number of Candidates:* Displays the total number of candidates participating in the election.
2. *Number of Voters:* Shows the total number of registered voters.
3. *Total Voters:* Indicates the count of all registered voters.
4. *Voters Voted:* Reflects the number of voters who have cast their votes.

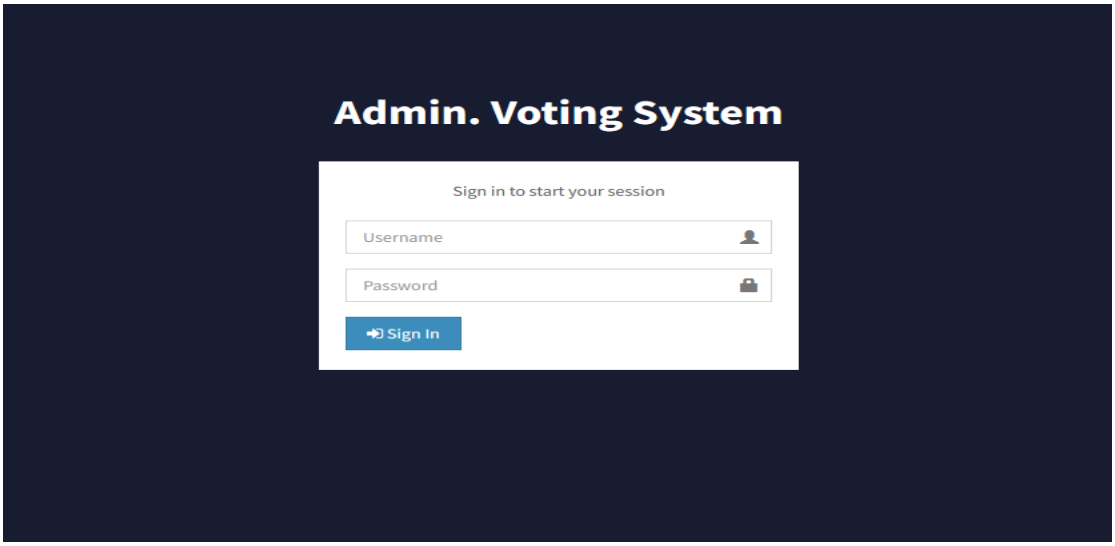

Fig. 6 Login Admin Page of the system

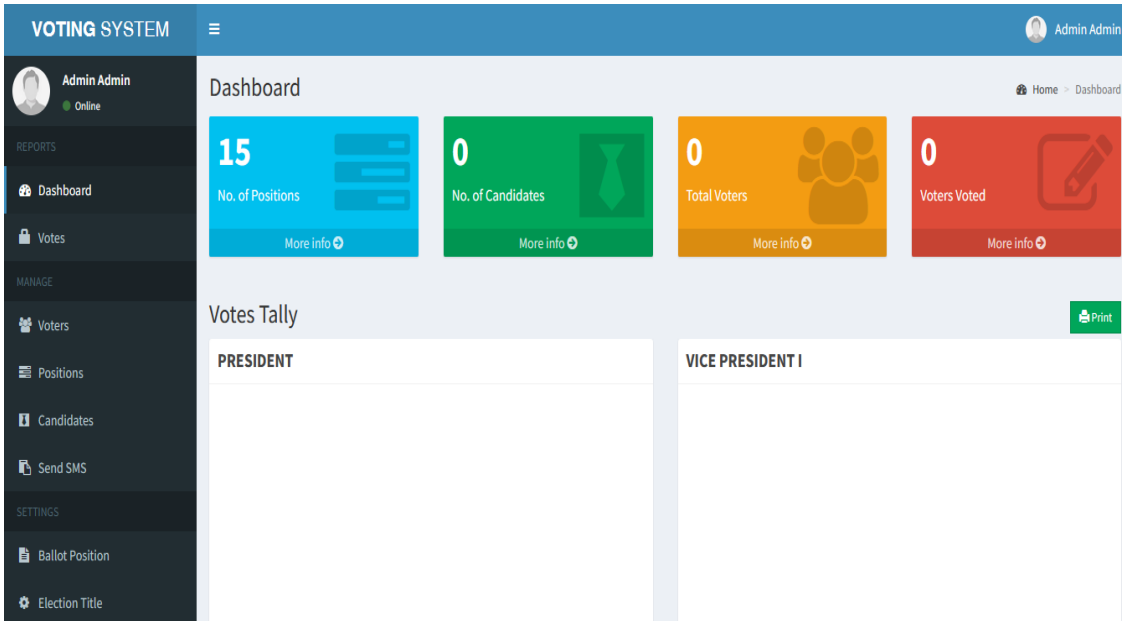

Fig. 7 Dashboard of the application

The dashboard serves as the primary navigation point, providing quick insights into the election's current status and facilitating access to other modules.

A. Votes Module

This module is dedicated to managing and viewing voting data. It presents the specific post or role for which votes are

cast (position), the name of the candidate running for each position (candidate), and the total number of votes received by each candidate (vote). This module is crucial for tracking vote counts and providing real-time insights into the electoral landscape.

B. Voters Module

The Voters Module enables efficient management of voter information. It allows admins to upload an Excel sheet containing detailed voter information, including the voter's given name (first name), family name (last name), a photograph of the voter for identification purposes (photo), and a unique identification number assigned to each voter (voter ID). This module simplifies the registration process and ensures that all voter data is accurately captured and securely stored.

C. Position Module

This module provides detailed information about each position in the election. It includes a description of the role or responsibilities associated with each position (description of the position) and the maximum number of votes a candidate can receive for each position (maximum vote). The Position Module ensures that all participants understand the roles and voting limits.

D. Candidates Module

The Candidates Module displays comprehensive information about the candidates. It includes the position (the electoral position each candidate is running for), photo (a photograph of the candidate), first name (the candidate's given name), last name (the candidate's family name), and platform (the candidate's campaign platform or key issues they support). This module aids in candidate identification and provides voters with the necessary information to make informed decisions.

E. Send SMS Module

This module is designed for communication with voters. It allows admins to send SMS messages containing first name (the voter's given name), last name (the voter's family name), voter ID (the unique identification number assigned to the voter), and email address (the voter's email address). The Send SMS Module facilitates direct communication and ensures that voters receive important information regarding the election.

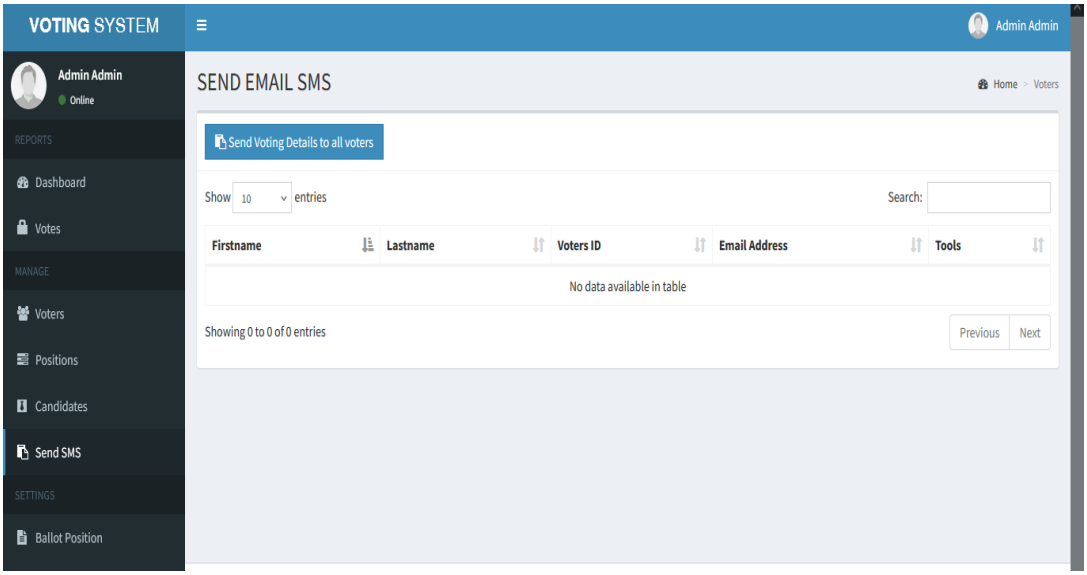

Fig. 8 SMS/Email module of the application

F. Ballot Position Module

The Ballot Position Module presents the ballot layout, showing the position of all candidates. It provides a textual representation of where each candidate is positioned on the ballot. This module helps organize the ballot and ensures that candidates are correctly displayed for voters.

The voting system integrates these modules to create a cohesive and efficient electoral management platform. By offering comprehensive tools for managing candidates, voters, and votes, the system ensures that elections are conducted smoothly and transparently. Each module plays a specific role in the overall process, contributing to a well-organized and user-friendly voting experience.

VII. CONCLUSION

The design and implementation of the electoral voting system, as outlined in this paper, demonstrate a significant advancement in optimizing electoral processes through the integration of modern technological solutions. The system's robust architecture - comprising comprehensive modules for managing votes, candidates, and voters - ensures a streamlined and efficient electoral process. By automating vote counting and providing a user-friendly interface for both

administrators and voters, the system enhances the overall integrity and reliability of elections. The incorporation of an admin dashboard with detailed metrics, including the number of votes, voters, positions, and candidates, offers valuable insights into the electoral process, facilitating better decision-making and management [24]. The system's ability to manage voter registration through an Excel upload and the secure email delivery of voting credentials reflects an emphasis on accessibility and security, addressing common concerns in electoral systems [25]. Moreover, the single-vote-per-post restriction and automatic vote counting reduce the potential for human error and ensure a fair and transparent voting process. The system's modular design - encompassing functionalities for sending SMS notifications, displaying ballot positions, and managing candidate details - provides a comprehensive solution to the challenges of modern electoral management [25]. In conclusion, this electoral system not only optimizes the voting process but also upholds principles of electoral integrity and voter engagement. Future developments could explore further enhancements in online voting security and user experience to continue advancing the field of electoral system optimization.

REFERENCES

[1] M. O. Achieng and E. Ruhode, "The adoption and challenges of electronic voting technologies within the South African context," *Int. J. Managing Inf. Technol. (IJMIT)*, vol. 5, no. 4, pp. 1-12, 2013.

[2] P. Norris, *Why Elections Fail*, Cambridge: Cambridge Univ. Press, 2015. [Online]. Available: https://doi.org/10.1017/CBO9781107280908.

[3] R. M. Alvarez and T. E. Hall, *Electronic Elections: The Perils and Promises of Digital Democracy*, Princeton, NJ: Princeton Univ. Press, 2008. [Online]. Available: http://www.jstor.org/stable/j.ctt7ss68.

[4] M. N. Franklin *et al*., *Voter Turnout and the Dynamics of Electoral Competition in Established Democracies Since 1945*, Cambridge: Cambridge Univ. Press, 2004.

[5] E. Gidengil, "The Decision to Vote or to Abstain," *Oxford Research Encyclopedia of Politics*, Oct. 26, 2016. [Online]. Available: https://oxfordre.com/politics/view/10.1093/acrefore/9780190228637.001.0001/acrefore-9780190228637-e-69.

[6] A. Blais, "What affects voter turnout?" *Annu. Rev. Polit. Sci.*, vol. 9, pp. 111-125, 2006. [Online]. Available: https://doi.org/10.1146/annurev.polisci.9.070204.105121.

[7] S. C. Stokes, "Political parties and democracy," *Annu. Rev. Polit. Sci.*, vol. 2, pp. 243-267, 1999. [Online]. Available: https://doi.org/10.1146/annurev.polisci.2.1.243.

[8] M. Duverger and D. W. Brogan, *Political Parties: Their Organization and Activity in the Modern State*, B. North and R. North, Trans., London: Methuen & Co. and John Wiley & Sons, 1954.

[9] P. Norris, R. W. Frank, and F. Martínez i Coma, "Measuring electoral integrity around the world: A new dataset," *PS: Polit. Sci. Polit.*, vol. 47, no. 4, pp. 789-798, 2014. [Online]. Available: https://doi.org/10.1017/S1049096514001061.

[10] D. Malik, K. Tripathi, and Jyotsna, "Enhancing the security of online voting system using defined biometrics," in *Proc. IEEE 3rd Int. Conf. Tech., Eng., Manag. Societal Impact using Marketing, Entrepreneurship, and Talent (TEMSMET)*, Mysuru, India, 2023, pp. 1-8, doi: 10.1109/TEMSMET56707.2023.10150198.

[11] K. F. Hashim *et al*., "Online political engagement using Twitter among Malaysian parliamentary members," in *Proc. IEEE 7th Int. Conf. Info. and Commun. Technol. (ICoICT)*, Kuala Lumpur, Malaysia, 2019, pp. 1-5, doi: 10.1109/ICoICT.2019.8835321.

[12] A. Delis *et al*., "Pressing the button for European elections: Verifiable e-voting and public attitudes toward internet voting in Greece," in *Proc. 6th Int. Conf. Electron. Voting: Verifying the Vote (EVOTE)*, Lochau/Bregenz, Austria, 2014, pp. 1-8, doi: 10.1109/EVOTE.2014.7001141.

[13] F. Bélanger and L. Carter, "The digital divide and internet voting acceptance," in *Proc. 4th Int. Conf. Digital Soc.*, Saint Maarten, Netherlands Antilles, 2010, pp. 307-310, doi: 10.1109/ICDS.2010.5.

[14] S. S. Chaeikar *et al*., "Security principles and challenges in electronic voting," in *Proc. IEEE 25th Int. Enterp. Distrib. Object Comput. Workshop (EDOCW)*, Gold Coast, Australia, 2021, pp. 38-45, doi: 10.1109/EDOCW52865.2021.00030.

[15] I. Singh *et al*., "Enhancing security and transparency in online voting through blockchain decentralization," *SN Comput. Sci.*, vol. 5, p. 921, 2024. [Online]. Available: https://doi.org/10.1007/s42979-024-03286-2.

[16] C.-H. Ho and S.-H. Lin, "Electoral voting protocol - a quorum-based approach for replica control," in *Proc. 7th Int. Conf. Parallel Distrib. Syst. (Cat. No. PR00568)*, Iwate, Japan, 2000, pp. 463-469, doi: 10.1109/ICPADS.2000.857730.

[17] N. Fauzia *et al*., "An efficient implementation of electronic election system," in *Proc. 10th Int. Conf. Comput. Inf. Technol.*, Dhaka, Bangladesh, 2007, pp. 1-6, doi: 10.1109/ICCITECHN.2007.4579420.

[18] S. Dinesh Kumar *et al*., "Theoretical analysis of voting systems," in *Proc. Int. Conf. Commun. Electr. Syst. (ICCES)*, Coimbatore, India, 2016, pp. 1-5, doi: 10.1109/CESYS.2016.7889932.

[19] D. Stockemer and M. Wigginton, "The (complex) effect of internet voting on turnout: Theoretical and methodological considerations," *Policy & Internet*, vol. 16, pp. 607-627, 2024. [Online]. Available: https://doi.org/10.1002/poi3.393.

[20] S. M. T. Toapanta *et al*., "Model of shared secret applied to a voting system for the national electoral council of Ecuador," in *Proc. Int. Conf. Comput. Inf. Telecommunication Syst. (CITS)*, Beijing, China, 2019, pp. 1-5, doi: 10.1109/CITS.2019.8862061.

[21] K. Dommett, G. Kefford, and S. Kruschinski, "Theoretical framework," *Data-Driven Campaigning and Political Parties: Five Advanced Democracies Compared*, Oxford Academic, 2023. [Online]. Available: https://doi.org/10.1093/oso/9780197570227.003.0002.

[22] B. J. Alfonsi, "E-voting advocates hold out hope," *IEEE Distributed Syst. Online*, vol. 5, no. 3, 2004. [Online]. Available: doi: 10.1109/MDSO.2004.1285879.

[23] S. M. Toapanta *et al*., "Ensuring the blind signature for the electoral system in a distributed environment," in *Proc. IEEE Int. Conf. Power, Intelligent Comput. and Syst. (ICPICS)*, Shenyang, China, 2020, pp. 211-215, doi: 10.1109/ICPICS50287.2020.9202348.

[24] M. Farhan *et al*., "Disrupting the ballot box: Blockchain as a catalyst for innovation in electoral processes," in *Proc. Int. Conf. Adv. Computing, Commun., Electr. Smart Syst. (iCACCESS)*, Dhaka, Bangladesh, 2024, pp. 1-6, doi: 10.1109/iCACCESS61735.2024.10499541.

[25] W. Salman, V. Yakovlev, and S. Alani, "Analysis of the traditional voting system and transition to the online voting system in the republic of Iraq," in *Proc. 3rd Int. Congr. Hum.-Comput. Interact., Optim. Robotic Appl. (HORA)*, Ankara, Turkey, 2021, pp. 1-5, doi: 10.1109/HORA52670.2021.9461387.

[26] G. Jiang and H. Zhang, "Analysis of the driving role of computer science and technology on the development of the Internet of Things," in *Proc. Int. Conf. Comput. Inf. Sci. and Artif. Intell. (CISAI)*, Kunming, China, 2021, pp. 892-895, doi: 10.1109/CISAI54367.2021.00179.