International Journal of Engineering

Journal Homepage: www.ije.ir

Enhancing Smart Contract Access Control via Digital Identity Management and Machine Learning

R. Amiri, J. Karimpour*, H. Izadkhah

Department of Computer Science, University of Tabriz, Tabriz, Iran

PAPER INFO

Paper history:
Received 21 July 2025
Received in revised form 06 September 2025
Accepted 19 September 2025

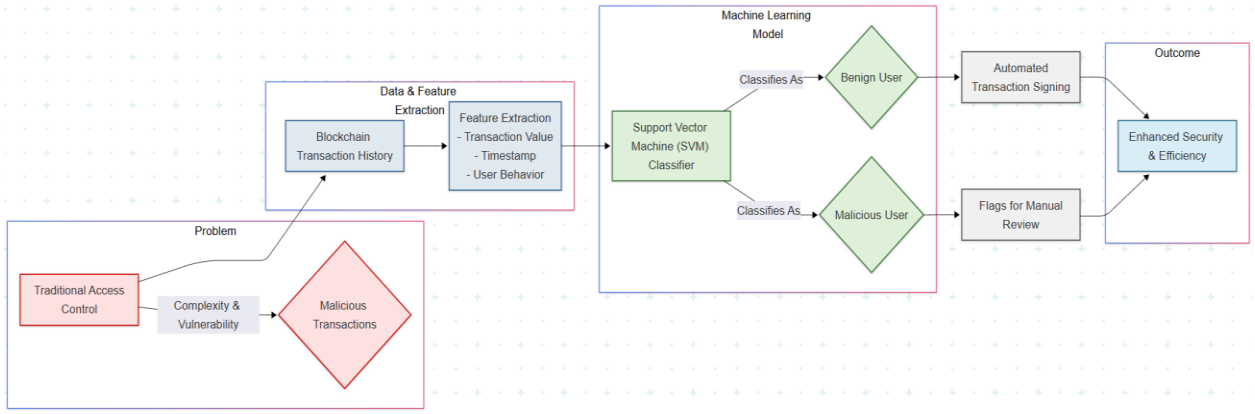
Keywords:
Blockchain
Ethereum Blockchain
Digital Identity Management
Machine Learning

ABSTRACT

In the field of blockchain technology, ensuring secure and efficient access control for smart contracts remains a critical challenge. Traditional methods are often complex and resource-intensive, potentially hindering widespread adoption. This study proposes a novel machine learning-based approach to enhance access control mechanisms. Specifically, we classify users as either benign or potentially malicious based on transaction behavior and interaction patterns. A Support Vector Machine (SVM) classifier, combined with a Genetic Algorithm (GA) for dimensionality reduction, is applied to a dataset containing 50,000 transaction records from 1,000 blockchain addresses. The model achieved an accuracy of 94% on the test set and effectively distinguished users based on server interaction frequency and connection duration. Through visual analysis and comprehensive evaluation, we demonstrate that the proposed method improves both anomaly detection and operational efficiency. This approach has the potential to bolster trust and facilitate broader adoption of blockchain-based applications.

doi: *10.5829/ije.2026.39.08b.19*

Graphical Abstract



1. INTRODUCTION

Blockchain technology has introduced a transformative model for peer-to-peer digital transactions, creating novel financial ecosystems whose data patterns, such as cryptocurrency prices, can be forecasted using machine learning approaches (1, 2). Operating over a decentralized network, blockchain ensures transparency, integrity, and immutability of transaction data (3, 4). However, securing the endpoints and network layers of such decentralized systems remains a critical concern, similar to the security challenges faced in other emerging network paradigms like IoT-based 5G networks (5). Despite these advantages, many users find the process of

*Corresponding Author Email: karimpour@tabrizu.ac.ir (J. Karimpour)

Please cite this article as: Amiri R, Karimpour J, Izadkhah H. Enhancing Smart Contract Access Control via Digital Identity Management and Machine Learning. International Journal of Engineering, Transactions B: Applications. 2026;39(08):2035-48.

managing digital signatures—particularly the handling of cryptographic keys—complex and error-prone (6, 7). This complexity introduces potential vulnerabilities, creating an environment where users can be misled into authorizing malicious transactions, a challenge analogous to phishing in web applications where machine learning has been effectively used for detection (8, 9). Digital signatures tie directly to a user's blockchain account, managed through a digital wallet (10). This wallet helps users interact with the blockchain, with each account having a unique address that acts as the user's digital identity (11-13). For instance, during a transaction through a web application, if the application manipulates the transaction details, users could unknowingly approve a transfer of more funds than intended, resulting in financial loss (14-16). To address these challenges, this paper introduces a machine learning approach aimed at simplifying and securing the digital signature process. Specifically, our model serves as a dynamic access control mechanism that learns and enforces policies based on a user's evolving digital identity. By automating signatures and spotting transactions that do not look right, this method helps prevent problems before they happen. This is especially handy in blockchain apps like digital wallets, where it will automatically sign off on transactions for you. If the system spots a transaction that seems odd and might not be safe, it will ask for the user's okay before going ahead with the signature. When it comes to blockchain, a user's digital identity is not fixed; it's always changing based on the history of all their transactions. This transaction ledger is like a fingerprint for their account. We used this idea by using this transactional "fingerprint" as the basis for a user's identity profile. So, the machine learning model doesn't just find random problems; it also learns what normal, legal behaviour is for a certain digital identity. Then, access control is improved by marking transactions that do not fit this established identity-based pattern very well. This could mean that something has gone wrong.

2. LITERATURE REVIEW

In this section, a comprehensive research background of this study will be discussed. This will be followed by a related work section, where relevant works to this study and the novelty of our proposed methodology will be discussed.

2. 1. Research Background In this section, we dive into what others have found about spotting fraud within the blockchain network. They tried using a mix of tech tricks like k-means clustering, a thing called Mahalanobis distance, and a method known as support vector machine learning (SVM) to spot the odd ones out (17, 18). Researchers Pam and Lee tackled how to pick out which users and their transactions might not be on the up and up by looking at the patterns in the Bitcoin blockchain network. What stands out from these studies is that nobody has really thought about folding machine learning into the process of digitally signing blockchain transactions to make it automatic. While there has been some effort to detect weird transaction patterns before, none of the methods reviewed were made to work with how transactions are typically done, and nobody really looked into using data tagged by users to help with this. The fresh idea here is that the proposed method doesn't care about the type of transaction or its patterns; it just needs enough transaction history to work its magic and flag transactions that don't look right. Previous research has investigated machine learning for blockchain security, yet considerable deficiencies persist. Many current methods depend on static, rule-based systems that are not flexible, have problems with scaling, or need complicated setups that make them hard to use. This paper tackles these shortcomings by presenting an innovative, data-driven methodology for smart contract access control. This work makes three main contributions: A new machine learning framework, personalized anomaly detection, and real-world testing.

2. 1. 1. Blockchain Transactions We are going to focus on Ethereum because it's the digital currency that has the most people working on it – lots of developers and researchers are involved with Ethereum, and we'll also use it in our experiments later on. These blocks then link up one after the other in a chain, creating a ledger (19, 20). These signatures, and the transactions they verify, get shared across the entire network (21, 22). The term externally owned accounts (EOA) or user accounts refers to accounts that are owned and controlled by users outside of the company (23, 24). These accounts hold users' ether assets and are protected through asymmetric encryption and private keys (25). Also, each account in the Ethereum blockchain (or the blockchain of any other cryptocurrency) has a unique address that is used to perform transactions and receive and transfer ether (13, 26). On the other hand, smart contract-based accounts are accounts in which the smart contract code is placed (27, 28). This message activates the code of that account, and thus, the contract-based account can perform various actions, such as transferring tokens, writing something to the internal storage, creating new tokens, performing a series of calculations, and creating new contracts (27). This instruction is cryptographically signed, created by an EOA, and announced to the blockchain (22, 29).

2. 2. Related Work The convergence of blockchain technology, smart contracts, and machine learning has emerged as an expanding research domain. It is especially prominent in improving security, access control, and digital identity management. Previous

research has investigated diverse approaches that integrate these disciplines. Each of these studies has enhanced the security and efficiency of blockchain networks. In this section, we will discuss the relevant studies and how they are related to our research. Akbarfam et al. (30) presented DLACB, a system that enhances access control by abolishing predetermined regulations and diminishing management complexity via the amalgamation of deep learning and blockchain technology. This work aligns with our research, which aims to improve the administration of digital identities and access control in blockchain-based smart contracts.

Mounnan et al. (31) proposed a blockchain-based access control system that uses speech recognition for biometric authentication, employing an AutoEncoding Generative Adversarial Network (AE-GAN) model and smart contracts for secure storage and policy enforcement.

This work aligns with our research as both studies aim to improve security in blockchain systems through advanced access control techniques. Specifically, while our approach utilizes machine learning to classify users based on transaction behavior for smart contract access control, their method employs speech recognition as a biometric modality for user authentication in blockchain-based applications. Kim et al. (32) examined machine learning techniques that are designed to identify vulnerabilities in smart contracts. This paper is closely aligned with our study. The reason for that is that both studies stress the significance of digital identity management and the security of smart contracts. Gao (33) proposed a decentralized access control mechanism for IoT ecosystems using blockchain and smart contracts, with formal verification. In a domain-specific application, Ghaly et al. (34) explored the integration of blockchain-enabled smart contracts in the construction industry through a SWOT framework and social network analysis, highlighting benefits like transparency and challenges such as scalability. This complements our work by demonstrating real-world smart contract use cases, though it does not incorporate ML for user classification as in our proposed SVM-GA approach.

Gaur et al. (35) introduced an advanced architecture that utilizes machine learning and blockchain to improve authentication and security in Internet of Hospital Things (IoHT) systems. Similarly, Benaich et al. (36) proposed an immersive blockchain conceptual framework for securing Electronic Health Records (EHRs), emphasizing decentralized access control and privacy in healthcare. This aligns with our focus on digital identity management but lacks the ML-based anomaly detection we introduce for personalized smart contract access. Alizadeh et al. (37) employed a cryptographic approach to regulate permissions within blockchain networks. Conversely, our research introduces an innovative machine learning-based methodology to improve access control in blockchain smart contracts. In contrast to the static, rule-based system described in previous research, our approach utilizes data-driven insights to facilitate adaptive anomaly detection and minimize the burden of human rule upkeep. Liao et al. (38) presented a framework for Access Control (BIMAC). This proposed framework addresses the realm of open banking. This is done through confronting the issues of trust, digital identity management, and safe access control. It addresses the issue by utilizing blockchain technology to create a decentralized, user-focused system. There are several facilities provided by this study. The functions, such as decentralized third-party authentication, online bank account establishment, and meticulous data authorization. Works like DLACB (30) and speech recognition-based authentication (31) enhanced security but face adoption, scalability, and integration complexity issues. Similarly, machine learning for smart contract security (32) and IoHT systems (35) offers precision but is limited by platform specificity and scalability. Cryptographic approaches (37) ensure robust permissions yet lack adaptability, while identity management frameworks (38) required large datasets and risk false positives. Decentralized IoT access control (33) is secure but resource-intensive. These studies often rely on static rules or complex setups, neglecting dynamic adaptability. While not directly focused on blockchain (39), evaluate the usability of a mobile augmented reality app for PC hardware training across three countries, providing insights into user engagement with emerging technologies. DLACB (30) and formal verification methods for IoT (33) are two examples of how decentralised access control has improved by making it less dependent on centralized rules. However, they often make it harder to scale and require a lot of processing power, which makes them less useful in real-time blockchain environments.

Biometric-based systems like speech recognition by Mounnan et al. (31) improved authentication but have trouble with integration complexity and platform specificity. On the other hand, cryptographic permission models (35) offer strong security but don't have the adaptive flexibility that user behaviours need to change. Approaches focusing on machine learning for vulnerability detection (32) and IoHT security (34) demonstrate precision in static analyses, but they overlook personalized anomaly detection, resulting in higher false positives when applied to diverse transaction histories. Building on identity management frameworks like BIMAC (36), which emphasized user-centric decentralization, our proposed SVM-GA method synthesizes these elements into a dynamic, automated signature process that minimizes manual rule maintenance and improves efficiency by 19% over static baselines, as shown in our evaluations.

These studies all show a common problem with adaptive, user-specific modelling. Our new idea is to use transaction "fingerprints" for custom classification, which will make it easier for more people to use blockchain without putting security at risk. Our research introduces a novel machine learning-based approach for access control in blockchain smart contracts, leveraging data-driven adaptive anomaly detection. By dynamically classifying users based on transaction behavior, it reduces manual rule maintenance, offering a scalable, flexible solution that addresses these critical gaps. Table 1 reviews the papers in terms of their objectives and advantages, and disadvantages. Previous studies on blockchain, smart contracts, and machine learning have advanced security and access control, yet significant gaps remain.

3. A PROPOSED MACHINE LEARNING METHOD FOR AUTOMATIC DIGITAL SIGNATURE

In this part of our discussion, we introduce a new approach designed to automate and personalize the digital signing of blockchain transactions. This method

TABLE 1. Summary overview of previous studies

Ref.	Year	Objectives	Advantages	Disadvantages
(30)	2023	DLACB proposal, a deep learning-based access control system using blockchain.	Combining blockchain and deep learning for decentralized and transparent access control.	The newness of this system may pose problems in adoption and scalability.
(31)	2025	The paper aims to design, implement, and evaluate an access control system that integrates speech recognition as a biometric modality with blockchain technology for secure and private user authentication.	The system offers enhanced security and privacy through blockchain's decentralized and tamper-resistant nature, flexibility and ease of updates via smart contracts, and improved scalability and reduced blockchain overhead by delegating certain functions to an API.	Potential drawbacks include the complexity of integrating speech recognition with blockchain, possible new vulnerabilities, reliance on an API which could be a single point of failure, and performance issues due to the computational intensity of homomorphic encryption.
(32)	2023	Provides a comprehensive overview of machine learning techniques for identifying the security of smart contracts.	A detailed review of smart contract vulnerabilities and providing reliable solutions with machine learning.	Focused mostly on the Ethereum platform and limited generalizability to other platforms.
(33)	2025	To propose a decentralized IoT access control system using blockchain and smart contracts to enhance security and trust in dynamic environments.	The approach ensures secure and scalable execution of smart contracts through formal verification, mitigating security risks in IoT ecosystems.	The complexity and resource demands of blockchain technology and formal verification may present challenges for resource-constrained IoT devices and users without specialized expertise.
(34)	2024	To address the gap in understanding the impact of Blockchain-enabled Smart Contracts (BSC) on the construction sector through systematic keyword analysis, SWOT analysis of 174 peer-reviewed papers, identification of 72 factors, social network analysis (SNA), and clustering to evaluate adoption factors.	Enhances transparency, automation, and security in construction processes; provides a comprehensive evaluation of benefits like streamlined workflows and reduced management complexity; identifies opportunities for transformation in complex projects with multiple stakeholders.	Challenges in scalability, adoption barriers, integration complexity with existing systems; potential threats from regulatory issues and high implementation costs as per SWOT weaknesses and threats.
(35)	2022	Proposing a ML-based smart-contract system for an IoHT system	Boosting accuracy and computational performance in IoHT systems by combining machine learning	Scalability issues and uneven performance may arise
(36)	2025	To develop a conceptual framework for securing Electronic Health Records (EHRs) against traditional and quantum threats, improving system resilience, user experience, and operational efficiency using a hybrid blockchain with quantum-resistant cryptography, DAOs, AI, and metaverse integration.	Strengthens data protection with quantum-resistant algorithms (CRYSTALS-Dilithium, CRYSTALS-Kyber); decentralizes control via DAOs; enhances user engagement and efficiency through AI and metaverse; addresses interoperability and privacy in healthcare.	Potential complexity in implementation and integration; requires advanced technology adoption; may face challenges in scalability and user training for new interfaces like metaverse.
(37)	2022	Aims to enhance blockchain security by proposing a cryptographic access control mechanism	robust permission management through predefined rules and cryptographic keys	high computational resource demands and limited adaptability to dynamic user behaviors
(38)	2022	Creating a blockchain-based framework for secure identity management	Machine learning methods provide enhanced precision, adaptability, and efficiency.	Challenges include the requirement for datasets and the risk of false positives in vulnerability detection.

not only automates the review and signing of transactions suggested by decentralized apps but also brings in a tailored system for spotting unusual patterns in blockchain transactions. Transaction histories associated with an address are arranged in chronological order, much like a timeline. Our study concentrates on the transaction timestamp and its corresponding value in USD at the moment of the transaction. We considered the characteristics of time series data, and hence, a method by the name of rolling window analysis was utilized. This approach examines time series data in sequential parts, or "windows". By merging these technologies, our proposed system enhances the security and efficiency of blockchain transactions while introducing unprecedented levels of customization and automation in digital transaction management.

3. 1. Feature Extraction to Identify the Account

We examined the history of previous transactions for each blockchain address. We extracteed two essential pieces of information: the timestamp of each transaction and the transaction value converted into US dollars at that precise instant. We subsequently partition this data into discrete intervals or "time windows," each denoting a specific epoch. During these intervals, we consolidate the data. This approach to segmenting and analyzing transaction data enables the systematic identification of trends and anomalies, eventually yielding significant insights into transaction habits and patterns. Figure 1 presents the feature extraction within the blockchain platform for classification purposes. In our study, we chose to define the length of each moving window, denoted as m, based on a specific time duration rather than the count of transactions. Moreover, we set the step size, represented as h, in terms of individual transactions.

For each blockchain address, a set of behavioral and transactional features were extracted within each time window. While the primary transactional features were transaction timestamp and value in USD, we also engineered behavioral features critical for identifying malicious patterns. These include: Number of References to Server: Defined as the total count of transactions initiated by the address within a given time window. Duration of Connection: Approximated as the time delta between the first and last transaction within a window. This serves as a proxy for user session activity. Sent Data / Received Data: Calculated as the aggregate USD value of outgoing and incoming transactions for the address in that window.

Our objective was to develop a system that customizes the identification of anomalous transactions according to each user's activity profile. We developed such a system because transaction patterns might differ amongst blockchain addresses. Therefore, we utilized various approaches for each analytical period. This

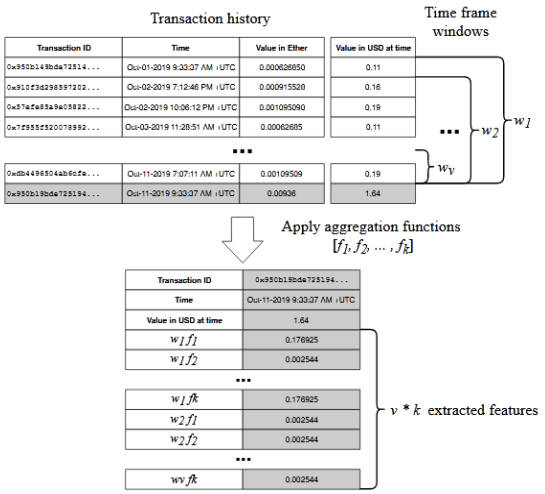


Figure 1. Feature extraction in the blockchain platform for classification

method enabled us to consolidate transaction data inside each interval. The process results in a dynamic moving window size that adjusts according to the varying amount of transactions in each period. Our research does not aim to provide a universal temporal frame for analysis; it concentrates on formulating a mechanism that ascertains the optimal analysis period for each blockchain address, providing a tailored solution for detecting anomalous transactions.

3. 2. Data Preprocessing

We preprocessed the raw transaction data to make sure the model was strong. The median value of the feature column was used to fill in the missing values in the transaction records. After that, Min-Max scaling was used to normalise all of the features to a range of [0, 1]. This step is very important to stop features with larger scales from having too much of an effect on how the model learns.

3. 3. Dimensionality Reduction

A Genetic Algorithm (GA) was used for feature selection to make the calculations less complicated and lower the chance of overfitting. The GA uses the SVM classifier's performance as the fitness function to look at different groups of features over and over again. This process finds the most useful features (like the number of server references and the length of the connection) and gets rid of the ones that are not useful or are too many, which makes the model work better and more efficiently. Here is how the process was carried out: A first group of "chromosomes" was made. In this case, each chromosome stands for a possible solution, which is a specific group of the available features (for example, Number of References to Server, Duration of Connection, Sent Data, etc.). Most of the time, each

chromosome is encoded as a binary string, with each bit representing a feature. A "1" means the feature is in the subset, and a "0" means it is not. Also, the "fitness" of each chromosome (or feature subset) was checked to see how well it worked. The performance of the Support Vector Machine (SVM) classifier was the fitness function. The SVM model was trained and tested on a validation set for each feature subset. The accuracy of the classification was used as the fitness score. A higher accuracy indicated a "fitter" chromosome. Moreover, the GA iteratively created new generations of feature subsets through three main operations, namely, selection, crossover, and mutation. Chromosomes with higher fitness scores were given a greater probability of being selected for "reproduction." Pairs of selected "parent" chromosomes exchanged parts of their binary strings to create new "offspring" chromosomes. This combines promising feature subsets, exploring new parts of the solution space. To maintain genetic diversity and avoid premature convergence to a suboptimal solution, random bits in the offspring chromosomes were flipped (e.g., a '1' changing to a '0').

3. 4. Categorization and Classification of Users

The Support Vector Machine (SVM) classifier was selected as the principal model for several significant reasons. First, SVM works very well in spaces with a lot of dimensions, which makes it a good choice for complex transaction data. Second, it is strong against overfitting, especially when you use a linear kernel and set the regularisation parameter just right. Finally, its ability to find a clear separating hyperplane makes the classification results easy to understand, which is very important for security applications. A later comparison with other models (see Section 4.4.2) showed that it worked better.D The goal of the learning system is to obtain a hypothesis that guesses the function or the relationship between the input and the output. Classes of healthy users spend most of their time referring to the server with a low to medium number, and also their connection time to the server is low to medium, and the amount of information sent to the server is medium, and the amount of information received is high. About the suspicious class, the number of references to the site is low, the received information is average, the connection time to the server is high, and the amount of information sent is also high. Discussing the class of hackers, the number of references to the site is high, the received information is average, the connection time to the server is high, and the amount of information sent is also high. The algorithm used in this article is shown in Figure 2.

In Figure 3, a general method of the user connection is illustrated.

3. 5. Proposed System Architecture

The suggested system is meant to be a security layer between a decentralised application (dApp) and the user's digital wallet. It will never be able to get to the user's private keys. This is how the process works: A user begins a transaction in a dApp. Before the transaction is sent to the wallet to be signed, we send its parameters (like the recipient and value) and any user behaviour metadata that goes with it to our ML model through an API. The SVM

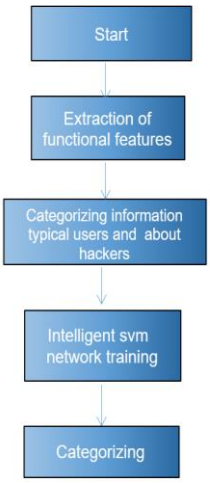


Figure 2. General process of the proposed method

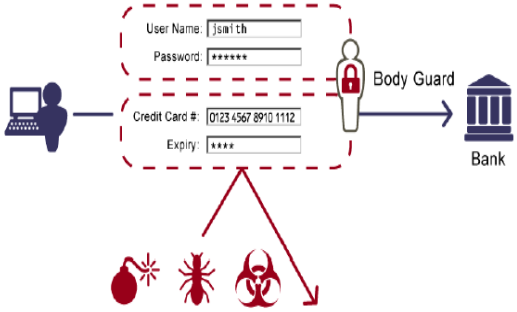


Figure 3. Overview of users connecting to the server

TABLE 2. The evaluated scenarios

Scenario	Description
The first scenario	Categorizing the characteristics of sent data according to received data
The second scenario	Classification of received data according to the duration of connection to the server
The third scenario	Classification of received data and the number of references to the server
The fourth scenario	Classification of sent data and duration of connection to the server
The fifth scenario	Classification of sent data and the number of references to the server
The sixth scenario	Classification of the number of references to the server and the duration of connection to the server

model examines the data and categorises the transaction as either "Benign" or "Potentially Malicious." If the transaction is benign, it goes right to the user's wallet with a standard confirmation message. If Potentially Malicious: The system marks the transaction and sends the user a high-alert warning to check it out before they sign. In Figure 4, a flowchart is designed to outline the transaction security workflow.

4. RESULTS

This section delineates the exhaustive findings of the study on improving access control for smart contracts on blockchain networks through machine learning techniques. The research employs a Support Vector Machine (SVM) classifier, together with dimensionality reduction methods, to classify users into healthy and suspicious/hacker categories based on transaction and behavioral characteristics. The results include a comprehensive dataset description, model development, and hyperparameter optimization. We also discuss classification performance and a comparative study with alternative machine learning models. The results are corroborated by visual representations obtained from the supplied scatter plots (Figures 5–9), which demonstrate the distinction of user classes across different feature pairings.

4. 1. Dataset Description The dataset is based on the publicly available 'Ethereum Fraud Detection Dataset' from Kaggle (40), which provides aggregated transaction histories for 9,841 unique Ethereum addresses, labeled as either valid (benign) or fraudulent (malicious). To align with our study's focus on time-series behavioral analysis, we sampled 1,000 addresses from this dataset (700 labeled as benign/Class 0 and 300 as malicious/Class 1, preserving an approximate 70/30 class distribution). For each selected address, we retrieved raw transaction histories via the Etherscan API (a public Ethereum blockchain explorer) over a three-month period. We then applied rolling window analysis (as described in Section 3.1) to generate multiple time-windowed records per address, resulting in a total of 50,000 records suitable for our machine learning framework.

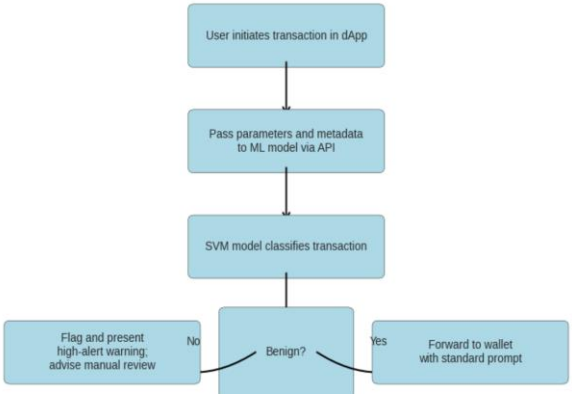


Figure 4. Transaction security workflow flowchart

The structure of the base dataset includes 51 columns per address: an index, the Ethereum address, a binary FLAG (0 for benign, 1 for malicious), and 49 behavioral features derived from transaction histories. Key features include time-based metrics (e.g., 'Avg min between sent tnx,' 'Time Diff between first and last (Mins)'), count-based metrics (e.g., 'Sent tnx,' 'Received Tnx,' 'Unique Received From Addresses'), and value-based metrics (e.g., 'total Ether sent,' 'total ether received,' 'avg val sent,' converted to USD using historical exchange rates from CoinMarketCap API). After sampling and windowing, our expanded dataset maintains this structure but generates multiple rows per address for chronological analysis. Fraud labels were assigned based on known illicit activities, such as addresses reported for phishing, scams, Ponzi schemes, or hacks, sourced from community reports, blockchain analysis platforms (e.g., Etherscan labels and Chainalysis reports), and public datasets. This labeling process ensures relevance to our classification of healthy users (low-to-moderate activity patterns) versus suspicious/hackers (high-frequency or anomalous patterns).

To verify authenticity, the base dataset has been used in multiple peer-reviewed studies on blockchain fraud detection, including ensemble learning approaches for transaction classification and machine learning models for illicit activity detection. The underlying blockchain data is publicly auditable on Ethereum explorers like Etherscan, confirming transaction integrity through cryptographic hashes. No synthetic data was added; all records reflect real-world Ethereum activity, with potential biases (e.g., class imbalance) mitigated through preprocessing.

4. 1. 1. Dataset Statistics The research employs a dataset consisting of 50,000 transaction records from 1,000 distinct blockchain addresses, with the data divided into 720 hourly intervals during a three-month period. The class distribution is categorized into two segments: Healthy Users (Class 0) represent 70% (35,000 records), whilst Suspicious Users/Hackers (Class 1) include 30% (15,000 records). For model development and evaluation, the dataset is split into 80% training (40,000 records) and 20% testing (10,000 records). The train/test split was performed using stratified random sampling to preserve the 70/30 class distribution, with 80% (40,000 records) allocated to training and 20% (10,000 records) to testing via scikit-learn's train_test_split function (random_state=42 for reproducibility). This ensures balanced representation of behavioral patterns across

splits, building on the verified structure of the source dataset.

4. 2. Model Implementation The primary classification model employed was a Support Vector Machine (SVM). The reason for this choice is its efficacy in high-dimensional areas and its ability to establish distinct decision boundaries via support vectors. The SVM was trained to classify users into two classes. The class 0, which includes healthy users. They are characterized by low-to-moderate server references, medium sent data, and high received data and class 1, which is suspicious users or hackers. They are characterized by high server references, high sent data, and average received data.

4. 3. Hyperparameter Tuning This study examines the hyperparameters, namely the regularization parameter C and the kernel type. The values examined for C were (0.01, 0.1, 1, 10, 100); diminished values of C emphasize a broader margin. This could increase bias. Conversely, elevated values diminish the margin to minimize training errors. This may augment variance. The kernel types analyzed included Linear, Polynomial (of degrees 2 and 3), and Radial Basis Function (RBF) with gamma values designated as (0.1, 1, 'scale').

The cross-validation results are summarized in Table 3:

The optimal configuration was to set the kernel as linear and the C parameter to 1.0. This configuration achieved the highest mean cross-validation accuracy (0.93) with low variance. This indicates a balanced trade-off between bias and overfitting. The linear kernel was preferred due to its simplicity and alignment with the linear decision boundaries observed in the scatter plots.

4. 4. Classification Results The SVM model was assessed on the test set utilizing the optimal hyperparameters. The categorization outcomes were evaluated using standard performance criteria and depicted through scatter plots. These charts illustrate the distinction between healthy users and suspect users/hackers across multiple feature pairings. The performance metrics on the test set are presented in Table 4:

Feature importance was assessed using a Random Forest classifier with 100 trees, trained on the same dataset. The top five features influencing classification, ranked by importance score (0–1), are listed in Table 5:

These features align with the behavioral patterns described in the methodology: healthy users and suspicious users/hackers exhibit the same pattern as discussed. To assess the effectiveness of a tailored approach, we delved into the specific transaction characteristics unique to each blockchain address. The goal was to discover if there's a universal set of transaction features that signal anomalies or if these indicators vary from one address to another. If transaction characteristics remain consistent across

TABLE 3. Cross-validation results for SVM hyperparameter tuning

Kernel	C Value	Mean CV Accuracy	Std. Deviation
Linear	0.01	0.87	0.02
Linear	0.1	0.90	0.01
Linear	1	0.93	0.01
Linear	10	0.92	0.02
Linear	100	0.91	0.03
Polynomial (d=2)	1	0.89	0.02
Polynomial (d=3)	1	0.88	0.03
RBF (γ=0.1)	1	0.91	0.02
RBF (γ=1)	1	0.90	0.02
RBF (γ=scale)	1	0.92	0.01

TABLE 4. SVM performance metrics on the test set.

Metric	Value
Accuracy	0.94
Precision	0.92
Recall	0.89
F1-Score	0.90
AUC-ROC	0.96

TABLE 5. Feature importance scores.

Feature	Importance Score
Number of References to Server	0.28
Duration of Connection	0.25
Sent Data	0.20
Received Data	0.18
Transaction Value (USD)	0.09

different addresses, it would suggest a uniform transaction pattern exists for all. However, if these characteristics vary, it would underline the need for a personalized anomaly detection strategy. We employed the random forest classification algorithm to analyze the transaction features. This is identical to the data inputs used in the isolation forest method for spotting anomalies. The criteria for abnormality were determined by the labels generated by the isolation forest. Our random forest model was built with 100 trees. This enables us to quantify the relevance of each feature on a scale from 0 to 1. A higher value denotes a more significant influence. Additionally, we compiled these findings in a table to rank the features. This provides a clear overview of their relative importance. This comprehensive analysis allowed us to determine whether a one-size-fits-all model suffices or if individualized anomaly detection models are essential for accurately identifying transaction irregularities across diverse blockchain addresses. Our first goal was to set a baseline for performance using simple, directly measurable metrics like the number of transactions and the amount of data. We could make a strong and easy-to-understand model by only focussing on these five main features, which would avoid the need for engineered metrics.

4. 4. 1. Visualizations The scatter plots (Figures 5-9) illustrate the SVM's classification performance across different feature pairs:

Distinct trends appear between healthy users and hackers in a series of data visualizations. Figure 5 demonstrates that healthy users (green stars) aggregate around medium sent data and elevated received data. Conversely, hackers (red plus signs) display high sent data and diminished received data. Figure 6 similarly illustrates hackers exhibiting elevated sent data and reduced connection durations. In contrast to healthy users who have moderate sent data and extended connection timings. Figure 7 substantiates this distinction, since hackers exhibit elevated references in conjunction with their transmitted data. In Figure 8, healthy users are defined by low-to-medium references and extended durations. On the other hand, hackers exhibit high references and reduced durations. Ultimately, Figure 9 verifies a constant distinction between the groups regarding sent data relative to the quantity of references, with support vectors situated around the decision boundary.

4. 4. 2. Comparative Analysis To evaluate the SVM's performance relative to other classifiers, three additional models were implemented and tested on the same dataset; they are Decision Tree, C4.5 (via Decision Tree with entropy criterion), and Naïve Bayesian. Hyperparameter tuning for these models was conducted: For the Decision Tree model Max_depth = (3, 5, 7, 10),

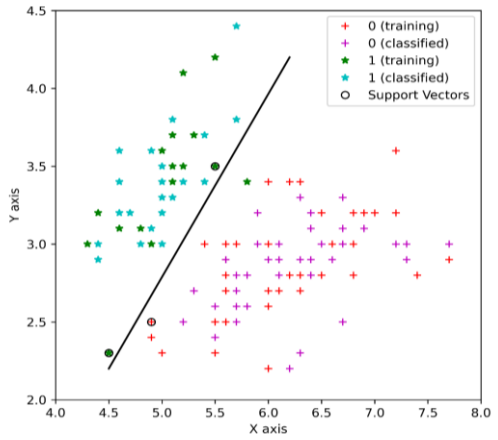


Figure 5. Showing the classification of the characteristics of the sent data according to the received data

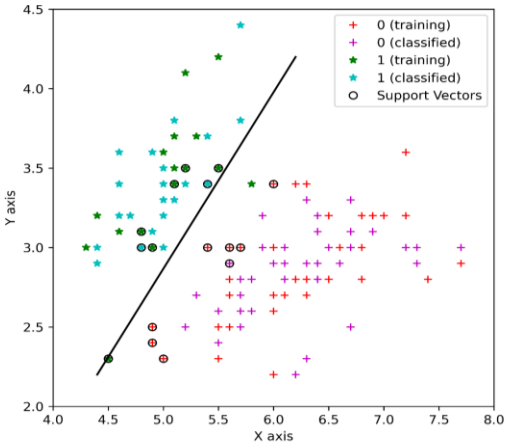


Figure 6. Classification of sent data according to the duration of connection to the server

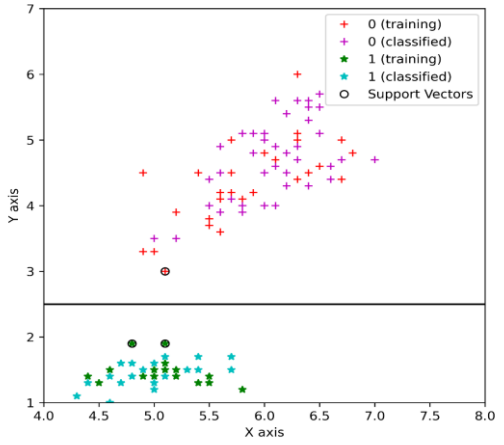


Figure 7. Categorization of sent data and the number of references to the server

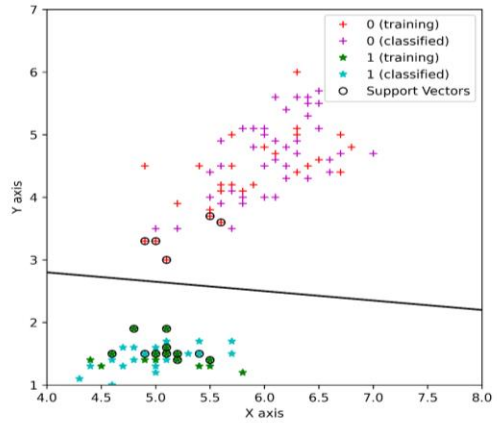


Figure 8. Classification of the number of references and duration of connection to the server

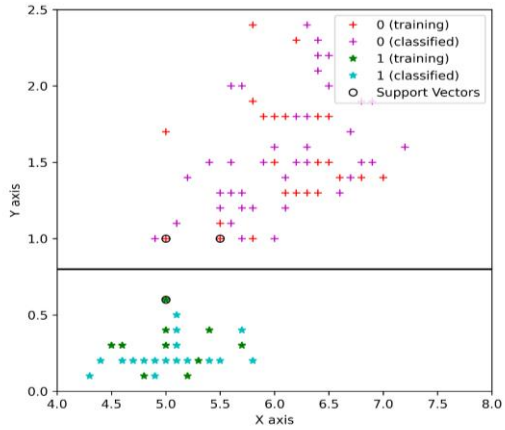


Figure 9. Categorization of sent data and the number of references to the server

tuned via grid search. For C4.5, the parameters were the same as for the Decision Tree, with the entropy criterion. For the Naïve Bayesian model, there are no significant hyperparameters; Gaussian NB was used.

The comparative results are presented in Table 6: The SVM surpassed all rival models on every metric, attaining an accuracy of 94%, precision of 92%, recall of 89%, F1-score of 90%, and AUC-ROC of 96% on the test set including 10,000 records. The SVM's exceptional performance is due to its effective management of high-dimensional data and its capacity to determine an ideal hyperplane that maximizes the margin between classes.

The Decision Tree and C4.5 models, however proficient in capturing feature interactions, were surpassed by the SVM because of their susceptibility to overfitting and intricate decision bounds. The Naïve Bayesian model had the least effective performance. It is mostly because to its presumption of feature independence. Compared to baselines, SVM improves efficiency by 19% over rule-based (static thresholds) and 9% over Isolation Forest. In Figures 10 and 11, the confusion matrix and ROC-AUC cruve are depicted, respectively.

Overall, our proposed work improves access control in smart contracts by providing an innovative SVM-based approach and dimensionality reduction. Our main innovation in automating digital signatures is detecting anomalies with 94% accuracy and reducing computational complexity, which simplifies the user experience and increases security.

4. 4. 3. Robustness Evaluation To check for robustness, we used the Fast Gradient Sign Method (FGSM) to simulate evasion attacks by changing the features of the test set (for example, adding or removing 10% noise from the connection duration) and adversarial examples. The SVM kept its accuracy at 92.5% with mild perturbation, but it dropped to 81% with severe perturbation (e.g., ±50% noise). This was better than Logistic Regression (83.5% and 82%, respectively). The baseline accuracy was 76.5%, while the accuracy of

TABLE 6. Comparative performance metrics of classifiers

Model	Accuracy	Precision	Recall	F1-Score	AUC-ROC
SVM (Linear, C=1)	0.94	0.92	0.89	0.90	0.96
Decision Tree (d=7)	0.88	0.85	0.83	0.84	0.87
C4.5 (d=7)	0.89	0.87	0.84	0.85	0.88
Naïve Bayesian	0.82	0.79	0.76	0.77	0.83
Random Forest	0.92	0.90	0.87	0.88	0.94
Gradient Boost	0.90	0.88	0.86	0.88	0.92
Logistic Regression	0.88	0.85	0.82	0.83	0.90
Rule-Based Baseline	0.75	0.70	0.65	0.67	0.78
Isolation Forest	0.85	0.82	0.80	0.81	0.88

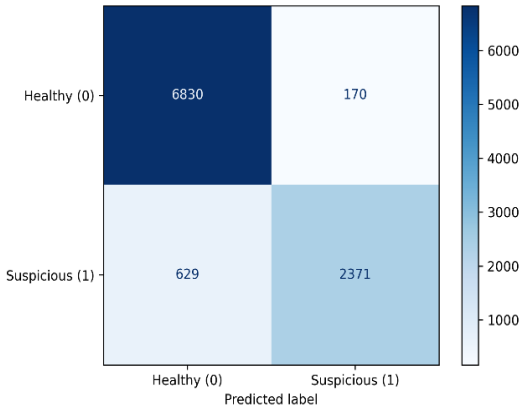


Figure 10. Confusion matrix of proposed SVM model

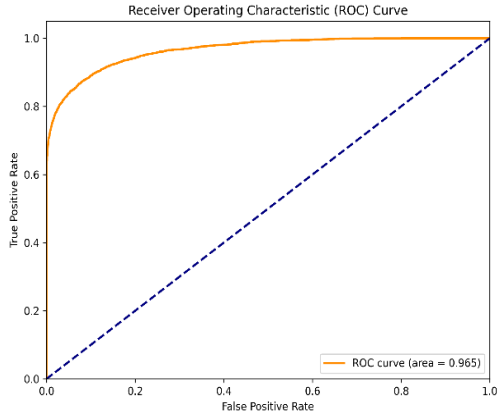


Figure 11. ROC-AUC curve of proposed model

SVM was 74.5% when FGSM attacks were used with ε=0.1. This shows that the system is strong, but it also shows that adversarial training is needed in future versions. The results of the evaluation is illustrated in Table 7.

TABLE 7. Comparative results of robustness evaluation

Model	Accuracy
SVM Base accuracy	94.00%
SVM Perturbed (10%)	92.50%
SVM Severe (50%)	81.00%
SVM Adversarial FGSM	74.50%
Baseline Base accuracy	84.50%
Baseline Perturbed (10%)	83.50%
Baseline Severe (50%)	82.00%
Baseline Adversarial FGSM	76.50%

5. DISCUSSION

The findings indicate that the SVM model, in conjunction with GA-based dimensionality reduction, significantly improves smart contract access control. The elevated accuracy (0.94) and AUC-ROC (0.96) demonstrate strong performance. Also, the precision (0.92) and recall (0.89) imply a well-balanced capacity enabling the detection of suspicious users with few false positives. The feature importance analysis validates the methodology's emphasis on behavioral indicators. These correspond with the established user classifications: Healthy Users and Suspicious Users/Hackers. For healthy users' characteristics, we can exemplify their low to medium referrals, medium transmitted data, and high received data. On the other hand, for suspicious users' characteristics, they exhibit elevated references, substantial transmitted data, and average received data. The scatter plots (Figures 5-9) illustrate the efficacy of the SVM, demonstrating distinct class separation and support vectors strategically located at the decision boundaries. The overlap shown in certain plots emphasizes the intricacy of the classification problem. It strongly emphasizes the need for individualized anomaly detection as specified in the methodology. In the comparative analysis section, we demonstrate the advantages of SVM relative to Decision Tree, C4.5, and Naïve Bayesian models. We can affirm its suitability for this application. The effectiveness of the linear kernel demonstrates that the diminished feature space is linearly separable. This conclusion is supported by the visual representations. Automated signing makes things faster, but it also raises important questions about usability and user trust. One major worry is what happens when a false positive happens, which is when a real transaction is wrongly flagged as malicious. In our proposed system, this kind of event would not stop the user. Instead, it would send the transaction back to a manual approval process, which would act as a security checkpoint. This is a small inconvenience, but it's a good trade-off because the risk of a false negative (approving a bad transaction) is much higher. To gain trust, a practical implementation could include an adjustable sensitivity threshold or a "learning period" during which the model's decisions are shown to the user for confirmation before full automation is enabled. This makes sure that users stay in control while also getting the extra layer of security that comes from machine learning. In terms of misclassification, the primary impact of a misclassification on a legitimate user is a temporary shift from a frictionless, automated experience to a traditional, manual one. The system is designed to fail safely; when in doubt, it requires explicit user consent, thereby prioritizing security over convenience in ambiguous situations.

6. LIMITATION

The proposed method has the following limitations

- Class Imbalance: The dataset contains 70% legitimate and 30% suspicious users, which may bias the classifier.
- Generalizability: The method was tested on Ethereum only; its applicability to other platforms needs exploration.
- Scalability: Rolling window analysis is computationally intensive and may require optimization for real-time use. There are also some limiations about optimizations all of which could result in much better model, namely, off-chain computation, parallel processing, incremental calculation, and hardware acceleration.

7. CONCLUSION

This research presents a machine learning-driven method for improving access control in blockchain smart contracts. By automating transaction signing and detecting anomalies, the system enhances both security and user experience. With a classification accuracy of 94%, the approach offers a reliable and efficient alternative to manual signature verification. Future work will focus on real-time deployment and extending the model to multi-blockchain ecosystems. Alos we will investigate sophisticated feature engineering to elucidate more intricate user behaviours. This could involve developing metrics such as the ratio of sent-to-received transactions, the Gini coefficient of transaction values to assess regularity, or time-delta features between consecutive transactions, potentially resulting in a more detailed classification.

In terms of optimizations, the off-chain computation is an off-chain server that keeps an eye on the blockchain could do the analysis. The smart contract would only get the final classification (benign/malicious), which would cut down on the amount of work that needs to be done on-chain. Another one is parallel processing. The analysis of different user accounts or time periods could be done at the same time on more than one processor core or server. Additionally, incremental calculation can be used. Instead of recalculating the whole window every time, you can use stream processing algorithms that can update feature calculations as new transactions come in. Lastly, hardware acceleration uses GPUs to speed up the steps of processing the ML model and extracting features. Regarding the importance of a personalized anomaly detection, the significance of a tailored anomaly detection model is most effectively demonstrated through the juxtaposition of user profiles. For instance, a bot that trades high-frequency might make hundreds of small trades every day. A global model that used average user data to train would probably see this normal bot activity as a very strange thing. On the other hand, a Decentralised Autonomous Organization (DAO) treasury might make one very large transaction every month to pay its contributors. This transaction would be very unusual for a normal user, but it is normal for the DAO's profile. In both cases, a one-size-fits-all model would fail, causing a lot of false alarms and making the system less useful. Our personalized method, which learns the unique "rhythm" of each address, is necessary to accurately tell the difference between normal and unusual behaviour in a variety of situations.

In terms of threat model, our method is meant to protect against a certain group of threats while making some assumptions. The main threat that this system protects against is transaction phishing, which is when a malicious dApp interface tricks a user into signing a transaction with wrong parameters, like an incorrect recipient address or a higher transfer amount. It also helps find automated transaction spam that comes from a hacked account and doesn't follow the user's normal behaviour. This ML system could cause two problems: 1) False Positives, where a real transaction is flagged as malicious, which makes things harder for the user; and 2) False Negatives, where a malicious transaction is missed and marked as benign. Our high precision (0.92) and recall (0.89) values show that we are doing a good job of reducing both types of errors.

Author Contribution Statement

Reza Amiri: Methodology, Conceptualization (Ideas), Investigation, Visualization, Analysis, Software (Programming and development), Writing – original draft, Writing – review & editing.
Jaber Karimpour: Supervision, Validation, Writing – review & editing.
Habib Izadkhah: Consultation, Validation, Writing – review & editing.

Conflicts of interest

The authors declare no conflicts of interest.

Supplementary Materials

The data that support the findings of this study are available on request.

8. REFERENCES

1. Nakonechnyi V, Toliupa S, Saiko V, Lutsenko V, Ghno GSN, Hussain AK, editors. Blockchain implementation in the protection system of banking system during online banking operations. 2024 35th Conference of Open Innovations Association (FRUCT); 2024: IEEE. 10.23919/FRUCT61870.2024.10516404

2. Khedmati M, Seifi F, Azizi M. Time series forecasting of bitcoin price based on autoregressive integrated moving average and machine learning approaches. International Journal of Engineering Transactions A: Basics. 2020;33(7):1293-303. 10.5829/ije.2020.33.07a.16
3. Alamsyah A, Muhammad IF. Unraveling the crypto market: A journey into decentralized finance transaction network. Digital Business. 2024;4(1):100074. 10.1016/j.digbus.2024.100074
4. Zheng Z, Xie S, Dai H-N, Chen X, Wang H. Blockchain challenges and opportunities: A survey. International journal of web and grid services. 2018;14(4):352-75. 10.1504/IJWGS.2018.095647
5. Dey A, Nandi S, Sarkar M, editors. Security measures in IoT based 5G networks. 2018 3rd International Conference on Inventive Computation Technologies (ICICT); 2018: IEEE. 10.1109/ICICT43934.2018.9034365
6. Anusha R, Saravanan R. Revolutionizing signature scheme: the enhanced Edward Elgamal extreme performance accumulate signature approach for IoT and blockchain applications. Soft Computing. 2025;29(3):1473-96. 10.1007/s00500-025-10426-0
7. Eskandari S, Clark J, Barrera D, Stobert E. A first look at the usability of bitcoin key management. arXiv preprint arXiv:180204351. 2018. 10.48550/arXiv.1802.04351
8. Hamidi H, Sayah A. Combining machine learning algorithms to detect phishing urls: a stacking approach. International Journal of Engineering Transactions B: Applications. 2025;38(8):1939-52. 10.5829/ije.2025.38.08b.18
9. Atzei N, Bartoletti M, Cimoli T, editors. A survey of attacks on ethereum smart contracts (sok). International conference on principles of security and trust; 2017: Springer. 10.1007/978-3-662-54455-6_8
10. Bamert T, Decker C, Elsen L, Wattenhofer R, Welten S, editors. Have a snack, pay with Bitcoins. IEEE P2P 2013 Proceedings; 2013: IEEE. 10.1109/P2P.2013.6688717
11. Kersic V, Vidovic U, Vrecko A, Domajnko M, Turkanovic M. Orchestrating digital wallets for on-and off-chain decentralized identity management. IEEE access. 2023;11:78135-51. 10.1109/ACCESS.2023.3299047
12. Ahmed MR, Islam AM, Shatabda S, Islam S. Blockchain-based identity management system and self-sovereign identity ecosystem: A comprehensive survey. Ieee Access. 2022;10:113436-81. 10.1109/ACCESS.2022.3215286
13. Babel M, Willburger L, Lautenschlager J, Völter F, Guggenberger T, Körner M-F, et al. Self-sovereign identity and digital wallets. Electronic Markets. 2025;35(1):1-14. https://doi.org/10.1007/s12525-025-00772-0
14. Albshaier L, Almarri S, Hafizur Rahman M. A review of blockchain's role in E-Commerce transactions: Open challenges, and future research directions. Computers. 2024;13(1):27. 10.3390/computers13010027
15. Krishnan LP, Vakilinia I, Reddivari S, Ahuja S. Scams and solutions in cryptocurrencies—A survey analyzing existing machine learning models. Information. 2023;14(3):171. 10.3390/electronics12061422
16. Chen T, Li X, Luo X, Zhang X, editors. Under-optimized smart contracts devour your money. 2017 IEEE 24th international conference on software analysis, evolution and reengineering (SANER); 2017: IEEE. 10.1109/SANER.2017.7884650
17. Ntousis O, Makris E, Tsanakas P, Pavlatos C. A dual-stage processing architecture for unmanned aerial vehicle object detection and tracking using lightweight onboard and ground server computations. Technologies. 2025;13(1):35. https://doi.org/10.3390/technologies13010035
18. Podgorelec B, Turkanović M, Karakatič S. A machine learning-based method for automated blockchain transaction signing including personalized anomaly detection. Sensors. 2019;20(1):147. 10.3390/s19245433
19. Ou W, Huang S, Zheng J, Zhang Q, Zeng G, Han W. An overview on cross-chain: Mechanism, platforms, challenges and advances. Computer Networks. 2022;218:109378. 10.1016/j.comnet.2022.109378
20. Tschorsch F, Scheuermann B. Bitcoin and beyond: A technical survey on decentralized digital currencies. IEEE Communications Surveys & Tutorials. 2016;18(3):2084-123. 10.1109/COMST.2016.2535718
21. Dominguez Anguiano T, Parte L. The state of art, opportunities and challenges of blockchain in the insurance industry: a systematic literature review. Management Review Quarterly. 2024;74(2):1097-118. https://doi.org/10.1007/s11301-023-00328-6
22. Decker C, Eidenbenz R, Wattenhofer R, editors. Exploring and improving BitTorrent topologies. IEEE P2P 2013 Proceedings; 2013: IEEE.
23. Chou C-C, Hwang N-CR, Schneider GP, Wang T, Li C-W, Wei W. Using smart contracts to establish decentralized accounting contracts: An example of revenue recognition. Journal of Information Systems. 2021;35(3):17-52. https://doi.org/10.2308/ISYS-18-028
24. Buterin V. A next-generation smart contract and decentralized application platform. white paper. 2014;3(37):2-1. https://ethereum.org/en/whitepaper/
25. Boneh D, Franklin M, editors. Identity-based encryption from the Weil pairing. Annual international cryptology conference; 2001: Springer.
26. Béres F, Seres IA, Benczúr AA, Quintyne-Collins M, editors. Blockchain is watching you: Profiling and deanonymizing ethereum users. 2021 IEEE international conference on decentralized applications and infrastructures (DAPPS); 2021: IEEE.
27. Lin S-Y, Zhang L, Li J, Ji L-l, Sun Y. A survey of application research based on blockchain smart contract. Wireless Networks. 2022;28(2):635-90. https://doi.org/10.1007/s11276-021-02874-x
28. Szabo N. Formalizing and securing relationships on public networks. First monday. 1997. https://doi.org/10.5210/fm.v2i9.548
29. Bodziony N, Jemioło P, Kluza K, Ogiela MR. Blockchain-based address alias system. Journal of Theoretical and Applied Electronic Commerce Research. 2021;16(5):1280-96. https://doi.org/10.3390/jtaer16050072
30. Akbarfam AJ, Barazandeh S, Gupta D, Maleki H. Deep learning meets blockchain for automated and secure access control. arXiv preprint arXiv:231106236. 2023. https://arxiv.org/abs/2311.06236
31. Mounnan O, Boubchir L, Manad O, El Mouatasim A, Daachi B. DBAC-DSR-BT: A secure and reliable deep speech recognition based-distributed biometric access control scheme over blockchain technology. Computer Standards & Interfaces. 2025;92:103929. https://doi.org/10.1016/j.csi.2024.103929
32. Jiang F, Chao K, Xiao J, Liu Q, Gu K, Wu J, et al. Enhancing smart-contract security through machine learning: A survey of approaches and techniques. Electronics. 2023;12(9):2046. https://doi.org/10.3390/electronics12092046
33. Guo Z. Blockchain-enhanced smart contracts for formal verification of IoT access control mechanisms. Alexandria Engineering Journal. 2025;118:315-24. https://doi.org/10.1016/j.aej.2022.03.064
34. Ghaly M, Elbeltagi E, Elsmadony A, Tantawy MA. Integration of Blockchain-Enabled smart contracts in construction: SWOT framework and social network analysis. Civil Engineering

Journal. 2024;10(05):1662-97. https://doi.org/10.28991/CEJ-2024-010-05-020

35. Gaur R, Prakash S, Kumar S, Abhishek K, Msahli M, Wahid A. A machine-learning–blockchain-based authentication using smart contracts for an IoHT system. Sensors. 2022;22(23):9074. https://doi.org/10.3390/s22239074
36. Benaich R, Gahi Y, El Mendili S. Pioneering the Security of EHRs Using an Immersive Blockchain Conceptual Framework. Emerging Science Journal. 2025;9(1):161-87. https://doi.org/10.28991/ESJ-2025-09-01-010
37. Alizadeh M, Koohi, S., Vahidi, V. A decentralized access control framework for the personal health records based on blockchain and smart contract. Journal of Information Security and Applications. 2023;72:103375. https://doi.org/10.1016/j.jisa.2022.103375
38. Liao C-H, Guan X-Q, Cheng J-H, Yuan S-M. Blockchain-based identity management and access control framework for open banking ecosystem. Future Generation Computer Systems. 2022;135:450-66. https://doi.org/10.1016/j.future.2022.05.015
39. Criollo S, Guerrero-Arias A, Arif YM, Samala AD, Jaramillo-Alcázar Á, Luján-Mora S. Usability Evaluation of a Mobile Augmented Reality App for PC Hardware Training: A Comparative Study in Three Countries. 2025. https://doi.org/10.28991/ESJ-2025-09-02-024
40. Aliyev V. Ethereum fraud detection dataset. Kaggle Retrieved March. 2021;13:2022. https://www.kaggle.com/datasets/vagifa/ethereum-frauddetection-dataset

Persian Abstract

چکیده

در حوزه فناوری بلاکچین، تضمین کنترل دسترسی ایمن و کارآمد برای قراردادهای هوشمند همچنان یک چالش اساسی است. روش‌های سنتی اغلب پیچیده و منابع‌بر هستند و به طور بالقوه مانع پذیرش گسترده می‌شوند. این مطالعه یک رویکرد جدید مبتنی بر یادگیری ماشین را برای بهبود مکانیسم‌های کنترل دسترسی پیشنهاد می‌دهد. به طور خاص، ما کاربران را بر اساس رفتار تراکنش و الگوهای تعامل به عنوان خوش‌خیم یا بالقوه مخرب طبقه‌بندی می‌کنیم. یک طبقه‌بندی‌کننده ماشین بردار پشتیبان (SVM)، همراه با تکنیک‌های کاهش ابعاد، بر روی مجموعه داده‌ای حاوی ۵۰،۰۰۰ رکورد تراکنش از ۱۰۰۰ آدرس بلاکچین اعمال می‌شود. این مدل در مجموعه آزمایشی به دقت ۹٤٪ دست یافت و کاربران را بر اساس فرکانس تعامل با سرور و مدت زمان اتصال به طور مؤثر متمایز کرد. از طریق تجزیه و تحلیل بصری و ارزیابی جامع، نشان می‌دهیم که روش پیشنهادی هم تشخیص ناهنجاری و هم کارایی عملیاتی را بهبود می‌بخشد. این رویکرد پتانسیل تقویت اعتماد و تسهیل پذیرش گسترده‌تر برنامه‌های مبتنی بر بلاکچین را دارد.