Received: 25 July 2025 | Revised: 23 October 2025 | Accepted: 5 December 2025 | Published online: 28 December 2025

REVIEW

Artificial Intelligence and Applications
2025, Vol. 00(00) 1–15
DOI: 10.47852/bonviewAIA52026923

BON VIEW PUBLISHING

Blockchain Platforms for Developing Smart Contracts and Their Computational Performance Evaluation: A Systematic Literature Review

Alock Gupta[1,*] and Kamlesh Lakhwani[1]

[1] *Department of CSE, JECRC University, India*

Abstract: An independent, trusted third party or governing body is no longer necessary to conduct secure financial transactions because of blockchain technology. The topic of smart contracts and their ability to facilitate additional computational progress has risen to the forefront of academic and industry conversations in response to the dizzying rate of growth in blockchain technology. The scholarly work takes into account the material that has been assessed by experts and aims to explain the fundamental idea and provide a comprehensive computational analysis of relevant literature. Such an approach contributes to the advancement of decentralized applications (dApps) by providing technical insights into their development frameworks. The initial section presents a brief overview of smart contracts, including their conceptual foundations, system architecture, and application domains. Furthermore, in a detailed review of existing platforms for developing smart contracts, it was found by comparison that the Tron and CoreDAO blockchains offer the most computationally efficient platforms to enhance the quality-of-services (QoS) in decentralized environments. These low-cost transaction models support the creation of resource-efficient smart contracts. In addition, this study includes a simulation work that considers the blockchain transactions as a dataset to train an artificial intelligence model that would support the computational prediction of the success and failure of the transactions.

Keywords: smart contract, Ethereum, Polkadot, Solana, Tron, Binance Smart Chain (BSC)

1. Introduction

Blockchain is an immutable ledger of digital transactions that is shared across several computers in a decentralized manner. Owing to its distributed and decentralized nature, blockchain technology is risk-free. Consequently, it is not feasible to hack a single component of the system in isolation from the others. The workflow of blockchain-based transactions is illustrated in Figure 1. Because future computing demands are expected to be higher, centralized systems will face challenges in terms of scalability. Data security, availability, access control, and privacy are fundamental issues in computer science study. As a data storage system, blockchain makes it very difficult, if not impossible, to edit, hack, or defraud. A reliable third party is no longer needed to process financial transactions because of smart contracts built on blockchain technology. With the introduction of smart contracts, blockchain technology has found applications in numerous fields, including healthcare, transportation, voting, public donation management, wills, and real estate transactions. Smart contracts digitally sign applications, making them unchangeable. To keep track of assets and transactions throughout a company's network, distributed ledger technology (blockchain) is a great help. Assets might be tangible objects like furniture or cars or intangible concepts like goodwill or money (intellectual property, patents, copyrights, and branding). There are numerous benefits of using a smart contract that is based on the blockchain, including reduced risk and improved efficiency for everyone involved. Blockchain networks are perfect for any type of data storage or distribution because of their immutable ledger characteristic, which means that only authorized users can access the data.

Blockchain-based smart contracts ensure ease of use through technological upgradation via consistent and embedded decentralization, autonomous execution, and accuracy. Figure 2 presents the evolution of powerful blockchain platforms. This paper considers cryptocurrency, which was listed as crypto from the year 2013 to 2025.

In 2013, Ethereum was invented, as indicated by its symbol. However, Ethereum contracts took a lot of time. Then, dash was released in 2014 to provide efficient services. The demand for user-defined tokens increased. Subsequently, the Waves cryptocurrency was listed in 2016. It has been observed that Ethereum, dash, and wave are very costly. Moreover, these blockchains could not rapidly execute transactions. To improve the transaction cost foundation, optimized consensus mechanisms and efficient gas management strategies were introduced. The Tron founders considered the role of decentralization and high-throughput transaction processing as key factors in blockchain scalability. In 2020, all cryptocurrency prices increased. Several cryptocurrencies such as BNB Smart Chain, Polkadot, Matic, and Solana came into existence and became popular in less time. In 2021, the metaverse was invented, and Sandbox and MANA were in demand. The buy and sell of digital assets such as NFT and digital land were made fluently using BNB and Matic. The era of 2022 was focused on yield farming, where holders were expected to get revenue. Several projects such as Tron, Pancake, CoreDAO, and PI network focused on field farming. In 2023, there were international marketplaces, such as Young Parrot and OpenSea. Some popular NFT brands assure the reliability

***Corresponding author:** Alock Gupta, Department of CSE, JECRC University, India. Email: Alok.20phen030@jecrcu.edu.in

Figure 1
Process flow of blockchain-based transactions

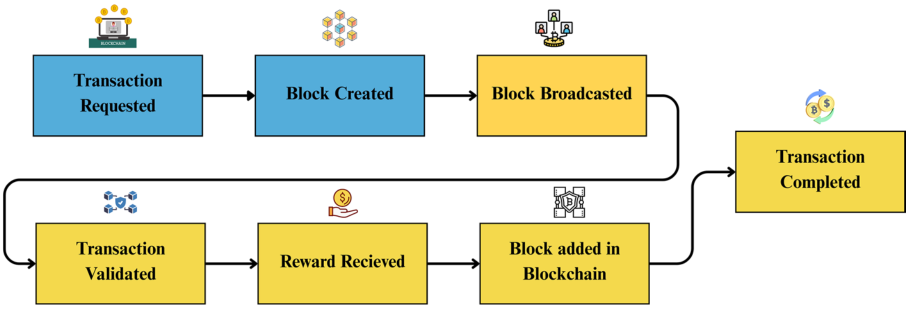
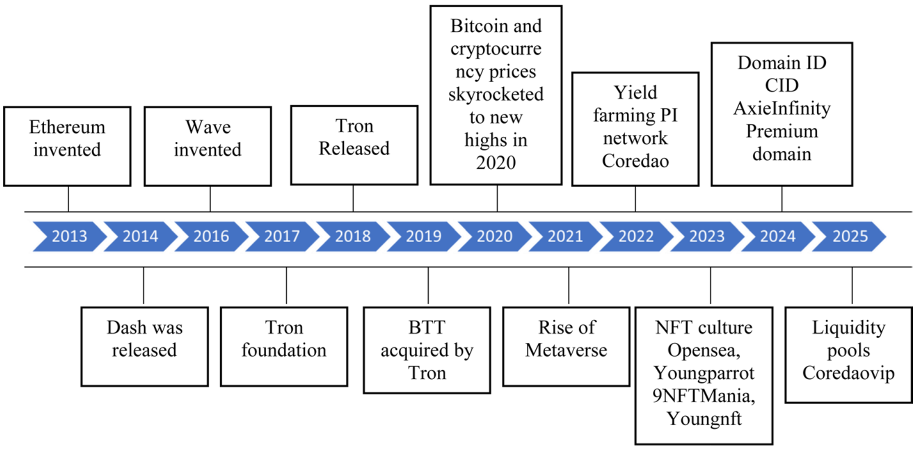


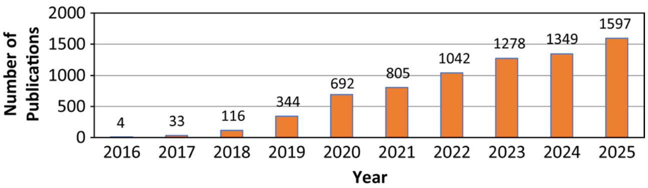
Figure 2
Evolution timeline of significant blockchain platforms



of NFT. In 2024, considering the use case of digital assets, there were provisions for domain booking for branding. CoreID and Axie Infinity provided ecosystems to buy domains. A premium domain is a system that is focused on domain parking for branding. In 2025, Bitcoin's value increased above $100000, but Alt coin's value did not increase. In this era, CoreDAO took the lead by introducing a decentralized governance framework aimed at improving network scalability and security. This system is a volunteer-based system that focuses on blockchain education and considers liquidity pooling practices on DEX systems. It provided an ecosystem for decentralized financing by circulating a limited supply of Coredaovip in 9NFTMANIA NFT holders and promoted the concept of tokenomics.

Studies on blockchain-based contracts commenced a few years ago, but they continue to produce vast and new contributions due to their techniques in various applications. Figure 3 demonstrates this point, in which the number of publications (authentic articles, assessment articles, books, etc.) posted year over year illustrates an upward trend in this discipline. Figure 3 shows that there was substantial growth during the last six years. The goals of this research could not have been met

Figure 3
Number of publications (data collected from Web of Science) on blockchain and smart contracts (topic – 1,994)



without reviewing the most up-to-date literature from reputable sources, which were published between 2015 and 2025.

Web of Science and other major databases were searched for any relevant articles. This study found the most relevant articles by searching for phrases like "blockchain technology," "smart contracts," "Ethereum and smart contracts," "blockchain problems," and "barriers in smart contracts." Scopus and Science Citation Indexes were used to index vast majority of papers cited in this research. As various studies were done on these topics at a very fast rate in Western culture, there is a need to study the topic in the Indian context. This area of research in today's scenario is very popular because every user trusts this technology due to its various features like immutability and transparency.

In the past, influential reviews were conducted by many distinguished authors. However, these review papers only focused on the Ethereum blockchain platform. Moreover, they did not consider factors such as overhead and performance. In the case of Ethereum for committing , transaction execution is very slow, resulting in high latency and low throughput. Consequently, Ethereum cannot use such kind of application in a realistic environment in the context of an enterprise application like supply chain management. Blockchain adoption is hindered by high transaction costs, sluggish transaction rates, complicated designs, and the lack of knowledge and experience, while smart contracts are hindered by legal difficulties and high transaction costs. There is a requirement to consider recently published research to resolve such issues. The need for online systems is growing daily.

Nowadays, working from home through online modes is trending. Therefore, everyone is dependent on online systems. However, there is still a need to maintain a smart contract that provides a cost-effective, high-performance, and secure solution for commercializing online systems. With the support of existing and available peer-reviewed literature, the researcher framed the structure and operation of blockchain smart contracts for constructing decentralized apps. In addition, the researcher is investigating current platforms for producing smart contracts and is offering the most efficient platforms for improving quality of service (QoS) in terms of minimizing transaction latency and maximizing transaction execution speed for creating low-cost smart contracts.

For those working on decentralized applications in the future, this review offers a high-level overview of smart contracts, discussing their concept, design, and potential uses. As an alternative to Ethereum, Waves, and Bitcoin, research is looking at high-performance blockchains that can execute the contracts with cheaper transaction fees. Therefore, the next online system smart contract needs be efficient in terms of both time and money. In its last section, the review paper delves into the challenges and potential outcomes of further research on decentralized blockchain smart contracts. Previous studies on blockchain technology based on contracts have been reviewed in the articles by Swain and Chouhan [1] and Shrivastava et al. [2] from a variety of fields, including healthcare and business. While some writers concentrated on public blockchains, others deemed their work to be more private [3].

Sizan et al. [4] and Jyoti et al. [5] considered the applicability of blockchain in real life. The role of blockchain in Internet of Things (IoT) has been discussed [6]. Ethereum-based smart contracts are also elaborated [7]. Some of the existing research focused on deep learning-based prediction models for crypto price prediction [8, 9]. The present review paper has considered the objective, findings, and limitations of previous research works. Real-life applications of blockchain, future trends, tools and techniques, attacks and vulnerability, different types of smart contracts, and barriers in the adoption of different blockchains are analyzed in different studies.

The blockchain ecosystem elaborates working of smart contracts in the blockchain. A comparison of existing blockchains such as Ethereum, Wave, and Steller has been made. Different smart contract platforms such as Ethereum, Polkadot, Solana, Tron, and BSC chain are elaborated with their technical specifications such as blockchain type, type, native coin, consensus mechanism, transaction per second, main-net launch data, market cap, market price, and token standard. Immutable and transparent smart contracts are presented with their parameters such as cost, time, security, and chances of mistakes. Moreover, the application of smart contracts is considered with tools and techniques.

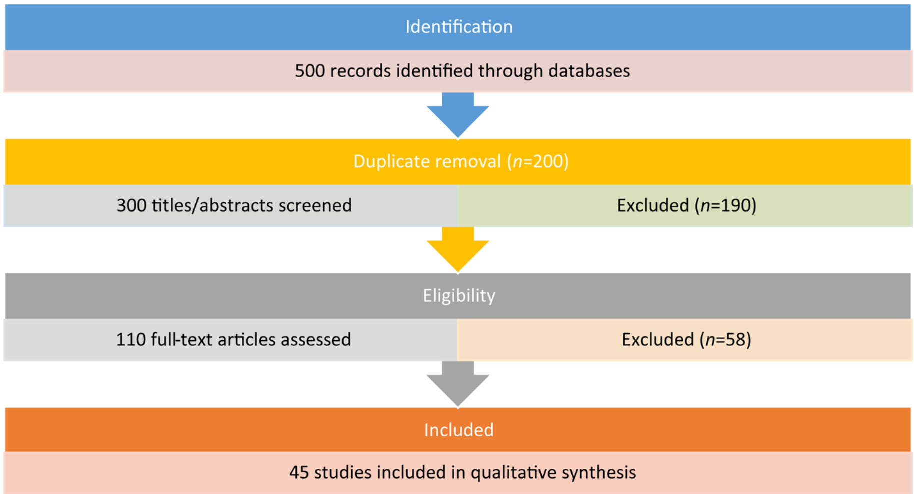
Part of the simulation is about finding an adaptation barrier for blockchain technology. The likelihood of success and failure may now be predicted with the help of a model trained using deep learning. Consideration was given for creating a dataset of TRX transactions for use in simulations. Data preparation, data filtering, configuration, training, and testing are the steps that lead to the calculation of the accuracy measures, including F1-score, recall, and precision. The dataset has been divided into 70% for training and 30% for testing during the simulation. The accuracy and mistake rates of a traditional machine learning model and a deep learning-based model that takes ANN into account were compared.

The review consists of five sections. Section 1 covers an introduction to blockchain technology and analyzes its benefits and drawbacks. Additionally covered are the ins and outs of the cryptocurrency transaction process and the utilization of blockchain technology. In addition, from 2013 to 2025, cryptocurrency—which was defined as "blockchain technology"—was considered by new blockchain platforms. The essentials and requirements of blockchain technology are also covered. Subsequently, in Section 2, we cover existing literature on blockchain technology based on smart contracts in many domains, including healthcare and business. The blockchain ecosystem's smart contracts and their underlying operating mechanism are introduced in Section 3. Moreover, we will compare and contrast the technical specifications of various blockchains, including their kind, native coin, consensus mechanism, and transaction per second, and demonstrate how smart contracts function. In Section 4, the steps for finding obstacles to blockchain adaption are outlined. To train the models to estimate the likelihood of success and failure, a deep learning model is utilized. It has been contemplated to create a dataset of TRON transactions for use in simulations. Various characteristics are taken into account when the simulation is running. After the data have been preprocessed, filtered, configured, trained, and tested, the accuracy parameters are determined. The dataset has been divided into 70% for training and 30% for testing during the simulation. A comparison is made between the standard machine learning model and a deep learning-based learning model that takes ANN into account in terms of accuracy and error. Section 5 then wraps up the research scope.

2. Literature Review

Using smart contracts on a permissioned blockchain infrastructure, Swain and Chouhan [1] presented a transparent auction method. To enhance trust, transparency, and security in land management systems, Shrivastava et al. [2] suggested a land registry framework that is blockchain-based and incorporates smart contracts and efficient consensus. With the introduction of BlockPres IPFS by Khan et al. [3], a decentralized and secure prescription management system that makes use of blockchain technology is introduced. For IoT applications, Sizan et al. [4] thoroughly evaluated various blockchain platforms. In their comprehensive analysis of Ethereum blockchain development tools, Jyoti et al. [5] pinpointed the advantages and disadvantages of these tools for deploying smart contracts. The smart contract executions of EVMs and Web Assembly virtual machines were compared by Zhang et al. [6]. For a comprehensive review of the methods used by blockchain systems to execute smart contracts, see Liu et al. [7]. By analyzing smart contracts from an efficiency and security perspective, Singh et al. [8] found weaknesses and offered solutions to fix them. Using blockchain technology and the automation of smart contracts, Naik et al. [9] suggested a decentralized platform for ridesharing. In 2024, Ta and Do [10] studied the price of Ethereum gas and modeled the blockchain's performance under several levels of contract complexity. For the purpose of detecting health insurance fraud, Kaafarani et al. [11] created a decision-making model based on smart contracts to choose blockchain platforms. A comprehensive analysis of smart contract platforms, uses, and inherent problems was provided by Sharma et al. [12]. The Smart Contract Broker method was suggested by Park et al. [13] to improve reusability in blockchain systems. Agrawal et al. [14] showcased a smart contract-based blockchain architecture for supply chain collaboration. With an emphasis on performance, development tools, and contract administration, Khan et al. [15] examined dApp frameworks. In their study, Madhwal et al. [16] created a logistics proof of a delivery model that relies on smart contracts. By comparing the two blockchain technologies' performance, Abhishek et al. [17] drew comparisons between Hyperledger Fabric and Ethereum. The importance of smart contracts in maintaining the authenticity of medical prescriptions is emphasized in the evaluation of PBFT-based blockchain systems by Garcia et al. [18]. In their all-encompassing review of smart contract uses, Lin et al. [19] covered every possible field. Using an AI-driven consensus mechanism, Kim [20] suggested a blockchain smart contract approach to tackle degree certificate forgeries. Vacca et al. [21] thoroughly examined several tools and approaches for developing smart contracts and blockchain technology. Blockchain smart contracts were discussed in Khan et al. [22], along with their uses, difficulties, and potential future developments. A hybrid on-chain and off-chain smart contract architecture was implemented and evaluated by Solaiman et al. [23]. Khan and Naz [24] developed a decentralized system for

Figure 4
PRISMA diagram



managing learning in 2021. A permissioned blockchain smart contract architecture with high performance concurrency was proposed by Jin et al. [25].

The PRISMA framework was used to find, evaluate, and incorporate papers that were relevant to the review to ensure that it was thorough and methodical. The following search terms were used: ("Blockchain" and "Smart Contract") or ("Blockchain Platforms" and "Computational Performance") or ("Ethereum" and "Tron" and "Polkadot"). To be considered for inclusion, articles had to be written in English and to focus on smart contracts built on the blockchain, their computational performance, and comparative analyses across platforms. Excluded from consideration were studies lacking an empirical evaluation, duplicate entries, or studies having a solely conceptual basis. From the original set of 500 records, 300 survived deletion of duplicates, 110 survived screening for title and abstract, and 45 survived synthesis inclusion requirements. The PRISMA flow diagram depicting the systematic screening from identification to inclusion is shown in Figure 4. Key recent studies on blockchain smart contract applications and their evaluation are summarized in Table 1.

2.1. Comparison features of existing literature

Table 2 shows the features that are explained in the previous research. It is considered the base paper of this research and focuses on different features that are used for real-life applications and future trends. The description contains different tools and techniques of smart contracts. Previous studies are about immutable contracts and transparent and destroyable contracts. We are considering expectation disorder and call stack vulnerability. It consists of unbounded computational power-intensive operation and barriers in adoption.

2.2. Research gap

Key gaps in the current literature are discussed here:

1) The bulk of currently available blockchain-based smart contracts performs badly in terms of execution time and has significant overhead, as shown by the related study.
2) While several studies have looked at how to best use blockchain platforms for developing DAPPs, the vast majority of these studies have ignored the importance of taking transaction costs into account.
3) According to the research, the Tron decentralized blockchain applications may benefit from the increased throughput and decreased transaction latency of the blockchain platform. However, most current researchers have ignored meta-heuristic approaches.

Previous research has mostly concerned itself with the security of Bitcoin storage mediums. The transactional overhead and throughput have not received a lot of attention [26–28].

3. Platform and Application

3.1. Blockchain ecosystem

As a cautionary tale, it is crucial to remember that blockchain applications do not simply take off overnight. Proper ideas can only be realized by bringing together a diverse group of creative thinkers and technical professionals who can put those ideas into action. "Blockchain ecosystem" is a term used to describe how blockchain may be used to automate business processes across several organizations. As a rule, they are employed to expedite the process of signing a contract so that all parties may be confident in the outcome without any delays. Christidis et al. (2016) highlighted the role of smart contracts in enabling decentralized automation.

Figure 5 shows the working process of smart contracts in the blockchain. Here, a data event is passed to the present response condition. The contract transaction set phase takes place. The contract transaction set consists of contact status, and significant information is stored in blocks. Finally, automatic execution takes place by considering the preset response rule. A distributed ledger may have several benefits, including decentralization, independence, enhanced flexibility, audit trail, and transparency.

3.2. Mathematical expression for finding blockchain's criteria

3.2.1. Get difficulty from the blockchain

The blockchain is the definitive source for mining difficulty because it records these data in each block. The following encoding may be deciphered using the Bitcoin source code: it is assumed that byte position 73 in the block represents the positive number x and that byte positions 74 and 75 represent the positive number y in

Table 1
Literature review

Ref	Author	Year	Objective	Methodology	Finding	Limitations
[1]	Swain and Chouhan	2025	Develop auction mechanism using smart contracts	Smart contracts on permissioned blockchain	Improves auction transparency	Limited to permissioned platforms
[2]	Shrivastava et al.	2025	Secure land registry with blockchain	Blockchain with efficient consensus	Enhances land system trust and transparency	Scalability not addressed
[3]	Khan et al.	2025	Secure prescription management	Blockchain + IPFS	Privacy and data integrity improved	Not tested in real-world settings
[4]	Sizan et al.	2025	Review blockchain for Industry 5.0 IoT	Literature review	Lists platform suitability and challenges	Lacks experimental validation
[5]	Jyoti et al.	2025	Analyze Ethereum development tools	Systematic analysis	Identifies tool strengths and limitations	Ethereum-specific focus
[6]	Zhang et al.	2024	Compare WASM vs. EVM for smart contracts	Performance benchmarks	WASM shows better performance	Focuses only on metrics
[7]	Liu et al.	2024	Review smart contract execution mechanisms	Survey and review	Broad overview of VMs and gas usage	No empirical data
[8]	Singh et al.	2024	Analyze efficiency/security in smart contracts	Security analysis	Lists vulnerabilities and mitigation methods	Trade-offs not quantified
[9]	Naik et al.	2024	Blockchain-based ride-sharing	Contract automation	Improves transparency and cost-efficiency	User behavior not studied
[10]	Ta and Do	2024	Simulate Ethereum gas costs	Simulation-based study	Offers contract cost optimization tips	Based on simulated data
[11]	Kaafarani et al.	2023	Health fraud detection with smart contracts	Decision-making model	Boosts fraud detection	Uncertain generalizability
[12]	Sharma et al.	2023	Review platforms and challenges	Literature review	Explores legal and scalability concerns	No empirical evidence
[13]	Park et al.	2023	Improve smart contract reusability	Contract broker system	Enhances modular development	Adoption unmeasured
[14]	Agrawal et al.	2023	Supply chain collaboration framework	Blockchain framework	Enhances traceability and efficiency	Industry-specific insights only
[15]	Khan et al.	2023	Review of dApp frameworks	Critical review	Highlights limitations in performance and tools	No performance data
[16]	Madhwal et al.	2022	Proof-of-delivery in logistics	Model development	Boosts logistics efficiency	Limited test scenarios
[17]	Abhishek et al.	2022	Compare Ethereum and Hyperledger Fabric	Performance evaluation	Hyperledger is better for enterprises	Only two platforms compared
[18]	Garcia et al.	2022	Medical prescription integrity	PBFT blockchain implementation	Prevents prescription fraud	Scalability not assessed
[19]	Lin et al.	2022	Survey smart contract applications	Literature survey	Covers broad application areas	No technical validation
[20]	Kim	2022	Prevent certificate forgery with AI smart contracts	AI-driven consensus model	Validates certificate authenticity	AI complexity not addressed

big-endian notation. Therefore, difficulty, as stored on the blockchain, is: d = (216 − 1) * (2208 − 8(x − 3))/y.

3.2.2. Difficulty adjustment

Typically, the network will discover a new solution every 10 min after readjusting the mining difficulty. There is a readjustment every 2016th block.

3.2.3. Network speed

Given that a miner will have some level of success once every 10 min, the overall network computing speed S(d) in hash/second is: S(d) = (232/600) * d = 7158278d. Using the current example difficulty of 1590896927258, the implied network speed is: 11.38*1018 hash/s or 11.38 EH/s (exa hashes/s).

3.2.4. Implied market share

Given the speed of the network, a miner may determine its percentage of the market using its hash rate, s, and the current difficulty, d, as: f = s/7158278d = 1. 397 * 10 − 7 * (s/d).

3.2.5. Time to find a block

There is no "progress" in performing hashes. It is likely that the next hash computed is that the winner stays constant, regardless of how many computations have been conducted before $1/(d*2^{32})$. Therefore,

Table 2
Comparison of features

Ref	Real-life application	Future trends	Tools and techniques of smart contracts	Attacks and vulnerabilities	Immutable and trasparent smart contract	Destroyable contracts	Exception disorder	Call stack vulnerability	Unbounded computational power-intensive operations	Barrier in adoption
Khan et al. [22]	✓		✓							✓
Iuliano and di Nucci [29]	✓		✓							
Vidal et al. [30]			✓	✓	✓					✓
Alaba et al. [31]	✓	✓	✓							✓
Singh et al. [32]	✓	✓	✓							✓
Wu et al. [33]		✓	✓	✓		✓	✓	✓	✓	

for a given hash rate s, the duration until the next block follows an exponential distribution with the cumulative density function (t is in s) below: P(tblock < t) 1 – e ^ (-ts/232d).

3.2.6. Price volatility analysis

$$E\left(\sigma t\right) = f\left(w\left(\sqrt{\sum_{i=1}^{n}\frac{\left(r_1-\sum_{n}^{r_i}\right)^2}{n-1}} + \left(1-w\right)\left(\sigma\,'\,t-1\right)\right|\sigma\,'\,t-1 = \sqrt{\sum_{i=1}^{n}\frac{\left(r_1-\sum_{n}^{r_i}\right)^2}{n-1}}\right), \quad (1)$$

where t is the standard deviation in time t, ‘t-1 is the standard deviation in the previous period with low volume, ri is the price returns, n is total price time integer observations, and w is the percentage factor weighting of transaction priority and execution cost.

3.3. Smart contract platforms

For the year 2025, this study looked at the finest smart contract systems. While several of them promise to be scalable, none of them had the opportunity to serve a big number of customers. Blockchain technology, originally introduced through Bitcoin’s peer-to-peer electronic cash system, laid the foundation for decentralized platforms [34]. In the foreseeable future, Ethereum and Polkadot will be the only major participants. The new PoS network will remain the dominant smart contract architecture in the foreseeable future until another project defeats Ethereum and replaces it within the following year. The six best smart contract platforms are the following:

1) **Ethereum:** The smart contract platform Ethereum was created in 2013 by Vitalik Buterin and his co-founders. After raising $16 million in a token sale, the Ethereum Foundation oversaw the project’s deployment two years later as a decentralized solution.
2) **Polkadot:** Gavin Wood, one of Ethereum’s co-founders, established another smart contract ecosystem called Polkadot. He opted to construct his blockchain network after finding that ETH was not going to be as safe and scalable as he had expected.
3) **Solana:** The Solana project aims to provide scalability for smart contracts without compromising their decentralization or security. However, this project is new and needs more attention, but the security and real-life applicability of the project are appreciable.
4) **Tron:** The Tron platform is an open-source blockchain OS with smart contract functionality, a consensus mechanism based on proof-of-stake principles, and its own coin, Tronix. Tron was formed by H.E. Justin Sun in September 2017.
5) **BSC:** It was revealed by CEO Changpeng Zhao that the Binance Smart Chain was being launched as an alternative to Ethereum in the DeFi industry. In contrast to the original Binance Chain, BSC enables smart contracts and is compatible with the EVM, making it a separate blockchain.
6) **CoreDAO:** These platforms gained rapid traction due to their high-speed consensus mechanisms, low fees, and interoperability. Among them, CoreDAO and Tron emerged as a particularly robust platform, combining efficient transaction processing with decentralized governance and smart contract optimization, positioning it as a

Figure 5
Working diagram of smart contracts in the blockchain

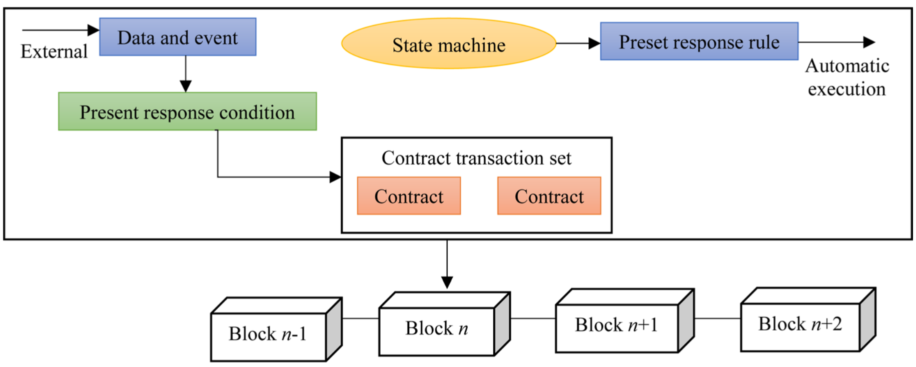
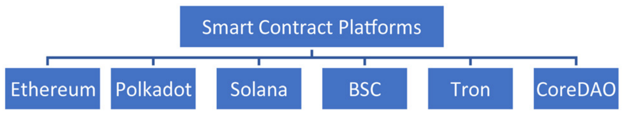


Figure 6
Best six smart contract platforms



leading choice for enterprise-grade applications and DeFi solutions. It evaluates current platforms and identifies CoreDAO and Tron as the most effective solutions for enhancing QoS by reducing transaction overhead, maximizing execution speed, and enabling affordable smart contract deployment [35–37].

Figure 6 shows the best six smart contract platforms that are frequently used to process smart contracts. It shows technical specifications such as blockchain type, native coin, transaction speed, and current price in the case of Ethereum, Tron, and CoreDAO.

3.4. Immutable and transparent smart contracts

The immutability of smart contracts increases confidence and the difficulty of fixing faults in the code. As a result of their immutability, these contracts will remain on the blockchain in perpetuity. Because of this, they are essentially unchanging. Once a piece of code has been deployed in a smart contract, it cannot be altered. Because the contract cannot be altered or reversed, the immutability characteristic is seen as a blessing by many. It ensures that the contract will be performed precisely as coded. By eliminating middlemen in circumstances where the contract terms can be examined openly, smart contracts may enhance this. They make it possible to create contracts that are both immutable and easily accessible. Smart contracts reduce the formality and expenses associated with conventional techniques while maintaining their validity and trustworthiness. Even though all of the data are publicly accessible, encryption may be used to ensure privacy. As a result, only those who have been granted access to the decryption keys may access the private data stored on the blockchain.

The use of smart contracts eliminates the need for intermediaries, saves a great deal of time and money, and is written inside each block of code. When doing research, buying and selling properties require an intermediary, everyone involved must be physically present, and a substantial investment of time and resources is required to complete the transaction. Unfortunately, fraud can still happen. By facilitating the use of smart contracts, the blockchain network gets rid of all of these problems. “Smart contracts” are preprogrammed computer programs that execute themselves in response to certain events. As shown in Table 3, smart contracts provide several benefits over more conventional forms of contracts. Some of the following discrepancies were also noted by the writers of the book. Table 3 shows a variety of smart contract uses. These are the most important applications, although new ones are being developed daily [29, 38–41].

Table 3
Traditional versus smart contracts

S. No	Parameter	Traditional contracts	Smart contracts
1	Need of middlemen	Yes	No
2	Cost	Expensive	In-expensive
3	Need of paper work	Yes	No (digitally signed)
4	Time	Few days	Few minutes
5	Chances of mistakes/ frauds	Fair	No
6	Security	Poor	Excellent

Table 4
Applications of smart contracts

Sr. No.	Applications
1	Buying and selling property
2	Healthcare sector – maintain patient health records securely and transparently
3	Supply chain management – buyers and manufacturers can track information
4	Public donations – tracking public donations, maintaining transparency, and preventing donation misusages
5	Voting – prevents massive election expenditures and restricts voting scams
6	Leasing/selling vehicles – without the need for middlemen and saves time and money
7	Online gambling/lottery – prevents online gambling/lottery fraud by making the whole system transparent

Table 4 shows the application areas where smart contract technology is frequently used. These areas might be healthcare, property, supply chain management, and online gambling and lottery.

Notably, smart contracts are not supported by Bitcoin’s infrastructure. The Ethereum and Neo protocols, for instance, are widely used to construct smart contracts, a feature introduced in Blockchain 2.0. Table 4 lists several different ways where smart contracts may be put to use. Some of the technologies described in Table 5 are needed to create a smart contract.

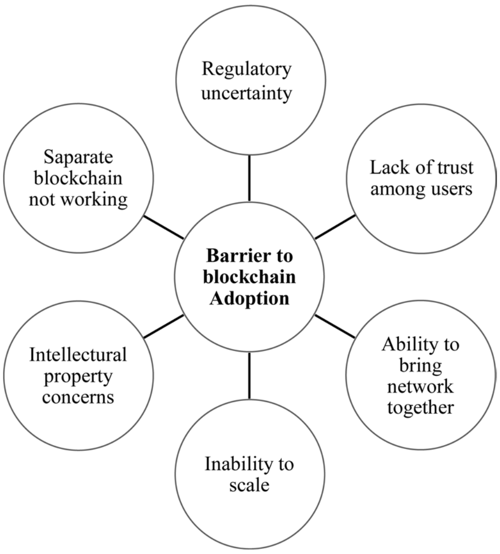
3.5. Barriers to adopting blockchain

To enhance the way the company innovates, these adoption hurdles may be considered as limits that must be addressed and examined. It is very uncommon for regulators to raise extra concerns when approving a new blockchain product, such as that of a bank that is

Table 5
Tools and techniques of smart contracts

Sr. No.	Tools (Ethereum smart contracts)	Description
1	Metamask plugin	A browser-based tool that lets you access and creates new Ethereum addresses and transfer transactions
2	Solidity/JavaScript/Node.js	Programming language
3	Ethereum wallet credentials	Account of Ethereum
4	Blockchain knowledge	It is the technology behind the wall
5	Truffle	Command-line tool for developing and testing Ethereum smart contracts
6	Web3.js	A library that helps in interacting with a local or remote node
7	Ether.js	This is the alternative to Web3.js
8	Remix IDE	Open-source debugging and compiling tool

Figure 7
Biggest barrier to blockchain adoption



utilizing the technology. This interaction with the regulator may serve as a springboard for future innovations in the form of newer, better goods. Figure 7 shows the biggest barrier to the adoption of blockchain. It explicitly identifies scalability, regulatory compliance, and awareness as major barriers to blockchain adoption.

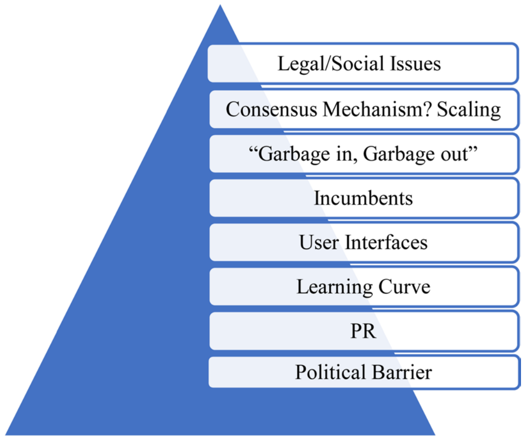
It may be difficult for financial institutions and banks to explain to authorities how data are exchanged between the members of the blockchain. Banks and insurance companies, with their heavy reliance on fiduciary duties, tend to be more conventional. They must adhere to a certain set of conventions. Regulators in these fields are also more risk-averse, and regulatory structures in many nations are decades behind where they should be. Because all transactions need to be broadcasted and documented on all nodes, scalability difficulties resurface, which might reduce the total number of transactions that can be completed. Every blockchain includes consensus protocols designed to keep all of the nodes in lockstep with one another. A transaction can only be added to the blockchain once, which slows down the network. Despite Bitcoin's reputation as a secure and decentralized network, it can only process seven transactions per second. As many as 1000 transactions per second may be processed on less decentralized public blockchains such as EOS. The expense of preserving data in many locations is likely to be higher than when the data are held in a single location. This database is solely replicated for disaster recovery and high availability (HA/DR). Redundancy in HA/DR ensures that the system is highly available, i.e., provides an SLA to the business's expectations. Duplicating the databases is a necessary step in ensuring the system's redundancy. You may think about it like this: let us say that one of your databases does not operate for any number of reasons, including a power outage, hacker intrusion, or system upgrade. When this occurs, the system will automatically look for the duplicated database.

3.6. Barriers to adopting smart contracts

To some blockchain experts, smart contracts' acceptance will expand as the web develops. In a future where everything is interconnected and has to communicate with each other, smart contracts may be a lifesaver.

In contrast, other experts believe that compelling use cases, which are yet lacking, will lead to wider acceptance. Crucial to this procedure is an appreciation for the nature of smart contracts, the functionality of blockchain technology, and the limitations of decentralization. One way to limit the transactions in a database is by using smart contracts, and these contracts cannot leave the database in which they exist. Figure 8 shows the issues in the adoption of smart contracts, which are the user interface, legal issues, learning curve, and political barriers.

Figure 8
Barriers to smart contract adoption



4. Simulation

Because blockchain is still a novel and controversial idea, there have been various obstacles to its widespread implementation. In addition, there is a need to educate people about blockchain technology in a number of nations. Many individuals are wary of adopting blockchain technology because of the many scams and frauds that are linked to it. Another factor affecting blockchain's use is the scarcity of technical tools and internet connectivity. One of the main problems with blockchain technology is how little people know about it. Businesses may face challenges with implementing blockchain technology, as they would with any other new innovation. There are problems with scalability and energy consumption with the current blockchain system. Every participant in a blockchain-based transaction needs access to the same global ledger in order for the transaction to be executed.

To complicate matters even further, the technology's lack of scalability, interoperability, blockchain developers, lack of standards, and high energy needs have all contributed to its sluggish acceptance. There has been a lack of uptake. There must be widespread acceptance for a blockchain ecosystem to operate. There is a lack of expertise due to the immaturity of blockchain. There is a dearth of people with the knowledge and expertise to build and utilize it, credibility among users, and amounts of money. Figure 8 shows the process of the identification of the blockchain adaptation barrier, where the barrier is identified and the final matrix is developed. Internal consistency is identified after the elimination of transitivity. Finally, the total direct influence matrix, influence relation map, and performance sensitivity analysis are obtained using Fuzzy MICMAC if there is no internal consistency. Figure 9 illustrates the integration of literature review, Delphi technique, and expert opinion to establish a self-structural interrelationship matrix. It proceeds through reachability analysis, performance partitioning, and consistency checks, culminating in the fuzzy MICMAC methodology for calculating direct and indirect influence relationships and conducting sensitivity analysis.

4.1. Dataset description

The simulation experiment was conducted using 35,000 transaction records retrieved from the TronScan public API between

Figure 9
Identification process of a blockchain adaptation barrier

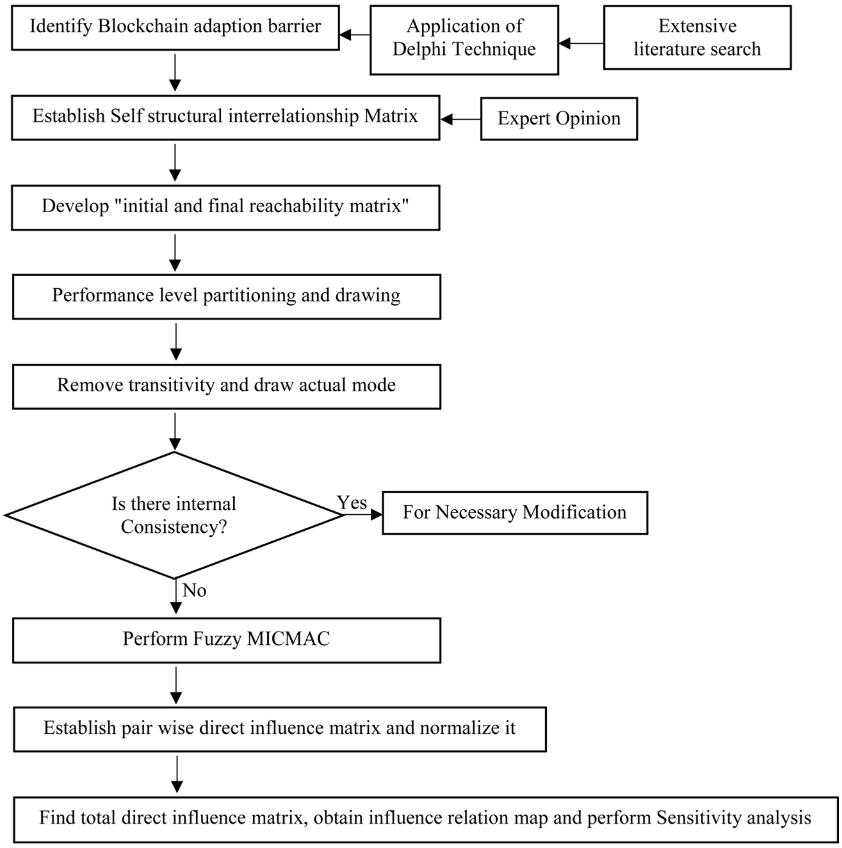
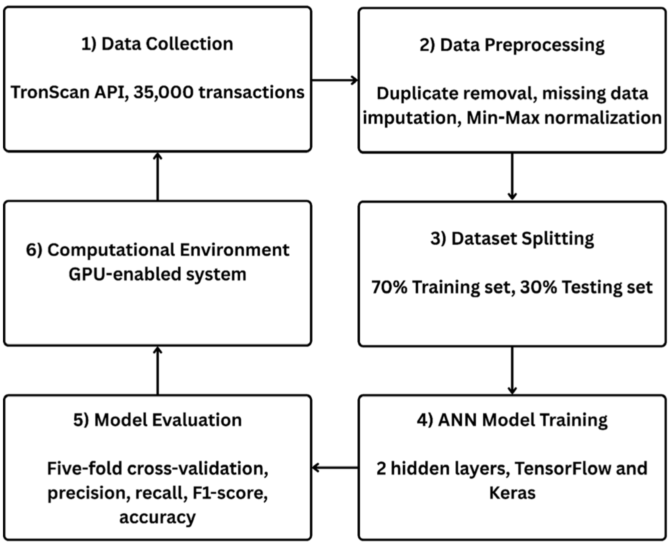


Figure 10
Simulation workflow for blockchain transaction analysis



2024 and 2025. Each record contained the following attributes: transaction hash, block number, sender, receiver, token type, transaction amount, status, and result. The dataset exhibited a class imbalance, with approximately 72% successful transactions and 28% failed transactions, requiring normalization and resampling strategies before training the classification model.

Figure 10 illustrates the step-by-step simulation workflow implemented in this study. The process begins with the collection of 35,000 transaction records via the TronScan API, followed by comprehensive data preprocessing involving duplicate removal, missing value imputation, and min–max normalization. Partitioning the cleaned dataset into training and testing subgroups is the next step. Using the TensorFlow and Keras frameworks, an artificial neural network (ANN) model is trained across 100 epochs with an Adam optimizer. The model is setup with two ReLU-activated hidden layers and a sigmoid output layer. For the sake of consistency and repeatability, the whole process is run in a computing environment that makes use of GPU acceleration.

4.2. Data preprocessing

To ensure that the analyses were accurate and consistent, the data had to be preprocessed. To manage missing values, we used mean substitution for numerical attributes and mode substitution for categorical ones. We also deleted duplicate entries. The min–max scaling technique was used to normalize each numerical feature, ensuring that all values were within the range and improving model convergence. To keep class proportionality, the dataset was stratified-sampled into two parts: one for model training and one for testing. Each part received 70% of the whole dataset.

To ensure high-quality input for model training, preprocessing was conducted in the following steps:

1) **Duplicate removal:** redundant transaction records were eliminated.
2) **Handling missing values:** null attributes were either imputed with median values or removed when excessively sparse.
3) **Categorical encoding:** categorical fields such as token type,

contract method, and transaction type were encoded using **one-hot encoding**.

4) **Numerical normalization:** features like gas limit, block size, and transaction value were scaled using **min–max normalization** to a [0, 1] range.
5) **Class balancing:** to prevent bias toward majority classes, the **Synthetic Minority Oversampling Technique (SMOTE)** was applied during training.

4.3. Feature set

The following attributes were selected as model input features:

Category	Features used
Transaction metadata	Timestamp, block_number, transaction_value, fee_limit, gas_used, energy_consumed
Address attributes	sender_activity_score, receiver_activity_score (derived from past behavior)
Contract context	contract_type (transfer/staking/voting/approval), token_standard (TRC-10/TRC-20)
Execution outcome	Label: success (1)/failure (0)

Derived metrics such as gas-to-fee ratio and historical sender reliability score were also computed to enrich the input space.

4.4. ANN architecture

ANN was built using the TensorFlow and Keras frameworks in Python 3.10. An output layer with a sigmoid activation function completed the model's architecture, which also included an input layer with 7 neurons and 2 hidden layers with 32 and 16 neurons (activated by ReLU). We used the Adam optimizer for 100 iterations with a batch size of 64 and a learning rate of 0.001.

Figure 11 shows the design of the ANN model that was utilized in the simulation investigation. The network is built out of an input layer that corresponds to transaction features and has seven neurons. Then, there are two hidden layers that use the ReLU function: one layer has 32 neurons, and the other has 16 neurons. Information flows from the input to the output because the connections between neurons are fully linked.

4.5. Software and hardware environment

The experiments were conducted on a Windows/Linux machine equipped with the following:

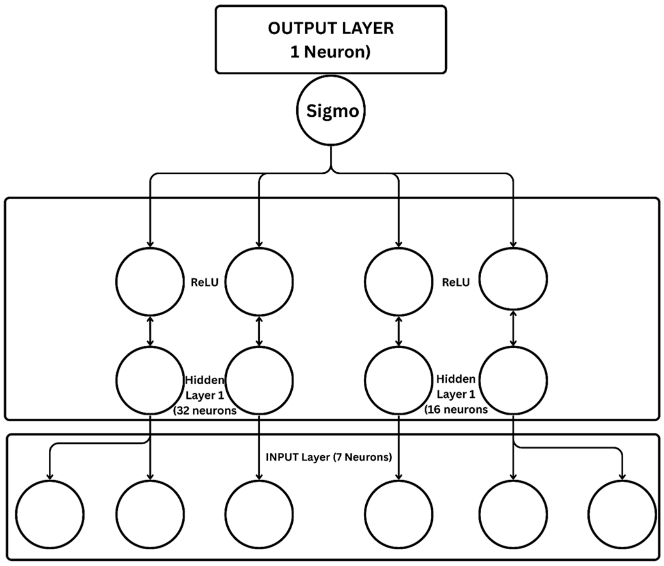
Component	Specification
Programming language	Python 3.10
Frameworks	TensorFlow 2.x, Scikit-Learn, Pandas, NumPy
Hardware	Intel i7/AMD Ryzen CPU, 16 GB RAM (with optional NVIDIA GPU acceleration when available)

Model training and testing were performed in Jupyter Notebook and exported for reproducibility.

4.6. Simulation of Tron transaction accuracy after deep learning

One way to construct a deep learning model is using the data provided by Tron transactions. The study would examine whether TRX

Figure 11
Architecture of the ANN model



transactions on the blockchain are successful or not by making use of an ANN-based deep learning approach. As part of the study, statistical data from the TRON scan have been analyzed, 70% has been used for training, and 30% has been used for testing. Figure 12 shows the simulation work that considers the dataset of transactions from the Tron scan. This dataset is passed to an ANN-based deep learning model for preprocessing, filtering, configuring, training, and testing. After testing, phase accuracy parameters are evaluated.

Training and testing rely on data preparation, which eliminates irrelevant or irrelevant information from the dataset. The model's validity would be confirmed by comparing the two learning systems' accuracy parameters. Sender, receiver, transaction type, status, amount, and token type are just a few of the many variables taken into account by deep learning. Some examples of tokens are Tron and wink. Trigger Smart Contracts, Vote, Stake TRX, Transfer TRX, and Unstake TRX are examples of transaction types. One of two possibilities exists: the status is either verified or not. The outcome might be either SUCCESS or FAILURE. Work on improving machine learning algorithms to anticipate transaction success or failure in a Tron scan environment has been considered. As a result of the user-defined filtering system, anomalies have been reduced. As a result, there has been a notable decrease in the mistake rate. Furthermore, this has increased the likelihood of achieving high levels of accuracy. After training and testing, a confusion matrix is obtained and precision (PR), recall value (RE), computed accuracy (CA), and F1-score (F1) are calculated to confirm accuracy, considering Figure 13.

5. Result and Discussion

During the simulation, 35000 records were considered for training. In addition, 15000 records were tested. Certain records were associated with success and others with failure, as shown in Table 6.

A total of 1,802 transactions were successful, while 1,992 were unsuccessful. Table 7 shows the confusion matrix that was produced through testing and training based on machine learning. During training, the ANN model was employed. From a total of 12543 transactions, only 10802 were approved for processing.

Table 8 shows the calculation of accuracy, precision, recall, and F1-score in the case of ANN-based learning.

Figure 12
ANN based on deep learning of Tron transactions

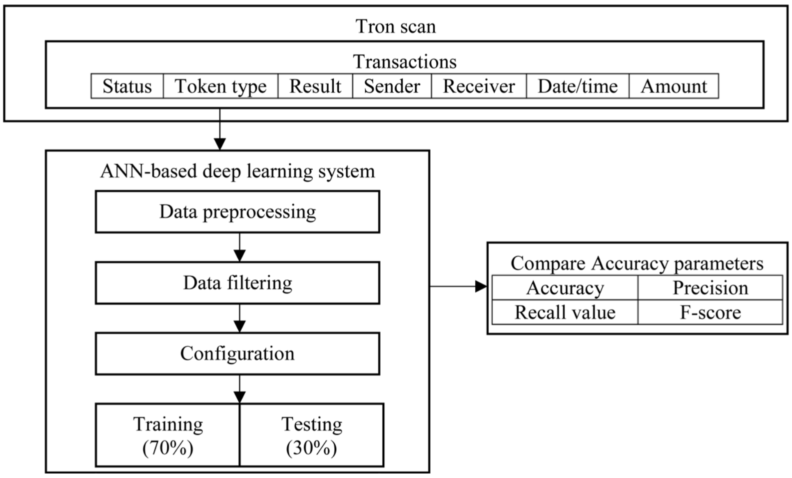


Figure 13
Predicted and true classes

		True class		
Predicted Class		**EFR**	**LFR**	**Total**
	EFR	True Positives	False Positives	**Predicted EFR**
	LFR	False Negatives	True Negatives	**Predicted LFR**
		True EFR	**True LFR**	

Table 6
Accuracy and inaccuracy when it comes to traditional ML methods

True value	**12801**
False value	2199
Total	15000
Accuracy	85.34%
Error	14.66%

Table 7
Traditional ANN-based learning research confusion matrix following simulations

	Success	Failure	Total
Success	10802	1741	12543
Failure	458	1999	2457
Total	11260	3740	15000

5.1. Simulation result for the ANN-based deep learning model after the filtering process

This dataset originally contained 10,000 records, out of which 9,221 valid records were retained after preprocessing. For training purposes, 7,500 of the filtered data points were tested. A total of 2,500 records also undergone testing. Table 9 displays the accuracy, mistake, fake, and real values.

A total of 11,602 transactions were successful, while 2,234 were unsuccessful. The confusion matrix in Table 10 is the result of the tests conducted after training using machine learning. During training, the ANN model was employed. Out of 12543 transactions, 11602% were approved, while the others were rejected. For the ANN-based simulation, the confusion matrix is displayed in Table 10.

Table 11 shows the calculation of the accuracy parameters in the case of the ANN-based deep learning model.

5.2. Cross-validation

The ANN model trained on the preprocessed data consistently achieved a mean accuracy of 92.2% with a standard deviation of ±0.3% across the folds. Additional evaluation metrics, including precision, recall, and F1-score, were similarly stable, demonstrating the model's reliable predictive capability under different data splits. This validation process affirms that the reported performance is not an artifact of data partitioning but reflects true model strength.

Here, we have a direct comparison between the raw dataset and the cleaned version. Table 12 displays a comparison of accuracy.

5.3. Statistical significance testing

To statistically validate the improvements in accuracy observed after data preprocessing and model optimization, a paired t-test was performed. The null hypothesis assumed no difference in

Table 8
Predicting learning accuracy in an ANN environment

Class	N (truth)	N (classified)	Accuracy	Precision	Recall value	F1-score
1	11260	12543	85.34%	0.86	0.96	0.91
2	3740	2457	85.34%	0.81	0.53	0.65

Table 9
Error and accuracy of the ANN-based deep learning model after the filtering process

True value	**13836**
False value	1164
Total	15000
Accuracy	92.24%
Error	7.76%

Table 10
Confusion matrix of the ANN-based deep learning model after the filtering process

	Success	**Failure**	**Total**
Success	11602	941	12543
Failure	223	2234	2457
Total	11825	3175	15000

model accuracy between using the original unfiltered dataset and the preprocessed dataset. The ANN model's accuracy on the unfiltered data averaged 85.3% ± 0.5%, whereas preprocessing increased accuracy to 92.2% ± 0.3%. The paired t-test results showed this increase to be statistically significant ($p < 0.05$), confirming that the preprocessing steps contribute materially to enhancing classification accuracy. Table 11 provides a detailed summary of accuracy, precision, recall, and F1-scores across the folds, reinforcing the statistical robustness of the performance gains.

5.4. Comparative analysis

Figure 14 compares the success and failure accuracy values of the filtered dataset to those of the unfiltered dataset, illustrating a 6.9% performance gain. Redundant captions were merged, and textual explanations were added immediately following each figure to enhance interpretive clarity.

Taking into account Table 12 as a whole, the results indicate a consistent improvement in QoS performance across evaluated platforms. Evidence suggests that compared to unfiltered datasets, filtered datasets provide more accurate results.

Figure 15 compares the success and failure precision values of the filtered dataset to those of the unfiltered dataset, taking Table 13 into account. Results show that compared to the unfiltered method, the filtered dataset provides more accurate results. Table 14 shows the comparison of recall value.

The contrast between the filtered success and failure recall value and the unfiltered value is shown in Figure 16, which is based on Table 14. The filtered dataset outperforms the unprocessed dataset in terms of recall value. Table 15 shows the comparison of the F1-score.

Figure 17 compares the success and failure F1-scores of the filtered dataset to those of the unfiltered dataset, taking Table 15 into account. Compared to the filtered method, the F1-score of the unfiltered data is superior.

Table 12
Evaluation of precision in the two instances

Result	**Accuracy without data filtering**	**Accuracy after data filtering**
Success	85.34%	92.24%
Failure	85.34%	92.24%

Figure 14
Comparison of accuracy

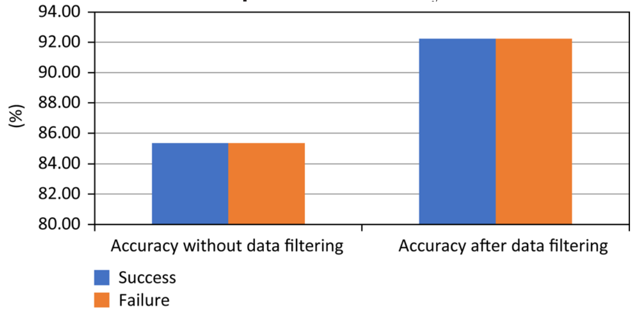


Figure 15
Comparison of precision

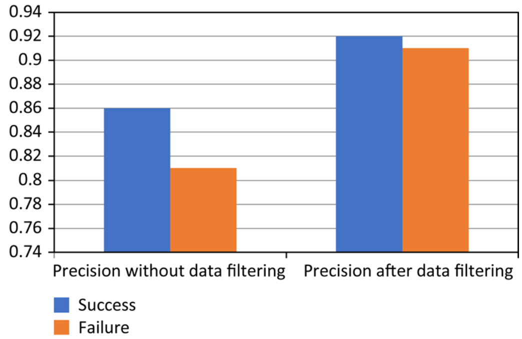


Table 13
Comparison of precision

Result	**Precision without data filtering**	**Precision after data filtering**
Success	0.86	0.92
Failure	0.81	0.91

6. Conclusion and Future Scope

Although this study identifies Tron and CoreDAO as computationally efficient platforms for implementing smart contracts, the findings are limited by the exclusive focus on a single blockchain ecosystem. Other major platforms such as Ethereum, Solana, and Polkadot employ distinct consensus mechanisms, fee models, execution speeds, and developer ecosystems, all of which may

Table 11
Calculation of accuracy parameters of the ANN-based deep learning model after the filtering process

Class	**N (truth)**	**n (classified)**	**Accuracy**	**Precision**	**Recall value**	**F1-score**
1	11825	12543	92.24%	0.92	0.98	0.95
2	3175	2457	92.24%	0.91	0.70	0.79

Table 14
Recall value

Result	Recall without data filtering	Recall after data filtering
Success	0.96	0.98
Failure	0.53	0.70

Figure 16
Comparison of recall value

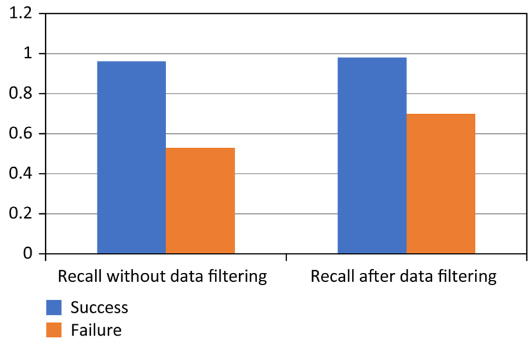
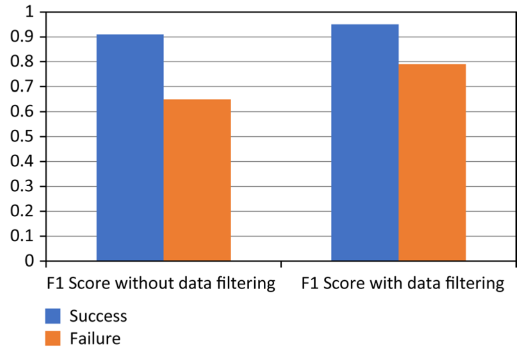


Table 15
F1-score

Result	F1-score without data filtering	F1-score with data filtering
Success	0.91	0.95
Failure	0.65	0.79

Figure 17
Comparison of F1-score



differently influence performance outcomes. Moreover, variations in smart contract development languages and network maturity affect benchmarking consistency and generalizability. Future work should broaden the comparative analysis across multiple blockchain platforms under standardized experimental conditions, also considering trade-offs in security, decentralization, and throughput. In addition, longitudinal studies assessing dynamic network loads will provide critical insights into performance stability over time. Such comprehensive cross-platform evaluation is necessary to robustly validate whether Tron's superior computational efficiency persists across diverse consensus algorithms and operational environments.

In conclusion, after implementing many blockchain platforms for developing smart contracts and after analyzing their relative merits and demerits, the Tron and CoreDAO blockchain platforms were found most effective in terms of transaction time, cost, and mechanism. There exist several blockchain technologies that are used for developing different smart contract platforms. The factors that influence the applicability of blockchain are transaction time, transaction cost, and consensus mechanism. Some blockchain technologies are more reliable and secure, but such technologies are taking a lot of time. After doing a comparative analysis, these blockchains are the most efficient platforms for enhancing the QoS and render low-cost smart contracts for decentralized applications. The real-life usage of such blockchains influences their worldwide acceptance. The complexity of building a smart contract using such a block is also a significant factor that can decide which blockchain should be used in the development of a smart contract. Many scams and frauds are also associated with such blockchain technology that is why many people hesitate to adopt it. The lack of technical instruments and internet availability is also influencing the use of blockchain.

Ethical Statement

This study does not contain any studies with human or animal subjects performed by any of the authors.

Conflicts of Interest

The authors declare that they have no conflicts of interest to this work.

Data Availability Statement

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

Author Contribution Statement

Alock Gupta: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Writing – review & editing, Visualization, Project administration. **Kamlesh Lakhwani:** Conceptualization, Methodology, Validation, Investigation, Resources, Writing – review & editing, Supervision.

References

[1] Swain, S., & Chouhan, V. (2025). A smart contract solution for transparent auctions on permissioned blockchain platform. *Computers and Electrical Engineering*, *121*, 109859. https://doi.org/10.1016/j.compeleceng.2024.109859

[2] Shrivastava, A. L., Dwivedi, R. K., & Prakash, S. (2025). Revolutionizing security in land management system with integration of efficient consensus and smart contract enabled blockchain in land registry procedure. *SN Computer Science*, *6*(1), 76. https://doi.org/10.1007/s42979-024-03436-6

[3] Khan, A., Litchfield, A., Alabdulatif, A., & Khan, F. (2025). BlockPres IPFS: Performance evaluation of blockchain based secure patients prescription record storage using IPFS for smart prescription management system. *Cluster Computing*, *28*(4), 255. https://doi.org/10.1007/s10586-024-05054-6

[4] Sizan, N. S., Dey, D., Abu Layek, Md., Uddin, M. A., & Huh, E.-N. (2025). Evaluating blockchain platforms for IoT applications in Industry 5.0: A comprehensive review. *Blockchain: Research and Applications*, *6*(3), 100276. https://doi.org/10.1016/j.bcra.2025.100276

[5] Jyoti, A., Gupta, P., Gupta, S., Khatter, H., & Mishra, A. (2025). Inherent insights using systematic analytics of developments

tools in Ethereum blockchain smart contract. *Recent Advances in Electrical & Electronic Engineering*, *18*(2), 135–146. https://doi.org/10.2174/0123520965249434231024111732

[6] Zhang, Y., Zheng, S., Wang, H., Wu, L., Huang, G., & Liu, X. (2024). VM matters: A comparison of WASM VMs and EVMs in the performance of blockchain smart contracts. *ACM Transactions on Modeling and Performance Evaluation of Computing Systems*, *9*(2), 1–24. https://doi.org/10.1145/3641103

[7] Liu, Y., He, J., Li, X., Chen, J., Liu, X., Peng, S., …, & Wang, Y. (2024). An overview of blockchain smart contract execution mechanism. *Journal of Industrial Information Integration*, *41*, 100674. https://doi.org/10.1016/j.jii.2024.100674

[8] Singh, R., Gupta, A., & Mittal, P. (2024). A systematic literature review on blockchain-based smart contracts: Platforms, applications, and challenges. *Distributed Ledger Technologies: Research and Practice*, *5*(1), 1–36. https://doi.org/10.1145/3704741

[9] Naik, M., Singh, A. P., & Pradhan, N. R. (2024). Decentralizing ride-sharing: A blockchain-based application with smart contract automation and performance analysis. *Multimedia Tools and Applications*, *84*(24), 29003–29030. https://doi.org/10.1007/s11042-024-20317-5

[10] Ta, M. T., & Do, T. Q. (2024). A study on gas cost of Ethereum smart contracts and performance of blockchain on simulation tool. *Peer-to-Peer Networking and Applications*, *17*(1), 200–212. https://doi.org/10.1007/s12083-023-01598-3

[11] Kaafarani, R., Ismail, L., & Zahwe, O. (2023). An adaptive decision-making approach for better selection of blockchain platform for health insurance frauds detection with smart contracts: Development and performance evaluation. *Procedia Computer Science*, *220*, 470–477. https://doi.org/10.1016/j.procs.2023.03.060

[12] Sharma, P., Jindal, R., & Borah, M. D. (2023). A review of smart contract-based platforms, applications, and challenges. *Cluster Computing*, *26*(1), 395–421. https://doi.org/10.1007/s10586-021-03491-1

[13] Park, J., Jeong, S., & Yeom, K. (2023). Smart contract broker: Improving smart contract reusability in a blockchain environment. *Sensors*, *23*(13), 6149. https://doi.org/10.3390/s23136149

[14] Agrawal, T. K., Angelis, J., Khilji, W. A., Kalaiarasan, R., & Wiktorsson, M. (2023). Demonstration of a blockchain-based framework using smart contracts for supply chain collaboration. *International Journal of Production Research*, *61*(5), 1497–1516. https://doi.org/10.1080/00207543.2022.2039413

[15] Khan, D., Memon, M. M., Hashmani, M. A., Simpao, F. T., Sales, A. C., & Santillan, N. Q. (2023). A critical review on blockchain frameworks for dApp. *International Journal of Technology Management and Information System*, *5*(1), 1–10.

[16] Madhwal, Y., Borbon-Galvez, Y., Etemadi, N., Yanovich, Y., & Creazza, A. (2022). Proof of delivery smart contract for performance measurements. *IEEE Access*, *10*, 69147–69159. https://doi.org/10.1109/ACCESS.2022.3185634

[17] Abhishek, P. M., Narayan, D. G., Altaf, H., & Somashekar, P. (2022). Performance evaluation of Ethereum and Hyperledger Fabric blockchain platforms. In *2022 13th International Conference on Computing Communication and Networking Technologies*, 1–5. https://doi.org/10.1109/ICCCNT54827.2022.9984288

[18] Garcia, R. D., Ramachandran, G., & Ueyama, J. (2022). Exploiting smart contracts in PBFT-based blockchains: A case study in medical prescription system. *Computer Networks*, *211*, 109003. https://doi.org/10.1016/j.comnet.2022.109003

[19] Lin, S.-Y., Zhang, L., Li, J., Ji, L., & Sun, Y. (2022). A survey of application research based on blockchain smart contract. *Wireless Network*s, *28*(2), 635–690. https://doi.org/10.1007/s11276-021-02874-x

[20] Kim, S.-K. (2022). Blockchain smart contract to prevent forgery of degree certificates: Artificial intelligence consensus algorithm. *Electronics*, *11*(14), 2112. https://doi.org/10.3390/electronics11142112

[21] Vacca, A., di Sorbo, A., Visaggio, C. A., & Canfora, G. (2021). A systematic literature review of blockchain and smart contract development: Techniques, tools, and open challenges. *Journal of Systems and Software*, *174*, 110891. https://doi.org/10.1016/j.jss.2020.110891

[22] Khan, S. N., Loukil, F., Ghedira-Guegan, C., Benkhelifa, E., & Bani-Hani, A. (2021). Blockchain smart contracts: Applications, challenges, and future trends. *Peer-to-Peer Networking and Applications*, *14*(5), 2901–2925. https://doi.org/10.1007/s12083-021-01127-0

[23] Solaiman, E., Wike, T., & Sfyrakis, I. (2021). Implementation and evaluation of smart contracts using a hybrid on- and off-blockchain architecture. *Concurrency and Computation: Practice and Experience*, *33*(1), e5811. https://doi.org/10.1002/cpe.5811

[24] Khan, M., & Naz, T. (2021). Smart contracts based on blockchain for decentralized learning management system. *SN Computer Science*, *2*(4), 260. https://doi.org/10.1007/s42979-021-00661-1

[25] Jin, C., Pang, S., Qi, X., Zhang, Z., & Zhou, A. (2021). A high performance concurrency protocol for smart contracts of permissioned blockchain. *IEEE Transactions on Knowledge and Data Engineering*, *34*(11), 5070–5083. https://doi.org/10.1109/TKDE.2021.3059959

[26] Tripathi, G., Ahad, M. A., & Casalino G. (2023). A comprehensive review of blockchain technology: Underlying principles and historical background with future challenges. *Decision Analytics Journal*, *9*, 100344. https://doi.org/10.1016/j.dajour.2023.100344.

[27] Chu, H., Zhang, P., Dong, H., Xiao, Y., Ji, S., & Li, W. (2023). A survey on smart contract vulnerabilities: Data sources, detection and repair. *Information and Software Technology*, *159*, 107221. https://doi.org/10.1016/j.infsof.2023.107221

[28] Uddin, M., Obaidat, M., Manickam, S., Laghari, S. U. A., Dandoush, A., Ullah, H., & Ullah, S. S. (2024). Exploring the convergence of metaverse, blockchain, and AI: Challenges and opportunities. *WIREs Data Mining and Knowledge Discovery*, *14*(6), e1556. https://doi.org/10.1002/widm.1556

[29] Iuliano, G., & di Nucci, D. (2024). *Smart contract vulnerabilities, tools, and benchmarks: An updated systematic literature review*. arXiv. https://arxiv.org/abs/2412.01719

[30] Vidal, F. R., Ivaki, N., & Laranjeiro, N. (2024). Vulnerability detection techniques for smart contracts. *Journal of Systems and Software*, *217*, 112160. https://doi.org/10.1016/j.jss.2024.112160

[31] Alaba, F. A., Sulaimon, H. A., Marisa, M. I., & Najeem, O. (2024). Smart contracts security application and challenges: A review. *Cloud Computing and Data Science*, *5*(1), 15–41.

[32] Singh, R., Gupta, A., & Mittal, P. (2024). A systematic literature review on blockchain-based smart contracts: Platforms, applications, and challenges. *Distributed Ledger Technologies: Research and Practice*, 3704741. https://doi.org/10.1145/3704741

[33] Wu, G., Wang, H., Lai, X., Wang, M., He, D., & Chan, S. (2024). A comprehensive survey of smart contract security: State of the art and research directions. *Journal of Network and Computer Applications*, *226*, 103882. https://doi.org/10.1016/j.jnca.2024.103882

[34] Nakamoto, S. (2008). *Bitcoin: A peer-to-peer electronic cash system*. Retrieved from: https://www.klausnordby.com/bitcoin/Bitcoin_Whitepaper_Document_HD.pdf

[35] Sivaram, T., & B, Saravanan. (2024). Recent developments and challenges using blockchain techniques for peer-to-peer energy

trading: A review. *Results in Engineering*, *24*, 103666. https://doi.org/10.1016/j.rineng.2024.103666

[36] Gupta, S., & Kishan, B. (2025). A performance-driven hybrid text-image classification model for multimodal data. *Scientific Reports*, *15*(1), 11598. https://doi.org/10.1038/s41598-025-95674-8

[37] Kamble, S. S., Gunasekaran, A., Subramanian, N., Ghadge, A., Belhadi, A., & Venkatesh, M. (2023). Blockchain technology's impact on supply chain integration and sustainable supply chain performance: Evidence from the automotive industry. *Annals of Operations Research*, *327*(1), 575–600. https://doi.org/10.1007/s10479-021-04129-6

[38] Avasthi, S., Jain, K., Prakash, A., & Sanwal, T. (2024). A blockchain architecture for secure decentralized system and computing. In *International Conference on Power Electronics and IoT Applications in Renewable Energy and Its Control*, 415–420. https://doi.org/10.1109/PARC59193.2024.10486648

[39] Kaur, K., Venkatesh, R., Pundir, S., Brindha, S., Rexy, V. A. M., & Singh, R. (2024). Machine learning for fraud detection in blockchain transaction. In *International Conference on Knowledge Engineering and Communication Systems*, *1*, 1–6. https://doi.org/10.1109/ICKECS61492.2024.10616821

[40] Bartoletti, M., Benetollo, L., Bugliesi, M., Crafa, S., Dal Sasso, G., Pettinau, R., …, & Zunino, R. (2025). Smart contract languages: A comparative analysis. *Future Generation Computer Systems*, *164*, 107563. https://doi.org/10.1016/j.future.2024.107563

[41] Liu, Y., Liu, A., Xia, Y., Hu, B., Liu, J., Wu, Q., & Tiwari, P. (2023). A blockchain-based cross-domain authentication management system for IoT devices. *IEEE Transactions on Network Science and Engineering*, *11*(1), 115–127. https://doi.org/10.1109/TNSE.2023.3292624

How to Cite: Gupta, A., & Lakhwani, K. (2025). Blockchain Platforms for Developing Smart Contracts and Their Computational Performance Evaluation: A Systematic Literature Review *Artificial Intelligence and Applications*. https://doi.org/10.47852/bonviewAIA52026923