Why Is My Transaction Risky? Understanding Smart Contract Semantics and Interactions in the NFT Ecosystem

Yujing Chen[1], Xuanming Liu[1], Zhiyuan Wan[1*†], Zuobin Wang[1], David Lo[2], Difan Xie[3], Xiaohu Yang[1]
[1]*The State Key Laboratory of Blockchain and Data Security, Zhejiang University*
[2]*School of Computing and Information Systems, Singapore Management University*
[3]*Hangzhou High-Tech Zone (Binjiang) Institute of Blockchain and Data Security*
{chenyujing, hinsliu, wanzhiyuan, wangzuobin, yangxh}@zju.edu.cn,
davidlo@smu.edu.sg, xiedifan@bcds.org.cn

***Abstract*—The NFT ecosystem represents an interconnected, decentralized environment that encompasses the creation, distribution, and trading of Non-Fungible Tokens (NFTs), where key actors, such as marketplaces, sellers, and buyers, utilize smart contracts to facilitate secure, transparent, and trustless transactions. Scam tokens are deliberately created to mislead users and facilitate financial exploitation, posing significant risks in the NFT ecosystem. Prior work has explored the NFT ecosystem from various perspectives, including security challenges, actor behaviors, and risks from scams and wash trading, leaving a gap in understanding the semantics and interactions of smart contracts during transactions, and how the risks associated with scam tokens manifest in relation to the semantics and interactions of contracts. To bridge this gap, we conducted a large-scale empirical study on smart contract semantics and interactions in the NFT ecosystem, using a curated dataset of nearly 100 million transactions across 20 million blocks on Ethereum. We observe a limited semantic diversity among smart contracts in the NFT ecosystem, dominated by proxy, token, and DeFi contracts. Marketplace and proxy registry contracts are the most frequently involved in smart contract interactions during transactions, engaging with a broad spectrum of contracts in the ecosystem. Token contracts exhibit bytecode-level diversity, whereas scam tokens exhibit bytecode convergence. Certain interaction patterns between smart contracts are common to both risky and non-risky transactions, while others are predominantly associated with risky transactions. Based on our findings, we provide recommendations to mitigate risks in the blockchain ecosystem, and outline future research directions.**

***Index Terms*—Ethereum, Blockchain, Transaction, Smart Contract, Scam, NFT.**

I. Introduction

A Non-Fungible Token (NFT) is a digital certificate of ownership, immutably recorded on a blockchain like Ethereum. While NFTs are commonly associated with digital assets like images and videos, their application has expanded to physical assets, such as postage stamps [1], [2], gold [3], real estate [4], and tangible artwork [5], reflecting their growing popularity in diverse markets. In cryptocurrency, an NFT functions analogously to traditional proof-of-purchase mechanisms, such as invoices or receipts. What distinguishes NFTs is their inherent verifiability and their ability to enable trustless transactions [6]. Verifiability ensures transparent ownership transfer records on the blockchain, enabling clear provenance tracking. Moreover, NFT allows exchanges of digital assets without mutual trust, as both the asset transfer and the cryptocurrency payment are executed atomically within a single, secure transaction on the blockchain. In 2023, the trading volume of NFTs reached approximately 11.8 billion USD in cryptocurrency [7].

The NFT ecosystem is the interconnected, decentralized environment surrounding the creation, distribution, and trading of NFTs [8]. Key actors, such as marketplaces, sellers, buyers, and content creators, leverage blockchain transactions and decentralized applications (DApps) to facilitate secure, transparent, and trustless exchanges of NFTs. Practitioners utilize smart contracts to build DApps, automating relevant processes like NFT creation and transfer in marketplaces [9]. Consequently, smart contracts interact with one another to facilitate NFT transactions. *Scam tokens* are fraudulent or deceptive cryptocurrency assets, implemented as smart contracts, deliberately created to mislead users and facilitate financial exploitation [10]. Scam tokens can typically be classified into three categories: 1) *Rugpull* tokens [11], [12] refer to malicious tokens whose creators deliberately withdraw liquidity or abandon corresponding projects after attracting user investment, leaving holders with worthless assets; 2) *Honeypot* tokens [13] lure users into purchasing assets that appear tradable but cannot be resold due to restrictive conditions embedded in token contracts, such as excessive transaction fees, blacklists, or non-standard token logic; 3) *Ponzi* tokens [14] sustain their value by relying on the continuous inflow of funds from new investors to pay returns to earlier ones, with the scheme collapsing when the inflow of new investment slows, resulting in significant losses for the majority of holders. In 2023, scam tokens caused financial losses of 5.6 billion USD [15], posing significant risks in the blockchain ecosystem.

Previous studies have explored the NFT ecosystem from various perspectives, including security challenges [8], user activities on prominent NFT marketplaces [16], and on-chain

*Zhiyuan Wan is the corresponding author.
†Also with Hangzhou High-Tech Zone (Binjiang) Institute of Blockchain and Data Security.

2643-1572/25/$31.00 ©2025 IEEE
DOI 10.1109/ASE63991.2025.00172

behaviors of NFTs [17]. Despite these efforts, a critical gap remains in understanding the semantics of smart contracts in the ecosystem, and how they interact during transactions. Further research has investigated specific risks in the NFT ecosystem, such as risks from rug pulls [18], wash trading [19], [20], promotion scams [21], and phishing scams [22]. However, how the risks associated with scam tokens manifest in relation to the semantics and interactions of smart contracts remain unclear. An in-depth understanding of smart contract semantics and interactions could provide valuable insights into the development practices of smart contracts, as well as inform the design of strategies and tools for detecting and mitigating risks in the blockchain ecosystem.

To address the gaps, we conducted a large-scale empirical study to explore the semantics and interactions of smart contracts in NFT transactions, using a curated dataset comprising nearly 100 million NFT transactions distributed across 20 million blocks on the Ethereum blockchain. We investigated the following research questions:

RQ1. What are the smart contracts involved in the transactions of the NFT ecosystem?

To understand the semantics of smart contracts in the ecosystem, we grouped smart contracts involved in NFT transactions into distinct semantic clusters with respect to their bytecode and source code, and explored the evolution of the number of deployed smart contracts across these semantic clusters over time. We identified 2,737 semantic clusters of smart contracts in the NFT ecosystem. Nonetheless, the semantic diversity of smart contracts in the ecosystem is limited, as the top 50 largest clusters account for 84.9% of the total smart contracts. Proxy, token, and DeFi contracts dominate the top 10 largest semantic clusters. Notably, minimal proxy contracts have experienced widespread adoption over an extended period, while financial utility contracts, due to their persistent presence, serve as a foundational layer in the long-term infrastructure of the ecosystem.

RQ2. How do the smart contracts interact during NFT transactions?

To capture the interactions among smart contracts during transactions, we measured the frequency and complexity of interactions across the semantic clusters of smart contracts in the NFT ecosystem. We observed that the semantic clusters with the most frequent interactions during transactions primarily comprise smart contracts related to marketplaces, proxy registry, and proxy. In the meantime, the transactions tend to involve a limited number of interactions among smart contracts, with a median of four contracts per transaction. Moreover, the top ten most frequent interaction patterns among smart contracts in transactions exhibit varying levels of complexity, typically associated with specific token operations and business processes.

RQ3. How do scam token risks manifest with respect to the semantics and interactions of smart contracts during NFT transactions?

To answer RQ3, we conducted a comparison of the bytecode of scam and non-scam tokens, and characterized the interactions of smart contracts in the risky transactions that involve scam tokens. We found that token contracts demonstrate considerable bytecode-level diversity, with scam tokens exhibiting notable bytecode convergence. Certain frequently observed interaction patterns between smart contracts are prevalent in both risky and non-risky transactions, while others are predominantly associated with risky transactions, characterized by isolated surges in transaction volumes over distinct short intervals from April 2018 to July 2024.

Based on the findings, we discuss the implications and provide recommendations for mitigating risks in the NFT ecosystem. In addition, we outline several research avenues, including real-time monitoring of proxy contracts, and the integration of code- and interaction-level features of smart contracts to enhance fraud detection and transaction risk assessment. This paper makes the following contributions:

- We present the first large-scale empirical study of smart contract semantics and interactions in the NFT ecosystem on Ethereum, and how the risks associated with scam tokens manifest with respect to the semantics and interactions of smart contracts during transactions.
- We curate a dataset that comprises 99,212,864 and 148,411,324 external and internal transactions in the NFT ecosystem, respectively, as well as 225,350 ERC721 contracts for future investigation by others.
- We provide practical recommendations for mitigating risks in the NFT ecosystem, and outline avenues of future research.

Our replication package is available online: https://doi.org/10.5281/zenodo.15550314.

II. Background

EVM-Based Blockchains. The Ethereum Virtual Machine (EVM) serves as the computational backbone of Ethereum and other EVM-compatible blockchains, enabling the execution of smart contracts in a decentralized and trustless manner. Designed as a Turing-complete virtual machine, the EVM allows developers to deploy and execute self-executing code written in programming languages such as Solidity and Vyper, which are compiled into EVM bytecode. The EVM has been widely adopted by numerous blockchain platforms, including Binance Smart Chain, Polygon, Avalanche C-Chain, and Fantom.

EVM-based blockchains operate under an account-based model, distinguishing between *externally owned accounts* (EOAs), which are controlled by users, and *contract accounts* (CAs), which are governed by smart contracts. Transactions on an EVM-based blockchain modify the global state of the blockchain, which is maintained and validated by a distributed network of nodes operating under Proof-of-Work or Proof-of-Stake consensus mechanisms.

Ethereum Tokens. Ethereum supports the creation and management of fungible and non-fungible assets through standardized token protocols, which are implemented as smart contracts that define and enforce token issuance, transfer, and interaction rules. The most widely adopted Ethereum token protocols include ERC20, which facilitates fungible token

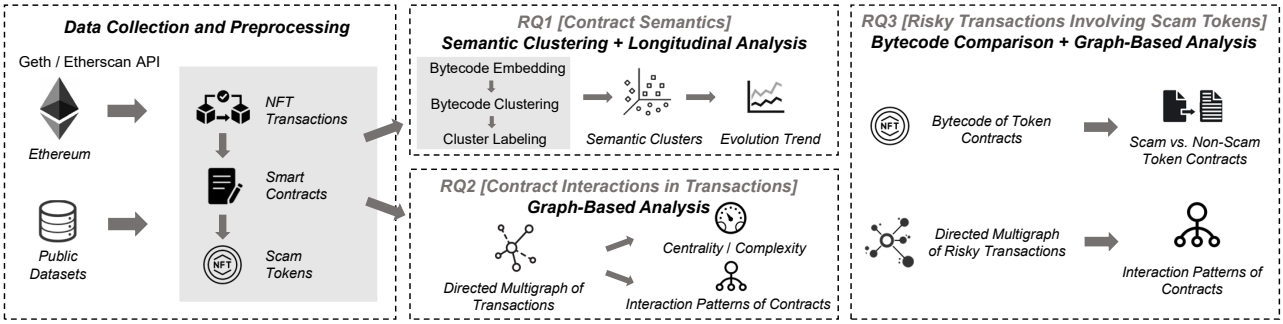


Fig. 1: Overview of research methodology.

implementations for DeFi and utility tokens, and ERC721, which enables the creation of unique, non-fungible tokens (NFTs) for digital ownership applications. Ethereum tokens play a foundational role in tokenized economies, powering DeFi protocols, NFT marketplaces, governance mechanisms, and cross-chain interoperability.

External and Internal Transactions on Ethereum. Ethereum, as an EVM-based blockchain, facilitates transactions that encompass token transfers, contract deployment, and execution, which can be broadly classified into *external* and *internal* transactions [23]. External transactions originate from EOAs, and are uniquely identified by transaction hashes as recorded on chain. EOAs serve as the entry point for the execution of external transactions, enabling the invocation of smart contracts or the transfer of Ether. In contrast, internal transactions, commonly referred to as the *message calls* in the execution model of Ethereum, occur during the execution of smart contracts. Unlike external transactions, internal transactions are not explicitly recorded on-chain as standalone entities. Instead, reconstructing internal transactions requires the execution traces that include contract invocations (e.g., via `CALL` or `DELEGATECALL` opcode) and *event logs* (e.g., ERC20 or ERC721 `Transfer` event) which serves to emit structured information that helps track and identify specific contract activities.

III. Methodology

This section outlines the data we collected, and the methodology used to address the research questions, as depicted in Fig. 1.

A. Data Collection and Preprocessing

NFT Transactions. We synchronized Ethereum mainnet blocks up to June 1, 2024, using Geth [24], from block 0 to block 20,000,000. From the event logs recorded in the synchronized blocks, we filtered out ERC721 `Transfer` events and extracted the corresponding external transactions. For each external transaction, we collected the transaction hash, the EOA address that initiated it, and the CA address invoked by the EOA. As a result, we collected 99,212,864 external transactions. To further collect relevant internal transactions, we located 148,411,324 internal transactions from the XBlock-ETH dataset [25] associated with the collected external transactions. For each internal transaction, we extracted their timestamp, as well as the `from` and `to` addresses representing from where and to which account the transaction was sent. Additionally, we extracted the execution order between internal transactions.

Smart Contracts. Based on the 5,965,248 addresses of CAs involved in our NFT transaction data, we further retrieved their corresponding bytecodes from the XBlock-ETH dataset. We also collected the source code of the CAs whenever available via the API provided by Etherscan [26]. As a result, we collected the source code for 5,108,420 CAs, while the remaining 856,828 were not open-sourced.

Scam Tokens. Starting from a public dataset of scam tokens on Ethereum in a recent study [10], we labeled the 225,350 ERC721 contracts involved in our transaction data. Specifically, we labeled 3,144 ERC721 contracts as scam tokens, including 2,600 rugpull, 392 honeypot, and 152 Ponzi tokens.

B. Data Analysis

We seek to answer the following research questions:

RQ1. What are the smart contracts involved in the transactions of the NFT ecosystem?

To understand the semantics of the smart contracts in the NFT ecosystem, we first captured the *linguistic topics* [27] of smart contracts involved in the NFT-relevant transactions, which reveal the intention of code in smart contracts. Specifically, we analyzed both bytecode and source code (wherever available) of smart contracts by following three steps: 1) ***Bytecode Embedding***. We began with the bytecodes of the 5,965,248 smart contracts in our dataset, aiming to capture bytecode embeddings representing smart contracts. Initially, we identified duplicate smart contracts with identical bytecodes, leaving 213,625 unique smart contracts. We then disassembled each of the 213,625 smart contract bytecodes into a sequence of EVM opcodes using Go Ethereum v1.14.7 [24]. With opcode sequences as input, we further applied the BGE embedding model [28] to learning bytecode representation for each smart contract, outputting a 1024-dimensional representation vector. The BGE embedding model has demonstrated its efficacy in generating embeddings for subsequent clustering and classification tasks in recent studies (e.g., [29], [30], [31]). 2) ***Bytecode Clustering***. To group smart contract bytecodes,

we used the non-parametric clustering algorithm HDBSCAN [32], which has demonstrated its efficacy for code clustering in recent studies (e.g., [33], [34]). HDBSCAN incorporates the idea of hierarchical clustering, thus can automatically select an appropriate density threshold. Specifically, we employed the implementation of the HDBSCAN algorithm from version 0.8.40 of the `hdbscan` library [35]. The Silhouette score [36] is a widely used metric for evaluating the quality of clustering, quantifying how well each data point fits within its assigned cluster by comparing intra-cluster cohesion and inter-cluster separation, with higher values indicating more distinct and well-separated clusters. To optimize the Silhouette score, we explored multiple combinations of the parameters `min_cluster_size` and `min_samples`, ultimately selecting a value of five for both parameters. The remaining parameters are set to their default values. As a result, out of the 5,965,248 smart contract bytecodes, we identified 2,737 clusters encompassing 5,478,013 smart contracts, with 525,262 contracts (8.8%) remaining unclustered. The Silhouette score of 0.827, which is close to the maximum score of 1.0, indicates that each cluster is tightly grouped and well separated from the other cluster. 3) ***Cluster Labeling***. Through bytecode clustering, we partitioned the smart contracts based on their bytecode embeddings into 2,737 clusters. For each cluster i, we assigned a label by utilizing the *linguistic topic* derived from the available source code associated with the bytecodes in that cluster. Specifically, we first retrieved the source code file(s) associated with all bytecodes within each cluster. Next, following prior work [27], we constructed a term-document matrix A_i by extracting the vocabulary from the source code of each cluster. In the term-document matrix A_i, each cluster is represented by a vector of term occurrences, where terms correspond to words appearing in the source code of the cluster. We then calculated the relevance of term t_j to the current cluster i using the formula:

$$\text{rel}(t_j, i) = \text{sim}(t_j, A_i) - \frac{1}{|\mathcal{A}|} \sum_{A_k \in \mathcal{A}} \text{sim}(t_j, A_k) \qquad (1)$$

Consequently, we used the top-n list of the most relevant terms from cluster i as its linguistic topic as a reference for labeling. We then randomly selected smart contracts with their source code available from each cluster, and scrutinized the source code to infer the underlying semantics of the smart contracts in each cluster. For the 428 clusters (15.71%) lacking open-sourced smart contracts, we referred to the information provided for the smart contracts on Etherscan.

Moreover, we investigated the evolution of the numbers of deployed smart contracts across semantic clusters over time. Specifically, we aggregated the smart contracts based on their deployment timestamps in three-month intervals and categorized them according to the corresponding semantic clusters.

RQ2. How do the smart contracts interact during NFT transactions?

To capture the interactions among smart contracts during NFT transactions, we introduced a directed multigraph with attributed nodes and edges, denoted as $G = (V, E, \mathbf{X}_V, \mathbf{X}_E)$. The directed multigraph G consists of (i) a node set V that represents CAs and EOAs on Ethereum associated with NFT transactions, (ii) a set of directed edges $E = \{(u, v)\}$ that represents invocations from EOAs to CAs and invocations among CAs during the transactions, (iii) a feature vector $x_v \in \mathbf{X}_V$ for each node $v \in V$ to represent node attributes, and (iv) a feature vector $x_e \in \mathbf{X}_E$ for each edge $e \in E$ to represent edge attributes.

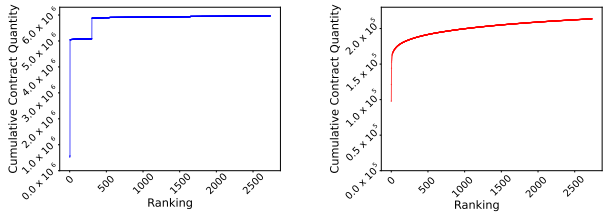
Given a collection of transactions, we build the directed multigraph as follows. For each transaction, the involved Ethereum accounts, including both EOAs and CAs, constitute the nodes $v \in V$, while the invocations between these accounts during the transactions represent the directed edges $e \in E$ in the multigraph. We then leveraged the feature vectors $\mathbf{X}_V$ to capture the semantic characteristics of the nodes. Specifically, for each node representing CA, the feature vector $x_v \in \mathbf{X}_V$ captures its corresponding bytecode embedding, as well as the cluster label, as identified in RQ1. Furthermore, we utilized the feature vectors $\mathbf{X}_E$ of edges to capture transaction-level features. Specifically, for each edge e representing an invocation in a transaction, the feature vector $x_e \in \mathbf{X}_E$ captures transaction hash, transaction timestamp, and its position in the invocation sequence during the transaction.

Based on the constructed directed multigraph, we performed an in-depth analysis of the interactions among smart contracts in NFT transactions. Specifically, we first aggregated the nodes (smart contracts) in the multigraph based on the semantic clusters identified in RQ1, and then measured the centrality of these semantic clusters in the interactions of smart contracts during transactions. We further measured the complexity of interactions between smart contracts in the transactions, and employed frequent subgraph mining to extract interaction patterns of smart contracts in the NFT ecosystem.

RQ3. How do scam token risks manifest with respect to the semantics and interactions of smart contracts during NFT transactions?

To answer RQ3, we started with clustering the 255,350 token contracts in our dataset based on their bytecodes, including 3,144 scam tokens. Specifically, like RQ1, we first employed the BGE embedding model to derive bytecode representations for each token contract, followed by the application of HDBSCAN to group token contracts with respect to their bytecodes. We then compared the distributions of scam and non-scam tokens across the resulting bytecode clusters of token contracts.

Subsequently, we characterized the interactions of smart contracts during risky transactions involving scam tokens. Specifically, we first sampled subgraphs that represent transactions involving scam tokens from the directed multigraph constructed in RQ2, serving as the risky transactions in the NFT ecosystem. We then applied frequent subgraph mining to extract interaction patterns of smart contracts in these risky transactions. Additionally, we analyzed the evolution of transaction volumes associated with the extracted interaction patterns of smart contracts in risky transactions.



(a) Total (b) Deduplicated

Fig. 2: Cumulative number of smart contracts across semantic clusters in the NFT ecosystem.

IV. Results

A. Contract Semantics in the NFT Ecosystem (RQ1)

Figure 2 illustrates the cumulative distributions of numbers of both (a) total and (b) deduplicated smart contracts per cluster, with clusters ordered by their deduplicated sizes and unclustered smart contracts as an independent group. Among the 2,737 clusters of smart contracts, the top 10 semantic clusters cover over 30% of the deduplicated contracts but more than 75% of the total contracts, indicating that most contracts concentrate on a few dominant semantic patterns while the remainder exhibit semantic diversity in the NFT ecosystem. Furthermore, as illustrated in Figure 2a, the cumulative number of smart contracts exhibits two sharp increases upon the inclusion of the clusters ranked 4 and 303, suggesting extensive code duplication within these clusters, which is consistent with the high code clone ratios in smart contracts on Ethereum reported in previous studies [37].

We further conducted an in-depth analysis of the top ten clusters containing the largest number of unique smart contracts in the NFT ecosystem, situated at the leftmost end of the CDF curve. Table I provides a summary of the top ten clusters, which account for 30.4% of the smart contracts in our dataset. Out of the ten clusters, two do not provide accessible source code (clusters rank 6 and 8), thereby preventing the derivation of corresponding semantic topics from source code (see the Appendix [38] for detailed linguistic topics). Among the top ten clusters of smart contracts, four predominantly comprise proxy contracts, three are associated with DeFi protocols, and the remaining three correspond to NFT assets and protocols. We made the following observations, with example contracts provided in the Appendix [38]:

EIP-1967 Proxy Contracts define a standard for proxy contract storage slots, which specifies how to store the address of the implementation of a contract in a predictable and secure way when using proxy contracts:

- Manifold EIP-1967 Proxy Contracts (❶ and ❸). The two clusters both comprise EIP-1967 proxy contracts provided by the *Manifold* protocol [39]. *Manifold* provides toolkits that enable digital creators to mint and manage NFT collections, as well as configurable widgets, APIs, and frameworks that facilitate developers to build NFT experiences [40]. The semantic distinction between the two smart contracts primarily lies in their delegation methods. Specifically, the smart contract from Cluster ❶ uses `Address.functionDelegateCall` with a security check that verifies the target contract address before executing the delegate call. In contrast, the smart contract from Cluster ❸ uses `delegatecall`, which bypasses such a security check.
- Generic EIP-1967 Proxy Contracts (❺). The contracts in Cluster 5 also adhere to the EIP-1967 standard. Meanwhile, they offer greater flexibility by allowing the logic contract address and initialization parameters to be specified as constructor arguments during deployment.

EIP-1167 Proxy Contracts (❹). EIP-1167 defines a minimal proxy contract, which provides a standardized way to deploy lightweight contracts that delegate their execution to another contract, often referred to as the *master* contract [41]. The EIP-1167 proxy contracts, with minimal bytecode, contain the logic to delegate calls to existing contracts, thus forwarding any transactions or function calls to the corresponding master contracts. The cluster consists of 4,463,176 contracts, with 1,867 unique bytecode, indicating extensive code cloning related to EIP-1967 proxy contracts in the NFT ecosystem.

NFT Contracts define and manage NFTs, representing ownership of items such as art, collectibles, music, or in-game items:

- JPEG NFT Contracts (❻). JPEG mining [42] refers to a type of NFT creation in which actual image data is directly uploaded to the blockchain, contrasting with conventional NFTs that typically store only metadata on-chain. During the minting process, miners upload the image data and receive an NFT in return. Contracts in this cluster are generated as part of the JPEG mining process and include information related to the image data of the NFT.
- ERC721 NFT Contracts (❾). The cluster comprises NFT contracts built upon the implementation of the ERC721 standard by OpenZeppelin [43], which provides essential functionalities such as token minting and transfer, as well as extended features, including metadata support and approval mechanisms. ERC721 is the most widely used standard for representing NFTs on the Ethereum blockchain. Each ERC721 token is inherently unique, capable of carrying distinct meta-

TABLE I: Top ten semantic clusters with the most smart contracts in the NFT ecosystem.

Label	# Deduplicated Contracts	Total (%)
❶ Manifold EIP-1967 Proxy A	23,425	0.39%
❷ Uniswap v3 Liquidity Pool	22,864	0.38%
❸ Manifold EIP-1967 Proxy B	8,592	0.14%
❹ EIP-1167 Proxy	1,867	74.82%
❺ Generic EIP-1967 Proxy	1,523	0.02%
❻ JPEG NFT	1,496	0.03%
❼ ERC721A NFT	1,387	0.02%
❽ Vault	1,255	0.02%
❾ ERC721 NFT	1,054	0.02%
❿ Financial Utility	760	0.01%

data, ownership, and provenance.

- ERC721A NFT CONTRACTS (❼). As the NFT ecosystem continues to expand, an increasing number of projects demand scalable mechanisms for large-scale token issuance. However, the standard ERC721 implementation incurs substantial gas costs when minting tokens in batches. To address this limitation, the ERC721A standard [44] was introduced as a gas-efficient alternative, specifically designed to optimize batch minting while remaining interface-compatible with ERC721. The cluster consists of NFT contracts adopting ERC721A, which implement efficient batch minting along with additional optimizations.

DeFi Contracts enable decentralized financial services without reliance on traditional intermediaries, extending the utility and financialization of NFTs:

- UNISWAP V3 LIQUIDITY POOL CONTRACTS (❷). Uniswap v3 [45] introduces an interaction model with NFTs through the concept of concentrated liquidity, wherein each liquidity position is represented as an NFT. Unlike previous versions of Uniswap, where liquidity providers (LP) received fungible LP tokens, Uniswap v3 assigns an ERC721-compliant NFT to each liquidity position, encoding the owner's specific parameters such as token pair, fee tier, and price range. This design enables more granular control over capital allocation and turns each position into a composable, tradable asset on-chain. As a result, these position NFTs can be transferred, collateralized in DeFi protocols, or integrated into broader NFT marketplaces and financial infrastructure.
- VAULT CONTRACTS (❽). ERC721 vault contracts, instantiated via the `mint` function of a `VaultFactory` contract, are designed to custody individual NFTs and issue ERC20 tokens representing fractional ownership. These vaults enable decentralized management and trading of fractionalized NFT assets.
- FINANCIAL UTILITY CONTRACTS FOR NFTS (❿). Financial smart contracts involve NFT-backed lending, collateral management, and other financial functionalities. Contracts in this cluster implement mechanisms for interest and debt calculation, loan issuance and repayment, as well as auction-based liquidations and collateral redistribution.

Figure 3 illustrates the quarterly deployment dynamics of the top ten contract clusters in the NFT ecosystem. Cluster ❹, primarily consisting of minimal proxy contracts, dominates in both longevity and volume, reflecting the widespread adoption of cost-efficient, scalable deployment patterns in the NFT ecosystem. Cluster ❷, representing Uniswap V3 Liquidity Pool contracts, reached 13,806 deployments between January 2023 and June 2024, whereas standard NFT contracts such as ERC721A and ERC721 (Clusters ❼ and ❾) appeared only 106 times, reflecting a broader trend of increasing integration between NFTs and DeFi that enables novel use cases. In contrast, Cluster ❿, encompassing financial utilities for NFTs such as lending and auction contracts, maintains a relatively low but stable deployment rate, indicating its role as a long-term infrastructure layer rather than a trend-driven component.

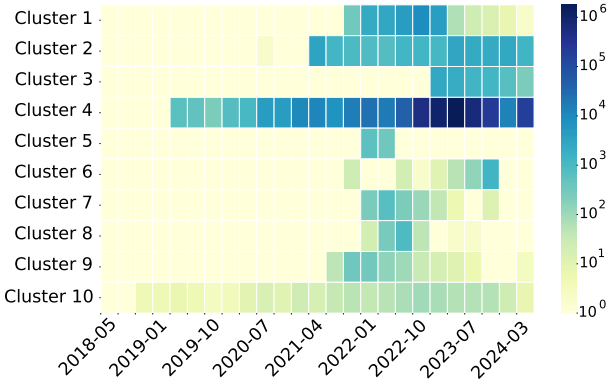


Fig. 3: Evolution in the numbers of deployed smart contracts across the top ten semantic clusters.

We do not observe any deployments of the ten semantic clusters of smart contracts prior to April 2018.

> **Finding 1:** The semantic diversity of smart contracts in the NFT ecosystem is limited, with the top 50 largest clusters comprising 80.2% and 84.9% of the deduplicated and total smart contracts, respectively. Proxy, token, and DeFi contracts dominate the top 10 largest semantic clusters. Notably, minimal proxy contracts have seen widespread adoption over a prolonged period, while financial utility contracts, with their sustained presence, function as a crucial layer in the long-term infrastructure of the ecosystem.

B. Contract Interactions in NFT Transactions (RQ2)

In this section, we present the results of our analysis of the interactions between smart contracts in NFT transactions, based on the constructed directed multigraph.

1) Centrality of Smart Contracts in Interactions: Table II summarizes the top five semantic clusters of smart contracts in the NFT ecosystem with the highest degree in the directed graph, including two clusters associated with NFT marketplaces, two related to proxy registries, and one dedicated to proxies (see the Appendix [38] for detailed linguistic topics). The two clusters associated with NFT marketplaces server distinct functions: Smart contracts in MARKETPLACE A are primarily concerned with high-level operations related to NFT marketplace activities, such as managing sales, pricing strategies, and user roles. In contrast, MARKETPLACE B focuses on the technical infrastructure necessary for processing offers and executing orders, including data handling, execution flows, and error management mechanisms. In the case of the proxy registry-related clusters, which function as centralized registries for proxy contracts, the smart contracts in PROXY REGISTRY A handle sophisticated state management and execution of proxy operations based on specific conditions. Meanwhile, PROXY REGISTRY B focuses on the core functionality of proxies, including delegated calls and the management of ownership and access control for proxy

TABLE II: Top five semantic clusters of smart contracts with the highest degree of interaction occurring during transactions.

Label	Degree	# Contracts
MARKETPLACE A	66,092,098	29
MARKETPLACE B	56,193,686	7
PROXY REGISTRY A	48,231,893	19
PROXY	47,640,194	795,872
PROXY REGISTRY B	45,361,729	5

contracts. In addition, the smart contracts in the PROXY cluster are primarily concerned with the technical aspects of implementing proxy patterns, focusing on the upgradeability and versioning of smart contracts.

We also observed that the smart contracts that remain unassigned to any semantic clusters have a cumulative degree of 306,999,918 in the directed multigraph. The top three unclustered contracts by degree are *OpenSea: Conduit*, *ENS: Old ETH Registrar Controller*, and *ENS: Registry with Fallback*. The high degrees of these contracts indicate their frequent interactions with other smart contracts in the NFT ecosystem, suggesting the active and prominent role of OpenSea and Ethereum Name Service (ENS) in the NFT ecosystem.

Finding 2: The semantic clusters with the highest degrees of interaction during transactions in the NFT ecosystem are primarily composed of smart contracts related to NFT marketplaces, proxy registries, and proxies. Furthermore, unclustered contracts associated with OpenSea and Ethereum Name Service also demonstrate frequent interactions during transactions.

2) Interaction Statistics and Patterns: According to the statistics derived from the directed multigraph, the median number of interacting smart contracts per transaction is 3 (min: 1, mean: 5.1, max: 603), while the median number of interactions between these contracts is 4 (min: 1, mean: 7.5, max: 12,903). On one hand, the relatively low median values indicate that, on average, transactions in the NFT ecosystem typically involve limited interactions between smart contracts. On the other hand, the wide ranges in both the number of contracts involved (from 1 to 603) and the frequency of interactions (ranging from 1 to 12,903) suggests that a subset of transactions might involve more complex and intricate interactions between contracts.

Moreover, we identified 771,839 distinct interaction patterns between smart contracts in NFT transactions, with the ten most frequently observed patterns depicted in Figure 4a. We made the following observations:

☛ **Pattern 1: Minimal Interactions (35.1%)** Pattern 1 demonstrates the direct interaction between EOAs and NFT contracts, occurring in over 34 million transactions (35.1%) in our dataset. Notably, 59.6% (1357 out of 2276) of the identified semantic clusters of smart contracts in RQ1 are involved in these interactions.

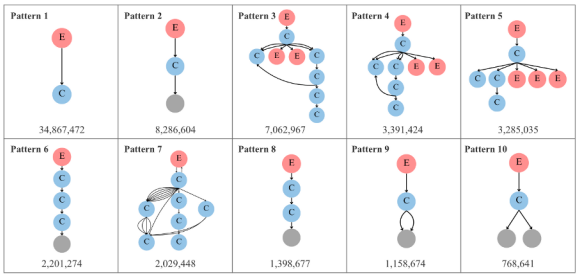
☛ **Pattern 2 and 8: Chained Interactions (8.4% and 1.4%)** Pattern 2 and 8 are observed in three types of transactions: 1) an EOA initiates the first NFT smart contract that implements certain extended functionalities, which subsequently calls additional smart contract(s) from distinct semantic clusters; 2) an EOA interacts with a proxy initially, which then delegates the call to an NFT contract, and in some cases, triggers further calls to other contracts; 3) an EOA initially interacts with smart contracts that handle the logic for NFT management, such as NFT auction and NFT staking, ultimately triggering NFT transfers.

☛ **Pattern 3, 4 and 5: Interactions via Marketplace (7.1%, 3.4% and 3.3%)** As the leading NFT marketplace, *OpenSea* has launched two representative protocols for decentralized trading of NFTs, *Wyvern* [46] in 2017 and *Seaport* [47] in 2022. Patterns 3 and 4 depict frequent interactions between smart contracts in NFT transactions utilizing the *Wyvern* protocol, while Pattern 5 illustrates interactions in transactions involving the *Seaport* protocol. A comparison of these patterns suggests that transactions utilizing the *Seaport* protocol tend to be more streamlined, characterized by shorter call chains in smart contracts, and exhibit greater flexibility, particularly with multi-asset transactions that involve multiple users, as compared to the *Wyvern* protocol.

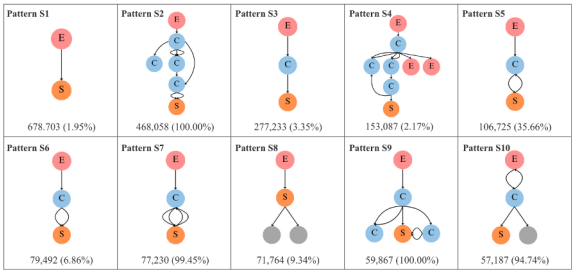
☛ **Pattern 6: *Gnosis Safe* Involved Interactions (2.2%)** *Gnosis Safe* is a platform that offers a Smart Account solution for Ethereum, which provides multi-signature wallets that require multiple parties to approve transactions, thereby enhancing security for users to manage digital assets [48]. Pattern 6 frequently occurs in the transactions involving *Gnosis Safe* where the first interacted contract functions as a proxy, and the second contract, which serves as the proxy's logic contract, primarily consists of NFT contracts with diverse extend features. The third invoked contract provides methods for the creation and interaction with proxy contracts within the *Gnosis Safe* ecosystem, while the fourth contract implements the core functionality of Gnosis Safe. In these transactions, users typically purchase NFTs and transfer payment in Ether through a proxy, while the NFT contract subsequently sends a portion of the payment received from the users to the *Gnosis* proxy of a specific platform to cover the fee.

☛ **Pattern 7: Interactions via ENS (2.1%)** Pattern 7 is frequently observed in transactions involving the Ethereum Name Service (ENS). A typical example of this pattern involves an EOA initiating a transaction by invoking the *ETH Registrar Controller* contract, which subsequently interacts with several ENS-related smart contracts, including: 1) the *Base Registrar Implementation* contract, which facilitates the registration of Ethereum-based domain names; 2) the *Registry with Fallback* contract, representing the ENS registry with an integrated fallback mechanism; and 3) the *Public Resolver* contract, which resolves ENS domains to various associated resources.

☛ **Pattern 9: Repeated Interactions (1.2%)** Pattern 9 is frequently observed in interactions involving *unclustered* smart contracts in NFT transactions. As Pattern 9 illustrates, a smart



(a) Throughout all NFT transactions



(b) In transactions involving scam tokens (the corresponding *risky transaction ratios* are indicated in parentheses)

Fig. 4: Top ten most frequent interaction patterns between smart contracts in NFT transactions, where EOAs, CAs, and scam tokens are denoted as E, C, and S, respectively. Grey nodes represent either CAs or EOAs.

contract repeatedly invokes another contract in a transaction, typically for purposes such as performing NFT batch operations, confirming and validating states before and after NFT transfers, or setting the state of NFTs.

👉 **Pattern 10: Branching Interactions (0.8%)** Pattern 10 is frequently observed in transactions involving ether transfers to EOAs, such as those related to the purchase of NFTs, payment to the buyer, or payment of platform fees. For example, in a transaction [49] associated with an NFT auction, the auction contract first transfers the Ether received from the buyer to the seller, and then calls the NFT contract to facilitate the transfer of the token to the buyer.

> **Finding 3:** NFT transactions typically involve a limited number of interactions between smart contracts, with a median of four contracts per transaction. Among the 771,839 distinct interaction patterns identified between smart contracts in NFT transactions, the top ten most frequent patterns exhibit varying levels of complexity and are associated with specific NFT operations and business processes.

C. Risky Transactions Involving Scam Tokens (RQ3)

This section presents the results of our comparison between the bytecodes of scam and non-scam tokens in the NFT ecosystem, along with the interaction patterns between smart contracts in transactions involving scam tokens.

1) Scam vs. Non-Scam Tokens: We identified 1,350 distinct clusters from the bytecodes of 255,350 ERC721 smart contracts (i.e., scam and non-scam tokens) in the NFT ecosystem. The bytecode clusters of these token contracts vary significantly in size, ranging from 5 to 97,319. Scam tokens span 45 out of the 1,350 bytecode clusters, with varying ratios across the clusters. Table III shows the distribution of the 45 clusters with scam tokens across various intervals of scam token ratios. Specifically, among the 45 clusters, 20 clusters consist entirely of scam tokens (100%), while an additional 18 clusters have a high scam token ratio between 60% and 100%. Together, these two categories of clusters account for 2,326 scam tokens, and represent the majority of scam tokens in our dataset. In contrast, two clusters exhibit a moderate scam token ratio between 40% and 60%, comprising 93 scam tokens. Meanwhile, five clusters fall within the [0%, 40%) range, with 714 out of 725 scam tokens in this group being unclustered. This distribution suggests that scam tokens are not randomly dispersed but tend to concentrate within a limited number of semantically similar clusters, reflecting strong code-level homogeneity among scam tokens.

We further analyzed the available source code of representative bytecode clusters across different scam token ratio intervals, and derived the following observations:

- **Clusters with 100% Scam Tokens.** We selected a representative cluster containing 51 scam tokens, all of which were labeled as rugpull tokens. As exemplified by the leftoverz [50] contract, the scam tokens in this cluster typically extend the ERC721 standard by incorporating two additional features: (i) a *whitelist* mechanism that facilitates NFT airdrops to a designated group of users, and (ii) a *payment splitter* component that allocates and distributes payments among multiple recipients according to predefined shares.
- **Clusters with [60%,100%) Scam Tokens.** We selected a representative cluster containing 1,426 token contracts, with 96% of which being labeled as scam tokens. As exemplified by the *EtherElephants* [51] contract, the token contracts in this cluster tend to be a direct fork of the `ERC721Full` contract, which is an *outdated* reference implementation of the ERC721 standard provided by OpenZeppelin [43].
- **Clusters with [0,60%) Scam Tokens.** Among the seven clusters with scam token ratios between 0% and 60%, the source code of the tokens exhibits considerable diversity in the ways the ERC721 standard is extended. The semantic variations in the included tokens suggest a deliberate effort by scammers to explore diverse contract-level mechanisms for executing fraudulent schemes.

TABLE III: Statistics of token clusters across ranges of scam token ratios.

Scam Token Ratio (%)	# Clusters	# Scam Tokens	# Total Tokens
100%	20	321	321
[60%, 100%)	18	2,005	2,143
[40%, 60%)	2	93	197
[0, 40%)	5	725 (714 is unclustered)	77,813

Finding 4: Token contracts in the NFT ecosystem exhibit significant bytecode-level diversity, distributed across 1,350 distinct clusters, with scam tokens predominantly concentrated in 45 (3.3%) of these clusters. Among the 45 clusters, 38 demonstrate scam token ratios exceeding 60%, indicating a high concentration of fraudulent activity in specific bytecode patterns.

2) Interaction Patterns in Risky Transactions: We identified 22,306 interaction patterns between smart contracts from the 3,089,245 risky transactions in the NFT ecosystem, with the ten most frequently observed patterns depicted in Figure 4b. Through analysis of these patterns, we made the following observations:

👉 **Patterns S2, S7, S9 and S10 are predominantly associated with risky transactions, with each pattern exhibiting a risky transaction ratio exceeding 90%.** Patterns S2, S7, S9 and S10 exhibit high risky transaction ratios of 100%, 99.45%, 100%, and 94.74%, respectively. The predominance of these patterns in risky transactions suggests that certain interactions among smart contracts are highly indicative of malicious behavior in the NFT ecosystem. We also noticed that these patterns are grounded in specific contract-level workflows, including minting, structured transfers, and auction-based bidding mechanisms. For example, Pattern S2 reflects user interactions during the minting process of the *Codex Record* NFT collection [52]. Pattern S7 is derived from a characteristic transfer behavior observed in the *LucidSight* collection [53]. Patterns S9 and S10 are associated with the auction mechanisms implemented in the *Marble* [54] and *CryptoFlower* [55] collections, respectively. Notably, Pattern S10 captures a bidding workflow: an EOA places a bid by transferring a fixed amount of ETH and subsequently receives a refund for any surplus once the auction concludes.

👉 **Patterns S1, S3, S4, S6 and S8 are shared across both risky and non-risky transactions, with each pattern exhibiting a risky transaction ratio below 10%.** Patterns S1, S3, S4, S6 and S8 observed in risky transactions correspond to Patterns 1, 2, 4, 9, and 10 identified in RQ2, with each pattern exhibiting a relatively low risky transaction ratio of 1.95%, 3.35%, 2.17%, 6.86%, and 9.34%, respectively. The relatively low risky ratios associated with these five interaction patterns reflect their dual-use nature, as they are adopted by both legitimate users and malicious actors in their transactions in the NFT ecosystem. The shared usage of such interaction patterns across transactions may limit the discriminative utility of these patterns in distinguishing between risky and non-risky transactions in the NFT ecosystem. It may require the incorporation of contextual and semantic features when designing effective detection mechanisms for risk transactions exhibiting such patterns.

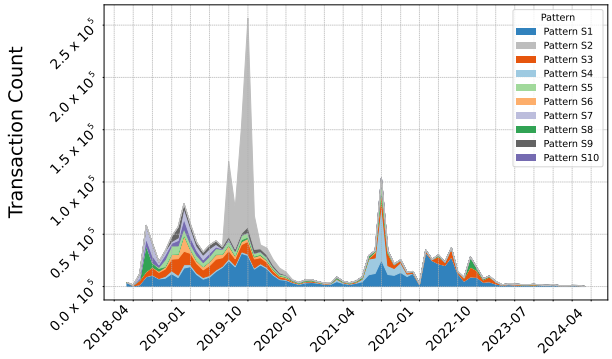
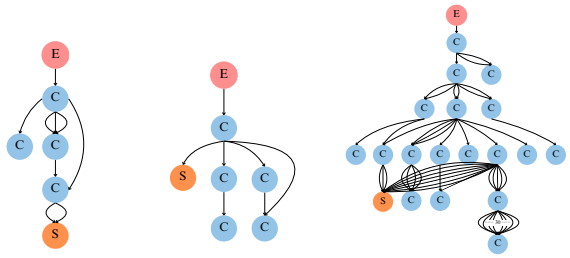


Fig. 5: Evolution in transaction volumes for the ten most frequent interaction patterns of scam tokens.

Figure 5 presents the evolution of risky transaction volumes associated with the above ten most frequently observed interaction patterns, spanning the period from April 2018 to July 2024. Among these patterns, Pattern S1 demonstrates sustained activity throughout the entire timeframe, suggesting that the structural generality and flexibility of the interaction pattern have contributed to its continued use in scam campaigns. In contrast, Pattern S2 exhibits a sharp and isolated surge in transaction volume between late 2019 and mid-2020, peaking at over 200,000 transactions. This suggests a targeted exploitation of specific contract-level mechanisms, likely related to the minting logic of the rugpull-like Codex Record collection. In addition, Patterns S7 and S10 exhibit modest yet distinct spikes during earlier periods in the timeframe, suggesting their involvement in localized or short-lived scam activities. Notably, the transaction volumes associated with all patterns demonstrate a consistent decline after early 2022, which may be attributed to the emergence of more effective detection mechanisms, shifts in attacker strategies beyond scam tokens, or a reduced economic incentive for conducting large-scale exploitation.

Moreover, certain interaction patterns between smart contracts are predominantly observed in transactions involving specific scam tokens. In particular, 10,893 patterns are exclusively linked to rugpull tokens, 1,160 to Ponzi tokens, and 1,559 to honeypot tokens. Figure 6 presents three representative patterns, which appear in 468,058, 2,202, and 1,424 transactions involving rugpull, Ponzi, and honeypot, respectively. These representative patterns are more frequently observed in risky transactions of their corresponding scam types, where they often exhibit relatively higher frequencies within each category.



(a) Rugpull (b) Honeypot (c) Ponzi

Fig. 6: Interaction Patterns between Smart Contracts Predominantly Observed in Rugpull-, Honeypot-, and Ponzi-Related Transactions.

> **Finding 5:** The ten most frequently observed interaction patterns between smart contracts in risky transactions can be classified into two categories: 1) those that are prevalent in both risky and non-risky transactions, and 2) those predominantly associated with risky transactions, characterized by isolated surges in transaction volumes across distinct short intervals from April 2018 to July 2024. The transaction volumes associated with all patterns demonstrate a consistent decline after early 2022.

V. Discussion

A. Implications

We reflect on our findings from the research questions, offering practical recommendations to mitigate risks in the NFT ecosystem. We also highlight the avenues of future research.

Impact and Risks of EIP-1967 Proxy Contracts. In RQ1, we observed that EIP-1967 proxy contracts represent three of the top ten semantic clusters of smart contracts deployed on Ethereum in the NFT ecosystem. Meanwhile, proxy contracts are among the top five most interactive smart contracts during NFT transactions, as observed in RQ2. Given the prevalence of EIP-1967 proxy contracts in the NFT ecosystem, their reliability and stability are critical to the overall security and operation of the NFT ecosystem. Malicious actors may exploit proxy contracts to conceal malicious logic, misleading blockchain explorers (e.g., Etherscan) and monitoring tools into mistakenly identifying them as legitimate logic contracts [56], particularly in the context of EIP-1967. Specifically, malicious proxies can store the address of a legitimate logic contract in the designated slot defined by EIP-1967, while redirecting execution to a separate logic contract that contains malicious logic at runtime [57]. Future work could put efforts into developing a real-time monitoring system of logic contracts behind proxies during NFT transactions, enabling the early detection of anomalous or suspicious activities.

Prevalence of Cloned EIP-1167 Proxy Contracts. In RQ1, we observed that EIP-1167 contracts, acting as minimal proxies and are not upgradeable by design, account for 74.82% of the deployed smart contracts in the NFT ecosystem, far exceeding the the 19.54% share of non-upgradeable proxies reported on Ethereum [57]. The majority of these cloned EIP-1167 proxy contracts are configured to delegate calls to a smart contract known as *XEN Torrent* [58]. XEN Torrent facilitates bulk minting of XEN tokens through a mechanism involving Virtual Minting Units (VMUs) [59]. Users can create VMUs by interacting with XEN Torrent, which are then used to claim XEN tokens. The centralization around XEN Torrent could expose the NFT ecosystem to systemic risks if vulnerabilities in these contracts are exploited.

Fraud Detection and Transaction Risk Assessment. In RQ3, we observed that scam tokens are distributed across 45 out of 1,350 (3.3%) bytecode clusters, with 38 of these clusters exhibiting high scam token ratios exceeding 60%, indicating a semantic convergence of scam tokens in their bytecodes. Such convergence could inform the development of cluster-based approaches for scam token detection in the NFT ecosystem. Moreover, previous studies [10], [60], [61], [62] have proposed graph-based models for fraud detection in the Ethereum ecosystem, where tokens are represented as nodes in the graphs. Future work could explore the potential of using the distinct differences in the bytecodes between scam and non-scam tokens to enhance graph-based fraud detection models, particularly in the process of node embedding.

We also identified four interaction patterns–S2, S7, S9, and S10–between smart contracts in risky transactions, exhibiting risky transaction ratios existing 90% (RQ3). These interaction patterns exhibit strong discriminatory power in differentiating between risky and non-risky transactions, and could therefore serve as behavioral signatures of fraudulent activities. Future work can incorporate these patterns as high-level semantic features into risk assessment frameworks for transactions in the NFT ecosystem.

B. Threats to Validity

Our study exclusively focuses on the ERC721 token standard, which enables a more targeted investigation, but may limit the generalizability of our findings to other NFT token standards, such as ERC1155. In RQ1, the labeling of semantic clusters of smart contracts was independently conducted by two authors to reduce personal biases; however, the process may still involve a degree of subjectivity. To mitigate this threat, a blockchain expert was consulted to validate the labeling results, thereby enhancing the reliability of the annotations. Another potential threat to validity arises from the lack of sufficient off-chain data, particularly in the analysis presented in RQ3. Risky transactions may involve complex, multi-step interactions that occur off-chain before assets are transferred to centralized exchanges, mixers, or cross-chain bridges. The inability to access comprehensive off-chain information limits our capacity to fully trace and analyze these transactions, potentially omitting factors that influence transaction patterns and their associations with illicit activities.

VI. Related Work

Some researchers have conducted empirical studies to explore the NFT ecosystem from various perspectives, including security challenges [8], user activities on the prominent NFT marketplace [16], and on-chain behaviors of NFTs [17]. Specifically, Das et al. [8] explored the security challenges in the NFT ecosystem with respect to the participating actors, including flaws in NFT marketplace design, threats from off-chain external entities, and trading malpractices by malicious users. White et al. [16] performed a longitudinal measurement study on the OpenSea NFT marketplace, focusing on the activities of buyers and sellers across various categories of NFT collections on OpenSea. A more recent work [17] conducted a large-scale measurement study of the NFT ecosystem, analyzing the on-chain behaviors of NFTs through the lens of events, participants, and marketplaces, as well as leveraging NFT trading data to investigate market manipulation, such as wash trading and arbitrage. Our work primarily focuses on the semantics and interactions of smart contracts during NFT transactions, thereby distinguishing it from the aforementioned studies that address other facets of the NFT ecosystem.

Other researchers have focused on specific risks in the NFT ecosystem, including rug pulls [18], wash trading [19], [20], promotion scams [21], and phishing scams [22]. Specifically, Huang et al. [18] characterized the symptoms of NFT rug pulls and proposed approaches for automatic detection and early warning of such scams. Regarding NFT wash trading, von Wachter et al. [19] proposed an approach to identify suspicious wash trading behaviors in transactions on the OpenSea NFT marketplace. Later, La Morgia et al. [20] performed a systematic analysis of NFT wash trading across six Ethereum-based NFT marketplaces, and measured the profitability of wash trading activities. As for NFT promotion scams, Roy et al. [21] conducted a longitudinal study on fraudulent NFT project promotions on Twitter, characterizing the associated Twitter accounts and the tactics used in these scams. They further proposed a machine learning model to identify fraudulent projects promoted on Twitter. In terms of NFT phishing scams, Yang et al. [22] explored the patterns and economic impact of phishing scams, as well as the NFT preference and post-theft behavior of scammers. They also proposed approaches for detecting NFT phishing accounts. In contrast to the aforementioned prior studies that focus on specific risks, our work investigates how scam token risks manifest through the semantics and interactions of smart contracts during transactions.

VII. conclusion and Future Work

In this work, we conducted a large-scale empirical study to explore the semantics and interactions of smart contracts in NFT transactions, using a curated dataset of nearly 100 million transactions across 20 million blocks on Ethereum. We characterize the semantics of smart contracts in the NFT ecosystem, analyze the frequency and complexity of interactions between smart contracts during NFT transactions, and explore how the risks of scam tokens manifest with respect to the semantics and interactions of contracts. Future work could put efforts into improving the security of proxy smart contracts through real-time monitoring of their corresponding logic contracts, the development of graph-based models for fraud detection by taking into account bytecode features, as well as the integration of smart contract interaction patterns into transaction risk assessment frameworks.

VIII. Acknowledgement

This research was supported by the National Science Foundation of China (No. 62472383), the Fundamental Research Funds for the Central Universities (No. 226-2025-00004), and the Open Research Fund of the State Key Laboratory of Blockchain and Data Security, Zhejiang University.

References

[1] Decrypt, "Now postage stamps are getting the nft treatment," 2021. [Online]. Available: https://decrypt.co/61963/now-postage-stamps-are-getting-the-nft-treatment,

[2] PR Newswire, "Usps certifies casemail as first blockchain generated epostage," https://www.prnewswire.com/news-releases/usps-certifies-casemail-as-first-blockchain-generated-epostage-301267842.html, 2021.

[3] CoinTelegraph, "You can now buy gold-backed nfts with the mining carbon footprint offset," https://cointelegraph.com/news/you-can-now-buy-gold-backed-nfts-with-the-mining-carbon-footprint-offset, 2021.

[4] Blockchain App Factory, "Real estate tokenization," https://www.blockchainappfactory.com/real-estate-tokenization.

[5] Flipkick, "Flipkick," https://www.flipkick.io, accessed: 2025-05-25.

[6] Q. Wang, R. Li, Q. Wang, and S. Chen, "Non-fungible token (nft): Overview, evaluation, opportunities and challenges," *arXiv preprint arXiv:2105.07447*, 2021.

[7] CoinGecko, "2023 annual crypto industry report," 2024, accessed: 2025-05-25. [Online]. Available: https://assets.coingecko.com/reports/2023/CoinGecko-2023-Annual-Crypto-Industry-Report.pdf

[8] D. Das, P. Bose, N. Ruaro, C. Kruegel, and G. Vigna, "Understanding security issues in the nft ecosystem," in *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, 2022, pp. 667–681.

[9] S. Yang, J. Chen, and Z. Zheng, "Definition and detection of defects in nft smart contracts," in *Proceedings of the 32nd ACM SIGSOFT International Symposium on Software Testing and Analysis*, 2023, pp. 373–384.

[10] C. Wu, J. Chen, Z. Zhao, K. He, G. Xu, Y. Wu, H. Wang, H. Li, Y. Liu, and Y. Xiang, "Tokenscout: Early detection of ethereum scam tokens via temporal graph learning," in *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, 2024, pp. 956–970.

[11] "Rugpull," https://coinmarketcap.com/alexandria/glossary/rug-pull, 2024.

[12] F. Cernera, M. La Morgia, A. Mei, and F. Sassi, "Token spammers, rug pulls, and sniper bots: An analysis of the ecosystem of tokens in ethereum and in the binance smart chain ({{{{{BNB}}}}})," in *32nd USENIX security symposium (USENIX security 23)*, 2023, pp. 3349–3366.

[13] C. F. Torres, M. Steichen *et al.*, "The art of the scam: Demystifying honeypots in ethereum smart contracts," in *28th USENIX Security Symposium (USENIX Security 19)*, 2019, pp. 1591–1607.

[14] "Ethereum's top gas guzzlers are ponzi schemes," https://cryptonews.net/news/ethereum/384739/, 2024.

[15] The Business Times, "Losses from crypto scams grew 45% in 2023, fbi says," https://www.businesstimes.com.sg/companies-markets/banking-finance/losses-crypto-scams-grew-45-2023-fbi-says, 2024, accessed: 2025-05-25.

[16] B. White, A. Mahanti, and K. Passi, "Characterizing the opensea nft marketplace," in *Companion Proceedings of the Web Conference 2022*, 2022, pp. 488–496.

[17] J. Huang, P. Xia, J. Li, K. Ma, G. Tyson, X. Luo, L. Wu, Y. Zhou, W. Cai, and H. Wang, "Unveiling the paradox of nft prosperity," in *Proceedings of the ACM on Web Conference 2024*, 2024, pp. 167–177.

[18] J. Huang, N. He, K. Ma, J. Xiao, and H. Wang, "Miracle or mirage? a measurement study of nft rug pulls," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 7, no. 3, pp. 1–25, 2023.

[19] V. von Wachter, J. R. Jensen, F. Regner, and O. Ross, "Nft wash trading: Quantifying suspicious behaviour in nft markets," in *International Conference on Financial Cryptography and Data Security*. Springer, 2022, pp. 299–311.

[20] M. La Morgia, A. Mei, A. M. Mongardini, and E. N. Nemmi, "A game of nfts: Characterizing nft wash trading in the ethereum blockchain," in *2023 IEEE 43rd International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2023, pp. 13–24.

[21] S. S. Roy, D. Das, P. Bose, C. Kruegel, G. Vigna, and S. Nilizadeh, "Unveiling the risks of nft promotion scams," in *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 18, 2024, pp. 1367–1380.

[22] J. Yang, J. Liu, D. Lin, J. Wu, B. Huang, Q. Li, and Z. Zheng, "Who stole my nft? investigating web3 nft phishing scams on ethereum," *IEEE Transactions on Information Forensics and Security*, 2024.

[23] S. Wu, D. Wang, J. He, Y. Zhou, L. Wu, X. Yuan, Q. He, and K. Ren, "Defiranger: Detecting price manipulation attacks on defi applications," *arXiv preprint arXiv:2104.15068*, 2021.

[24] "Geth," 2023, https://geth.ethereum.org.

[25] P. Zheng, Z. Zheng, J. Wu, and H.-N. Dai, "Xblock-eth: Extracting and exploring blockchain data from ethereum," *IEEE Open Journal of the Computer Society*, vol. 1, pp. 95–106, 2020.

[26] Etherscan, "Etherscan," https://etherscan.io, accessed: 2025-05-25.

[27] A. Kuhn, S. Ducasse, and T. Gîrba, "Semantic clustering: Identifying topics in source code," *Information and software technology*, vol. 49, no. 3, pp. 230–243, 2007.

[28] BAAI, "bge-large-en-v1.5," https://huggingface.co/BAAI/bge-large-en-v1.5, accessed: 2025-05-25.

[29] C. Zhang, H. Zhong, K. Zhang, C. Chai, R. Wang, X. Zhuang, T. Bai, Q. Jiantao, L. Cao, J. Fan, Y. Yuan, G. Wang, and C. He, "Harnessing diversity for important data selection in pretraining large language models," in *The Thirteenth International Conference on Learning Representations*, 2025. [Online]. Available: https://openreview.net/forum?id=bMC1t7eLRc

[30] X. Liang, H. Wang, S. Song, M. Hu, X. Wang, Z. Li, F. Xiong, and B. Tang, "Controlled text generation for large language model with dynamic attribute graphs," in *Findings of the Association for Computational Linguistics ACL 2024*, L.-W. Ku, A. Martins, and V. Srikumar, Eds. Bangkok, Thailand and virtual meeting: Association for Computational Linguistics, Aug. 2024, pp. 5797–5814.

[31] X. Zheng, Z. Wan, S. Liu, K. Yang, D. Lo, and X. Yang, "Gnncontext: Gnn-based code context prediction for programming tasks," *IEEE Transactions on Software Engineering*, 2025.

[32] R. J. G. B. Campello, D. Moulavi, and J. Sander, "Density-based clustering based on hierarchical density estimates," in *Advances in Knowledge Discovery and Data Mining*, J. Pei, V. S. Tseng, L. Cao, H. Motoda, and G. Xu, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 160–172.

[33] W. Liang, X. Ling, J. Wu, T. Luo, and Y. Wu, "A needle is an outlier in a haystack: Hunting malicious pypi packages with code clustering," in *2023 38th IEEE/ACM International Conference on Automated Software Engineering (ASE)*. IEEE, 2023, pp. 307–318.

[34] Z. Ma, M. Jiang, F. Luo, X. Luo, and Y. Zhou, "Surviving in dark forest: Towards evading the attacks from front-running bots in application layer," in *34rd USENIX Security Symposium (USENIX Security 25)*, 2025.

[35] "hdbsan," https://pypi.org/project/hdbscan/, accessed: 2025-05-25.

[36] K. R. Shahapure and C. Nicholas, "Cluster quality analysis using silhouette score," in *2020 IEEE 7th international conference on data science and advanced analytics (DSAA)*. IEEE, 2020, pp. 747–748.

[37] Z. Wang, Z. Wan, Y. Chen, Y. Zhang, D. Lo, D. Xie, and X. Yang, "Clone detection for smart contracts: How far are we?" *Proceedings of the ACM on Software Engineering*, vol. 2, no. FSE, pp. 1249–1269, 2025.

[38] Anonymous, "Replication package for 'why is my transaction risky? understanding smart contract semantics and interactions in the nft ecosystem'," https://doi.org/10.5281/zenodo.15550314, 2025.

[39] Manifold, "Manifold," https://docs.manifold.xyz/manifold-for-developers, 2023, accessed: 2025-05-25.

[40] Alchemy, "Manifold," https://www.alchemy.com/dapps/manifold, 2024, accessed: 2025-05-25.

[41] P. Murray, N. Welch, and J. Messerman, "Eip-1167: minimal proxy contract," *ethereum improvement proposals*, no. 1167, 2018.

[42] Buterin Cards, "Buterin cards," https://www.buterin.cards, 2024, accessed: 2025-05-25.

[43] OpenZeppelin, "Erc721," https://docs.openzeppelin.com/contracts/4.x/api/token/erc721, 2024, accessed: 2025-05-25.

[44] "Erc721a," https://github.com/chiru-labs/ERC721A, 2021.

[45] H. Adams, N. Zinsmeister, M. Salem, R. Keefer, and D. Robinson, "Uniswap v3 core," https://app.uniswap.org/whitepaper-v3.pdf, 2021, accessed: 2025-05-25.

[46] Wyvern Protocol, "Wyvern protocol," https://wyvernprotocol.com/docs/protocol-components, 2019, accessed: 2025-05-25.

[47] OpenSea, "Seaport," https://docs.opensea.io/docs/seaport, 2024, accessed: 2025-05-25.

[48] Safe, "What is safe?" https://docs.safe.global/home/what-is-safe, 2025, accessed: 2025-05-25.

[49] https://etherscan.io/tx/0xbbdf118861b0539bd9f19958dd1bd437c39b10c2bc06c96307f796da2fb2f5e2, accessed: 2025-05-25.

[50] "leftoverz," https://etherscan.io/address/0xb8f9bfC712E77F9DfFE22EA8f9ADaE8d4314d0D2#code, accessed: 2025-05-25.

[51] "Cryptotoon," https://etherscan.io/address/0xBF8a84DE5DcC0bd5792026BFDeBFc75d9675A363#code, accessed: 2025-05-25.

[52] "Codexrecordproxy," https://etherscan.io/address/0x8853B05833029e3Cf8d3Cbb592f9784FA43d2a79#code, accessed: 2025-05-25.

[53] "Mlbnft," https://etherscan.io/address/0x8c9b261Faef3b3C2e64ab5E58e04615F8c788099#code, accessed: 2025-05-25.

[54] "Marblenft," https://etherscan.io/address/0x1d963688FE2209A98dB35C67A041524822Cf04ff#code, accessed: 2025-05-25.

[55] https://etherscan.io/address/0x65FCFB6870C744Ec181e4F64a7F41A0Cfd76B845#code, accessed: 2025-05-25.

[56] F. Community, "Evasion techniques: Report on the continuous monitoring," https://github.com/apehex/web3-evasion-techniques/blob/main/report/forta.pdf, 2023, accessed: 2025-05-25.

[57] M. Zhang, P. Shukla, W. Zhang, Z. Zhang, P. Agrawal, Z. Lin, X. Zhang, and X. Zhang, "An empirical study of proxy smart contracts at the ethereum ecosystem scale," in *2025 IEEE/ACM 47th International Conference on Software Engineering (ICSE)*, 2025, pp. 620–620.

[58] https://etherscan.io/address/0x0a252663dbcc0b073063d6420a40319e438cfa59#code, accessed: 2025-05-25.

[59] XEN Crypto, "Xen crypto: Whitepaper," 2022, accessed: 2025-05-25. [Online]. Available: https://www.xencrypto.io/wp-content/uploads/2022/12/REP-final-20221227T163457Z.pdf

[60] S. Hu, T. Huang, K.-H. Chow, W. Wei, Y. Wu, and L. Liu, "Zipzap: Efficient training of language models for large-scale fraud detection on blockchain," in *Proceedings of the ACM Web Conference 2024*, ser. WWW '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 2807–2816. [Online]. Available: https://doi.org/10.1145/3589334.3645352

[61] W. Li, Z. Liu, X. Li, and S. Nie, "Detecting malicious accounts in web3 through transaction graph," in *Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering*, ser. ASE '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 2482–2483. [Online]. Available: https://doi.org/10.1145/3691620.3695344

[62] Z. Ding, J. Shi, Q. Li, and J. Cao, "Effective illicit account detection on large cryptocurrency multigraphs," in *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, ser. CIKM '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 457–466. [Online]. Available: https://doi.org/10.1145/3627673.3679707