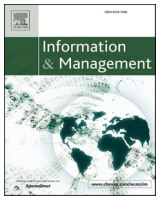
Information & Management 63 (2026) 104242

Contents lists available at ScienceDirect

Information & Management

journal homepage: www.elsevier.com/locate/im



Token fraud identification and implications for post-crowdfunding performance

Ziyi Xiong [a,*], Rong Liu [b], Hemang Subramanian [b]

[a] Department of Information Systems and Security, Coles College of Business, Kennesaw State University, Kennesaw, Georgia, 30144, USA
[b] Department of Information Systems and Business Analytics, College of Business, Florida International University, Miami, Florida, 33199, USA

ARTICLE INFO

Keywords:
Web3 application
Fraud detection
Web3 token fraud
Graph neural network
Application performance

ABSTRACT

Tokens issued by emerging Web3 applications serve multiple roles, including crowdfunding, payment, and governance, during the development of these applications. However, Web3 token fraud damages the trust of stakeholders, potentially contributing to the failure of these applications. We present an end-to-end mechanism for identifying wallet accounts suspected of Web3 token fraud and analyze the impact of such fraud on the performance of Web3 applications post-crowdfunding. First, we develop novel graph neural network models to identify fraudulent wallet accounts within evolving on-chain transaction networks using a crowd-reported fraud dataset. Next, we construct a dynamic ex ante fraud risk profile for each Web3 application by aggregating account-level fraud predictions. Finally, we evaluate the impact of risk profiles on Web3 application performance. Our results indicate a nuanced effect of Web3 token fraud. Web3 token fraud influences both application usage and user base expansion negatively. A prior surge in application usage may intensify the risk of token fraud, while earlier user base expansion can potentially alleviate this risk.

1. Introduction

As blockchain technology automates business processes through smart contracts, it has introduced a new type of digital infrastructure in which "meta trust" facilitates the direct exchange of value between users without the need for trusted intermediaries [1]. Web3 applications (a.k.a. DApps) are built with this infrastructure. Unlike traditional platforms (e.g., Amazon, Airbnb) that hold centralized authority, Web3 applications rely on their users to govern the application life cycle, participate in its development, and co-create value through crypto tokens, underpinning decentralization. As a result, users in Web3 ecosystems hold a more central position than in any other traditional ecosystem. Recent years have witnessed a surge of Web3 applications. In 2016–21, over 7200 Web3 applications have funded themselves by issuing crypto tokens (referred to as *Web3 tokens* hereafter) through initial coin offerings (ICOs). These ICOs account for a total investment of 35 billion dollars [2]. As of Q3 2024, over 17,000 Web3 applications have been deployed on various blockchains, with >17 million active users [3]. The Web3 industry has grown to a 3 trillion dollar economy with >500 global exchanges [4].

Despite the boom in the Web3 industry, it has been plagued by varieties of Web3 token fraud (or *token fraud* for simplicity) due to the lack of regulation, substantial volume, technical complexity, and security vulnerability of Web3 applications [5,6]. *Web3 token fraud* involves illicit transfers of Web3 tokens through various tactics, such as phishing, hacking, and exploitation, resulting in financial losses for victims [6]. According to the REKT Database [7], over 3800 significant token fraud events have been reported, involving hundreds of Web3 tokens and affecting more than $80 billion in funds. Moreover, token fraud is estimated to have caused $24.2 billion in financial losses, representing 0.34 % of total on-chain transactions [8]. The prevalence of these threats is further evidenced by the FBI's report of a year-over-year increase in token fraud in U.S. through 2022 [9]. Beyond substantial financial loss to users, Web3 token fraud raises serious concerns about transaction security, thus damaging trust in the emerging Web3 ecosystems and hindering their widespread adoption [10,11].

Extensive research has examined traditional fraud schemes such as phishing and hacking [12–16], and a growing body of work focuses on fraud in cryptocurrencies such as Bitcoin (BTC) or Ethereum (ETH) [17-20]. However, the existing methods are not well suited to the Web3 context for several reasons. First, traditional methods often focus on detecting deceptive messages, yet fraudsters of Web3 tokens may not

* Corresponding author.
E-mail addresses: zxiong@kennesaw.edu (Z. Xiong), rong.liu2@fiu.edu (R. Liu), hsubrama@fiu.edu (H. Subramanian).

https://doi.org/10.1016/j.im.2025.104242
Received 2 February 2024; Received in revised form 29 August 2025; Accepted 31 August 2025
Available online 1 September 2025

directly engage with victims. Second, BTC/ETH fraud detection methods utilize publicly accessible transactions recorded on blockchains (refer to as *on-chain transactions*, or simply *transactions*, unless otherwise specified). However, these methods typically assume homogeneous transaction networks, as BTC and ETH function purely as cryptocurrencies with stable transaction patterns [17,19]. In contrast, Web3 ecosystems consist of thousands of Web3 tokens with frequent entry and exit, creating rapidly evolving transaction networks. Third, unlike BTC or ETH, Web3 tokens often serve multifunctional roles, such as payment, product usage rights, and governance [21], enabled by complex smart contracts. This versatility and technical complexity expose them to a broad range of fraud risks, ranging from traditional threats such as phishing and hacking to smart contract exploits that allow unauthorized asset transfers without direct user interaction [22] and to application-layer attacks such as redirecting user funds by modifying an application's wallet addresses. These unique characteristics of Web3 token fraud necessitate customized fraud detection techniques.

Motivated by this uniqueness, we pose two related research questions in the Web3 context: (1) *how can Web3 token fraud be detected*? and (2) *how does Web3 token fraud affect the performance of Web3 applications*? To answer these questions, we propose a novel end-to-end mechanism that includes three steps: (1) identifying fraudulent wallet accounts (refers to as *accounts* hereafter) involved in Web3 token transactions using public on-chain data; (2) constructing a risk profile for each Web3 application; and (3) evaluating the impact of token fraud on application performance.

In the first step, we develop a graph neural network (GNN) model to identify fraudulent accounts within evolving transaction networks, which change as tokens and users join or leave the system. Our model incorporates observable ex ante signals as node and edge features, such as token retention, open-source development (OSD), team size, and expert ratings, which prior research has shown to be indicative of project credibility, governance quality, and eventual success or failure [23,24]. With over 159,845,654 transactions from 2017 to 2020 and a dataset of 761 crowd-reported fraudulent accounts, we employ a time-based nested cross validation procedure [25] to train the model on one year's data and test it on the subsequent year to ensure model generalizability. Our GNN model outperforms benchmarking models by 8–34 % in the F1 score of the fraud class and by 4–41 % in PRC (area under the precision-recall curve).

In the second step, we use the trained model to estimate the fraud probability for each account (i.e., account-level fraud risk). Although accounts are anonymous and lack personally identifiable information, we posit that transactional footprints offer valuable insights into irregularities and the potential risk to other parties. We then construct dynamic risk indicators for each Web3 token by aggregating the risks associated with accounts that have transacted with the token within a specific time period. These risk indictors collectively form a risk profile (i.e., application-level fraud risk) for the Web3 application issuing the token.

Last, in the third step, we assess the impact of application-level fraud risk on the performance of Web3 applications post crowdfunding. We focus on widely used dimensions of application performance in prior research—user base and application usage [26–28]. Our results indicate that this risk affects both application usage and user base expansion negatively. Interestingly, we observe that a previous surge in transactions may intensify the token fraud risk, while a prior increase in user base can potentially alleviate this risk.

This paper makes several contributions to the literature and practices of the emerging Web3 economy. First, our study extends the information system (IS) literature on cybersecurity [12,15,17,19] by contextualizing fraud detection within Web3 applications. To our knowledge, we are among the first to study both fraud detection and its ex ante effect on key application network metrics such as user base and application usage. We design a generalizable approach customized to detecting Web3 token fraud based on micro-level blockchain transactions in large, dynamic on-chain networks. Second, while prior research focused on individual fraudulent behavior [12,29,30], the current paper aggregates account-level prediction to a large web3 application context. Our novel fraud risk measure allows us to construct dynamic ex ante risk profiles for Web3 applications, supplementing existing susceptibility analyses of information systems based on organizational or application factors [31]. While prior research emphasizes financial losses caused by fraud, our study highlights the broader implications of Web3 token fraud for the sustainable performance of Web3 applications. Our novel mechanism adds new empirical evidence of the long-lasting negative impact of token fraud on Web3 applications performance, with effects persisting for up to 15 weeks. Our work also complements prior research on factors influencing the post-ICO success of blockchain ventures [2,32,33] by revealing how fraud and application-level risk profiles can threaten the long-term viability of Web3 applications.

Finally, our work provides practical insights for analysts, developers, investors, and ventures in Web3 ecosystems. For analysts and developers, we offer an effective tool for identifying suspicious accounts and scrutinizing trading parties based on public on-chain transactions. Similarly, our proposed application-level fraud risk indictors can aid investors in selecting Web3 tokens to minimize risk in their token portfolios. For ventures, our fraud detection model can be used to monitor Web3 token transactions continuously, allowing proactive mitigation of security vulnerabilities. At the policy level, our study resonates with the increasing initiatives to curb crypto fraud at state and national levels, as well as among the G-20 nations.[1]

2. Background and related work

In this section, we first describe the concepts of Web3 tokens and compare them with Bitcoin (BTC) and Ethereum (ETH). Then we introduce the details of Web3 token fraud and discuss its differences from BTC/ETH fraud and traditional fraud to highlight the unique design requirements for Web3 token fraud detection. We also review studies relevant to these topics.

2.1. Web3 tokens and comparison with Bitcoin/Ethereum (BTC/ETH)

Web3 refers to a broad range of emerging decentralized blockchain-based applications. Unlike centralized Web2 platforms, such as Amazon, Web3 applications allow users to control their data and monetize the content they create [34]. To raise capital, Web3 applications typically sell crypto tokens directly to a large crowd of people. In return, token holders are entitled to either ownership of the applications or rights to consume products or services in the future. Thus, each token represents a unit of value tied to the associated Web3 application and may be traded for other assets on cryptocurrency exchanges. These tokens are governed by smart contracts, which are self-executing programs designed to create and manage Web3 tokens. More specifically, smart contracts maintain the mapping between token accounts and the tokens they hold.

An illustrative example of a Web3 application is *Decentraland*,[2] a blockchain-based metaverse platform. Decentraland allows users to purchase, trade, and develop virtual real estate using its native token, MANA. Beyond virtual real estate, Decentraland offers functionalities such as trading non-fungible tokens (NFTs), hosting interactive social events, and enabling community-driven governance. MANA token holders can vote on proposals and decisions affecting the platform,

[1] The G20 leaders meeting in New Delhi, India, called for increased cooperation among nations to combat fraud and to build sustainable public digital infrastructure amid the rise of public and private cryptocurrencies. Refer to page 22 here: https://www.mea.gov.in/Images/CPV/G20-New-Delhi-Leaders-Declaration.pdf.

[2] https://decentraland.org.

exemplifying the decentralized principles of Web3 applications.

The performance of a Web3 application can be assessed by its ICO success during crowdfunding and subsequent token transactions post-ICO. Prior literature has examined features that signal application quality and attribute to successful ICOs [32,35]. As summarized in Appendix A, Table A-1, these features include team capability [33], project scope [24], open-source development [23], founders' commitment or "skin in the game" [2], and expert evaluation [33,36]. Following successful ICOs, about 20 % of the total tokens can be listed in decentralized crypto exchanges (DEXs) or centralized crypto exchanges (CEXs) for wider circulation [2]. CEXs, such as Binance and Krake, rely on private central databases to execute off-chain transactions without accessing blockchain to reduce latency and transaction cost (i.e., gas fee), whereas DEXs, such as Uniswap, use automated market makers protocols to determine token prices and execute on-chain transactions on the blockchain [4,37]. Several studies have analyzed post-ICO performance in terms of on-chain wallets and wallet transfers, off-chain trading volume, and the buy-and-hold returns of tokens [2,38–40]. Other studies have looked at the impact of cybercrimes on the behavior of token investors [6]. Yet few studies have investigated the influence of Web3 token fraud on the growth and sustainability of Web3 applications after ICO.

Last, it is important to note that Web3 tokens differ fundamentally from Bitcoin and Ethereum in both functionality and technical complexity. Bitcoin and Ethereum serve primarily as digital currencies, and their transactions are confined to their respective blockchains. In comparison, Web3 tokens, as illustrated by Decentraland, can have a wide range of use, such as acting as local currencies, granting rights to future cash flows, or providing access to products or services [21]. Accordingly, the underlying smart contracts governing these tokens are typically more complex, making them prone to security vulnerabilities. Moreover, Web3 tokens often interact with services across multiple applications, exposing them to sophisticated fraud schemes that exploit application interdependencies. In 2022, the Nomad Bridge exploit targeted a messaging protocol used by several DApps. By manipulating the smart contract logic, attackers drained $190 million worth of tokens from wallets interacting with DApps reliant on the Nomad infrastructure [41].

2.2. Web3 token fraud and comparison with other types of fraud

With the growth of the cryptocurrency market, a wide variety of fraud schemes have emerged, including phishing [17-19], hacks, and exploits [42-44], resulting in massive financial losses. For instance, in Q1 2024, the blockchain sector suffered a loss of $407 million due to exploits and hacks [45]. Hornuf et al. [46] classified Web3 token fraud schemes into two categories: internal and external fraud. Internal fraud is initiated by token issuers, such as exit fraud, where issuers disappear with raised funds instead of building legitimate businesses. External fraud, originating outside token issuers, includes schemes such as pump-and-dump, phishing, and hacking. Our study targets detecting external Web3 token fraud based on-chain token transactions, as internal fraud often lacks associated on-chain activities. Moreover, we exclude pump-and-dump, as it involves a small portion of tokens traded on crypto exchanges[3] and requires additional data, such as rush orders and token prices, from the exchanges for detection [42,47].

Fig. 1 shows the typical stages of Web3 token fraud covered by our study. In the *Bait and Hook* stage, fraudsters use deceptive techniques to lure victims into revealing their private data (e.g., private keys) or to entice them to approve asset transfer transactions. Previous studies find that Web3 applications with highly successful ICOs or open-source development (OSD) are particularly susceptible to such fraud schemes [42,46,48]. In the *Transfer Asset* stages, victims are tricked into invoking

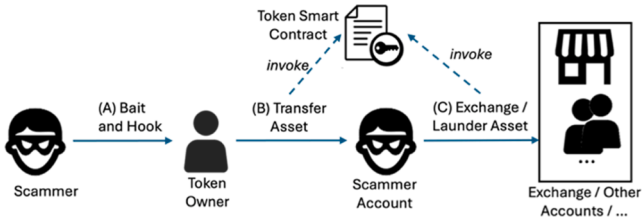


Fig. 1. Stages of token fraud.

smart contracts to transfer their tokens to fraudsters' accounts or approving the transfer transactions initiated by fraudsters. In some cases, fraudsters exploit vulnerabilities in smart contracts or Web3 applications to steal the assets directly without interacting with victims. For example, CoinDash fell victim to such an attack in 2017 [49]. A malicious attacker changed the official wallet address on CoinDash's website, diverting funds to the attacker's wallet. This incident highlights how the technical quality of a Web3 application can significantly influence its susceptibility to fraud. In the final stage, *Exchange/Launder Asset*, to conceal the illicit nature of stolen assets, fraudsters invoke token smart contracts to move the assets to various destinations. For instance, fraudsters can convert the stolen tokens into cash or Bitcoin via exchanges or deposit them into privacy-preserving blockchains such as Secret Network to conceal their origin [22].

Web3 token fraud differs from both traditional fraud and BTC/ETH fraud in how they progress through these stages, as summarized in Table 1. During the *Bait and Hook* stage, traditional fraud typically uses electronic channels such as websites, SMS, e-mails, and other social engineering techniques to attract victims [15,16]. In contrast, BTC/ETH or Web3 token fraud employ more sophisticated methods such as exploiting wallet vulnerabilities or tricking users into signing transactions [50]. While BTC/ETH fraud often involves interaction between the fraudsters and the victims, Web3 token fraud expands the threat landscape by introducing new risks, such as smart contract exploits and application-layer attacks, alongside traditional threats such ase phishing and hacking. Detecting fraud in this context requires methods that capture a wide range of threats across the application ecosystem, from technical vulnerabilities (e.g., code quality) to organizational factors (e.g., governance models). Project-specific factors, such as application quality and OSD, offer measurable indicators of a project's exposure to these risks and contribute to a holistic assessment of vulnerability [51]. Thus, integrating these observable, ex ante signals can significantly enhance the effectiveness of real-time fraud detection.

Table A-1 in Appendix A summarizes well-studied indicators for Web3 application quality [2,23,33]. High-quality applications are well managed and adopt higher development standards, making them less susceptible to token fraud. For instance, token fraud, such as hacking and exploits, may occur when fraudsters discover and take advantage of vulnerabilities or bugs in the code of Web3 applications [6]. High-quality applications are more likely to intervene promptly and halt the spread of token fraud once identified.

In the *Transfer Asset* phase, for traditional fraud, asset transfers remain hidden and inaccessible. In contrast, transfers of BTC/ETH and Web3 tokens are recorded on blockchains, and thus are transparent and traceable.[4] Blockchain's inherent features, such as tamperproof data structures and consensus protocols, provide robust transaction security by ensuring data integrity and preventing unauthorized modifications

[3] Not every Web3 token can be listed on a crypto exchange [2]. In our dataset, only 23% (562 out of 2427) of Web3 tokens have been traded on crypto exchanges.

[4] Strictly speaking, these transactions are executed on DEXs or peer-to-peer platforms. Transactions from CEXs, on other hand, are stored in private databases and are not publicly accessible. Since CEXs take custody of user wallets, they are less vulnerable to fraud types such as phishing and smart contract exploitation, but they are more prone to other forms of fraud such as insider trading.

Table 1
Comparison of traditional, BTC/ETH, and Web3 token fraud.

Fraud Stage	Traditional Fraud	BTC/ETH Fraud	Web3 Token Fraud	Requirements for Web3 Token Fraud Detection
(A) Bait and Hook	Use electronic channels (e.g., websites) to attract victims	Steal users' private keys or trick users to sign transactions	Fraudsters use a wide range of tactics, from traditional threats like phishing and hacking to smart contract exploits and application-layer breaches, even without direct interaction with victims	Web3 token characteristics, e.g., code quality, governance, and OSD, affect vulnerability
(B) Transfer Asset	Transfer records are private	Transfer BTC/ETH to fraudulent wallets with transactions saved on BTC/ETH blockchains	Transfer tokens to fraudulent accounts; transactions often span different applications in the same blockchain or multiple blockchains where the tokens operate[1]	Transactions form heterogenous networks, which continuously evolve with the entry and exit of Web3 tokens
(C) Exchange / Launder Asset	Private records	BTC/ETH are fungible and portable and can be easily cashed out	Tokens are exchanged for other assets directly or via exchanges, or laundered through mixers, cross-chain bridges, or other means [22]	Besides incoming transactions, outgoing transactions should be considered as they indicate patterns for asset laundering

[1] Cross-chain technology remains in its early stages of development [52], as evidenced by our data, which show that an overwhelming majority of Web3 tokens (86%) operate on the Ethereum blockchain only. Due to the limited availability of cross-chain web tokens, our method does not explicitly support multiple blockchain interactions.

(see Appendix B for more details). Web3 token fraud, however, introduces additional complexity as tokens can span multiple blockchains, such as Ethereum, Solana, and Polygon. Moreover, since many tokens coexist in a single blockchain, tracing the transaction flows of a specific Web3 token requires the use of heterogeneous graphs, where token attributes are attached to nodes and edges. In contrast, BTC/ETH fraud detection is limited to Bitcoin or Ethereum transactions only and homogeneous graphs can suffice [17,19,53]. Finally, the continuous entry of new Web3 tokens and the exit of existing ones cause their transaction networks to evolve far more dramatically than BTC/ETH networks. For instance, in our dataset, 1283 Web3 tokens were introduced in 2018 alone.

In the *Exchange / Launder Asset* stage, for traditional fraud, asset laundering is private and difficult to trace, while Bitcoin or Ethereum assets are fungible and portal, making them easy to cash out. In contrast, Web3 tokens, which are often traded on a limited number of exchanges, face liquidity constraints. Fraudsters usually employ sophisticated laundering techniques, such as routing funds through privacy-preserving blockchains, decentralized exchanges, or cross-chain bridges. Therefore, detecting fraudulent accounts requires monitoring both incoming and outgoing flows, as outgoing flows often reveal laundering activities while existing fraud detection primarily focuses on incoming transactions to identify fraudulent accounts [20,54].

2.3. Methods for fraud detection

Due to the distinct characteristics of Web3 token fraud (see the last column of Table 1), existing methods designed for detecting traditional and BTC/ETH fraud may not be effective in identifying Web3 token fraud. Table 2 summarizes the relevant literature on fraud detection.

Prior literature on traditional fraud has predominantly concentrated on the *Bait and Hook* stage, addressing critical topics such as susceptibility and methods for deception detection [12,13,15,16]. For example, studies have examined how users respond to deceptive information, such as phishing attempts, and framed fraud detection through the identification of deceptive communications. Detection strategies typically rely on content, portal, and propagation process analyses [17], using labeled data such as phishing websites and emails. However, these approaches are not directly applicable to blockchain contexts, which often lack labeled Bait and Hook content.

Existing studies on BTC/ETH fraud detection typically rely on transaction networks in a transductive setting and sophisticated feature engineering. These studies often create a network using historical transactions, generate a feature vector for each account in the fixed network through graph neural network embeddings, and train models to identify fraudulent accounts using the feature vectors [17,19,53]. Since BTC/ETH transactions are stored on their respective blockchains, homogeneous networks are sufficient for their analysis. For instance, graph convolutional network (GCN) and autoencoder have been used to derive node representations based on the statistics of edge information [17]. Other approaches automate feature extraction through random walks using skip-gram models [19,53]. In addition, some studies emphasize manual feature engineering. For instance, W. Chen et al. [17] designed a cascade feature extraction method based on transaction networks. A recent paper proposes a hybrid model that integrates manual feature engineering and time-series graph analysis [11]. However, since these approaches assume static networks and sufficient transaction history for each node, the trained models may struggle to generalize as networks evolve, as with the introduction of new Web3 tokens [61].

2.4. Summary of design requirements and research scope

As summarized in the last column of Table 1, Web3 token fraud detection presents distinct challenges compared with traditional fraud and BTC/ETH-related fraud, requiring a tailored approach. First, Web3 applications are vulnerable to a range of emerging fraud risks, including smart contract exploits and application-layer attacks, in addition to traditional threats such as phishing and hacking. Second, Web3 token transactions form massive heterogenous networks comprising millions of nodes and edges associated with thousands of Web3 tokens. The networks evolve continuously as new tokens enter and others exit. This dynamic nature precludes the transductive approaches used for BTC/ETH fraud detection and calls for inductive learning. Inductive methods generate embeddings based on a sampled local neighborhood without relying on a fixed global network structure to ensure generalizability [61]. Third, besides incoming transactions, an account's outgoing transactions should be tracked as they reveal patterns of asset laundering. However, most existing graph neural network (GNN) approaches focus exclusively on incoming edges when generating node features, as these align with the direction of information flow [61,62].

To address these challenges, we propose a custom inductive GNN model that evaluates a token's exposure to external fraud risk by integrating token transaction patterns with observable, ex ante indicators of application quality. These indicators are particularly valuable in the rapidly growing Web3 landscape where direct auditing of large-scale smart contracts is often impractical. We model these indicators as node and edge attributes within heterogeneous transaction networks and customize the GNN model to capture both incoming and outgoing transactions for effective fraud detection. To ensure model generalizability, we adopt a time-based nested cross-validation procedure [25] and test the trained models exclusively on newly added accounts after training. Moreover, we assess the impact of the estimated token fraud risk on the post-ICO performance of Web3 applications.

Table 2
Summary of related fraud detection approaches.

Studies	Methods	Datasets	Included in Benchmark?
Traditional fraud			
Characteristics of victims[12,55,15,56]	Factor analysis to determine victims' characteristics, such as vulnerability, coping response, susceptibility and so on.	Field study data, Survey data	No[a]
Characteristics of fraud and fraudsters[14,57,58]	Factor analysis and coding to extract features of the fraud websites, emails, etc.	Labeled websites	No[a]
		Cybercrime documents	No[a]
		Survey data	No[a]
BTC/ETH Fraud Detection			
Transaction analysis[17,59,53,29]	Supervised learning approaches based on-chain transactions (LightGBM, SVM, Random Forest)	Bitcoin/Ethereum transaction with labeled phishing accounts	Yes
	Unsupervised Trimmed K-Means approach (Trimmed K-Means)	Bitcoin transactions	Yes
Manual feature engineering[50,60,19]	A cascade feature extraction from transaction graphs (DElightGBM)	Ethereum transactions with labeled phishing accounts	Yes
	Network embedding method to extract latent features (Node2Vec)	Bitcoin/Ethereum transactions	Yes
A hybrid model[11]	Integrating manual feature engineering with LSTM model for account transaction time-series	Ethereum transactions with labeled phishing accounts	No[b]

[a] These models are not included into our benchmarking models as our context lacks corresponding datasets.
[b] LSTM is trained on transaction history and cannot be applied to new accounts without sufficient transaction data.

Table 3
Design objectives and justifications.

	Design objective	Justification
1	To be able to identify fraudulent crypto accounts	This objective enables us to estimate the likelihood of a crypto account being fraudulent and to analyze the impact of such likelihoods on the growth and sustainability of Web3 applications.
2	To evaluate the effectiveness of our model in identifying fraudulent token accounts	This evaluates the classification accuracy of our novel model compared to existing models.
3	To utilize the information obtained in (1) above to estimate token fraud risk and create risk profile for each Web3 application	This objective addresses the gap in existing literature and enables us to study the impact of token fraud on application performance post crowdfunding. This feature enables us to create an overall risk profile for each Web3 application by aggregating the account-level fraud likelihoods (see (1)-(2)).
4	To assess the impact of the risk profile of each Web3 application on its post-crowdfunding performance	Overall evaluation of the risk profile and its effects on (1) user base, (2) application adoption, and (3) token value after crowdfunding.

3. Design of the token fraud risk analysis framework

We follow the design science research method [63] to address these research questions. DSRM essentially has four steps: (1) setting the design objectives with justification, (2) designing the IT artifact, (3) implementation of the IT artifact, and (4) evaluating the artifact against an existing baseline. In Table 3, we highlight the design objectives of our solution and justify each design objective.

3.1. Design of our framework

We use a modular design to achieve the specific design objectives outlined in Table 3. Fig. 2 describes the four modules used in our design, as explained below:

(i) Data collection and feature extraction module: This module consists of three web-crawlers that crawl specific websites to extract information for analysis downstream.
(ii) Fraudulent account identification module: We use the collected data to train our novel inductive GNN model and benchmark with existing methods (see Table 2). With the trained model, we predict the fraud probability of each account.
(iii) Token risk estimation module: This module aggregates the account-level fraud risks and generates a risk profile for each Web3 application.
(iv) Evaluation module: We measure the impact of token fraud risk on the performance of Web3 applications post crowdfunding in this module.

Next, we discuss each component in detail.

3.2. Data collection and feature extraction module

Our dataset consists primarily of application information from ICObench.com and transaction records from etherscan.io. Table A-2 in Appendix A summarizes our data processing steps, along with records obtained after each step. We compile a distinctive dataset of Web3 applications funded through ICOs, sourced from various outlets, where each application is linked to a corresponding Web3 token. We wrote a robot to crawl data for Web3 applications listed on ICObench.com.[5] Each Web3 application has a separate web page with comprehensive information such as token ticker symbol, description, campaign information, financial data, founder retention rate (the percentage of tokens retained by the founders), expert evaluation, size of the team, access to the GitHub link for the application, and application scope.[6] Table 4 describes the raw features, the corresponding data sources, and their definitions.

[5] Although ICObench.com was no longer available, the application data have been achieved by many studies [2,86]. For instance, they can be downloaded from the supplementary material of Lyandres et al [2]. Our dataset can also be shared upon request.
[6] By "application scope," we are referring to the range of industries that a project encompasses. Our proposed model incorporates Web3 application features that indicate application quality. Since a Web3 application with a wider scope might attract a broader investor base and consequently secure more funds for future technical advancements, we consider the project scope a significant attribute of the application.

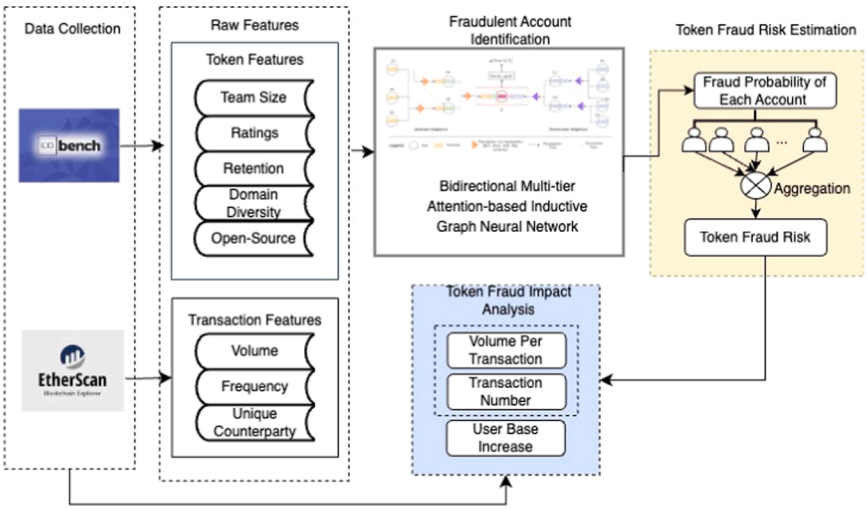


Fig. 2. Architecture of our end-to-end token fraud risk analysis framework.

Table 4
Data sources and raw features.

Features	Definition	Source
Web3 Application Information		
Token ticker	An abbreviation used to uniquely represent a token/application	ICObench
Project Scope	The number of industries that an application is involved with	
Retention rate	The percentage of the tokens that are reserved by application founders, i.e., "skin in the game."	
Experts rating (overall rating, team/ vision/ product rating)	Aggregated rating from evaluations of ICObench analysis and the ratings from external experts regarding a Web3 application's team, vision, and product	
Team size	The number of members in the application team	
Github	Whether the application open sources their development	
Token Transactions		
Token ticker	An abbreviation used to uniquely represent the transacted token	etherscan
Block index	The block index of each transaction	
Transaction volume	The number of tokens transferred in a transaction	
Wallet addresses	The wallet addresses of token senders and receivers in each transaction	
Reported fraudulent accounts	The wallet addresses of fraudulent accounts listed in etherscan.io	

The next dataset we collect is Web3 token transaction records[7] from etherscan.io, as 86 % of Web3 tokens in our sample operate on Ethereum. We used the etherscan API and a Selenium web-scraper to obtain all time-stamped transactions pertaining to each Web3 token. From each transaction record, we collected the sender and receiver wallet addresses. Also, etherscan.io maintains a roster of accounts labeled as fraudulent,[8] serving as the primary data source utilized by previous fraud detection studies. To detect the fraudulent accounts, we combined transaction data with application information to train a classification model. In line with previous studies, such as Abbasi et al. [64], H. Wen et al. [18], and Q. Yuan et al. [19], we use the list of fraudulent accounts on etherscan.io as the ground truth for our classifier training.[9]

[7] In addition to the "Token Transfer" transaction type, the other two types of transactions are normal transactions (ETH transfers between externally owned addresses) and internal transactions (ETH transfers through a smart contract as an intermediary), both of which are associated with Ethers (the native cryptocurrency of the Ethereum blockchain) and therefore beyond the scope of our research.

[8] The list of addresses related to phishing and hacks can be found at https://etherscan.io/tokens/label/phish-hack. The details of each fraudulent address can be validated through https://ethscamcheck.io.

[9] Fraud detection relies heavily on available data, since fraudulent activities are inherently hidden and often difficult to identify [17-19,30]. The use of crowd-reported incidents provides a practical solution to establish baseline patterns for fraudulent behavior. Similar approaches have been used in financial fraud detection and cybersecurity studies where known incidents act as reference points for model training [87,88].

3.3. Fraudulent account identification module

To enhance transaction security, it is critical to identify fraudulent accounts and alert associated parties. Our first step is to estimate the fraud probability for each account, following a supervised learning approach. After a detailed literature review (see Section 2.3), in Table 5, we summarize the complexity of this fraud detection task in Web3 applications context and propose design requirements along with design components corresponding to each task complexity.

Lack of user profiles. Due to user anonymity, we can only profile an account based on its transaction patterns and its neighbors and then use this profile to determine whether it is a fraudulent account. Since transactions connect accounts, it becomes intuitive to look at the data from a network perspective by regarding accounts as nodes and transactions as weighted directed edges [17]. As shown in Fig. 3a, an edge e (N1, N2) is established between nodes N1 and N2 when N1 (i.e., seller) transfers a token of type T to N2 (i.e., buyer). The transfer details (e.g., volume and time) are modeled as edge attributes. N1 and N2 can have multiple edges, each corresponding to a unique type of tokens transferred.

Web3 application quality as key indicator. As discussed in Section 2.2, high-quality Web3 applications are less susceptible to fraud such as hacking and exploitation, given their stronger technical and operational foundations, along with increased awareness and public scrutiny. Hence, as shown in Fig. 3a, we incorporate characteristics that signal Web3 application quality (see Appendix A, Table A-1) as additional edge attributes.

Integrating information from multi-tier neighbors. As shown in

Table 5
Task complexity and design requirements.

Task Complexity	Design Requirements	Design Components
Lack of user profiles	Extracting users' digital footprints from Web3 token transactions	Graph neural networks created from Web3 token transactions
Requiring information on the technical resilience and quality of Web3 applications	Incorporating features that signal the quality of Web3 applications into the fraud detection model	Including quality features of Web3 applications as attributes of nodes and edges
Considering both incoming and outgoing transactions to uncover patterns of fraud schemes and asset laundering	Automatically extracting features from multi-tier neighbors and enabling feature interactions	Multi-level bidirectional information propagation through an attention mechanism
Massive dynamic transaction networks	An incremental and generalizable approach that does not assume a fixed network structure	Inductive GNN learning and time-based nested cross validation procedure

Fig. 3a, we can observe tiers of neighbors for the focal node N1 based on the shortest paths between N1 and its neighbors. N1 may be influenced by its first-tier, second-tier, or even higher-order neighbors. Previous studies have found that neighbors' transaction information is useful in fraud detection and manually engineered features from transaction statistics [50]. For example, Chen et al. [17] proposed a cascading method to derive features from neighbors. For each node, this method calculates the sum, minimum, maximum, mean, and standard deviation of its first-tier incoming/outgoing neighbors' transaction counts as its first-tier neighbor features. Then similar aggregation can be applied recursively to derive second-tier neighbor features and beyond. However, this method leads to 10 first-tier neighbor features, 100 ($2 \times 5 \times 10$) second-tier neighbor features, and an exponentially increasing number of higher-order neighbor features. Furthermore, such simple aggregation does not consider the interactions between features. For example, fraudsters often make a large number of transactions within a short period, as evidenced by the transactions occurring within a single block or consecutive blocks. Thus, the interaction between transaction volume and transaction duration may be a good indicator of fraud. To address these challenges, we implement multi-level propagation to recursively generate features and allow feature interactions across multiple tiers (described shortly). This allows us to learn features more efficiently and better capture the complex relationships between features.

In addition, a node's spatial position within the network can provide valuable insights for identifying fraudulent accounts. Fig. 3b shows a subnet surrounding malicious nodes. It appears that malicious nodes either hold central positions within the subnet or transact with other central nodes, while most benign nodes are less central or even terminal nodes. One possible explanation for this is that fraudsters may use multiple accounts to launder stolen tokens. To capture this spatial information, we utilize a bidirectional model to generate features for an account based on its both upstream and downstream neighbors. Moreover, as neighbors vary in their importance, we use an attention mechanism to dynamically allocate attention among them. As illustrated in Fig. 3b, a typical malicious node has multiple neighbors, but only a few of them are fraudulent. The attention mechanism learns to assign higher weights to features from malicious neighbors in order to emphasize the common features of fraudulent nodes.

Massive dynamic transaction networks. Token fraud detection is an extremely unbalanced classification problem, with just a few hundred positive samples but millions of negative samples. The transaction network also evolves rapidly due to frequent entry and exit of Web3 tokens. This massive and dynamic network precludes transudative graph neural network models which require to observe the entire network. We adopt an inductive model that learns node features incrementally from randomly sampled local neighborhoods, without assuming a fixed network structure. For example, in Fig. 3a, features for node N1 can be learned from small subnets randomly sampled from multiple tiers of neighbors. Moreover, to ensure generalizability, following the time-based nest cross validation procedure, we train models on data from year t and then test the trained models on data from year $t+1$.

To fulfil these design requirements, we design a bidirectional multi-tier attention-based inductive graph neural network, shown in Fig. 4. For each node N1, our method transforms, propagates, and aggregates information from multiple tiers of neighbors along both source-to-target and target-to-source routes. For instance, as a source node, node A propagates its features to a target node B2. Then the features of node A and B2 are fused and continue to flow to another target node X. Similarly, for downstream neighbors to which X is connected, we propagate the information of the target nodes to X in a direction opposite to the transaction or token flows. For instance, for the edge of e (C1, D1), features of the target node D1 are propagated to the source node C1 and then to X. As a result, X congregates the information from multiple tiers of source and target neighbors. During the propagation, neighbors' information is transformed and aggregated by different aggregation functions. Following Chen et al. [17], we derive node features from the statistics of a node's own transactions (i.e., edges) and its neighbors' transactions by multi-layers of non-linear transformation. We also employ a multi-head attention mechanism [65] to allocate attention to different neighbors.

Formally, let $NS(i)$ denote the source nodes connected to node i, $NT(i)$ be the target nodes to which i is connected, and $e_{i,j}$ be the attribute of the edge between nodes i and j. Let $h_i^0 \in \mathbb{R}^H$ be the H-dimensional features derived from the original features x_i of node i (Eq. (1)). Node feature h_i^k is computed recursively through k propagation steps ($k \geq 1$) using Eqs. (2)–(7):

$$h_i^0 = W_0\, x_i \tag{1}$$

$$gs_i^k = M_s h_i^{k-1} + U_s\left[Agg_{j\in NS(i)}\left(W_s\left[h_j^{k-1}, e_{j,i}\right]\right)\right], \text{ where } Agg \in [sum, mean, min, \max] \tag{2}$$

$$gt_i^k = M_t h_i^{k-1} + U_t\left[Agg_{j\in NT(i)}\left(W_t\left[h_j^{k-1}, e_{i,j}\right]\right)\right], \text{ where } Agg \in [sum, mean, min, \max] \tag{3}$$

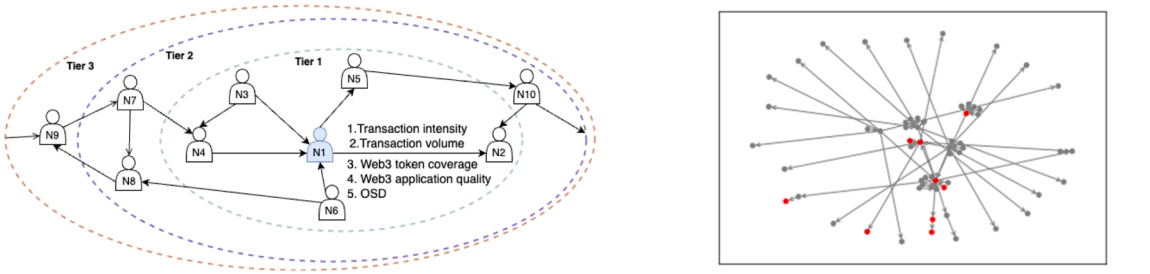
$$fs_i^k = L_s h_i^{k-1} + \sum_{j\in NS(i)}\left(att_{i,j}\, V_s\left[h_j^{k-1},\ e_{j,i}\right]\right),\ att_{i,j} = \frac{e^{s_{i,j}}}{\sum_{k\in NS(i)} e^{s_{k,j}}},\ s_{i,j} = \left(Q_s h_i^{k-1}\right)^T\left(O_s\left[h_j^{k-1}, e_{j,i}\right]\right)\Big/\sqrt{H} \tag{4}$$

$$ft_i^k = L_T h_i^{k-1} + \sum\nolimits_{j\in NT(i)}\left(att_j V_t\left[h_j^{k-1}, e_{i,j}\right]\right),\ att_{i,j} = \frac{e^{s_{i,j}}}{\sum_{k\in NT(i)} e^{s_{k,j}}},\ s_{i,j} = \left(Q_t h_i^{k-1}\right)^T\left(O_t\left[h_j^{k-1}, e_{i,j}\right]\right)\Big/\sqrt{H} \tag{5}$$

$$h_i^k = Z\left[gs_i^k,\ gt_i^k, fs_i^k, ft_i^k\right] \tag{6}$$

$$p(scam|i) = P\left[\ h_i^0,\ \ h_i^1,\ \ldots,\ \ h_i^k\right] + b. \tag{7}$$

For each source node j in $NS(i)$, we first transform the concatenated h_j^{k-1} and edge feature $e_{j,i}$ into an H-dimensional vector. Then we aggregate the vectors across all nodes in $NS(i)$ using four summary operators, concatenate the aggregated results, and transform them again to



(a) Illustrative Token Transaction Network (b) Subnet with Fraudulent Accounts (in red)

Fig. 3. Token transaction network.

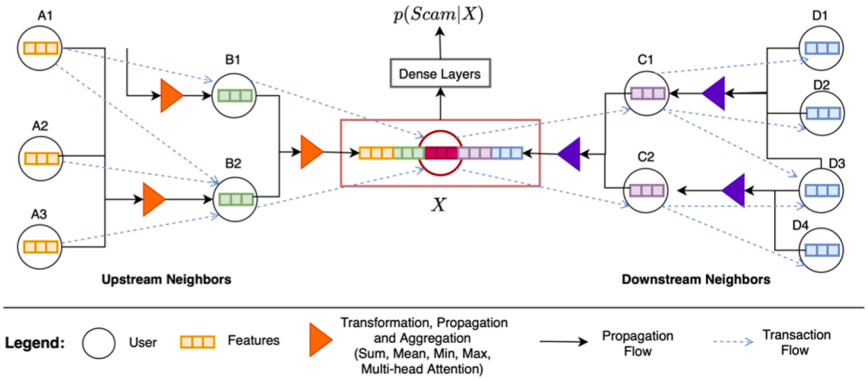


Fig. 4. Bidirectional multi-tier attention-based inductive graph neural network.

an H-dimensional vector. The result is combined with the transformed h_i^{k-1} to produce gs_i^k (Eq. (2)). A similar process is applied to target nodes $NT(i)$ to compute gt_i^k.

Next, we adopt the multi-head attention [65] mechanism to determine the attention that node i pays to each of its source and target neighbors. In each attention layer (i.e., head), the attention weight of node i to node j (i.e., $att_{i,j}$) is calculated as the scaled dot product between the transformed node feature h_i^{k-1} and the concatenation of h_j^{k-1} with the edge attribute $e_{j,i}$ (or $e_{i,j}$), normalized by Softmax function (Eqs. (4)–(5)). The feature attended to, fs_i^k (ft_i^k), is the attention-weighted sum of the transformed features of source (target) neighbors (including node i itself). For multi-head attention, we average fs_i^k and ft_i^k across all attention heads. In Eq. (6), we concatenate all generated features and transform the concatenated result to produce the H-dimensional latent node feature h_i^k.

Finally, in Eq. (7), a linear transformation is applied to the concatenated $h_i^0, h_i^1, \ldots, h_i^k$ to predict the final fraud risk. Although h_i^k integrates information from the previous propagation steps, h_i^{k-1}, we still include all earlier representations to mitigate potential information loss during the multi-layer transformation. Eqs. (1)–(7) include tunable parameter matrices: $W_0 \in \mathbb{R}^{H\times H1}$, $W_s, W_t, O_s, O_t, V_s, V_t \in \mathbb{R}^{(H1+H2)\times H}$, $M_s, M_t, L_s, L_t, Q_s, Q_t \in \mathbb{R}^{H\times H}$, $U_s, U_t, Z \in \mathbb{R}^{H\times 4H}$, and $P \in \mathbb{R}^{1\times (K+1)H}$, as well as a bias parameter b. Here $H1$ ($H2$) is the dimension of node (edge) attributes, H is the new dimension after transformation, and K is the number of propagation steps.

Next, we describe how the input features (x_i and $e_{i,j}$) are generated. As shown in Table 6, we derive features for each node based on its incoming and outgoing transactions separately, resulting in 54 features per each direction. For instance, after sorting all incoming transactions by timestamp, we calculate the time intervals between consecutive transactions and then compute the summary statistics of these intervals to capture the intensity of an account receiving tokens. The intuition is that fraudsters often perform transactions in rapid succession. We also measure the number of unique Web3 tokens involved in an account's transactions, as fraudsters tend to engage with a broad range of tokens. Similarly, for each pair of nodes, we compute the same set of features based on the transactions between them, resulting in 54 edge features. To enhance model efficiency, we exclude features with >50 % missing values (e.g., no std. for a single transaction), resulting in 58 node features and 24 edge features.

3.4. Token fraud risk estimation

We estimate token fraud risk for Web3 applications (i.e. application-level fraud) based on account-level fraud predictions. We first estimate the probability that an account is fraudulent using the token fraud detection module described in Section 3.3. Then we sort out the accounts that have transacted a Web3 token over a given period (i.e., in week t) as either token receivers or token senders and aggregate the account-level risks to an *application-level fraud risk* through aggregator functions. In this study, we adopt a set of functions, weighted mean (denoted as *FraudRiskAvg*), standard deviation (denoted as *FraudRiskStd*) and coefficient of variation (denoted as *FraudRiskCV*), to aggregate the account-level risks for each Web3 token over a given

Table 6
Input node and edge features.

Features	Raw data	Aggregators	# of Features	Intuition
Transaction intensity	Time intervals between consecutive transactions	Count, min, max, median, mean, std, sum	7	Fraudsters usually commit transactions with high intensity
Transaction volume	Volume attribute in transactions	Min, max, median, mean, std	5	Fraudulent transactions usually have large volume
Web3 token coverage	Web3 token ID	Unique count	1	Fraudsters may have interest in many tokens
Web3 application quality (See Appendix A, Table A-1)	Web3 App. quality characteristics (8)	Min, max, median, mean, std	40 (5 * 8)	Low quality web3 tokens may be vulnerable to token fraud; Large raised fund may attract fraudsters
Open-source development (OSD)	Web3 application OSD indicator	mean	1	OSD increases the risk of token fraud

period, generating an application-specific fraud risk profile in Eqs. (8)-(10),

$$FraudRiskIndicators: \left\{ FraudRiskAvg_{i,t} = \frac{\sum_{s \in U(i,t)} NodeRisk_{s,target} + NodeRisk_{s,source}}{2|U(i,t)|} \right. \tag{8}$$

$$FraudRiskIndicators: \left\{ FraudRiskStd_{i,t} = \sqrt{\frac{\sum_{s \in U(i,t)} (NodeRisk_s - \overline{NodeRisk})^2}{|U(i,t)|}} \right. \tag{9}$$

$$FraudRiskIndicators: \left\{ FraudRiskCV_{i,t} = \frac{FraudRiskAvg_{i,t}}{FraudRiskStd_{i,t}}, \right. \tag{10}$$

where $U(i,t)$ corresponds to the set of transactions associated with token i during time t, with $|U(i,t)|$ denoting the size of the set, and $NodeRisk_{s,\ target}$ $(NodeRisk_{s,\ source})$ indicates the fraud risk of the source (target) node in each transaction s of $U(i,t)$.

Intuitively, a high value of the *FraudRiskAvg* for a token implies that a significant proportion of its transacting accounts are likely to be fraudulent. Similarly, a high value of the *FraudRiskStd* suggests that the likelihood of fraudulent accounts associated with this token is widely dispersed. To put it differently, in the case of a random transaction, there is considerable uncertainty regarding whether the transacting parties are fraudulent or not.

3.5. Evaluating the impact of token fraud risk

In addition to measuring the risk to Web3 applications posed by token fraud, we also investigate the impact of such fraud risks on the performance of these applications. Their ultimate goal is to sustain themselves in the longer term after securing external financing. Considering the widespread popularity of ICO crowdfunding among Web3 applications, our analysis focuses specifically on those funded through ICOs. Rather than examining ICO success, we draw our attention to post-ICO performance, examining how Web3 token fraud risk influences the longer-term growth and viability of Web3 applications. As Web3 token fraud undermines the trust between members in Web3 ecosystems [10,11], it may threaten the long-term sustainability of these applications. Based on our proposed risk measurement, we would like to specifically address (1) whether and how the token fraud risk may affect the post-crowdfunding growth of Web3 applications and (2) whether the previous growth will moderate the impact of token fraud risk.

Prior literature has demonstrated the critical roles of application adoption and user acquisition in business growth [17,27,28,66,67]. We therefore describe the post-crowdfunding performance of Web3 applications on two dimensions: token usage and user base expansion. Although market price is another common metric of post-ICO performance of Web3 applications, we do not investigate it, for the following reasons: (1) Our analysis centers on-chain transactions that reflect the multifaceted roles played by Web3 tokens, such as product consumption and governance, beyond their use as cryptocurrency [21]. These on-chain activities are directly tied to the operational health of Web3 applications, aligning our focus with their application performance rather than with market values. (2) Our fraud risk metric is derived from on-chain transactions, but the market price of Web3 tokens is largely influenced by off-chain transactions, speculative trading, and exchange liquidity [68–71]. Disentangling these effects would require extensive modeling of non-transactional factors, which falls outside the scope of our study. (3) Market prices vary significantly across exchanges due to differences in liquidity and trading activity, introducing inconsistencies and potential biases in the analysis [37]. (4) Only 20 % of Web3 tokens are listed on exchange platforms, limiting the generalizability of market prices as a general performance indicator across broader Web3 ecosystems.

In line with existing literature on venture growth and financial markets [17], we aggregate the relevant data by the week. Correspondingly, we compute the weekly token fraud risk indicators ($FraudRiskIndicator_t$) formulated in Eqs. (8)–(10). We employ the panel vector autoregression (PVAR) model to answer the proposed questions. PVAR applies a statistical model to variable movements, implying genuine simultaneity among them. It treats all variables as jointly endogenous, without distinguishing between exogenous and endogenous ones [72]. It is "a useful means of summarizing time series facts" [73] and "helps us understand the complexity of dynamics between endogenous variables and capture the feedback loops" [74]. Specifically, PVAR is well suited for analyzing dynamic relationships between variables over time. For instance, a recent study applied PVAR analysis to explore the relationship between cryptocurrency prices and open-source development activities [74]. Similarly, we leverage PVAR to examine how token fraud risk may shape the future performance of applications, including their usage and user base growth. More importantly, the utilization of the PVAR model enhances the robustness of our study by effectively addressing concerns related to non-stationarity, spurious causality, endogeneity, serial correlation, and reverse causality [74,75]. The PVAR model is generally expressed by the equation

$$\boldsymbol{Y}_{i,t} = \sum_{k=1}^{s} \mathrm{A}_k \cdot \boldsymbol{Y}_{i,t-k} + \mathrm{B} \cdot Controls_{i,t} + \boldsymbol{u}_i + \boldsymbol{\varepsilon}_{i,t}, \tag{11}$$

where $\boldsymbol{Y}_{i,t}$ is a column vector that contains the variables of fraud risk and post-crowdfunding growth for project i at week t, $\boldsymbol{Y}_{i,t-k}$ is k-week lagged

dependent variables, s is the maximum lag order, and A_k and B are related coefficient vectors. The vector $Controls_{i,t}$ includes time-varying project-level covariates such as project age, which may influence both fraud risk and application performance. The panel data structure by nature incorporates $\boldsymbol{u}_i$ as time-invariant unobserved project-specific effects; $\boldsymbol{\varepsilon}_{i,t}$ is the error vector satisfying the normality and independence assumption of $E(\boldsymbol{\varepsilon}_{i,t}) = 0$ and $Cov(\boldsymbol{\varepsilon}_{i,m}, \boldsymbol{\varepsilon}_{i,n}) = 0$.

4. Data description and experiment setup

4.1. Data description

As described in Appendix A, Table A-2, we obtained a dataset of 6362 Web3 applications funded through ICOs as of February 2021 from ICObench.com. We added missing information from icodrops.com and trackico.io for about 10 % of the Web3 applications in our sample. We successfully matched 3642 Web3 applications with their token identifiers on the Ethereum blockchain and collected their transaction records from its inception until February 2021. As we focused on the post-crowdfunding stage, we excluded transactions before theICO end date and removed the Web3 applications that did not survive their ICO stage. Table 7A–C presents a summary statistic of the quality features of Web3 applications, transaction number, transaction volume,[10] and the numbers of nodes, edges, and fraudulent nodes in each year's transaction network.

As shown in Table 7A, our final dataset consists of 2427 Web3 applications with 26,641,703 accounts and 159,845,654 transaction records. Based on the list of 5839 cloud-reporting fraudulent accounts from etherscan.io, we pinned down 761 fraudulent accounts that had transacted our focal Web3 tokens associated with the Web3 applications in our samples.

To better understand the nature of Web3 token transactions, we analyzed a random sample of 1000 wallet addresses. For each address, we retrieved all associated transactions and consolidated all involved counterparties. As shown in Table 7D, this resulted in 7233 transactions involving 3821 unique addresses. Due to the encryption of transaction payloads, we can only infer the nature of these transactions based on the participating addresses. We manually reviewed these addresses to identify whether they were affiliated with CEX or DEX.[11] Our analysis revealed that 94.5 % of the addresses were not linked to any exchange platform and 77 % of the transactions occurred exclusively between the non-exchange addresses. These findings suggest that the majority of transactions may not be trading-related, but instead reflect the broad functionalities of Web3 tokens, such as crowdfunding, payment, product usage, and governance [21].

4.2. Inductive model training

Our data range from the year 2017 to 2020. As shown in Table 7C, the transaction network changes significantly year to year. To ensure model generalizability, we adopt a time-based nested cross-validation procedure [25], a well-established method for time series prediction. As shown in Fig. 5A, with every two years as a cycle, we train the model on the transaction data from the first year and test it on the data from the second year. For example, the model is first trained on the transaction data of 2017, and it predicts the fraud probability for new accounts in the transaction data of 2018. This process is repeated across subsequent cycles. Through such continuous train–test cycles, we can assess the consistency of all models' performance over time.

We follow a widely adopted inductive learning procedure to partition the transaction graph into training, validation, and test subsets [61, 76,77]. We use the 2017–2018 train–test cycle as an example to illustrate this procedure, which consists of three steps:

(1) *Splitting train, validation, and test positive nodes*. The 2017 train dataset includes 159 positive nodes. As illustrated in Fig. 5C, we randomly select 80 % of them for training and reserve the remaining 20 % for validation. The 2018 dataset consists of 295 newly added positive nodes, which are used exclusively for testing. Therefore, these test nodes are completely absent during training, strictly preventing any data leakage.

(2) *Sampling negative nodes and generating node batches*. Given our extremely unbalanced dataset, with over 2.5 million negative nodes, we subsample negative nodes to create batches. For training, we randomly sample 1000 negative nodes [12] using stratified sampling based on their node degrees and combine them with the positive training nodes to form a training node batch. Similarly, we generate a validation node batch with 250 randomly sampled negative nodes that have no overlap with the training batch. These training and validation node batches are generated on demand until model convergency. The testing node batch comprises the positive nodes along with 1000 randomly sampled negative nodes newly added in 2018.

(3) *Sampling a subnet for each node batch*. For each node batch, we need to sample a subnet for computing node embeddings by propagating information from neighbor nodes. Using the positive and negative nodes as seeds, we recursively sample up to S source and S target neighbors per seed node for K iterations [61]. This K-tier sampling enables K propagation steps as defined in Eqs. (1)–(7). Fig. 5B illustrates two sampled subnets for Node 1, generated by recursively sampling up to three neighbors over two iterations. Each subnet covers only a portion of the entire network, which helps reduce overfitting and improve generalizability [61]. We set $S = 50$ and $K = 2$, as this configuration yields the best validation performance. This aligns with Hamilton et al. [61], which indicates that increasing K and S produces marginal returns in performance at the cost of substantially higher runtime. The sampled subnets correspond to the training, validation, and test splits.

With sampled training subnets, we estimate the fraud probabilities and cross entropy loss for the seed nodes using Eqs. (1)–(7) and update the mode parameters to minimize the loss. Note that in each update, the model only sees a subnet and is trained to generalize across varying subnets. As a result, even when the transaction network evolves, the model remains capable of making correct predictions. The model is continuously evaluated on the evaluation subnets to monitor the training process. After training, the model is tested on the test subnet, and we report the performance of the seed nodes in the subnet. Table C-1 of Appendix C describes the hyper-parameter configurations of our model.

We follow the time-based nested cross-validation procedure to train three models using the data from 2017 to 2019. To make full use of training samples, we initialize the 2018 model with the learned parameters from the 2017 model and the 2019 model with the parameters from the 2018 model. This allows subsequent models to retain previously learned fraud patterns while being fine-tuned with new samples. These trained models are then used to predict the fraud probability for every account in the data from 2018 to 2020, respectively.

[10] Since the unit of measure in a transaction is $1/N$ tokens, where N is a large number defined in the token smart contract, we use the logged value to represent the large transaction volume.

[11] The list of exchanges can be found at https://etherscan.io/directory/Exchanges/Crypto_Exchanges.

[12] We experimented with different sizes for negatively sampled nodes (e.g., 500, 1000, 1500, 2000), but observed no significant changes in model performance across these variations.

Table 7
Descriptive statistics for key attributes of Web3 applications and accounts.

Attributes	Mean	Std. Dev.	Median	Min	Max	# of Obs.
(A) *Web3 applications*						
Raised Amount (log)	14.95	2.62	15.43	0.00	21.53	2427
Retention Rate	0.47	0.21	0.52	0.00	1.00	2427
Experts Ratings:						
Overall Rating	3.22	0.76	3.20	0.70	5.00	2427
Benchy Rating	3.17	0.75	3.20	0.60	5.00	2427
Team Rating	3.64	1.06	4.00	1.00	5.00	2427
Product Rating	3.40	1.03	3.60	1.00	5.00	2427
Team Size	13.77	8.41	12.00	1.00	75.00	2427
GitHub	0.55	0.50	1.00	0.00	1.00	2427
Project Scope	2.82	2.37	2.00	1.00	29.00	2427
(B) *Accounts*						
TransNum	12.00	2140.04	2.00	1.00	6372,594.00	26,641,703
TransAvgVol (log)	30.70	13.38	22.94	0.00	177.45	26,641,703

(C) *Transaction Networks*

Year	Edge	Node	Positive Nodes
2017	5171,721	2530,309	159
2018	15,020,358	6961,693	295
2019	14,943,477	6081,023	85
2020	40,630,560	14,676,793	222

(D)*Transactions involving Randomly Sampled Accounts*

Interaction Type	# Accounts (% of total)	# Transactions (% of total)
CEX-related	207 (5.4 %)	1498 (20.7 %)
DEX-related	34 (0.1 %)	173 (2.3 %)
Non-exchange	3580 (94.5 %)	5562 (77 %)
Total	3821	7233

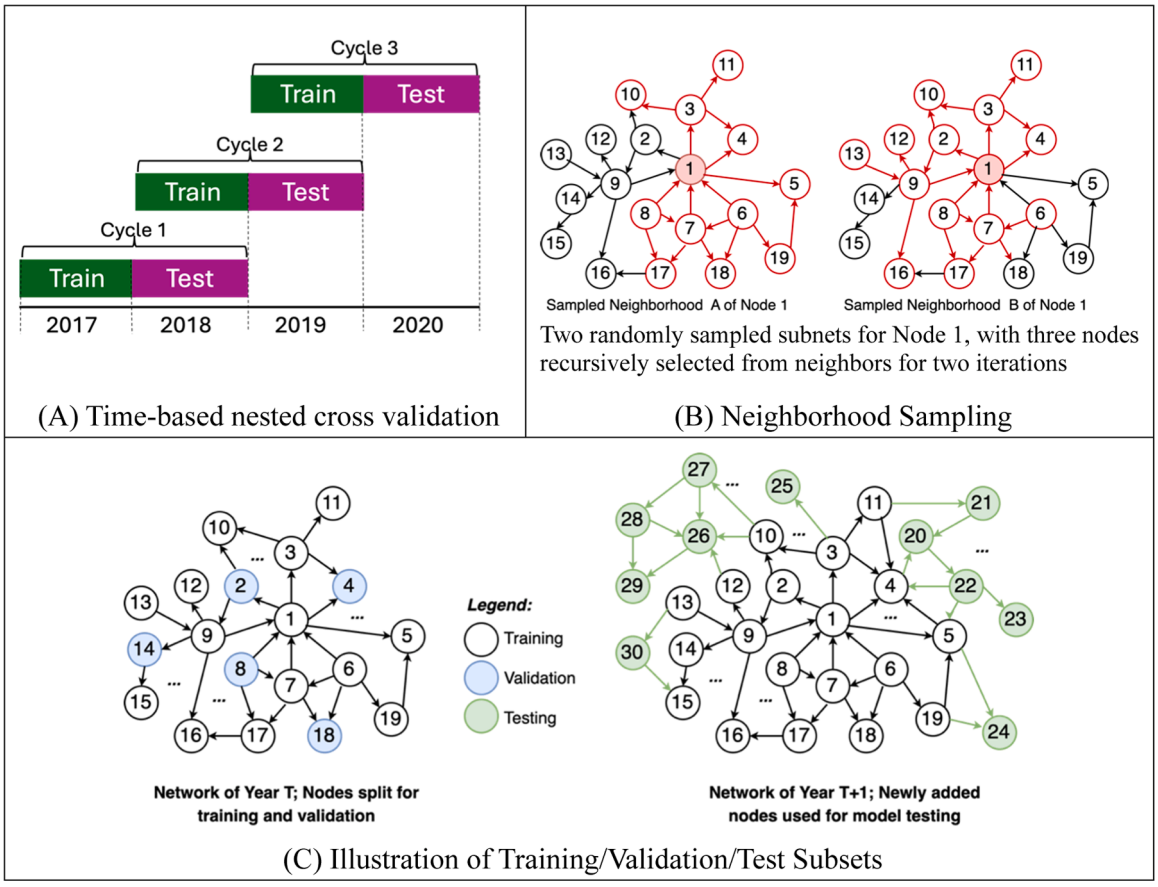


Fig. 5. Training, validation, and test data for inductive model training.

4.3. Benchmarking setup

To examine the effectiveness of our proposed token fraud detection model, we compare it with a number of benchmark models proposed by previous studies, including trimmed *K*-means [60], SVM [50,18], random forest [50], LightGBM [50], DElightGBM [50] and Node2Vec [19]. The selection of the benchmarking models was discussed in Section 2.3 (see Table 2). Since these models are unable to propagate neighbors' features automatically, for fair comparison, we generated the neighbors' features of each node following W. Chen, Guo, et al. [50] and concatenated them with the node's own features. The parameters of each baseline model were optimally determined by grid search. For reference, we provide the configurations of the baseline models in Appendix C, Table C-2. We exclude transductive graph neural network models such as graph convolution network (GCN) and graph attention network (GAT), because they depend on the entire network instead of

subnets, making it prohibitively expensive to train them on the massive network.

In line with previous studies [50,11,19], we consider commonly used measures for classification including *precision, recall,* and *F1 score* using a default threshold of 0.5. To measure the overall model performance regardless of the threshold, we also calculate the area under the *precision–recall curve* (PRC), which is preferred for models trained by extremely unbalanced datasets [78]. We do not consider area under the ROC curve (AUC), because it may inflate model performance for extremely unbalanced data [78,79].

5. Model performance and empirical results

5.1. Performance of token fraud detection model

Table 8 show the classification performance of each model. From the comparison, our proposed model consistently achieves the best results in almost all the metrics over the three train–test cycles. For the positive class, our model achieves an F1 score of 69.52 %, 58.13 %, and 69.82 % for the respective periods, outperforming the best baseline model by over 7 % in each case. Notably, the model achieves recall rates of 71.53 %, 69.41 %, and 72.77 %, which are >8–15 % higher than the best-performing baseline, indicating that our model is more effective in retrieving fraudulent accounts. Consistent with the positive class, for negative class, our model outperforms all the baseline models by up to 30 % in F1. Regarding overall performance, the PRC scores of our model exceed those of the best baseline models by 4–8 % for the three time periods. Note that during the period 2018–2019, due to the significantly reduced positive cases (85), all models exhibit lower performance than for other periods.

Recall that for fair comparison, we followed a cascading method [50] to generate features from each node's first- and second-tier neighbors in all the baseline models. Thus, the performance gain of our model can be attributed to our design components, including the use of inductive graph neural network learning, attention mechanisms, and effective feature propagation. In particular, we expanded the input features to include signals for Web3 application quality. To understand the effects of this extension, we performed an ablation study where these quality features are excluded from the models. As shown in Table 9, for the positive class, the average precision decreases by 6 %, recall by 3 %, and F1 score by 4.81 %. For the negative class, F1 score is reduced by 2.53 % and PRC by 7.8 %. These findings show the value of these quality features as critical inputs for improving model performance.

5.2. Impact of token fraud risk on post-crowdfunding performance

To comprehensively evaluate the dynamic relationship between token fraud risk and post-crowdfunding performance of Web3 applications, we generate risk profiles and post-crowdfunding performance metrics for each Web3 application based on the account-level fraud predictions. Starting the first day after the ICO crowdfunding event, we first follow the equations shown in Eqs. (8)–(10) to calculate three weekly risk indicators, namely *FraudRiskAvg, FraudRiskStd*, and *FraudRiskCV*, which describe the token fraud risk profile of each Web3 application in each week. Then we calculate the weekly transaction count (denoted as *TransCnt*) and the weekly average transaction volume (denoted as *AvgTransVol*) to indicate application usage, and the weekly count of active accounts (denoted as *AccCnt*) to indicate user expansion. The variables and their descriptive statistics are summarized in Tables 10 and 11, respectively. All performance metrics are log-transformed to reduce skewness.

With the four data panels, we conducted PVAR analysis to examine their evolving relationship using the empirical models in Eq (11). By performing Fisher-type root unit tests, we confirmed the absence of unit roots and the satisfaction of the stationarity assumption. With the lag order selection, we found the optimal lag order was 3, as it yielded the highest overall coefficient of determination (CD). Given this, projects with fewer than 4 weeks of post-crowdfunding data were automatically dropped by the PVAR implementation due to insufficient data for lagged terms. Following the PVAR analysis procedures suggested by previous studies [72,74], we applied a first-differencing procedure to remove panel fixed effects, effectively eliminating characteristics specific to individual Web3 applications that could otherwise bias the results. Also, we subtracted the cross-sectional mean from each variable to account for

Table 8
Model Benchmarking.

Models | Metrics (%)	Positive Class				Negative Class	F1 Average	PRC	
	Precision	Recall	F1	F1. Diff.	F1		Score	Diff.
2017 – 2018 (295 positive cases)								
Trimmed K-means [1]	9.52	8.13	8.75	–	75.61	42.18	N/A	–
Random Forest	48.35	44.75	46.48	0.00	84.96	65.72	45.82	0.00
SVM	52.33	53.22	52.77	6.29	85.91	69.34	47.83	2.01
LightGBM	61.40	47.46	51.99	5.51	86.93	69.46	59.67	13.85
DElightGBM	58.89	63.39	61.11	14.63	84.43	72.77	67.84	22.02
Node2Vec	67.21	56.27	61.26	14.78	89.74	75.50	70.22	24.40
Our model (Full)	**67.63**	**71.53**	**69.52**	**23.04**	**90.62**	**79.65**	**78.66**	**32.84**
2018 – 2019 (85 positive cases)								
Trimmed K-means	8.33	10.59	9.33		91.15	50.24	N/A	–
SVM	12.45	85.89	21.76	0.00	64.98	43.37	20.76	0.00
Node2Vec	23.33	49.41	31.70	9.94	90.50	61.10	27.74	6.98
Random Forest	42.70	44.70	43.68	21.92	95.08	69.38	32.60	11.84
LightGBM	44.14	57.65	50.00	28.24	95.04	72.52	47.82	27.06
DElightGBM	46.93	54.11	50.27	28.51	95.43	72.85	50.59	29.83
Our model (Full)	**50.00**	**69.41**	**58.13**	**36.37**	**95.75**	**76.22**	**55.21**	**34.45**
2019–2020 (222 positive cases)								
Trimmed K-means	38.01	21.60	27.54		88.44	57.99	N/A	–
SVM	51.00	36.15	42.31	0.00	89.91	66.06	31.96	0.00
Random Forest	47.13	53.99	50.33	8.02	88.47	69.40	50.12	18.16
Node2Vec	64.19	44.60	52.63	10.32	91.71	72.17	51.66	19.70
LightGBM	59.50	55.87	57.63	15.32	91.31	74.47	57.48	25.52
DElightGBM	65.12	59.62	62.25	19.94	92.37	77.31	65.88	33.92
Our model (Full)	**67.10**	**72.77**	**69.82**	**27.51**	**93.24**	**81.53**	**73.63**	**41.67**

[1] Trimmed *K*-means is an unsupervised clustering model. We determine the cluster with the most positive training cases as the "positive cluster" and assign each test sample to the nearest cluster based on its distances to cluster centroids. For example, in 2017–2018 data, the positive cluster consists of 112 accounts, with 12 correctly identified. Since each test sample is "hard" assigned to a cluster without a probability, we omit the calculation of PRC.

Table 9
Comparing full model with ablation (full – token features) model.

Year	Positive Class							F1 of Negative Class			PRC		
	Precision		Recall		F1								
	Ablation	Full	Ablation	Full	Ablation	Full	Diff.	Ablation	Full	Diff.	Ablation	Full	Diff.
2017 – 2018 (295 positives)	65.64	67.63	72.54	71.53	68.92	69.52	0.60	88.00	90.62	2.62	74.23	78.66	4.43
2018 – 2019 (85 positives)	43.59	50.00	60.00	69.41	50.50	58.13	7.63	93.03	95.75	2.72	40.50	55.21	14.71
2019–2020 (222 positive cases)	57.09	67.10	71.83	72.77	63.62	69.82	6.20	91.00	93.24	2.24	69.38	73.63	4.25
Average	55.44	61.58	68.12	71.24	61.01	65.82	4.81	90.68	93.20	2.53	61.37	69.17	7.80

Table 10
Descriptions for variables in the PVAR model.

Construct	Variable	Definition
$Performance_t$	$TransCnt_t$	The total number of transactions at week t
	$AvgTransVol_t$	The average transaction volume at week t
	$AccCnt_t$	The total number of active accounts at week t
$FraudRiskIndicator_t$	$FraudRiskAvg_t$	The token fraud risk indicator for week t using mean aggregation in Eq (8)
	$FraudRiskStd_t$	The token fraud risk indicator for week t using standard deviation aggregation in Eq (9)
	$FraudRiskCV_t$	The token fraud risk indicator for week t using coefficient of variance aggregation in Eq (10)

Table 11
Descriptive statistics of variables in the PVAR model.

Variable	Mean	Std. Dev.	Median	Min	Max	# of Obs.	# of Apps.
$TransCnt_t$ (log)	2.213	2.340	1.386	0	12.444	224,519	2427
$AvgTransVol_t$ (log)	27.776	22.648	29.719	0	177.445	224,519	2427
$AccCnt_t$ (log)	1.928	2.036	1.386	0	12.328	224,519	2427
$FraudRiskAvg_t$	0.125	0.158	0.054	0	1.000	224,519	2427
$FraudRiskStd_t$	0.152	0.176	0.046	0	0.707	224,519	2427
$FraudRiskCV_t$	2.764	58.687	0.608	0.010	4720.033	124,949[a]	2427

[a] The reduced # of obs. is due to the undefined std. dev. for tokens that only have a single transaction in a week.

Table 12
PVAR Estimation Results.

(1) FraudRiskAvg	(1) $FraudRiskAvg_t$	(2) $TransCnt_t$	(3) $AvgTransVol_t$	(4) $AccCnt_t$
t-1	0.2265***	**−0.1979*** **	**−0.1391*** **	**−0.1727*** **
	(0.0041)	(0.0200)	(0.0303)	(0.0165)
t-2	0.1521***	**−0.1399*** **	**−0.0923*** **	**−0.1089*** **
	(0.0041)	(0.0193)	(0.0304)	(0.0161)
t-3	0.1313***	**−0.0805*** **	**−0.0914*** **	**−0.0702*** **
	(0.0040)	(0.0190)	(0.0299)	(0.0159)
(2) TransCnt				
t-1	**0.0074*** **	0.3619***	0.0891***	0.0773***
	(0.0012)	(0.0095)	(0.0073)	(0.0061)
t-2	**0.0022***	0.1280***	0.0258**	0.0071
	(0.0012)	(0.0082)	(0.0080)	(0.0059)
t-3	−0.0019	0.1144***	0.0129*	0.0113**
	(0.0012)	(0.0077)	(0.0079)	(0.0058)
(3) AvgTransVol				
t-1	**0.0041*** **	−0.0423***	0.2563***	−0.0366***
	(0.0006)	(0.0031)	(0.0049)	(0.0026)
t-2	**0.0037*** **	0.0263***	0.2003***	0.0208***
	(0.0006)	(0.0030)	(0.0049)	(0.0026)
t-3	**0.0052*** **	0.0270***	0.1826***	0.0237***
	(0.0006)	(0.0029)	(0.0048)	(0.0025)
(4) AccCnt				
t-1	**−0.0090*** **	0.2225***	0.0016	0.4545***
	(0.0014)	(0.0119)	(0.0093)	(0.0085)
t-2	**−0.0035*** **	0.0572**	−0.0159*	0.1727***
	(0.0015)	(0.0104)	(0.0098)	(0.0081)
t-3	**−0.0014**	0.0277	−0.0138***	0.1262***
	(0.0014)	(0.0097)	(0.0097)	(0.0076)

time fixed effects, thereby isolating temporal dynamics in the data. To address potential endogeneity bias, we employed system-GMM, a robust estimation method that uses lagged values of endogenous variables as internal instruments [72]. This approach not only corrects for endogeneity by breaking the correlation between explanatory variables and the error term but also accommodates the dynamic relationships inherent in PVAR models. By leveraging both differenced and level equations, system-GMM improves estimation efficiency and ensures consistent, unbiased results. These methodological enhancements collectively affirm the robustness of our PVAR analysis.

Additionally, we performed a Granger causality test [80] for each equation of the underlying PVAR model. The Granger causality tests within the PVAR framework indicate that earlier instances of token fraud risk significantly predict later shifts in usage growth (*"TransCnt"* and *"AvgTransVol"*) and user base expansion (*"AccCnt"*). The results of the Granger causality test are displayed in Appendix D.

We have three risk indicators in the proposed risk profile. As we found that all these metrics show consistent results, for reporting purposes, we will only display the PVAR results from the *FraudRiskAvg* indicator. In our time series setup for the PVAR model, we use calendar week as the time variable, which controls for the market sentiment by nature. Therefore, we do not include ETH or BTC price as control variables. Instead, we include *Age* which is the number of weeks since the launch of the application. The corresponding model is presented as follows:

$$\begin{bmatrix} FraudRiskAvg \\ TransCnt \\ AvgTransVol \\ AccCnt \end{bmatrix}_{i,t} = \sum_{k=1}^{3} A_k \cdot \begin{bmatrix} FraudRiskAvg \\ TransCnt \\ AvgTransVol \\ AccCnt \end{bmatrix}_{i,t-k} + B \cdot Age_{i,t} + \boldsymbol{u}_i + \varepsilon_{i,t}. \quad (12)$$

5.2.1. PVAR estimation results

Table 12 provides the estimated dynamics abetween token fraud risk and post-crowdfunding performance obtained from the PVAR model shown in Eq. (12). We start by examining the interplay between *FraudRiskAvg* and the application usage metrics, *TransCnt* and *AvgTransVol*. The coefficients on row (1) and columns (2)–(3) in Table 12 indicate the Granger causal effect of *FraudRiskAvg* at times *t-3, t-2,* and *t-1* on *TransCnt* and *AvgTransVol* at time *t*. The significantly negative coefficients suggest that higher token fraud risk predicts a decline in application usage over time, as the heightened risk may scare users away. Conversely, the highlighted coefficients on rows (2)–(3) and column (1) show that prior increases in *TransCnt* and *AvgTransVol* at times *t-3, t-2*, and *t-1* predict a subsequent rise in *FraudRiskAvg* at time *t*, revealing a bidirectional dynamic. A Web3 application with high transaction frequency and volume is more likely to attract fraudsters, as its surge of weekly transaction count and average transaction volume makes it a more appealing target, thereby elevating token-fraud risk.

From the coefficients on row (1) and column (4), we observe a similar negative Granger causal effect of *FraudRiskAvg* at times *t-3, t-2,* and *t-1* on *AccCnt* at time *t*, implying that a rise in token fraud risk predicts a subsequent decline in user base expansion. Conversely, the coefficients on row (4) and column (1) indicate that *AccCnt* at times *t-3,*

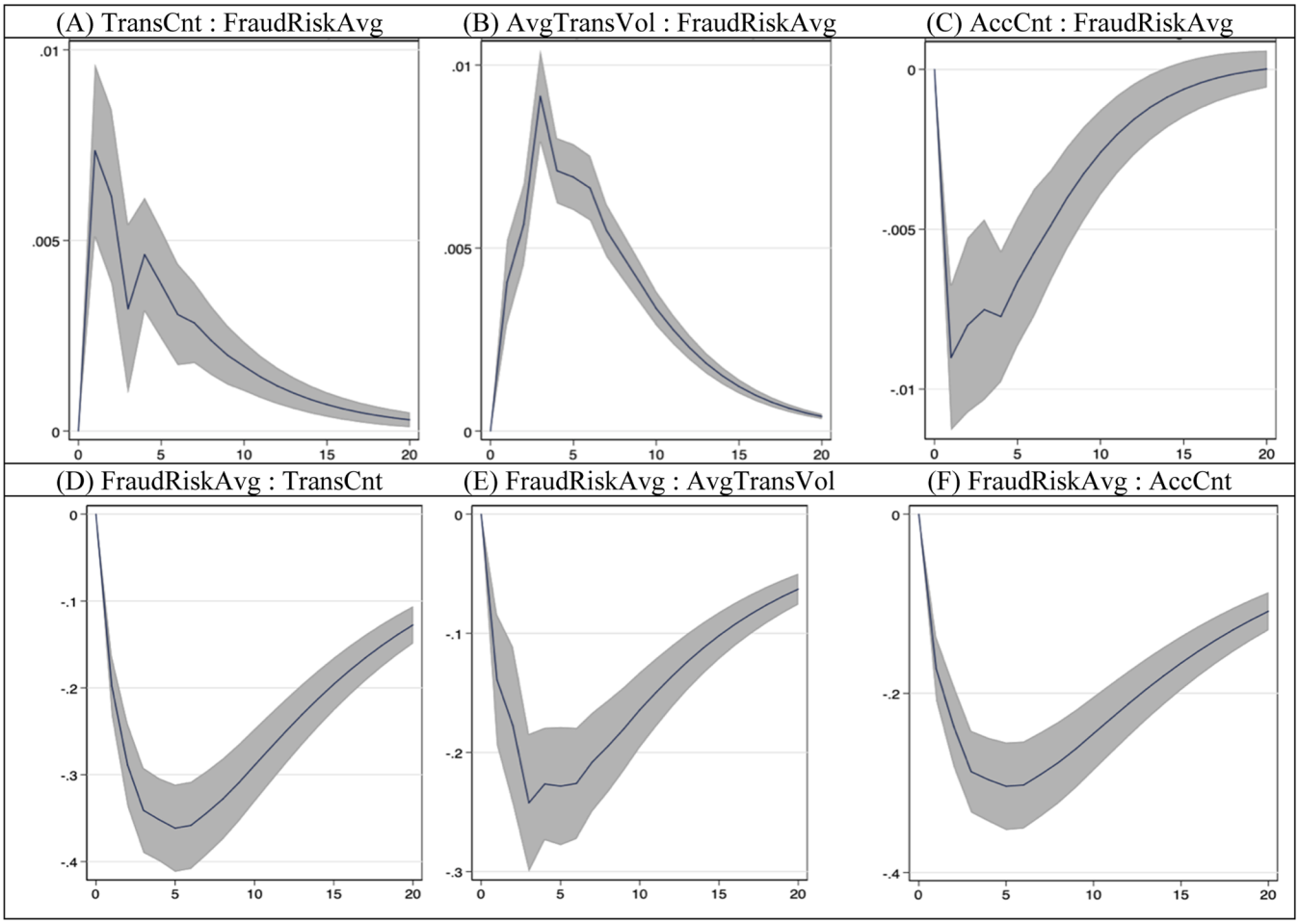


Fig. 6. Impulse response plots.

t-2, and *t-1* appears to exhibit a negative Granger causal relationship with subsequent *FraudRiskAvg* at time *t*. This can be explained by the fact that most ordinary crypto accounts associated with Web tokens engage in occasional and modest transactions. An increase in the number of active accounts may not result in a proportional rise in transaction volume; rather, it could reduce token fraud risk by diminishing the proportion of illicit accounts among all active accounts.

Additionally, Type I and Type II errors in the fraud detection model may introduce potential bias. Specifically, Type I errors (false positives) may falsely elevate a node's fraud risk and therefore increase the fraud risk for associated Web3 applications. In contrast, Type II errors (false negatives) may misclassify fraudulent nodes as benign, leading to an underestimation of fraud risk. These errors could impact the accuracy of our risk assessment and subsequent analyses. To correct such potential bias, we conduct the SIMEX procedure [81] to correct both the systems error and the measurement errors identified from the diagnosis step. The detailed process and results are described in Appendix E. The corrected coefficients obtained from this procedure remain consistent with the results shown in Table 12, thereby reinforcing the robustness of our findings.

5.2.2. Impulse response function plots

Furthermore, we also calculate the impulse response function (IRF) to analyze the long-term dynamics with a forecasting horizon of 20 weeks. IRF demonstrates the longer-term response of each of the PVAR endogenous variables to one unit shock of the other endogenous variable through Monte Carlo simulation with 100 repetitions. Figs. 6A–C show the response of token fraud risk to one unit increase in the performance metrics, *TransCnt, AvgTransVol,* and *AccCnt*. It is observed that the increases in the performance metrics of application usage, *TransCnt* and *AvgTransVol,* are followed by positive responses in token fraud risk, while an increase in user base (*AccCnt*) corresponds to a negative response. These patterns persist for up to 15 weeks. Figs. 6D–F reveal the responses of post-ICO performance metrics to one unit increase in token fraud risk. We observe that increases in token fraud risk are followed by a negative and instant response in *TransCnt, AvgTransVol,* and *AccCnt*, with effects persisting for up to 20 weeks. Generally, the IRF results are consistent with the findings from the PVAR regression results.

Overall, our analysis suggests two key dynamics. First, a surge in application usage is associated with an increased risk of Web3 token fraud. According to routine activity theory [82], the features of environmental settings, such as target suitability and the presence of guardians, impact criminal activities. Hence, the visibility of an IS application increases its risk of being attacked [31]. Higher transaction counts and volumes increase the visibility of Web3 application tokens, making them more attractive and suitable targets for fraud. Second, our findings reveal that an expanded and stable user base can mitigate token fraud risks. Routine activity theory also posits that the presence of capable guardians reduces the likelihood of attacks [82,31]. In Web3 applications, which are co-governed by their users, a larger and more stable user base enhances collective vigilance and governance capabilities. This strengthens the application's overall guardianship, fostering greater deterrence and resilience against fraudulent activities. Together, these findings highlight the dual role of usage growth and user base expansion in shaping the fraud risk landscape for Web3 applications.

Additionally, we conducted an initial analysis on the relationships between token fraud risk and token market values for 562 exchange-listed Web3 tokens, summarized in Appendix F. However, compared with the results for the 2427 tokens shown in Table 12, this analysis shows different interplays between these variables for this small subset. This inconsistency may stem from unclear interactions between on-chain and off-chain transactions. As discussed in Section 3.5, as our fraud risk indicators are derived from on-chain token transactions, whereas token market values are driven by the predominant off-chain transactions from centralized crypto exchanges [4,37]. Due to the unavailability of the off-chain data, we have reserved this issue for future research.

6. Discussion and conclusion

In this paper, we have addressed the threat of fraud in the Web3 industry, a sector experiencing remarkable growth but deeply plagued by the prevalence of token fraud. We designed and implemented an end-to-end framework to measure the risk to Web3 applications posed by Web3 token fraud. We proposed an inductive custom GNN model to predict fraud accounts based on-chain token transactions and Web3 application quality information. Our model can generalize to evolving transaction networks and consistently outperform existing benchmarking models in year-over-year evaluations. The predictions generated by our model were used to create real-time risk indicators for each Web3 application. Using these indicators, we further assessed the impact of token fraud on the performance of Web3 applications. Our empirical results showed that as token fraud risk increases, application usage and user base expansion tend to decline, with these negative effects often persisting for several months. Moreover, our findings suggest that the prior surges in application usage may exacerbate token fraud risk, whereas previous user base expansion can mitigate it.

Our study offers several methodological, theoretical, and practical implications for IS, fraud detection, and the recent Web3 economy.

Methodological Implications: Fraud detection is an enduring topic in IS research [12,15]. Our work introduces a generalizable method for detecting fraud within the emerging Web3 context, which involves complex and evolving Web3 token transaction networks. Our inductive learning approach echoes the active learning method suggested for financial fraud detection [64] and complements existing research on BTC/ETC fraud [17,29,19]. Moreover, while most fraud detection studies focus on identifying fraudulent behavior at the individual user level [13,29,30], our study takes a significant step further by aggregating the fraud risk from the individual level to the Web3 application level. To the best of our knowledge, this paper is among the pioneering works to quantify the fraud risk of an application, that is, a venture, based on fraudulent behavior of its users. Our proposed measure can specifically facilitate future studies on the trust issues of Web3 applications, addressing the call from IS scholars to examine user trust in the context of blockchain technology and applications [10,83–85].

Theoretical Implications: While previous literature often focuses on the financial losses caused by fraud, our study draws attention to the impact of Web3 token fraud on the future growth of Web3 applications and demonstrates the prolonged impact of token fraud on Web3 applications performance, with effects lasting up to 15 weeks. Given the high failure rate of Web3 applications and ongoing public skepticism toward the Web3 industry, our findings offer a potential route to examining the sustainability of Web3 ecosystems. Specifically, we investigate the post-crowdfunding success of Web3 applications by exploring the relationships between the token fraud risk and their application performance. While prior research has extensively analyzed the success factors related to an ICO crowdfunding [2,32,33], fewer studies have focused on post-crowdfunding success [2,39]. By addressing this gap, our work contributes to a deeper understanding of the factors that influence the sustainability of Web3 applications. By linking fraud risk to sustainability outcomes, this study emphasizes the importance of proactive fraud mitigation strategies and opens avenues for future research to explore governance models, user participation, and security mechanisms in shaping the long-term viability of Web3 ecosystems. Additionally, our empirical results extend the applicability of existing information systems susceptibility frameworks [31] to the Web3 context. This provides valuable insights into application-level risks and their implications for decentralized ecosystems.

Practical Implications: Finally, our work provides actionable insights for users, ventures, and regulators in Web3 industry to foster its growth and sustainability. Our proposed fraud risk indictors may help users select Web3 tokens, scrutinize trading parties, and stay vigilant during

token transactions. Ventures can leverage our fraud detection models to constantly monitor the trading behaviors of token holders and responsively mitigate vulnerabilities. Regulators can track the fraud risk across Web3 applications and take proactive measures to curb fraudulent behaviors.

This work is not without limitations, which may lead to future research. First, we take a data-driven approach to fraud detection but do not dive into the root causes of fraud. For example, future studies may explore the heterogeneity among Web3 applications or analyze the technical vulnerabilities of smart contracts to explore this area. Second, extending the idea of risk profiles, future studies can investigate additional functions (e.g., probabilistic aggregator) to generate alternative measures for assessing fraud risk in Web3 applications. Third, our preliminary study (see Appendix F) indicates inconsistent relationships between Web3 token fraud and token market values. Future research could examine fraud in both on-chain and off-chain transactions to assess its collective impact on token market values. Additionally, as the majority of tokens in our dataset operate on the same Ethereum and cross-chain Web3 activity remains limited [52], we do not explicitly model multi-blockchain interactions, which also offer a valuable research direction. In addition, graph-based models may be extended for fraud detection in similar context, if fraudulent behaviors exhibit a network structure. Finally, agent-based simulation could be valuable for exploring complex user interactions and the potential ripple effects between token transactions, prices, and application success in the context of fraud.

CRediT authorship contribution statement

Ziyi Xiong: Conceptualization, Writing – original draft, Writing – review & editing, Methodology, Investigation, Formal analysis, Data curation. **Rong Liu:** Writing – review & editing, Writing – original draft, Supervision, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Hemang Subramanian:** Writing – review & editing, Writing – original draft, Methodology, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.im.2025.104242.

References

[1] A. Kumar, R. Liu, Z. Shan, Is blockchain a silver bullet for supply chain management? Technical challenges and research opportunities, Decision Sciences 51 (1) (2020) 8–37.
[2] E. Lyandres, B. Palazzo, D. Rabetti, Initial coin offering (ICO) success and post-ico performance, Management Science (2022).
[3] Gherghelas, S. (2024b, October 8). State of the Dapp Industry Q3 2024. Dapp Industry Reports. https://dappradar.com/blog/state-of-the-dapp-industry-q3-2024/#Chapter-1.
[4] Chen, Y., Gurrola-Pérez, P., & Lin, K. (2023). A review of crypto-trading infrastructure. Available at SSRN 4560793. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4560793.
[5] M. Conti, E.S. Kumar, C. Lal, S. Ruj, A survey on security and privacy issues of bitcoin, IEEE Communications Surveys & Tutorials 20 (4) (2018) 3416–3452.
[6] L. Hornuf, P.P. Momtaz, R.J. Nam, Y. Yuan, Cybercrime on the ethereum blockchain, J. Bank. Fin. 175 (2025) 107419.
[7] REKT Database. (2024). . https://de.fi/rekt-database [Dataset].
[8] 2024 Crypto Crime Trends. (2024, January 18). Chainalysis. https://www.chainalysis.com/blog/2024-crypto-crime-report-introduction/.
[9] Cryptocurrency Fraud Report. (2024). Federal Bureau of Investigation. https://www.ic3.gov/AnnualReport/Reports/2023_IC3CryptocurrencyReport.pdf.
[10] Z. Shao, L. Zhang, S.A. Brown, T. Zhao, Understanding users' trust transfer mechanism in a blockchain-enabled platform: a mixed methods study, Decis. Support. Syst. 155 (2022) 113716.
[11] T. Wen, Y. Xiao, A. Wang, H. Wang, A novel hybrid feature fusion model for detecting phishing scam on Ethereum using deep neural network, Expert. Syst. Appl. 211 (2023) 118463.
[12] A. Abbasi, D. Dobolyi, A. Vance, F.M. Zahedi, The phishing funnel model: a design artifact to predict user susceptibility to phishing websites, Information Systems Research 32 (2) (2021) 410–436.
[13] A. Abbasi, F.M. Zahedi, D. Zeng, Y. Chen, H. Chen, J.F. Nunamaker Jr, Enhancing predictive analytics for anti-phishing by exploiting website genre information, Journal of Management Information Systems 31 (4) (2015) 109–157.
[14] A. Abbasi, Z. Zhang, D. Zimbra, H. Chen, J.F. Nunamaker Jr, Detecting fake websites: The contribution of statistical learning theory, MIS Quarterly (2010) 435–461.
[15] R.T. Wright, M.L. Jensen, J.B. Thatcher, M. Dinger, K. Marett, Research note—Influence techniques in phishing attacks: An examination of vulnerability and resistance, Information Systems Research 25 (2) (2014) 385–400.
[16] R.T. Wright, K. Marett, The influence of experiential and dispositional factors in phishing: An empirical investigation of the deceived, Journal of Management Information Systems 27 (1) (2010) 273–303.
[17] L. Chen, J. Peng, Y. Liu, J. Li, F. Xie, Z. Zheng, Phishing scams detection in Ethereum transaction network, ACM Transactions on Internet Technology (TOIT) 21 (1) (2020) 1–16.
[18] Wen, H., Fang, J., Wu, J., & Zheng, Z. (2021). Transaction-based hidden strategies against general phishing detection framework on Ethereum. 1–5.
[19] Yuan, Q., Huang, B., Zhang, J., Wu, J., Zhang, H., & Zhang, X. (2020). Detecting phishing scams on Ethereum based on transaction records. 1–5.
[20] Zhang, D., Chen, J., & Lu, X. (2021). Blockchain phishing scam detection via multi-channel graph classification. 241–256.
[21] L.W. Cong, Y. Xiao, Categories and functions of crypto-tokens, in: M. Pompella, R. Matousek (Eds.), The Palgrave Handbook of FinTech and Blockchain, Springer International Publishing, 2021, pp. 267–284.
[22] B. He, Y. Chen, Z. Chen, X. Hu, Y. Hu, L. Wu, R. Chang, H. Wang, Y. Zhou, TxPhishScope: Towards detecting and understanding transaction-based phishing on Ethereum, in: Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security, 2023, pp. 120–134.
[23] C. Fisch, Initial coin offerings (ICOs) to finance new ventures, J. Bus. Ventur. 34 (1) (2019) 1–22.
[24] S.T. Howell, M. Niessner, D. Yermack, Initial coin offerings: Financing growth with cryptocurrency token sales, Review of Financial Studies 33 (9) (2020) 3925–3974.
[25] C. Sun, P. Adamopoulos, A. Ghose, X. Luo, Predicting stages in omnichannel path to purchase: a deep learning model, Information Systems Research 33 (2) (2022) 429–445.
[26] A. Chen, Y. Lu, P.Y.K. Chau, S. Gupta, Classifying, Measuring, and Predicting Users' Overall Active Behavior on Social Networking Sites, Journal of Management Information Systems 31 (3) (2014) 213–253.
[27] Z. Gu, R. Bapna, J. Chan, A. Gupta, Measuring the impact of crowdsourcing features on mobile app user engagement and retention: a randomized field experiment, Management Science 68 (2) (2022) 1297–1329.
[28] Sangaralingam, K., Pervin, N., Ramasubbu, N., Datta, A., & Dutta, K. (2012). Takeoff and sustained success of apps in hypercompetitive mobile platform ecosystems: An empirical analysis. ICIS 2012 Proceedings.
[29] P. Monamo, V. Marivate, B. Twala, Unsupervised learning for robust Bitcoin fraud detection, in: 2016 Information Security for South Africa (ISSA), 2016, pp. 129–134.
[30] V. Van Vlasselaer, T. Eliassi-Rad, L. Akoglu, M. Snoeck, B. Baesens, GOTCHA! Network-based fraud detection for social security fraud, Management Science 63 (9) (2017) 3090–3110.
[31] J. Wang, M. Gupta, H.R. Rao, Insider Threats in a Financial Institution: Analysis of Attack-Proneness of Information Systems Applications, MIS Quarterly 39 (1) (2015) 91–112.
[32] H. Chitsazan, A. Bagheri, M. Tajeddin, Initial coin offerings (ICOs) success: Conceptualization, theories and systematic analysis of empirical studies, Technol. Forecast. Soc. Change 180 (2022) 121729.
[33] W. Xu, T. Wang, R. Chen, J.L. Zhao, Prediction of initial coin offering success based on team knowledge and expert evaluation, Decis. Support. Syst. 147 (2021) 113574.
[34] A. Murray, D. Kim, J. Combs, The promise of a decentralized internet: What is Web3 and how can firms prepare? Bus. Horiz. 66 (2) (2023) 191–202.
[35] J. Gan, G. Tsoukalas, S. Netessine, Initial coin offerings, speculation, and asset tokenization, Management Science 67 (2) (2021) 914–931.
[36] T. Dean, D.J. Daluwathumullagamage, A. Marsden, Predictability of ICO success and returns, Journal of Applied Business and Economics 22 (13) (2020) 20–36.
[37] S. Hägele, Centralized exchanges vs. decentralized exchanges in cryptocurrency markets: a systematic literature review, Electron. Mark. 34 (1) (2024) 33.
[38] H. Benedetti, L. Kostovetsky, Digital tulips? Returns to investors in initial coin offerings, Journal of Corporate Finance 66 (2021) 101786.
[39] C. Fisch, P.P. Momtaz, Institutional Investors and Post-ICO Performance: An Empirical Analysis of Investor Returns in Initial Coin Offerings (ICOs), Journal of Corporate Finance (2020).
[40] P.P. Momtaz, The pricing and performance of cryptocurrency, The European Journal of Finance 27 (4–5) (2021) 367–380.
[41] Hack Analysis: Nomad Bridge. (2023, January 11). Medium. https://medium.com/immunefi/hack-analysis-nomad-bridge-august-2022-5aa63d53814a.

[42] M. Bartoletti, S. Lande, A. Loddo, L. Pompianu, S. Serusi, Cryptocurrency scams: Analysis and perspectives, IEEe Access. 9 (2021) 148353–148373.
[43] M. Conti, A. Gangwal, S. Ruj, On the economic significance of ransomware campaigns: a Bitcoin transactions perspective, Comput. Secur. 79 (2018) 162–189.
[44] D.Y. Huang, M.M. Aliapoulios, V.G. Li, L. Invernizzi, E. Bursztein, K. McRoberts, J. Levin, K. Levchenko, A.C. Snoeren, D. McCoy, Tracking ransomware end-to-end, in: 2018 IEEE Symposium on Security and Privacy (SP), 2018, pp. 618–631.
[45] Gherghelas, S. (2024a, April 4). State of the Dapp Industry Q1 2024. Dapp Industry Reports. https://dappradar.com/blog/state-of-the-dapp-industry-q1-2024.
[46] L. Hornuf, T. Kück, A. Schwienbacher, Initial coin offerings, information disclosure, and fraud, Small Business Economics 58 (4) (2022) 1741–1759.
[47] M. La Morgia, A. Mei, F. Sassi, J. Stefa, The doge of wall street: Analysis and detection of pump and dump cryptocurrency manipulations, ACM. Trans. Internet. Technol. 23 (1) (2023) 1–28.
[48] P. Xia, H. Wang, B. Zhang, R. Ji, B. Gao, L. Wu, X. Luo, G. Xu, Characterizing cryptocurrency exchange scams, Comput. Secur. 98 (2020) 101993.
[49] Levy, A. (2017, July 17). Fraudsters just stole $7 million by hacking a cryptocoin offering. CNBC. https://www.cnbc.com/2017/07/17/coindash-website-hacked-7-million-stolen-in-ico.html.
[50] Chen, W., Guo, X., Chen, Z., Zheng, Z., & Lu, Y. (2020). Phishing scam detection on Ethereum: Towards financial security for blockchain ecosystem. The Twenty-Ninth International Joint Conference on Artificial Intelligence, 4506–4512.
[51] L. Wallace, M. Keil, A. Rai, Understanding software project risk: a cluster analysis, Information & Management 42 (1) (2004) 115–125.
[52] P. Han, Z. Yan, W. Ding, S. Fei, Z. Wan, A Survey on Cross-chain Technologies, Distributed Ledger Technologies: Research and Practice 2 (2) (2023) 1–30.
[53] J. Wu, Q. Yuan, D. Lin, W. You, W. Chen, C. Chen, Z. Zheng, Who are the phishers? Phishing scam detection on Ethereum via network embedding, IEEE Transactions on Systems, Man, and Cybernetics: Systems (2020).
[54] H. Huang, X. Zhang, J. Wang, C. Gao, X. Li, R. Zhu, Q. Ma, PEAE-GNN: Phishing detection on Ethereum via augmentation Ego-Graph based on graph neural network, IEEe Trans. Comput. Soc. Syst. (2024).
[55] N.A.G. Arachchilage, S. Love, A game design framework for avoiding phishing attacks, Comput. Human. Behav. 29 (3) (2013) 706–714.
[56] R.T. Wright, S.L. Johnson, B. Kitchens, Phishing susceptibility in context: a multilevel information processing perspective on deception detection, MIS Quarterly 47 (2) (2023).
[57] R. Chen, J. Wang, T. Herath, H.R. Rao, An investigation of email processing from a risky decision making perspective, Decis. Support. Syst. 52 (1) (2011) 73–81.
[58] R. Naidoo, A multi-level influence model of COVID-19 themed cybercrime, European Journal of Information Systems 29 (3) (2020) 306–321.
[59] J. Nicholls, A. Kuppa, N.A. Le-Khac, FraudLens: Graph Structural Learning for Bitcoin Illicit Activity Identification, in: Annual Computer Security Applications Conference, 2023, pp. 324–336.
[60] L. Nan, D. Tao, Bitcoin mixing detection using deep autoencoder, in: 2018 IEEE Third International Conference on Data Science in Cyberspace (DSC), 2018, pp. 280–287.
[61] W. Hamilton, Z. Ying, J. Leskovec, Inductive representation learning on large graphs, Adv. Neural Inf. Process. Syst. (2017) 30.
[62] Kipf, T. N., & Welling, M. (2017). Semi-supervised classification with graph convolutional networks (No.). arXiv. https://doi.org/10.48550/arXiv.1609.02907.
[63] A. Hevner, S. Chatterjee, A. Hevner, S. Chatterjee, Design science research in information systems, Design Research in Information Systems: Theory and Practice (2010) 9–22.
[64] A. Abbasi, C. Albrecht, A. Vance, J. Hansen, Metafraud: a meta-learning framework for detecting financial fraud, MIS Quarterly (2012) 1293–1327.
[65] J. Devlin, M.W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2018, pp. 4171–4186.
[66] F.F. Suarez, J. Kirtley, Dethroning an established platform, MIT. Sloan. Manage Rev. 53 (4) (2012) 35–41.
[67] von Briel, F., & Davidsson, P. (2019). Digital platforms and network effects: Using digital nudges for growth hacking. ICIS 2019 Proceedings.
[68] A.S. Hu, C.A. Parlour, U. Rajan, Cryptocurrencies: Stylized facts on a new investible instrument, Financ. Manage 48 (4) (2019) 1049–1068.
[69] D. Koutmos, Return and volatility spillovers among cryptocurrencies, Econ. Lett. 173 (2018) 122–127.
[70] Y. Sovbetov, Factors influencing cryptocurrency prices: Evidence from Bitcoin, Ethereum, Dash, Litcoin, and Monero, Journal of Economics and Financial Analysis 2 (2) (2018) 1–27.
[71] J. Wu, J. Liu, Y. Zhao, Z. Zheng, Analysis of cryptocurrency transactions from a network perspective: An overview, Journal of Network and Computer Applications 190 (2021) 103139.
[72] M.R. Abrigo, I. Love, Estimation of panel vector autoregression in Stata, Stata J. 16 (3) (2016) 778–804.
[73] D. Holtz-Eakin, W. Newey, H.S. Rosen, Estimating vector autoregressions with panel data, Econometrica: Journal of the Econometric Society (1988) 1371–1395.
[74] M. Petryk, L. Qiu, P. Pathak, The impact of open-source community on cryptocurrency market price: An empirical investigation, Journal of Management Information Systems 40 (4) (2023) 1237–1270.
[75] C.W.J. Granger, P. Newbold, Forecasting Economic Time Series, 2nd edition, Academic Press, 1986.
[76] J. Chen, T. Ma, C. Xiao, FastGCN: Fast learning with graph convolu-tional networks via importance sampling, in: International Conference on Learning Representations, 2018.
[77] R. Ying, R. He, K. Chen, P. Eksombatchai, W.L. Hamilton, J. Leskovec, Graph Convolutional Neural Networks for Web-Scale Recommender Systems, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018, pp. 974–983.
[78] Davis, J., & Goadrich, M. (2006). The relationship between Precision-Recall and ROC curves. 233–240.
[79] P. Branco, L. Torgo, R.P. Ribeiro, A survey of predictive modeling on imbalanced domains, ACM Computing Surveys (CSUR) 49 (2) (2016) 1–50.
[80] C.W. Granger, Investigating causal relations by econometric models and cross-spectral methods, Econometrica: Journal of the Econometric Society (1969) 424–438.
[81] M. Yang, G. Adomavicius, G. Burtch, Y. Ren, Mind the gap: Accounting for measurement error and misclassification in variables generated via data mining, Information Systems Research 29 (1) (2018) 4–24.
[82] L.E. Cohen, M. Felson, Social change and crime rate trends: a routine activity approach (1979). Classics in Environmental Criminology, Routledge, 2010, pp. 203–232.
[83] R. Beck, C. Müller-Bloch, J.L. King, Governance in the blockchain economy: a framework and research agenda, J. Assoc. Inf. Syst. 19 (10) (2018) 1.
[84] F. Hawlitschek, B. Notheisen, T. Teubner, The limits of trust-free systems: a literature review on blockchain technology and trust in the sharing economy, Electron. Commer. Res. Appl. 29 (2018) 50–63.
[85] L. Hughes, Y.K. Dwivedi, S.K. Misra, N.P. Rana, V. Raghavan, V. Akella, Blockchain research, practice and policy: Applications, benefits, limitations, emerging research themes and research agenda, Int. J. Inf. Manage 49 (2019) 114–129.
[86] P.P. Momtaz, CEO emotions and firm valuation in initial coin offerings: An artificial emotional intelligence approach, Strateg. Manage J. 42 (3) (2021) 558–578.
[87] Y. Fang, C. Zhang, C. Huang, L. Liu, Y. Yang, Phishing email detection using improved RCNN model with multilevel vectors and attention mechanism, IEEe Access. 7 (2019) 56329–56340.
[88] E.W. Ngai, Y. Hu, Y.H. Wong, Y. Chen, X. Sun, The application of data mining techniques in financial fraud detection: a classification framework and an academic review of literature, Decis. Support. Syst. 50 (3) (2011) 559–569.

Ziyi Xiong is currently an Assistant Professor of Information Systems in the College of Business at Kennesaw State University. Her research interests lie on blockchain, FinTech, machine learning and digital entrepreneurship.

Rong Liu is an associate professor in the College of Business at Florida International University. She has also worked as a research staff member at the IBM T. J. Watson Research Center and as an associate professor at Stevens Institute of Technology. She received her Ph.D. in Business Administration from Penn State University. Her research interests include deep learning, natural language processing, fintech, blockchain, and business process management. She has authored more than 60 journal and conference publications and holds 20 patents.

Dr. Hemang Subramanian is an Associate Professor of Information Systems & Business Analytics and the ATOM Blockchain Faculty director at FIU. His research interests include Blockchain system design, cryptocurrency price movements, and IT entrepreneurship. He has authored more than 15 articles in journals such as Information Systems Research, Organization Science, Communications of AIS, Communications of ACM, IEEE Software, JMIR, Journal of Database Management, Journal of Managerial Finance.