DRAINCLoG: Detecting Rogue Accounts with Illegally-obtained NFTs using Classifiers Learned on Graphs

Hanna Kim[1] Jian Cui[2] Eugene Jang[3] Chanhee Lee[3] Yongjae Lee[3] Jin-Woo Chung[3] Seungwon Shin[1]
[1] KAIST, South Korea [2] Indiana University Bloomington, USA [3] S2W Inc., South Korea
[1]{gkssk3654, claude}@kaist.ac.kr [2]{cuijian}@iu.edu
[3]{genesith, leemember, lee, jwchung}@s2w.inc

***Abstract*—As Non-Fungible Tokens (NFTs) continue to grow in popularity, NFT users have become targets of phishing scammers, called *NFT drainers*. Over the last year, $100 million worth of NFTs were stolen by drainers, and their presence remains a serious threat to the NFT trading space. However, no work has yet comprehensively investigated the behaviors of drainers in the NFT ecosystem.**

In this paper, we present the first study on the trading behavior of NFT drainers and introduce the first dedicated NFT drainer detection system. We collect 127M NFT transaction data from the Ethereum blockchain and 1,135 drainer accounts from five sources for the year 2022. We find that drainers exhibit significantly different transactional and social contexts from those of regular users. With these insights, we design *DRAINCLoG*, an automatic drainer detection system utilizing Graph Neural Networks. This system effectively captures the multifaceted web of interactions within the NFT space through two distinct graphs: the NFT-User graph for transaction contexts and the User graph for social contexts. Evaluations using real-world NFT transaction data underscore the robustness and precision of our model. Additionally, we analyze the security of *DRAINCLoG* under a wide variety of evasion attacks.

I. Introduction

With the emergence of blockchain technology, *Non-Fungible Tokens* (NFTs) have revolutionized the digital creator economy. NFTs are digital assets, such as art or collectibles, with unique identification codes and metadata [24]. NFTs have attracted numerous content creators and investors, and by 2021 the NFT market exploded, growing to around $22 billion [4].

As a result, scammers targeting NFTs have also emerged in the NFT ecosystem [30], [46]. Over $100 million in NFTs were stolen in one year by scammers, called *drainers*, with the majority of them using phishing scams [30]. NFT drainers continue to make headlines, causing significant damage to users [11]. For instance, a sophisticated phishing attack on Uniswap[1] (the largest Ethereum-based decentralized exchange) caused $8 million worth of damages to NFT users [28].

Efforts to combat NFT drainers have had limited effectiveness. The Opensea NFT marketplace implemented the policy of marking stolen NFTs as untradeable [42]. However, this is only effective when victims are able to notice and report the attacker. The cryptowallet Metamask implemented phishing warnings [12], but could be bypassed by certain drainers [6].

Even before NFT drainers, phishing attacks targeting cryptocurrency were already considered a significant threat to the trading security of Ethereum [16]. As a response to such attacks, researchers have proposed several methods to detect phishers. One line of research relies on using handcrafted user features to detect scammers [18], while other approaches employ network representation learning by utilizing Node2Vec [58] or Graph Neural Networks (GNN) [17], [32], [59] to capture scammers.

However, these methods of detecting cryptocurrency phishers are unsuitable for detecting NFT drainers for the following reasons. First, features that are essential for cryptocurrency phisher detection, such as those that describe the liquidation process, are not shown in NFT drainers. In addition, cryptocurrency-focused detection systems cannot leverage individual NFTs and fail to account for the multiple transaction types of NFTs. Thus, they cannot fully capture the differences between NFT drainers from regular users. This raises the need for an automatic detection system that captures various factors of the NFT ecosystem. This also motivates a comprehensive investigation on the transaction patterns of NFT drainers.

To the best of our knowledge, the existing literature has not explored detecting *NFT drainers* (NFT phishing scammers). To fill this gap, we aim to investigate the trading traits of NFT drainers and propose an automatic detector that identifies suspicious trading behaviors. To this end, we collect an extensive dataset of over 127 million NFT transactions from the Ethereum blockchain, spanning from January to December 2022. We also collect information on 1,135 reported drainers from various sources, including Twitter and Etherscan. Our analysis of these drainer accounts in the NFT ecosystem reveals their trading patterns to be distinctly different from regular users. Specifically, we pinpoint two crucial factors in

[1] uniswap.com

Network and Distributed System Security (NDSS) Symposium 2024
26 February - 1 March 2024, San Diego, CA, USA
ISBN 1-891562-93-2
https://dx.doi.org/10.14722/ndss.2024.24888
www.ndss-symposium.org

identifying drainers: their unique **NFT transaction context**, characterized by quickly selling NFTs at much lower prices, and **social context**, often linking with other users displaying similar trading anomalies.

However, capturing the intricate relationships between users and NFTs for drainer detection remains a challenging task. This is because the transaction history of millions of NFTs and the various types of interactions between millions of users must be considered. To address this challenge, we propose a novel GNN-based drainer detection system, *DRAINCLoG*. We use GNNs as they are well-suited for modeling relationships between users and NFTs in a graph structure. GNNs can incorporate both the features of nodes and edges, making them effective at identifying patterns of anomalous behavior among interconnected entities [60], [7], [51], [19]. The GNN-based structure allows *DRAINCLoG* to capture the user-to-user and NFT-to-user relationships effectively.

DRAINCLoG utilizes two types of graphs that uniquely model relationships to identify drainers: the *NFT-User graph* and the *User graph*. The NFT-User graph models interactions between users and NFTs, with two types of nodes and attributed edges, allowing us to obtain a representation of users' transaction context by considering each NFT's transaction history. The User graph models interactions between users, with attributed nodes and two types of edges, enabling us to capture the social context by integrating information on the users and their relationships. To obtain representations, we use customized GNNs for each NFT graph type. These representations are then fused to leverage information from both graph types. Our model significantly outperforms existing baselines in drainer detection. Overall, we believe that our study will inspire further research and practical efforts to improve NFT trading security. In summary, the contributions of the work are listed below:

- We collect 127M NFT transaction data from the Ethereum blockchain and 1,135 drainer accounts from five different channels. Drainer accounts are publicly available for future research [2].
- We present the first empirical study on NFT drainers and find that drainers have distinct characteristics and transaction patterns from regular users.
- We comprehensively capture the relationships between users and NFTs into novel graph structures. Our GNN-based model, *DRAINCLoG*, automatically detects drainers from these graphs.

II. Background and Motivation

A. Background

Non-Fungible Token (NFT) is a cryptographic asset on the blockchain with unique identification codes and metadata that distinguish them from each other. NFTs guarantee ownership of unique digital assets, such as images, video files, and even physical assets. By 2017, the Ethereum blockchain, which currently accounts for approximately 80% of the global NFT trading volume, became a popular hub for NFTs, with collectibles like CryptoPunks [8] gaining prominence. An NFT *collection* refers to a group of NFTs sharing similar features, but each NFT has unique variations that sets it apart. This uniqueness can lead to significant value differences among NFTs, even if they belong to the same collection.

[2]will be made available on acceptance

Fig. 1: Summary of NFT transaction types: ⓐ Mint, ⓑ Sale, ⓒ Gift, and ⓓ Burn

NFT Transaction Types We introduce four types of NFT transactions: *mint, sale, gift,* and *burn*, as depicted in Figure 1.

ⓐ *Mint*. An NFT is created by minting, the process of inscribing a digital asset to the blockchain. Minted NFTs can be listed and traded on NFT marketplaces, such as OpenSea [41] and Rarible [44].

ⓑ *Sale*. A sale is a process of transferring an NFT ownership to another account for payment. NFTs are typically traded with Ether, the native cryptocurrency of Ethereum, and sometimes fungible tokens. Users can partake in sales in two ways: buying and selling.

ⓒ *Gift*. A gift is a process of handing over ownership without any monetary exchange. Typically, gifting occurs between addresses that are related. Within the NFT ecosystem, there are various scenarios where NFTs are gifted. For example, gifts can be used between users to avoid monitoring when manipulating markets by wash trading. Users can partake in gift transfers in two ways: gifting-in and gifting-out.

ⓓ *Burn*. Burning is the process of sending NFTs to an inaccessible address, which will remove them from circulation. Burning is used for various purposes, such as adjusting the supply of NFTs, operating a collection's community, etc.

Stealing NFTs from victims' NFT wallets is commonly called **draining** [33], and we will use this term in this paper. The main goal of NFT drainers is to "drain" (steal) NFTs from victims, although cryptocurrencies and private information could also be targeted. NFT drainers commonly use phishing scams for their purpose. We detail the procedures of how NFT drainers operate by dividing them into three steps: (1) spreading phishing websites, (2) draining NFTs, and (3) cashing out drained NFTs. Figure 2 illustrates the process.

(1) Spread phishing websites. NFT drainers mainly use two methods to spread phishing websites to victims. First, they *1.1) use social media*, such as Twitter. They can make social media accounts masquerading as official accounts of popular NFTs, sometimes even compromising them. The drainers upload posts linking to scam sites on these channels. Another method is to *1.2) use phishing token airdrops*. Airdrops are commonly used in marketing campaigns to promote creators' projects by sending free tokens [54]. However, it can also be abused by drainers to trick victims. A drainer can send fake tokens to target wallets, tricking victims into clicking a phishing website link in the token's description.

(2) Drain NFTs from victims. The drainers steal NFTs from lured victims through two primary methods: *2.1) capturing login credentials of crypto wallets* or *2.2) exploiting an interface*

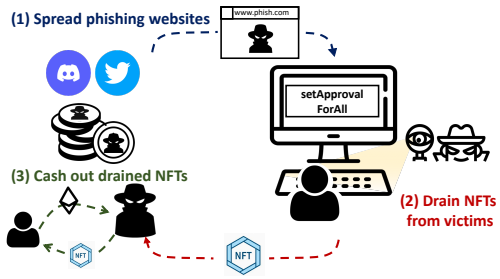


Fig. 2: The process of draining NFTs using phishing attacks: (1) spreading phishing websites, (2) draining NFTs from victims, and (3) cashing out drained NFTs.

function. The first method is similar to the traditional phishing methods that target victim passwords. When victims input their crypto wallet credentials, drainers can record the information using tools like keyloggers [50]. With the stolen credentials, drainers can access victim accountsto transfer NFTs into their own wallets. Alternatively, drainers can abuse smart contracts to deceive victims. Drainers can craft malicious smart contracts using functions such as *setApprovalForAll*, and lure victims to sign a transaction with the contract [57]. *SetApprovalForAll* is an important method used for NFT trading. When selling NFTs on a marketplace, a user needs to call the function to authorize the marketplace to transfer the NFTs from the seller's account to the buyer's. However, by calling the function on a phishing website, a victim grants the drainer permission to transfer all the victim's tokens [38]. On the blockchain, such an unauthorized transfer from the victim's account to the drainer's is logged as *gifts*.

(3) Cash out drained NFTs. Scammers targeting cryptocurrencies cash out stolen assets using crypto exchanges and sometimes through mixing services [59], [22]. Conversely, NFT drainers must sell the stolen NFTs by listing them on marketplaces [22]. To combat NFT scams, OpenSea [41], the largest NFT marketplace, made a policy of disabling the buying, selling, and gifting of stolen items [42].

B. Motivation

NFTs have attracted the attention of investors and criminals alike. NFT drainers using phishing scams continue to make headlines [11], [10]. For instance, a sophisticated phishing attack on Uniswap NFT holders caused damages of $8 million when users were tricked into approving malicious transactions [28], [52].

Alarmingly, there have been a wealth of tools made available to assist NFT draining. Software packages for NFT draining are being sold between $29 – $149 on dedicated websites [34] and the darkweb [20]. The accessibility of draining tools lowers the barriers to entry for future NFT drainers. To prevent such tools, MetaMask, a popular crypto wallet, updated to include a feature to warn users when transactions request the *setApprovalForAll* [12], [23] function. However, NFT drainers have also developed tools to bypass this update [6].

Although the damage caused by NFT drainers is increasing, no previous work has yet conducted an in-depth measurement study or proposed a detection method for NFT draining. Several methods have been proposed to detect phishers targeting cryptocurrency [32], [18], [17], but they are unsuitable for applying to the NFT ecosystem for the following reasons.

First, essential features to detect cryptocurrency phishers do not apply to NFT drainers. The ineffectiveness of previous features raises the need for a measurement study on NFT drainers to gain insights into how to detect phished NFTs. In Section IV, we investigate the activity of NFT drainers to identify trading traits that can detect NFT draining.

Second, existing methods cannot consider complex contexts within the NFT ecosystem. Unlike cryptocurrency, where coins are interchangeable and have equal value, NFTs are of distinct identities with dynamic values, making complex transaction patterns (factoring price and frequency). To fully understand a user's trading habits, the transaction history of each NFT the user trades should be considered. In Section V-B, we describe our NFT drainer detection system, *DRAINCLoG*, which includes a module specifically designed to leverage a user's NFT transaction context.

Third, previous studies on detecting cryptocurrency phishing cannot fully capture the relationship between NFT users. While cryptocurrency transactions are only limited to transfers, NFT transactions can be further classified into four categories. Our investigation shows that the distinction of transaction types between NFT users is a significant indicator when interpreting relationships between users. Section V-C explains how *DRAINCLoG* uses a module to capture user relationships in the NFT trading network.

III. Dataset Construction

This section summarizes our data collection approach and the datasets used in this study. We collect two datasets in our paper: NFT transaction data and NFT drainer accounts.

Fetching NFT transaction records from Ethereum blockchain: There are two types of addresses in the Ethereum blockchain: Externally Owned Accounts (EOA) and Smart Contracts. An EOA represents an account controlled by an individual. It can send transactions, interact with smart contracts, and manage digital assets on Ethereum. In this paper, we refer to EOAs simply as *users* or *accounts* to enhance readability. A smart contract is a code deployed on the blockchain executing actions and transactions based on predefined conditions.

We utilize *transfer logs* to get a token's transmission history. Transfer logs are written to the blockchain by smart contracts whenever token transfers occur. With this information, we can identify changes in ownership of tokens, such as who sends the tokens to whom.

We distinguish between two types of NFT transactions between users: sales and gifts. However, because the NFT transfer logs do not have payment information, evidence of payment must be found from other sources. During our research period, over 90% of the NFT trading volume took place on three platforms: Opensea, Blur, and X2Y2 [9]. A detailed analysis of the smart contracts used for NFT trading from these markets suggests that most NFTs traded on the marketplace are sold for Ether or various Fungible Tokens (FTs). If an NFT buyer pays in Ether, it is recorded through an

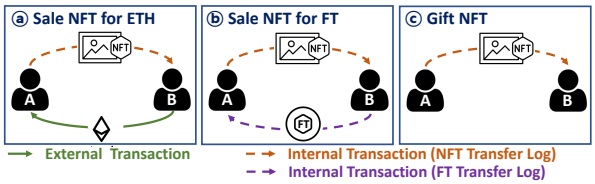


Fig. 3: Visualizations simplifying the NFT transaction types on the Ethereum blockchain. We classify each NFT transaction record between users into sales (ⓐ and ⓑ) or gifts (ⓒ) based on whether a payment was made.

TABLE I: Summary of collected NFT transaction records. Note that the number of mintings is larger than the number of NFTs due to the multiple mintings of spam NFTs.

Type	**Collection**
NFT	80,795,833
Address	4,733,670
EOA (Account)	4,640,645
Smart contract	93,025
Transaction	127,820,930
Mint	81,769,127
Sale	24,915,481
Gift	19,722,551
Burn	1,413,771
Period	January 1,2022 ∼ December 31, 2022

external transaction. If the buyer pays in FT, it is recorded in an FT transfer log. Hence, we gather both external transactions and transfer logs associated with both NFTs and FTs. To accomplish this, we run an Ethereum node and retrieve the data from the node using web3.py, an Ethereum-specific library to interact with the Ethereum blockchain.

Figure 3 provides an overview of how NFT transaction types appear on the Ethereum blockchain. If there is an NFT transfer log from user A to user B and an external transaction in which B sends Ether to A, we infer the NFT was sold for Ether (ⓐ). Instead, if there is an FT transfer log in which B sends FTs to A, we interpret this as the NFT being sold for FTs (ⓑ). For both sale types, we include indirect sales, such as when an NFT sender receives currency from the NFT receiver through a marketplace address. Otherwise, we regard that there was no payment for the NFT, and the NFT's transfer from A to B is considered as a gift (ⓒ).

To confirm the quality of our NFT transaction data, we analyzed weekly transaction statistics of our data, which include the number of sales, the number of gifts, and trading volume. Our data collection is supported by similar reportings from NFTGo [40], a renowned NFT analysis platform.

According to Elliptic[3], NFT draining grew rapidly and has been on the rise since January 2022 [30]. As a result, we focus on data with `block_timestamp` between January 1, 2022 to December 31, 2022. We obtain more than 127 million transactions from this period for 80 million NFTs. The data includes transactions from over 4 million unique accounts. Our NFT transaction data is summarized in Table I.

Crawling NFT drainer accounts: We crawled drainer accounts reported for NFT phishing scams from five websites: Twitter, ScamSniffer, Etherscan, CryptoScamDB, and Chainabuse.

TABLE II: Summary of collected drainer accounts.

Channel	**# active accounts**	**# drainer accounts**
Scamsniffer	817	797
Chainabuse	769	737
Twitter	728	682
Etherscan	128	128
CryptoscamDB	68	66
Total	1,230	1,135
Period	January 1,2022 ∼ December 31, 2022	

(1) Twitter [53] is utilized by NFT users as a channel to share information on drainer accounts. We first collect tweets that mention the phrase "NFT" and one of the following keywords: *phish, hack, drain, stole*. Then, we manually select tweets written by high-profile users who analyze scammers in the crypto-space professionally. (2) ScamSniffer [5] collects malicious accounts by visiting suspicious sites that trick users into making dangerous transactions. ScamSniffer checks whether accounts have suspicious behavior through other security services [47]. (3) Etherscan [25], an Ethereum block explorer, provides a list of Ethereum addresses reported for phishing/hacking. Etherscan reviews and assesses reports to prove the accounts are involved in scams or phishing activities [26]. (4) CryptoScamDB [2] collects reports on scams in the crypto space, including malicious accounts, URLs, and descriptions. CryptoscamDB manually scans the reports before adding addresses to the dataset [14]. We only select reports related to NFT draining by using the same criteria in (1) Twitter. (5) Chainabuse [1] provides scam reports with descriptions across multiple blockchains. Chainabuse has a spam detection system and attributes a confidence score to reports calculated by experts [13]. We collected Ethereum addresses with the 'Checked by Chainabuse' badge [15] reported under the NFT Scam category.

We gathered reports from the above channels based on the specified criteria until January 1, 2023. Note that some accounts were reported multiple times across different sites. During our data collection period, we identified 1,230 accounts involved in NFT transactions. Since not all reported accounts were successful, some reported accounts did not show drainer activity. As outlined in Section II-A, the act of draining NFTs from victims' wallets is categorized as *gifts*. Based on this, we define accounts that have at least one gifted-in NFTs as *drainers*. Using this definition, we identified 1,135 unique accounts labeled as drainers (summarized in Table II).

IV. Drainer Activity Characterization

Since drainers have malicious intent, they are likely to exhibit different behavior patterns in NFT trading from regular users. In this section, we look into distinguishable traits of drainers that motivate the design of our NFT drainer detection model. We conduct a measurement study with NFT transaction records from January 1, 2022, to July 31, 2022, including 645 drainer accounts. First, by comparing primary trading features with regular users, we verify that most drainer accounts are

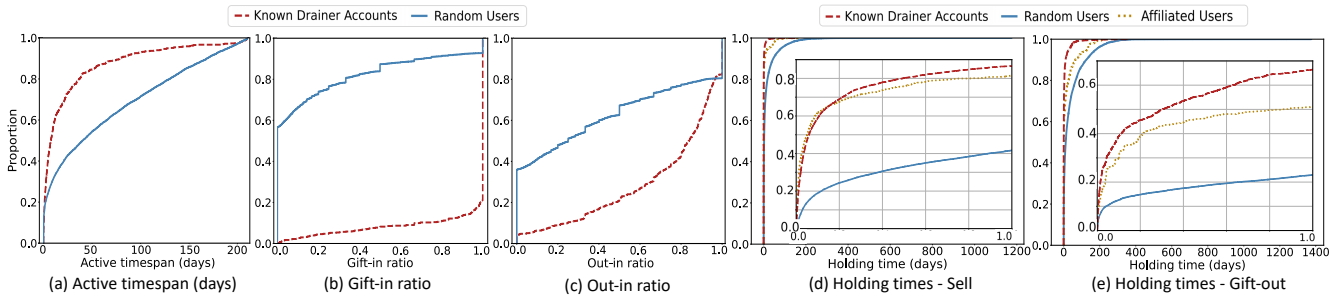


Fig. 4: CDFs of (a) active timespan, (b) gift-in ratio, and (c) out-in ratio. CDFs of holding times of different user types based on out-transaction types: (d) *sell* and (e) *gift-out*, where the y-axis denotes proportion of NFTs.

used only for draining. Then, we focus on their liquidation patterns and how they are distinct from regular transactions. We track drained NFTs and find that they have different **NFT transaction context** and **social context** from regular users.

A. Trading behavior of drainers

As mentioned in II-A, there are many efforts to prevent phishing scams in the NFT space, such as sharing information about drainers, phishing websites, and drained NFTs. If drainer accounts become known to be suspicious, it is nearly impossible for them to trade NFTs with benign users. Thus, we expect that drainers are more likely to 1) have short-lived accounts and 2) use their accounts only for draining.

In order to investigate these two assumptions, we analyze the trading activity of 645 NFT drainers. For comparison, we randomly extract 10,000 users from our data of 3 million regular users. From this set, users with only one transaction record were excluded since their active timespan cannot be calculated. This left 6,658 regular users to be used in the analysis. We compare these two groups along three dimensions.

First, we measure how long they traded by calculating the active timespan: the time difference (in days) between the first and last transaction recorded for each user. Figure 4(a) shows that drainers trade for a shorter period than regular users. This phenomenon is also similarly observed in Ether phishing accounts [18].

Second, we look into how NFT drainers utilize their accounts. Previous work to detect cryptocurrency phishers [32], [18] only focused on the amount and frequency of user transactions. However, multiple transaction types exist in NFT trading. This is an important consideration since we assume drainers will be gifted NFTs from victims (as opposed to buying or minting NFTs). Thus, we calculate the ratio of gifting-in to all in-transaction types (gifting-in, buying, and minting). In Figure 4(b), the majority of drainers have a relatively high proportion of NFTs gifted; 75.1% of drainers obtain NFTs only through gifting-in in the three in-transaction types.

In addition, we analyze how likely drainers are to transfer out. Drainers looking to liquidate do not wish to hold their NFTs. Therefore, drainers are likely to transfer their NFTs out. We measure *out-in ratio*, the ratio of the number of out-transactions to in-transactions. In Figure 4(c), we observe drainers have a higher out-in ratio than regular users. Regular users generally have a lower out-in ratio: 38.1% of them did not make any out-transactions at all. On the other hand, drainers have higher out-in ratios: 75.9% of them make out transactions on more than half of their NFTs. This suggests drainers have different intentions from regular users, who are more likely to use NFTs for collecting or investing.

TABLE III: Statistics on transactions of drained NFTs according to behaviors right after draining: *sell*, *gift-out*, and *hold*. # gifting is the number of giftings before the first sale after draining. % sold is the percentage of NFTs eventually sold.

Type	# gifting	# drained NFTs	% sold
Sell	0	11,195 (41.8%)	100%
Gift-out	1	6,426 (24.0%)	75.1%
	≥ 2	1,125 (4.20%)	39.3%
Hold	0	8,065 (30.1%)	0%
Total		26,811 (100%)	61.4%

By combining the observations above, we conclude that most drainer accounts are for draining purposes only. Additionally, we further analyze differences between drainers and regular users along 19 dimensions (more details in Appendix D) and utilize them in our detection method.

B. Liquidation behavior of drainers

In this section, we dive deep into the liquidation process of drained NFTs. The uniqueness of NFTs enables us to track the transaction history of each NFT. Since drainer accounts are only used for draining, we assume all gifted-in NFTs were stolen from victims.

Alternate accounts. We find a total of 26,811 NFTs that were gifted to drainers (Table III). Of these, 41.8% were sold directly, 28.2% were gifted-out to other users, and the rest (30.1%) remained in the drainer's wallets. We defined *affiliated users* as all accounts that were gifted drained NFTs from known drainer accounts.

Only 17.5% of drainers do not have affiliated users. In other words, most drainers (82.5%) have one or more affiliated users and use them to liquidate drained NFTs indirectly. We find 637 affiliated users including 15.4% that are related to two or more drainers. The most overlapped affiliated user is connected with 16 drainers. The fact that many drainers choose to liquidate through the same affiliated accounts suggests a close relationship between them. 60% of affiliated users get NFTs only through gifting-in, while the rest of them (40%) participate in buying and minting like regular users (Appendix D-A).

TABLE IV: Statistics of holding times of drained NFTs depending on user type along out-transaction types: *sell* and *gift-out*. Percent decrease is the measure of the decrease from $HT\_regular_{avg}$ to HT_drain as a percentage of $HT\_regular_{avg}$.

	# drained NFTs	
Case	**Sell**	**Gift-out**
$HT\_drain < HT\_regular_{avg}$	8,077 (88.9%)	6,210 (90.9%)
$HT\_drain = HT\_regular_{min}$	6,498 (71.4%)	5,034 (73.7%)
Total	9,085 (100%)	6,832 (100%)
Stats of percent decrease		
Mean	87.7%	87.5%
Standard deviation	29.5	29.8

From the result, we observe that most affiliated users exhibit similar behaviors to drainer accounts, but there are also a significant number of affiliated accounts that engage in general NFT trading.

Rapid liquidation: We now analyze how quickly drainers liquidate NFTs by measuring the holding times of each NFT. We define holding time as the timespan a user held ownership of an NFT, and measure it by taking the difference (in days) of the in-transaction and out-transaction. Holding times can vary greatly depending on the user's investment strategy and characteristics of the NFT, such as market price and rarity. We measure the holding times of all users that owned the NFT for each of the 18,746 drained NFTs with out-transactions. Note that the holding time can be longer than seven months (our collection period) because we refer to NFT transaction data before 2022 to minimize the bias. We compare drainer holding times with those of affiliated users and those of regular users along two out-transaction types: sell and gift-out.

Drainers and affiliated users show similar distributions which are noticeably different from that of regular users. Figure 4 (d) shows they sell more than 80% of NFTs within a day, which is twice more likely than regular users. They also have short holding times before gifting, but drainers gift NFTs much faster compared to affiliated users (Figure 4 (e)). By integrating these observations with the fact that up to 75% of NFTs sent to affiliated users are sold (Table III), we notice that drainers alternate between two strategies of liquidation: (1) directly selling NFTs quickly or (2) quickly gifting-out NFTs to affiliated users for selling.

For comparison with regular transactions, we calculate the average holding time ($HT\_regular_{avg}$) for each NFT, which serves as a reference to a drainer's holding time (HT_drain). Note that the ($HT\_regular_{avg}$) is calculated from the NFT's minting date up to July 2022. As shown in Table IV, 90% of HT_drains are shorter than $HT_regular_{avg}$s, with an average of 87.7% percent decrease regardless of the out-transaction types. We also observe that HT_drain is the minimum holding time for 70% of NFTs. From these results, we identify that drainers deviate from regular transaction patterns in terms of holding time.

Bargain prices: The findings from the above raise a question; *how can drainers liquidate their NFTs so quickly?* To answer this question, we compare the sales prices of drainers/affiliated users (Price_{drain}) with the market prices. However, unlike stocks or cryptocurrencies, each NFT has its own unique value. Also, market prices are susceptible to fluctuations based on supply and demand dynamics [37]. Thus, defining a market price for an NFT is feasible only when a sale occurs.

TABLE V: Statistics on sale price of drained NFTs compared to their market price (Price_{avg}, $\text{Price}_{closest}$). Only consider NFTs sold after draining with other sale records of more than one. *p.d.* denotes the measure of decrease from each market price to Price_{drain} as a percentage of the market price.

		Stats of p.d.	
Case	**# drained NFTs**	Mean	Std
$\text{Price}_{avg} > \text{Price}_{drain}$	8,214 (74.0%)	37.3%	24.9%
$\text{Price}_{closest} > \text{Price}_{drain}$	8,490 (76.5%)	39.1%	24.2%
Total	11,100 (100%)		

To instead provide a comparative market price, we employ two baselines: Price_{avg} and $\text{Price}_{closest}$. Price_{avg} represents the mean sales price from the NFT's minting date up to July 2022, and $\text{Price}_{closest}$ signifies the sales price from a transaction occurring nearest to the drainer's sale time. We observe that drainers sell 74% and 76% of their NFTs cheaper than the two baselines, respectively, with an average price decrease of 37% and 39% (Table V).

In summary, our analysis reveals distinct transaction and social contexts exhibited by NFT drainers compared to regular users. Specifically, drainers tend to have irregular **NFT transaction contexts**, such as quickly liquidating NFTs at prices lower than the market value. Additionally, drainers often have unique **social contexts**, such as making sales through affiliated users, and often are linked to the same affiliated users.

V. Design of NFT Drainer Detector

Section IV emphasizes the importance of understanding both transaction and social contexts for NFT drainer detection. However, capturing the intricate relationships between millions of users and NFTs for drainer detection remains a challenge.

To tackle the challenge, we designed a graph-based NFT drainer detection model, *DRAINCLoG*, depicted in Figure 5. The model creates a comprehensive representation of each user using a transaction context extractor and a social context extractor. Each extractor is trained to learn the relevant context on a *NFT-User graph* and a *User graph*, respectively. The NFT-User graph models NFT-to-user interactions using *NFT ownership edge attributes*, and the user transaction contexts can be fully captured by aggregating the interaction history of all of their owned NFTs. The User graph models comprehensive user-to-user interactions using *user node attributes* that detail user trading behaviors and two types of edges representing user interactions. The two contexts are then combined with user node attributes and fed into a classifier.

A. Feature Engineering

Before learning high-level user representations, we perform feature engineering based on observations in Section IV to obtain *NFT ownership edge attributes* and *user node attributes*.

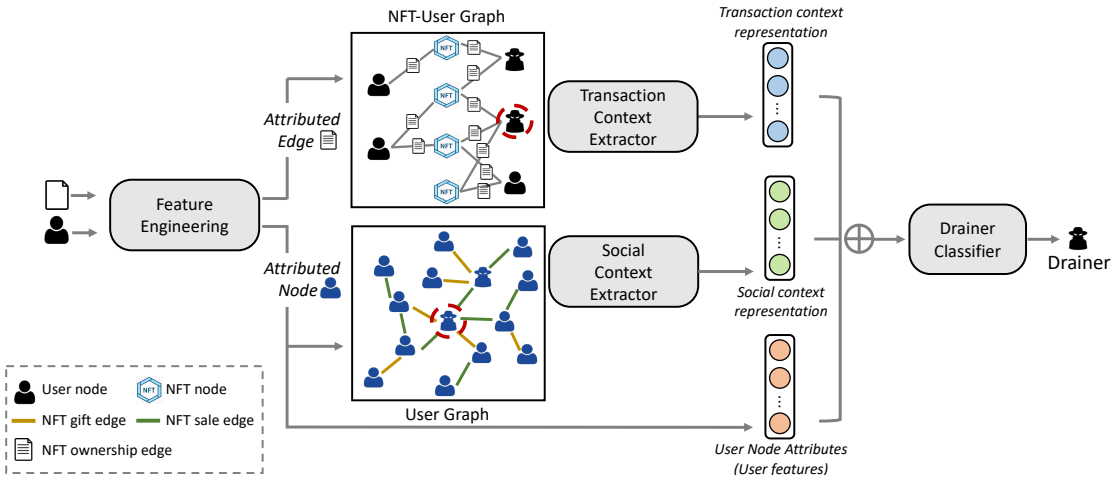


Fig. 5: The overall architecture of *DRAINCLoG*. First, NFT transaction records and user accounts obtain attributes through feature engineering and are used to construct an NFT-User graph and a User graph. The transaction context extractor and social context extractor are trained to extract information from each graph, respectively. The user features and the representations from two extractors are concatenated and fed into the classifier.

1) NFT ownership edge attributes: To capture different transaction behaviors, we create representations of how users interact with NFTs, which is used in the NFT-User graph. We use the following 7 features to represent NFT ownership:

1) **Holding time:** Holding time is the timespan a user held ownership of the NFT. If there is no out-transaction, we calculate it by taking the difference (in days) between the in-transaction and the last day of our collection period. As drainers tend to sell or gift NFTs faster than regular users, their holding times are shorter than those of regular users.
2) **In-transaction type & Out-transaction type:** Each transaction type is a categorical feature of how the user received the NFT (buy or gift-in) and what the user did with the NFT (sell, gift-out, or hold), respectively. Drained NFTs have the gift-in type.
3) **In-price & Out-price:** Each is the price (in Ether) the user sent to receive the NFT and the price the user received when sending the NFT out, respectively. It is zero if the transaction type is gifting, and -1 if no out-transaction was made. The in-price and out-price are significant because drainers sell their NFTs cheaper than regular transactions.
4) **Average holding time & Average sale price:** We include the NFT's average holding time and sales price across all owners, calculated over the entire lifespan of the NFT. This can be used as a reference in finding anomalies in holding times and pricing.

2) User node attributes: To comprehensively understand the social relationships between users, we created detailed representations of their trading behavior, which is used in the User graph. We introduce 19-dimensional user node attributes that take into account NFT characteristics such as collections and transaction types, which helps the model to identify distinct behavior patterns of drainers.

1) **Active timespan:** The time difference between the first and the last transaction. Drainers are more likely to have a short active timespan than regular users.
2) **Gift-in ratio:** The ratio of gifting-in to all in-transactions (minting, buying, gifting-in). Most drainers obtain NFTs only through gifting.
3) **Out-in ratio:** The ratio of out-transactions to in-transactions. Drainers have a higher out-in ratio than regular users.
4) **Number of each transaction type:** Transaction types considered are minting, buying, gifting-in, selling, and gifting-out. Drainers participate in selling and gifting-in more than other types. (Appendix D-A)
5) **Number of collections for each transaction type:** NFT users commonly form communities based on collections [39] and trade NFTs from similar collections. Unlike regular users, drainers tend to trade more kinds of collections than regular users. (Appendix D-B)
6) **Number of neighbors for each transaction type:** Accounts that made a transaction with each other are considered neighbors. Drainers tend to have more neighbors from gift-in transactions. Since minting produces no neighbor, it is excluded from the transaction types. (Appendix D-C)
7) **Frequency of gift-ins & sales:** The number of gifting-in/selling transactions divided by the active timespan. Gift-ins and sales occur more frequently for drainers. (Appendix D-D)

B. NFT Transaction Context Extractor

We construct a *NFT-User graph* to model ownership changes in NFTs and train an extractor for the graph to extract NFT transaction context of each user. From the NFT-User graph, we first obtain the transaction context of each NFT by aggregating its transaction history. Finally, to get a representation of the user's NFT transaction context, we aggregate the transaction contexts of all of their owned NFTs.

1) NFT-User graph construction: We construct an undirected graph $G(U, N, E_T)$. There are two types of nodes: user nodes U, NFT nodes N, and the u nodes and n nodes can

be connected with an attributed edge $e \in E_T$, called *NFT ownership attributes*. Additionally, $N_e(n)$ is the node n's one-hop neighbors (users who traded NFT n), $N_e(n) = \{u' \in U | (u', n) \in E_T\}$, and $N_e(u)$ is the set of NFT nodes in node u's one-hop neighbors, $N_e(u) = \{n' \in N | (n', u) \in E_T\}$.

2) NFT transaction context extraction: A transaction context h^N of an NFT node n is updated based on attributed edges connected to, denoted as $T_n = \{t_{u_1}, t_{u_2}, \ldots, t_{u_m}\}$, where u_i $(1 \leq i \leq m)$ is one of $N_e(n)$, and m is the size of $N_e(n)$. Then, we aggregate them with convolution layer as follows :

$$h_n^N = \sigma\left(W^N \cdot AGG_N(T_n)\right)$$

where σ is an activation function, and W^N is the trainable matrix for learning. We use mean-pooling as an aggregation function to represent the transaction context of each NFT.

Next, a user representation h^U is updated by all neighboring NFT nodes' transaction context vectors. For a user u, the representation h_{un}^U of each NFT $n \in N_e(u)$ is obtained as follows:

$$h_{un}^U = W^U \cdot concat(t_{un}, h_n^N)$$

where t_{un} is u's ownership of n, and h_n^N is the transaction context vector of n. We concatenate the two and apply a linear transformation with the learnable matrix W^U.

Finally, we integrate all of the NFT transaction context vectors $\{h_{n_1}^N, h_{n_2}^N, \ldots, h_{n_v}^N\}$ (v is the size of $N_e(u)$). To be sensitive to irregular NFT transaction patterns, we implement attention mechanisms to combine each NFT ownership vector. Specifically, we adapt a multi-head graph attention operation to reduce the effects of noise.

$$z_{un} = \mathrm{LeakyReLU}(a \cdot h_{un}^U)$$

$$h_u^U = \sum_{n \in N_e(u)} \alpha_{un} z_{un}$$

where z_{un} is an attention score calculated by taking the dot product of learnable weight a and applying $\mathrm{LeakyReLU}$. α_{un} is the normalized attention score using the softmax function. Lastly, we compute the final representation by averaging the attention head outputs.

Training. The final representation feeds into the classification layer for classification. The extractor updates the trainable parameters to learn features that distinguish drainers from regular users. We use cross-entropy loss as the loss function.

C. Social Context Extractor

Drainers have distinct motivations for engaging in NFT trading compared to regular users, leading to the formation of unique social connections. Most drainers choose to liquidate through affiliated accounts. Moreover, some affiliated accounts are used by multiple drainers, which suggests a close relationship between the co-users. Thus, the relationships between users are essential to detect drainers. To model user interactions, we construct a *User graph* and use it to train an extractor to learn the social context of users.

1) User graph construction: Trading between users can be constructed as a graph $G(U, E, R, X_U)$, where U is the set of user nodes, and E is labeled edges (u_i, r, u_j), where $r \in R = \{sale, gift\}$ is a relation type. Each user node has 19-dimensional user node attributes, X_U. If u_i transfers an NFT to u_j, then u_i and u_j are connected with an edge $e \in E$.

2) Social context extraction: Understanding user relationships in the NFT ecosystem relies heavily on transaction types. Therefore, we utilize transaction types as relational information between users. This approach allows us to capture relational dependencies and acquire more meaningful representations.

To accomplish this, we employ the R-GCN model (Relational Graph Convolutional Network) [48], which is specifically designed to handle graph structures with relational data. The R-GCN model has demonstrated strong performance across various tasks and is well-suited for our purposes.

From the user graph, we obtain a representation vector of a user node u updated by its neighboring user nodes. The propagation at $(l+1)$-th layer of R-GCN with L layers is as follows:

$$h_u^{l+1} = \sigma\left(W^l h_u^l + \sum_{r \in R} \mathrm{AGG_U}\left(\frac{1}{c_{u,r}} W_r^l h_v^l\right), \forall v \in N(u)_r\right)$$

where σ is an activation function, and W^l is a learnable matrix shared among all nodes at l-th sub-layer. $N(u)_r$ is neighboring user nodes under relation r of u. We use mean-pooling as the aggregation function $\mathrm{AGG_U}$.

Training. We train this module in the same process as the NFT transaction context extractor by feeding the outputs into the classification layer and using cross-entropy loss.

D. Drainer Classifier

Following the above operations, we obtain three types of features representing users: (1) NFT transaction context representation, (2) social context representation, and (3) user node attributes. We concatenate the three representations together to create our final representation, integrating comprehensive information learned from our graphs. In order to learn the differences between drainers and regular users, we feed the final representation to a classifier layer.

We choose a support vector machine (SVM) [29] as our classifier layer. SVM is a supervised machine learning model that uses classification algorithms. SVM has the advantage of reducing the chances of model overfitting, making the model highly stable. SVM is also powerful to deal with high dimensional features.

VI. Experiments

To validate our model in different aspects, we present several empirical evaluations in this section. Specifically, we seek to answer the following research questions:

- How effective is *DRAINCLoG* in detecting NFT drainers?

- How does each component affect performance?

- How robust is our model against evasion attacks?

TABLE VI: Dataset statistics for training and evaluation. Ratio refers to the ratio of drainers to sampled regular users. Each number in evaluation datasets is the average value over 5 runs.

Dataset		Ratio	# central nodes	# total nodes	# transactions
Training	D_0	1:80	52,245	2,010,384.0	24,745,525.0
Evaluation	D_1	1:10	6,006	2,087,436.0	28,375,070.6
	D_2	1:100	55,146	2,743,003.4	41,384,504.8
	D_3	1:1000	546,546	3,179,105.4	45,289,602.6

A. Datasets

In our experiments, we utilize a dataset consisting of NFT transaction data and accounts identified as NFT drainers.

Potential False Negative Filtering: In Section IV, we observe that drainers often gift stolen NFTs to *affiliated users*, and that some affiliated users receive NFT gifts from multiple drainers. From this observation, it can be suggested that the other accounts that gift NFTs to known affiliated users are also highly suspected to be related to drainers. However, these accounts, while suspicious, cannot be determined with certainty as drainers themselves. Therefore, we choose to exclude these suspicious accounts as well as affiliated users from the regular user category.

Training Dataset Construction: To create our training dataset, we first gathered accounts that engaged in transactions from January 1, 2022, and July 31, 2022. This was comprised of 3,137,221 accounts, with 645 of them being identified as drainer accounts. Due to the highly imbalanced ratio of regular users to drainers, directly using this data for training would not be effective. Therefore, we used two sampling strategies to select regular users for the training set.

First, we excluded accounts that have low activity. Specifically, we removed two categories of accounts: 1) accounts that had never received NFTs from other users and 2) accounts with zero active time. From the remaining 1,355,811 accounts after removal, we sampled 45,150 regular users (at a 1:70 ratio) to include in our training dataset.

Secondly, we additionally sampled "heavy" regular users, those with over 50 transactions. This strategy aims to address the potential model bias towards transaction quantity, since regular users tend to partake in fewer transactions than drainers (see Appendix C). Thus, we select an additional 6,450 heavy regular users (at a 1:10 ratio) into the training dataset.

Our final training dataset comprises 645 drainers and 51,600 (45,150+6,450) regular users. The specific ratios used in the sampling processes were empirically selected through experimental evaluation. The ideal number of regular users can depend on the class ratio in the evaluation dataset. For a deeper exploration of how the class ratio in the training dataset impacts results, refer to Section VII.

Evaluation Dataset Construction: For the evaluation dataset, we selected accounts that engaged in at least one transaction between August 1, 2022, and December 31, 2022. This yielded a total of 1,723,465 accounts, of which 490 were identified as drainers. Note that our dataset utilizes transaction records of these accounts from January 1, 2022 to December 31, 2022. Given the notable imbalance between drainers and regular users in the dataset, it is critical to evaluate our model under various scenarios. To achieve this, we created three separate datasets by adjusting the proportion of regular users in our tests to be 10, 100, and 1000 times the number of drainers. By doing so, we aim to understand our model's performance under varying levels of class imbalance. While higher ratios of regular users offer a representation more in line with real-world distributions, it is more lenient towards false negatives, which translates to drainer accounts that remained undetected.

To construct each dataset, we use the selected accounts as central nodes and include first and second-order neighbor nodes. Note these neighbor nodes are used only to enrich the graph, and are not used for training or evaluation. The dataset statistics are summarized in Table VI.

B. Baselines

As baselines, we use methods that effectively detect Ethereum phishing accounts. In addition, we also use other widely used graph-based models because *DRAINCLoG* is a graph-based model. The baselines can be divided into two categories: Feature-based and graph-based.

Feature-based:

- **Ether features** [18] use 119-dimensional statistical features previously used for Ethereum phishing account detection. The features mainly consist of first-order neighbor information.
- **E-GCN features** [17] are the initial node features used in E-GCN, a method proposed to detect Ethereum phishing accounts.
- ***DRAINCLoG* user features** are the initial node features used in our User-graph.

Graph-based:

- **Trans2Vec** [58] is a modified random walk-based graph embedding method with biases for neighbor sampling.
- **E-GCN** [17] applies a Graph Convolutional Network (GCN) to detect Ethereum phishing accounts.
- **GAT** [55] is a widely used GNN model. It learns node representations by aggregating neighbor nodes with an attention mechanism.
- **GraphSAGE** [27] is another GNN variant. It learns node representations by aggregating sampled neighbor node features.

To evaluate the effectiveness of our features, we implement each graph-based baseline twice: once using E-GCN features (denoted by the "E-" prefix) and once using *DRAINCLoG* user features (denoted by "N-" prefix).

C. Experimental Results

For implementation details, refer to Appendix A. Table VII shows that our model outperforms the baselines in all evaluation metrics, which proves the effectiveness of *DRAINCLoG* for NFT drainer detection.

TABLE VII: The results of experiments averaged over 5 runs on datasets $D_1 \sim D_3$. Pre., Rec., F1., and FP/TP mean precision, recall, F1 score, and the number of false positive/true positive, respectively.

Model	Dataset (ratio)	D_1 (1:10)				D_2 (1:100)				D_3 (1:1000)			
	Metrics	Pre.	Rec.	F1	FP/TP	Pre.	Rec.	F1	FP/TP	Pre.	Rec.	F1	FP/TP
Feature based	Ether features	0.875	0.227	0.361	15.9/111.1	0.429	0.227	0.297	148.0/111.2	0.072	0.227	0.109	1433.2/111.2
	E-GCN features	0.838	0.104	0.185	10.0/51.0	0.334	0.104	0.159	102.4/51.0	0.047	0.104	0.064	1045.4/51.0
	DRAINCLoG user features	0.976	0.618	0.757	7.4/302.4	0.779	0.618	0.689	86.2/304.2	0.277	0.627	0.385	801.8/307.2
Graph based	Trans2Vec	0.000	0.000	0.000	0.0/0.0	0.000	0.000	0.000	0.0/0.0	0.000	0.000	0.000	0.0/0.0
	E-GCN	0.832	0.037	0.071	3.7/18.1	0.349	0.037	0.067	33.6/18.0	0.055	0.037	0.044	311.5/18.1
	E-GAT	0.933	0.010	0.020	0.4/5.0	0.825	0.010	0.020	1.2/5.0	0.256	0.009	0.018	12.8/4.4
	E-GraphSAGE	0.980	0.157	0.271	1.6/77.0	0.867	0.157	0.265	12.0/77.2	0.435	0.157	0.231	99.9/76.9
	N-GCN	0.838	0.103	0.183	9.8/50.2	0.351	0.103	0.159	93.8/50.6	0.057	0.102	0.073	825.5/50.0
	N-GAT	0.982	0.411	0.580	3.8/201.4	0.811	0.411	0.546	47.4/202.6	0.323	0.415	0.363	426.3/203.4
	N-GraphSAGE	0.987	0.569	0.722	3.6/278.4	0.860	0.569	0.685	45.8/280.2	0.416	0.579	0.484	398.3/283.7
	DRAINCLoG	**0.989**	**0.622**	**0.763**	3.4/304.4	**0.878**	**0.621**	**0.727**	42.8/306.0	**0.448**	**0.628**	**0.523**	379.1/307.7

DRAINCLoG outperforms standard feature-based methods since it benefits from the NFT-specific features and high-level representations obtained from extractors. Many NFT-specific features can be instrumental in distinguishing NFT drainers from regular users but are ignored in the existing methods. On the other hand, previous approaches use features that may not fully apply to NFT trading. For instance, a significant feature in Ethereum phishing account detection [18] is whether an account has mixing services as neighbors, since phishers tend to liquidate with such services. However, since NFTs cannot go through mixing services, this feature is inapplicable for analyzing NFT trading.

One surprising finding is that graph-based methods without our NFT-specific attributes (E-GCN, E-GAT, and E-GraphSAGE) perform poorly compared to the feature-based methods. For instance, solely relying on the E-GCN features yields better results than the complete E-GCN method itself. Furthermore, both E-GCN and E-GAT tend to classify the majority of users as non-malicious, leading to an alarmingly low recall. However, this trend changes when we embed NFT-specific attributes into these graph-based methods. By doing so, the performance of the modified graph-based methods (N-GCN, N-GAT, and N-GraphSAGE) improved significantly, highlighting the importance of our feature engineering. In addition to the NFT-specific attributes, *DRAINCLoG* takes into account transaction types and utilizes NFT transaction context extracted from the NFT-User graph. This results in a multi-faceted representation that allows it to outperform other graph-based methods.

DRAINCLoG outperforms the baselines in NFT drainer detection, but we observe that precision and F1-score decrease as the dataset size increases. While we have addressed potential false negatives when constructing datasets, unreported drainers may still exist among what we categorize as regular users. Naturally, the decrease in precision might have come from correctly classifying unreported drainers as drainers. We will discuss the unreported drainers in Section VII-D.

D. Ablation Study

In this section, we delve into understanding how individual components within *DRAINCLoG* impact its overall performance. We assess the influence of each representation by removing them one-by-one. Specifically, we assess the representations from our feature engineering, *NFT Transaction Context Extractor* (TCE), and *Social Context Extractor* (SCE). In addition, to understand the importance of considering transaction types in the User graph, we experiment with a version of *DRAINCLoG* that employs GCN instead of R-GCN.

TABLE VIII: The results of ablation experiments on D_1 and D_3 datasets averaged over 5 runs.

Dataset (ratio)	D_1 (1:10)			D_3 (1:1000)		
Removed	Pre.	Rec.	F1	Pre.	Rec.	F1
DRAINCLoG user features	0.985	0.599	0.745	0.410	0.607	0.489
NFT transaction context	0.989	0.585	0.735	0.450	0.588	0.510
Social context	0.984	0.591	0.739	0.391	0.595	0.472
Relation in SCE	0.980	0.589	0.736	0.327	0.594	0.422
DRAINCLoG	0.989	0.622	0.763	0.448	0.628	0.523

The experimental results, presented in Table VIII show that optimal performance is achieved when all components are integrated. Exclusion of the NFT transaction context leads to a marked drop in recall. This underscores its importance in identifying drainers, which other components might miss. Additionally, the removal of the social context representation yields the lowest precision of all components. This suggests its pivotal role in preventing *DRAINCLoG* from misclassifying regular users as drainers. Excluding relations in the component leads to a drop in performance, especially as the number of regular users increases. This highlights the importance of transaction types for identifying NFT drainers.

We also examine the influence of *NFT transaction edge attributes* and *user node attributes* on *DRAINCLoG*'s performance. We grouped features that represent similar concepts together. For example, the number of transactions for each transaction type (5) was grouped into one, *the number of transactions*. Then, we trained and evaluated *DRAINCLoG* on the graph constructed without each group.

In Table IX, critical components in the TCE component include *average information*, *out-transaction type & out-price*, and *price*. Notably, discarding average information for each NFT significantly impairs performance. Its inclusion aids the model in differentiating between regular and non-regular trans-

TABLE IX: The results of ablation experiments averaged over 5 runs. The most impactful features of each attribute are highlighted in color. (*) indicates a set of features involving in and out directions, and (**) indicates a set of features involving all transaction types.

Type	Dataset (ratio)	D_1 (1:10)			D_2 (1:100)			D_3 (1:1000)		
	Removed *feature* group	Pre.	Rec.	F1	Pre.	Rec.	F1	Pre.	Rec.	F1
NFT ownership edge attributes	Holding time	0.989	0.605	0.751	0.878	0.605	0.717	0.440	0.609	0.511
	Transaction type*	0.986	0.634	0.772	0.867	0.634	0.732	0.426	0.640	0.511
	Price*	0.984	0.618	0.759	0.867	0.617	0.721	0.425	0.627	0.506
	Avg. price & holding time	0.988	0.595	0.743	0.869	0.595	0.706	0.429	0.600	0.500
	In-transaction type & In-price	0.986	0.616	0.758	0.865	0.616	0.719	0.428	0.624	0.507
	Out-transaction type & Out-price	0.986	0.618	0.759	0.865	0.617	0.720	0.417	0.627	0.501
User node attributes	Active timespan	0.986	0.609	0.753	0.850	0.609	0.710	0.392	0.618	0.480
	Gift-in ratio	0.988	0.620	0.762	0.864	0.620	0.722	0.414	0.627	0.499
	Out-in ratio	0.987	0.624	0.764	0.873	0.624	0.727	0.427	0.634	0.510
	# transactions**	0.990	0.558	0.714	0.887	0.558	0.685	0.467	0.569	0.513
	# collections**	0.985	0.595	0.742	0.842	0.595	0.697	0.376	0.604	0.464
	# neighbors**	0.985	0.597	0.744	0.842	0.597	0.699	0.375	0.608	0.464
	Freq. of gift-in & sell	0.987	0.612	0.755	0.871	0.611	0.718	0.431	0.622	0.509
DRAINCLoG		**0.989**	**0.622**	**0.763**	**0.878**	**0.621**	**0.727**	**0.448**	**0.628**	**0.523**

actions based on NFT transaction patterns. The price and out-transaction information also play a crucial role in detecting drainers, capturing their distinctive pattern of selling NFTs at a lower price. Another significant observation is the increased F1-score in datasets D_1 and D_2 upon removal of *transaction type*. However, this improvement was offset by a drop in precision, resulting in poor performance in F1-score on dataset D_3. This highlights the importance of considering transaction type to avoid misclassifying regular users as drainers, particularly as the number of users increases.

The user node attributes exhibited a greater influence on performance compared to the NFT transaction edge attributes. The most critical attributes are *the number of collections*, *the number of neighbors*, and *active timespan*. This aligns with drainer behavior of trying to steal NFTs from a wide range of users. Consequently, drainers tend to engage in trades involving numerous NFT collections and interact with a large number of neighbors within a brief period.

E. Robustness

If drainers notice that they are monitored by *DRAINCLoG*, they may purposely change their trading patterns to avoid detection. Therefore, we discuss and evaluate to what extent an adversary can deceive *DRAINCLoG*.

1) Evasion attacks: The following are the environmental assumptions and constraints for evasion attacks.

Assumptions. *DRAINCLoG* closely monitors every transaction and swiftly alerts marketplaces when a new drainer is detected. Meanwhile, victims can directly report the drainer if they realize they've been compromised. On confirming a drainer's malicious activities, marketplaces instantly freeze their NFT trades to prevent further sales of stolen assets. For a comprehensive version of the actual usage scenario of *DRAINCLoG*, refer to Section VII.

Constraints. To benefit from stolen NFTS, drainers must rapidly sell the NFTs at lower prices, particularly before marketplace bans. Also, the selling processes for stolen NFTs are influenced by both regular users and drainers, not solely by the drainers. Thus, the latter cannot afford to alter specific parameters related to selling of stolen assets - like *holding time, out-transaction type, out-price,* and *frequency of selling*. Under the assumptions and constraints, drainers essentially have two avenues for evasion.

1. Utilizing Multiple Accounts. A drainer could deploy multiple accounts. In fact, using affiliated users for selling NFTs can be a part of this type of attack. While *DRAINCLoG* can spot drainers making sales through affiliated users, their methods can become more sophisticated. For example, during liquidation, a drainer might use multiple auxiliary accounts to trade an NFT cyclically, maintaining its original value, before selling it to a regular user at a reduced price. However, it is commonplace for users to operate multiple wallet accounts for better asset management. For instance, an individual could relocate assets from other accounts to a different account for sale. This can pose a challenge for *DRAINCLoG* as it might struggle to differentiate between this layered attack strategy and legitimate trades. Nonetheless, these attackers are still pressured to liquidate their NFTs across multiple sales in a restricted timeframe to evade marketplace bans, which results in a conspicuous spike in trading volume. While *DRAINCLoG* currently does not provide countermeasures for this layered attack, a security operator (or wash trading detection system) might grow suspicious of an unexpected trading volume surge of particular NFTs, even if the individual transactions do not seem malicious.

2. Using a Single Account. A drainer can also mask its activity by changing the trading pattern of the drainer account itself without the use of additional accounts. It can make a series of low-value noise transactions to alter its user node attributes, NFT owner edge attributes, and inter-relationships in the graphs. Thus, we introduce four types of attacks designed to change these attributes and graphs significantly within the given constraints.

Attack 1. Mint NFTs. The easiest way that drainers can engage in innocuous-appearing NFT transactions is by minting

TABLE X: The results of evasion attacks on D_1 and D_3 datasets. We re-trained the classifier layer with 3% of evasion attackers and evaluated remaining evasion attackers on datasets D_1' and D_3'. The result for other values of X refers to Appendix E.

	Dataset (ratio)		D_1 (1:10)			D_1' (1:10)			D_3 (1:1000)			D_3' (1:1000)		
	L	X	Pre.	Rec.	F1	Pre.	Rec.	F1	Pre.	Rec.	F1	Pre.	Rec.	F1
Attack1	10	N/A	0.970	0.481	0.643	0.981	0.542	0.698	0.278	0.484	0.353	0.263	0.547	0.356
	30		0.962	0.386	0.551	0.979	0.507	0.668	0.238	0.393	0.296	0.254	0.517	0.341
	50		0.959	0.354	0.517	0.979	0.497	0.660	0.221	0.358	0.273	0.251	0.507	0.336
Attack2	10	N/A	0.966	0.349	0.513	0.979	0.601	0.744	0.218	0.356	0.270	0.269	0.607	0.373
	30		0.940	0.192	0.319	0.980	0.635	0.771	0.132	0.194	0.157	0.277	0.639	0.387
	50		0.919	0.139	0.241	0.981	0.663	0.791	0.097	0.137	0.114	0.287	0.669	0.401
Attack3	10	60	0.866	0.110	0.195	0.966	0.574	0.719	0.081	0.115	0.095	0.220	0.591	0.320
	30		0.852	0.098	0.176	0.965	0.625	0.758	0.074	0.104	0.086	0.222	0.635	0.328
	50		0.873	0.114	0.202	0.970	0.644	0.774	0.082	0.118	0.097	0.264	0.648	0.374
Attack4	10	60	0.551	0.020	0.039	0.952	0.425	0.587	0.017	0.023	0.019	0.171	0.474	0.251
	30		0.426	0.012	0.024	0.956	0.563	0.709	0.010	0.013	0.012	0.183	0.587	0.278
	50		0.430	0.012	0.024	0.961	0.634	0.764	0.011	0.014	0.012	0.207	0.651	0.314
	DRAINCLoG		**0.989**	**0.622**	**0.763**	**0.989**	**0.622**	**0.763**	**0.448**	**0.631**	**0.523**	**0.448**	**0.628**	**0.523**

NFTs. With this attack, drainers can alter numerous user node attributes, including gift-in and out-in ratio.

Attack 2. Increase Active timespan. An active timespan is one of the important features distinguishing drainers from regular users. Drainers can easily increase their active timespan by engaging in a transaction before initiating draining activity.

Attack 3. Send Ether to victim. Drainers can hide their activities by sending Ether to victims after stealing NFTs. This way, draining is recorded as a sale instead of a gift. This attack not only changes most of the user node and NFT ownership edge attributes but also alters both associated graphs.

Attack 4. Combination of three attacks. This is the combination of all three attacks.

2) Evaluation: To evaluate *DRAINCLoG*'s detection capabilities against evasion tactics, we adjusted previous evaluation datasets. These modifications were guided by the specific attack strategy and its attack level (L), where $L \in \{10, 30, 50\}$.

For Attack 1, we increased the number of minted NFTs by $L\%$ of gifted-in NFTs. For Attack 2, the active timespan was extended by $L\%$. For Attack 3, we changed $L\%$ of gifting-in transactions to buying transactions by sending X% of the average sale price of each NFT to those victims, where $X \in \{1, 10, 60\}$. Lastly, for Attack 4, we integrated the tactics of Attack 1 and Attack 2, both at level 50, with Attack 3. We set X to be less than 60 because drainers typically sell stolen NFTs at 40% below their average sale price (based on our findings in Section IV).

The experimental results, presented in Table X show that as the attack strategies progress from Attack 1 to Attack 4, the system's performance is increasingly compromised. Attack 2 poses a greater threat than Attack 1, suggesting that a short active timespan was a critical trait of drainers. Nevertheless, Attack 1 and Attack 2 lag behind even the lowest intensity of Attack 3 and Attack 4. This stems from their inability to modify the user and NFT relationships in the graphs.

On the other hand, Attack 3 and Attack 4 poses a significant challenge to our system's detection capabilities. This is because attackers adopting Attack 3 deviate from the definition of drainers we used in this study (accounts that steal NFTs). Since the model was trained to identify accounts that steal NFTs, it is not optimized for detecting cases in which attackers buy NFTs at lower prices. As a result, these attackers are considered an unseen data type within our model, causing difficulties in detecting them. Table E (in Appendix E) shows that Attack 3 is more effective in evading the detector when more Ether is sent to victims. However, this comes at a cost for the attacker since each draining operation will incur additional costs. Despite the effectiveness of these attacks, it is critical to highlight the resilience of our system; this method fails to fully deceive the updated system with the defense method (which will be introduced below).

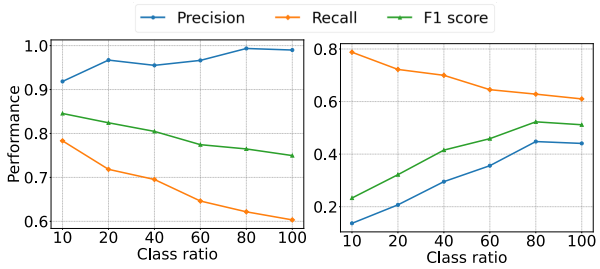
3) Defense: A proactive defense against evasive attacks entails periodically refining *DRAINCLoG* to recognize emerging drainer patterns. We update *DRAINCLoG* in a simplified manner, re-training only the last layer (SVM), while leaving two extractors intact. For the evaluation, we trained the SVM using a newly augmented dataset that combines the prior training dataset along with an additional 3% of evasion attackers. This updated model was then tested against the remaining evasion attackers. We trained and evaluated using a consistent ratio of drainers to regular users as detailed in Section VI-A.

In Table X, results of D_1' and D_3' show that *DRAINCLoG*'s performance significantly improved after updating the classifier with only 3% added attackers. Notably, we observe a pronounced increase in recall, indicating that *DRAINCLoG* is capable of detecting new types of drainers with only limited examples of such attackers. These results confirm that *DRAINCLoG* can effectively capture their hidden complex relationships within two graphs, which is difficult to change for attackers. However, the precision for D_3' remains suboptimal. In the next section, we will discuss ways to make *DRAINCLoG* more robust against new types of drainers.

VII. Discussion & Future Work

A. Evasion attack

For evasion attacks against *DRAINCLoG*, the main limitation of our approach lies in the case where the attackers



(a) D_1 (b) D_3

Fig. 6: Detection results based on the class ratio in the training dataset. The x-axis indicates the relative number of regular users to drainers.

mask their liquidation process through multiple sales using multiple accounts. Still, this type of attack yields a high trading volume of an NFT in a very short time and sometimes results in cycles in the NFT transaction graph. Thus, this can be distinguished from regular trades. To detect this type of attacker, *DRAINCLoG* can integrate with other protection systems, such as a wash trading detection system. It can capture their irregular trades by 1) detecting a rapid trade sequence using transaction velocity [56] or 2) finding cyclic trading patterns between users [56], [21].

Furthermore, we note that with a simple defense strategy, *DRAINCLoG* can detect evasive attackers that use a single account, but it still shows low precision. This low precision comes from the fact that they modified trading patterns to mimic regular users, and we re-train the model using them. To solve this problem, it is important to identify new characteristics of drainers that they cannot alter. We manually analyzed the blockchain transaction history and found that drainers tend to receive (steal) multiple NFTs from each of their victims nearly simultaneously. This phenomenon is attributable to the drainers' strategy; they distribute phishing websites, aiming for maximum reach and victim count, which leads to a synchronous influx of victims. However, drainers cannot change this pattern since they lack control over the moment each victim is compromised. Using this insight, we can consider integrating this characteristic into *DRAINCLoG*. Specifically, in the User Graph, we can add *interaction timespan* and *the number of NFTs traded* between users to edge attributes. In the NFT-User graph, we can add *blockchain timestamp* when the transaction occurs to the NFT ownership edge attributes. This information can help *DRAINCLoG* better distinguish evasive attackers from regular users.

B. Influence of training class ratio on performance

We trained our model to prioritize F1-score, a suitable metric for imbalanced datasets. However, the real-world significance of spotting unidentified drainers underscores the need for high recall. It should be noted that *DRAINCLoG* can be tailored for high recall by altering the ratio of drainers to regular users in the training dataset.

Figure 6 shows the influence of class ratio in the training dataset on overall performance. Given the limited reported drainer accounts available, we can increase the number of regular users. Within the regular user category in each dataset, we fixed the number of heavy regular users as ten times that of the drainers. *DRAINCLoG* trained with fewer regular users yields higher recall, facilitating the detection of previously undetected drainers. Conversely, as we incorporate more regular users into training, *DRAINCLoG* learns from a more diverse set, increasing precision. However, using a large number of regular users leads to a high-class imbalance in training, causing the model to focus more on avoiding false positives than on detecting drainers, which decreases recall. To address this limitation, we can expand drainer samples used for training by employing synthetic minority over-sampling techniques. This approach holds promise for enhancing both recall and precision.

C. Usage scenarios of DRAINCLoG

DRAINCLoG operates on a robust foundation consisting of a database enriched with Ethereum transaction data and a curated list of identified drainer accounts. This database undergoes real-time updates, fetching the latest transaction data directly from the blockchain. Also, it augments its drainer account list by integrating victim reports and phishing website detection systems. By analyzing data from phishing website detection systems, we can also collect on drainer accounts that have not yet started draining. Each time a transaction occurs, *DRAINCLoG* updates the profiles of the involved users and corresponding NFTs. The system can scan multiple targets simultaneously, assigning risk scores to each user profile. These evaluations are essential for enhancing the security measures of software crypto wallets and NFT marketplaces.

Specifically, *DRAINCLoG* can integrate with software crypto wallets—tools that facilitate interactions with blockchains. The following scenario can stop the draining caused by victims signing transactions with abused contracts. When a user tries to sign a transaction, *DRAINCLoG* could be used to cross-reference the recipient's account against its drainer list. If the recipient is on the list, the transaction is promptly halted, and the user is warned of this information. Otherwise, the risk score attached to the account is checked. Accounts with risk scores surpassing administrator-set thresholds trigger warnings to the user, prompting them to proceed with or abandon the transaction. This real-time decision is instrumental in preventing potential victimization.

On the other hand, marketplace administrators are notified with real-time updates on the drainer list and user risk scores. They can employ an automated banning mechanism that relies on high risk score thresholds (which ensures high precision). Alternatively, they can manually inspect accounts flagged by less stringent thresholds to increase recall. Once threats are confirmed, the marketplace can block the trading of stolen NFTs and notify affected users. The timely identification and banning of drainer accounts are important in undermining their revenue streams.

These administrators can subsequently incorporate newly identified drainer accounts into *DRAINCLoG*'s drainer list. If the count of such new additions surpasses a predetermined limit, *DRAINCLoG* can undergo a re-training phase using the expanded datasets to adapt and counteract the evolving strategies of drainers continually.

D. Analysis of false positives & negatives

• **False positive**: Of the 490,000 regular users evaluated (in D_3), 379 were flagged as drainers. However, unreported drainers might be included in false positives. To identify how many unreported drainers, *potential drainers*, exist, we manually verified each user using two indicators: 1) Connection with phishing attackers and 2) Possession of suspicious NFTs. The detailed criteria refer to Appendix B.

From our analysis, we identified 115 potential drainers. Taking these into account, the adjusted performance metrics were 0.615 precision, 0.699 recall, and 0.654 F1 score within the D3 dataset. We note that the actual number of unreported drainers could be higher.

One potential drainer sold 33 NFTs and gifted-out 19 NFTs from 79 gifted NFTs. Analyzing this user's Ethereum transaction history, we found records of Ether transfers with a reported phishing attacker that spanned 81 days. Their prolonged interaction suggests a potential relationship, raising suspicions about the user's activities in the NFT space.

On the other hand, some regular users were misclassified as drainers because their legitimate trading behaviors resemble those of actual drainers. We observed that the majority (88.8%) of false positives acquired NFTs solely via gifting-in (instead of buying or minting) and quickly sold them within their short active timespan. For instance, one user sold 253 NFTs, from which 241 were received as gifts, in just five days. Intriguingly, the user received 233 NFTs from another account over two days and swiftly sold most of them.

This behavior arises because sometimes individuals create several wallet accounts for better asset management to mitigate risks. Such practices can mislead *DRAINCLoG* into incorrectly categorizing them as drainers, especially if they rapidly sell NFTs after receiving them. However, a distinguishable pattern exists: during a draining attack, all NFTs of a victim are instantly transferred to the drainer, whereas benign users transfer their NFTs across multiple hours or days. Our model, unfortunately, overlooked this temporal distinction in user interactions. It's crucial to factor in the duration over which interactions occur between users in future refinements.

• **False negative**: We analyze drainers misclassified as regular users. False negatives have a lower out-in ratio than other drainers; 46.5% of them never sold an NFT, and 43.2% of them never gifted out an NFT. A key characteristic of drainers comes from when they liquidate or transfer the stolen NFTs to affiliated users. It seems that *DRAINCLoG* failed to detect them due to the lack of such processes.

E. Analysis of high-profile incidents

DRAINCLoG can detect drainers who conduct large-scale attacks. We discuss high-profile incidents that made headlines in the media, all of which were detected by *DRAINCLoG*.

In December 2022, an incident attributed to North Korean state-sponsored threat actors notably garnered significant media attention [43], [36]. These attackers made off with digital assets worth thousands of dollars. The attackers set up nearly 500 decoy websites, including renowned NFT collection sites and marketplaces. One particular drainer stole 1,055 NFTs

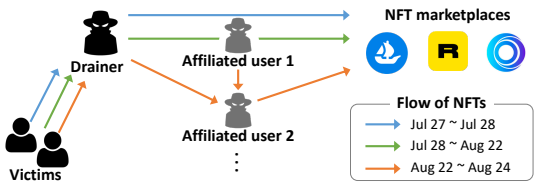


Fig. 7: Trading pattern of a drainer related to North Korean threat actors.

in total. This actor exhibited a unique liquidation method, as illustrated in Figure 7. On July 27, 2022, the NFTs stolen from victims began moving to this drainer's wallet. Over the next day, these stolen NFTs were swiftly offloaded on OpenSea. Starting July 28, newer stolen NFTs were transferred to an account, *affiliated user 1*, which proceeded to sell them. By August 22, the drainer started directing newer stolen NFTs to a new account, *affiliated user 2*, and any unsold NFTs of *affiliated user 1* were also transferred to the latter. This shift suggests a strategic maneuver to employ a secondary affiliated account for continued sales, especially when the previous one neared detection. Cumulatively, they employed a network of 15 affiliated accounts to optimize their sales strategy. Surprisingly, this account persisted in its draining activities and monetization until May 18, 2023. This persistence underscores the importance and timeliness of the real-time detection system.

Another noteworthy incident unfolded on October 21, 2022, which was started by the scammer known as *Monkey Drainer* [35], [49]. The drainer forged multiple accounts, mimicking influential Twitter accounts associated with the NFT community, such as those linked to the *RTFKT* collection and *Bored Ape Yacht Club* (BAYC) marketplace. They then disseminated posts that directed users to counterfeit NFT websites, baiting them with the promise of rewards or benefits. Over four days, assets amounting to roughly $3.5 million, including 251 NFTs, were stolen. After draining, the stolen NFTs were quickly shifted to four associated users, who in turn sold them shortly after acquisition.

VIII. Related Work

Suspicious behaviors in NFT markets: With the increasing popularity of NFTs, suspicious activities targeting NFTs are also rising. A few studies exist for analyzing security issues in the NFT ecosystem, such as wash tradings and shill bidding. However, to the best of our knowledge, we are the first to perform an in-depth study of NFT drainers and propose an NFT drainer detection system.

Das et al. [21] conducted a comprehensive study of design weaknesses originating from the NFT marketplaces and external entities. Also, they investigated various types of fraudulent user activities occurring in NFT marketplaces, such as counterfeit NFT creation, wash trading, and shill bidding. Von et al. [56] quantified market abuse in the NFT ecosystem with their proposed NFT wash trading detection algorithm. Roy et al. [45] conducted a longitudinal analysis of Twitter accounts that consistently promote fraudulent NFT collections through giveaway competitions and NFT phishing attacks.

Ethereum phishing scam detection: Ethereum phishing scam detection can be categorized into two main types: feature-based and graph-based approaches.

Chen et al. [18] extracted 119-dimensional statistical features to consider the 1-order neighbors of the node as well as the node itself. They used a LightGBM-based dual-sampling ensemble algorithm to classify phishing nodes.

Another line of research focuses on network representation. Wu et al. [58] proposed Trans2Vec, which is a modified random walk-based network embedding method with biases of transaction amount and timestamp for neighbor sampling. Chen et al. [17] introduced E-GCN, the first Ethereum phishing scam detection method based on Graph Neural Networks (GNN). They extracted 8-dimensional statistical features, such as in/out-degree, number of neighbors, etc., and fed them into a GCN for embedding. Li et al. [32] constructed edge representations from transaction records to capture the temporal relationship between users. The edge representations are aggregated into node representations and used to obtain structural features using GCN. Unlike the above works that approached using node classification, Zhang et al. [59] regarded the problem as a graph classification. They used hierarchical graph pooling layers to extract node-level representations, which were then aggregated to form graph-level representations.

However, the works discussed above are difficult to apply to NFT drainers detection due to the characteristics of the NFT ecosystem, as mentioned in Section II-A.

Graph Neural Network: In recent years, deep learning methods have achieved remarkable performance in various fields. Deep neural networks have also been applied to graph data to leverage the structural properties of graphs.

Graph Convolution Networks (GCNs) [31] is one of the most prominent graph neural network models. GCNs perform convolution operations on graph data and learn embeddings of nodes by aggregating features from neighboring nodes. Unlike GCNs, which uses information from adjacent nodes as is, Graph Attention Networks [55] utilize information from neighbors by using node attention. Multi-head attention is used to learn a number of attentions, and the node features obtained from each attention are concatenated to form a single feature. GraphSAGE [27] is an inductive GNN model, which generalizes the unobserved nodes. By incorporating node features into the learning algorithm and aggregator functions, it can learn the distribution of neighboring node features and the topological structure for the neighbors of each node.

IX. Conclusion

NFT phishing scams are a significant threat to the NFT trading ecosystem. Despite the increasing damages caused by NFT drainers, their behaviors are not well studied. To conduct an in-depth study on NFT drainers, we construct NFT phishing scam datasets. We verify that they have different transaction patterns compared to regular users. Based on our measurement results, we propose a detection model, *DRAINCLoG*, tailored to detect drainers in the NFT environment. *DRAINCLoG* is able to generate a user representation that considers NFT transaction context and social context. Evaluated on real-world NFT transaction data, we verify our model's effectiveness and robustness. We believe that our findings and detection method will contribute to the security NFT ecosystem.

Acknowledgement

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) [RS-2023-00215700, Trustworthy Metaverse: blockchain-enabled convergence research].

References

[1] "Chainabuse webpage," https://www.chainabuse.com/, [Online; accessed 14-September-2022].

[2] "Cryptoscamdb," https://cryptoscamdb.org/, [Online; accessed 14-September-2022].

[3] "Elliptic webpage," https://www.elliptic.co/, [Online; accessed 14-September-2022].

[4] "Nftgo," https://nftgo.io/, [Online; accessed 14-September-2022].

[5] "Scamsniffer webpage," https://scamsniffer.io/, [Online; accessed 14-September-2022].

[6] O. P. ADAM DARRAH and S. MITCHELL, "Flash report: Nft drainer claims to bypass cryptocurrency wallet update," https://www.zerofox.com/blog/flash-report-nft-drainer-claims-to-bypass-cryptocurrency-wallet-update/, 2022, [Online; accessed 22-November-2022].

[7] L. Akoglu and C. Faloutsos, "Anomaly, event, and fraud detection in large network datasets," in *Proceedings of the sixth ACM international conference on Web search and data mining*, 2013, pp. 773–774.

[8] R. Barber, "Nft statistics, facts & trends in 2022: All you need to know about non-fungible tokens," https://www.cloudwards.net/nft-statistics/, 2022, [Online; accessed 21-September-2022].

[9] T. Block, "Ethereum nft marketplace monthly volume," https://www.theblock.co/data/nft-non-fungible-tokens/marketplaces/nft-marketplace-monthly-volume, 2022, [Online; accessed 5-October-2023].

[10] D. V. Boom, "Nft investors lose $1.7m in opensea phishing attack," https://threatpost.com/nft-investors-lose-1-7m-in-opensea-phishing-attack/178558/, 2022, [Online; accessed 14-September-2022].

[11] ——, "Seth green loses $200k bored ape yacht club nft in phishing scam," https://www.cnet.com/personal-finance/seth-green-loses-200k-bored-ape-yacht-club-nft-in-phishing-scam/, 2022, [Online; accessed 14-September-2022].

[12] J. Brassell, "Metamask gets new 'set approval for all'' controls to boost security," https://www.beyondgames.biz/25286/metamask-gets-new-set-approval-for-all-controls-to-boost-security/, 2022, [Online; accessed 22-November-2022].

[13] Chainabuse, "Chainabuse public api (v1.2)," https://docs.chainabuse.com/docs/welcome-to-chainabuse-api, 2023, [Online; accessed 5-October-2023].

[14] ——, "Faq- are the reports open source? where do they go?" https://cryptoscamdb.org/faq, 2023, [Online; accessed 5-October-2023].

[15] ——, "Spam detection and reports acuracy," https://docs.chainabuse.com/docs/verifying-the-accuracy-of-reported-information, 2023, [Online; accessed 5-October-2023].

[16] H. Chen, M. Pendleton, L. Njilla, and S. Xu, "A survey on ethereum systems security: Vulnerabilities, attacks, and defenses," *ACM Computing Surveys (CSUR)*, vol. 53, no. 3, pp. 1–43, 2020.

[17] L. Chen, J. Peng, Y. Liu, J. Li, F. Xie, and Z. Zheng, "Phishing scams detection in ethereum transaction network," *ACM Transactions on Internet Technology (TOIT)*, vol. 21, no. 1, pp. 1–16, 2020.

[18] W. Chen, X. Guo, Z. Chen, Z. Zheng, and Y. Lu, "Phishing scam detection on ethereum: Towards financial security for blockchain ecosystem." in *IJCAI*, 2020, pp. 4506–4512.

[19] J. Cui, K. Kim, S. H. Na, and S. Shin, "Meta-path-based fake news detection leveraging multi-level social context information," in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 2022, pp. 325–334.

[20] A. DARRAH, "Flash report: Nft drainer claims to bypass cryptocurrency wallet update," https://www.zerofox.com/blog/flash-report-nft-drainer-claims-to-bypass-cryptocurrency-wallet-update/, 2022, [Online; accessed 14-September-2022].

[21] D. Das, P. Bose, N. Ruaro, C. Kruegel, and G. Vigna, "Understanding security issues in the nft ecosystem," *arXiv preprint arXiv:2111.08893*, 2021.

[22] T. Desk, "Here's what you can do if your nfts are stolen on opensea nft marketplace," https://indianexpress.com/article/technology/crypto/heres-what-you-can-do-if-your-nfts-are-stolen-on-opensea-nft-marketplace-8085992, 2022, [Online; accessed 14-September-2022].

[23] J. Ellis, "Metamask add feature to stop wallet drainer nft scams," https://nftevening.com/metamask-add-feature-to-stop-wallet-drainer-nft-scams/, 2022, [Online; accessed 22-November-2022].

[24] Ethereum.org, "Non-fungible tokens (nft)," https://ethereum.org/en/nft/, 2022, [Online; accessed 14-September-2022].

[25] Etherscan, "Etherscan webpage," https://etherscan.io/, [Online; accessed 14-September-2022].

[26] ——, "Report/flag address," https://info.etherscan.com/report-address/, 2023, [Online; accessed 5-October-2023].

[27] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *Advances in neural information processing systems*, vol. 30, 2017.

[28] R. Haqshanas, "Uniswap users fall victim to a usd 8m nft phishing attack, binance pulls false alarm," https://cryptonews.com/news/uniswap-users-fall-victim-to-a-usd-8m-nft-phishing-attack-binance-pulls-false-alarm.htm, 2022, [Online; accessed 14-September-2022].

[29] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intelligent Systems and their applications*, vol. 13, no. 4, pp. 18–28, 1998.

[30] S. Kaarus, "Elliptic: $100m in nfts stolen via scams in the past year," https://coingeek.com/elliptic-100m-in-nfts-stolen-via-scams-in-the-past-year/, 2022, [Online; accessed 14-September-2022].

[31] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[32] S. Li, G. Gou, C. Liu, C. Hou, Z. Li, and G. Xiong, "Ttagn: Temporal transaction aggregation graph network for ethereum phishing scams detection," in *Proceedings of the ACM Web Conference 2022*, 2022, pp. 661–669.

[33] lunaray, "New nft wallet-draining exploit-degen meta," https://medium.com/coinmonks/new-nft-wallet-draining-exploit-degen-meta-f81da02adb6f, 2022, [Online; accessed 14-September-2022].

[34] ——, "This nft scam as a service is a scam," https://cujo.com/nft-scam-as-a-service-a-scam/, 2022, [Online; accessed 14-September-2022].

[35] X. Luo, "Phishing scammer has drained $1m in crypto and nfts in past 24 hours, says on-chain sleuth," https://www.coindesk.com/business/2022/10/25/phishing-scammer-has-drained-1m-in-crypto-and-nfts-in-past-24-hours-says-on-chain-sleuth/, 2022, [Online; accessed 5-October-2023].

[36] C. LYONS, "North korean hackers stealing nfts using nearly 500 phishing domains," https://cointelegraph.com/news/north-korean-hackers-stealing-nfts-using-nearly-500-phishing-domains, 2022, [Online; accessed 5-October-2023].

[37] masa, "Nft collection meaning and how many nfts are in a collection?" https://bitkan.com/learn/nft-collection-meaning-and-how-many-nfts-are-in-a-collection-525, 2022, [Online; accessed 21-September-2022].

[38] Metamask, "What is a token approval?" https://metamask.zendesk.com/hc/en-us/articles/6174898326683#h_01GA1DZF6GH7EAJV86X6D839H6, 2022, [Online; accessed 14-September-2022].

[39] M. Nadini, L. Alessandretti, F. Di Giacinto, M. Martino, L. M. Aiello, and A. Baronchelli, "Mapping the nft revolution: market trends, trade networks, and visual features," *Scientific reports*, vol. 11, no. 1, pp. 1–11, 2021.

[40] NFTGo, "Market overview," https://nftgo.io/analytics/market-overview, 2023, [Online; accessed 5-October-2023].

[41] OpenSea, "Opensea webpage," https://opensea.io/, 2022, [Online; accessed 14-September-2022].

[42] ——, "What is opensea's stolen item policy?" https://support.opensea.io/hc/en-us/articles/4815371492499-What-is-OpenSea-s-stolen-item-policy, 2022, [Online; accessed 14-September-2022].

[43] R. Ramesh, "North korean hackers steal nfts via phishing websites," https://www.bankinfosecurity.com/north-korean-hackers-steal-nfts-via-phishing-websites-a-20803, 2022, [Online; accessed 5-October-2023].

[44] Rarible, "Rarible webpage," https://rarible.com/, 2022, [Online; accessed 14-September-2022].

[45] S. S. Roy, D. Das, P. Bose, C. Kruegel, G. Vigna, and S. Nilizadeh, "Demystifying nft promotion and phishing scams," *arXiv preprint arXiv:2301.09806*, 2023.

[46] N. SALHUANA, "Nft theft: Here's how the dark side of web3 gets away with it," https://nftnow.com/features/nft-theft-heres-how-the-dark-side-of-web3-gets-away-with-it/, 2022, [Online; accessed 14-September-2022].

[47] ScamSniffer, "Security check," https://docs.scamsniffer.io/docs, 2022, [Online; accessed 5-October-2023].

[48] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. v. d. Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks," in *European semantic web conference*. Springer, 2018, pp. 593–607.

[49] A. Shirinyan, "Notorious scammer "monkey drainer" steals $1 million in eth, here's how," https://u.today/notorious-scammer-monkey-drainer-steals-1-million-in-eth-heres-how, 2022, [Online; accessed 5-October-2023].

[50] C. Stouffer, "Nft scams: 10 types + how to avoid nft fraud," https://us.norton.com/internetsecurity-online-scams-nft-scams.html#, 2022, [Online; accessed 14-September-2022].

[51] R. Tan, Q. Tan, P. Zhang, and Z. Li, "Graph neural network for ethereum fraud detection," in *2021 IEEE International Conference on Big Knowledge (ICBK)*. IEEE, 2021, pp. 78–85.

[52] B. Toulas, "$8 million stolen in large-scale uniswap airdrop phishing attack," https://www.bleepingcomputer.com/news/security/8-million-stolen-in-large-scale-uniswap-airdrop-phishing-attack/, 2022, [Online; accessed 14-September-2022].

[53] Twitter, "Twitter webpage," https://twitter.com, [Online; accessed 14-September-2022].

[54] I. Vasile, "Crypto and nft airdrops: What are they, and how do they work?" https://beincrypto.com/learn/crypto-and-nft-airdrop/, 2022, [Online; accessed 14-September-2022].

[55] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[56] V. von Wachter, J. R. Jensen, F. Regner, and O. Ross, "Nft wash trading: Quantifying suspicious behaviour in nft markets," *arXiv preprint arXiv:2202.03866*, 2022.

[57] S. Waldman, "Metamask looks to reduce nft scams with updated "set approval for all" featurem," https://hypemoon.com/2022/7/metamask-set-approval-for-all-update-nft-scams/, 2022, [Online; accessed 14-September-2022].

[58] J. Wu, Q. Yuan, D. Lin, W. You, W. Chen, C. Chen, and Z. Zheng, "Who are the phishers? phishing scam detection on ethereum via network embedding," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2020.

[59] D. Zhang, J. Chen, and X. Lu, "Blockchain phishing scam detection via multi-channel graph classification," in *International Conference on Blockchain and Trustworthy Systems*. Springer, 2021, pp. 241–256.

[60] G. Zhang, Z. Li, J. Huang, J. Wu, C. Zhou, J. Yang, and J. Gao, "efraudcom: An e-commerce fraud detection system via competitive graph neural networks," *ACM Transactions on Information Systems (TOIS)*, vol. 40, no. 3, pp. 1–29, 2022.

APPENDIX A
IMPLEMENTATION DETAILS

The embedding size of both *NFT Transaction Context Extractor* (TCE) and *Social Context Extractor* is set to 64, and the learning rate is set to 6e-4 and 2e-3, respectively. In TCE, the number of attention heads is set to 8. The regularization parameter and gamma of SVM are set to 0.1 and 0.1, respectively. As the classifier for each baseline, we choose the one with better performance between SVM and lightGBM. Feature-based methods, E-GCN, and N-GCN use lightGBM, and the rest use SVM.

APPENDIX B
CRITERIA OF POTENTIAL DRAINER

C1. Users possessing *suspicious NFTs* that were banned from trading on OpenSea due to suspicious activities.

C2. Users who have consistently gifted NFTs to another account, where the receiving account holds *suspicious NFTs*.

C3. Users who engaged in multiple Ether or NFT gifts over time with accounts labeled as phishing attackers on Etherscan.

C4. Users who are newly reported.

APPENDIX C
DISTRIBUTION OF TRANSACTIONS

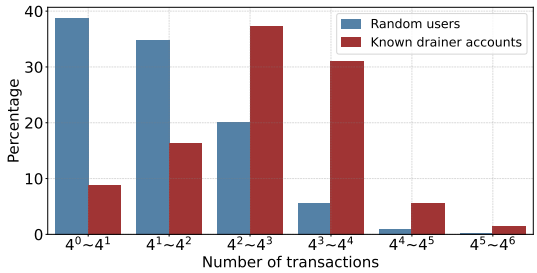


Fig. 8: Distribution of random users' transactions and drainers' transactions between January 1, 2022, and July 31, 2022.

APPENDIX D
FEATURE ANALYSIS

We analyze the activities of 645 drainers, 637 affiliated users, and a sample of 10,000 regular users who were active between January 1st and August 31st, 2022. We compare these three groups by plotting the cumulative distribution functions (CDFs) for 19 dimensions (Figure 9). To make these visualizations easier to interpret, we limit the x-axis of each graph to 100. We can observe that drainers and affiliated users exhibit distinct behavior patterns compared to regular users.

A. Number of transactions

It is apparent that drainers are more active in selling, gifting-in, and gifting-out transactions. However, they rarely participate in buying and minting transactions compared to regular users. These results suggest that the primary focus of drainers in the NFT ecosystem is on draining NFTs.

Upon closer examination, trends between drainers and affiliated users show little difference in selling, gifting-in, and gifting-out transactions. Interestingly, unlike drainers, affiliated users are more active in buying and minting NFTs. This behavior sets them apart from both drainers and regular users.

B. Number of collections

We find that the number of collections for each transaction type is generally smaller than the number of transactions. It is well known that NFT users tend to form communities based on specific collections and trade within those communities [39]. Although drainers have different intentions from regular users, they also trade a smaller number of collections than transactions. This is because they steal NFTs that are collected by regular users. However, due to their high levels of gifting-in, gifting-out, and selling transactions, drainers have a greater diversity of collections in those three types of transactions. As a result, they exhibit a significant difference from regular users in the number of NFT collections gifted-in, gifted-out, and sold.

In contrast, affiliated users are observed to actively participate in trading a wide range of collections across all types of transactions. This is because they are actively engaged in all types of transactions.

C. Number of neighbors

We analyze the number of neighbors a user has for each transaction type by considering the accounts with which a user has made a transaction as their neighbors. For buying and selling transactions, the distribution of neighbors is similar to that of the transactions themselves. However, for gifting-in and gifting-out transactions, the distribution of neighbors is significantly different. Most users gift NFTs to only a few neighbors, while they sell or buy NFTs with many neighbors. This suggests that NFT users have specific relationships through gifting NFTs, given that gifting is a process of transferring ownership without any payment.

This phenomenon is more pronounced in drainers than in regular users. When drainers steal NFTs, they may acquire all the tokens from each victim, resulting in a smaller number of gifting-in neighbors than the number of NFTs gifted-in. Additionally, drainers only gift NFTs to a few affiliated users, resulting in a much smaller number of gifting-out neighbors than the number of NFTs gifted-out.

In the case of affiliated users, their distributions of gifting-in and gifting-out are as expected, similar to those of drainers.

D. Ratio & Frequency & Active timespan

Drainers have a higher gift-in ratio than regular users and affiliated users. This is because they do not engage in buying and minting NFTs, but rather steal a large number of NFTs in a short period of time. This results in a high frequency of gifting-in transactions. Also, drainers are more likely to transfer out their NFTs than regular users. They also sell their NFT much more frequently in a short active timespan. The behavior of affiliated users falls between that of drainers and regular users. They are active in all types of transactions, particularly gifting-in, which results in a high gift-in ratio similar to drainers. However, their active period is not as short as drainers.

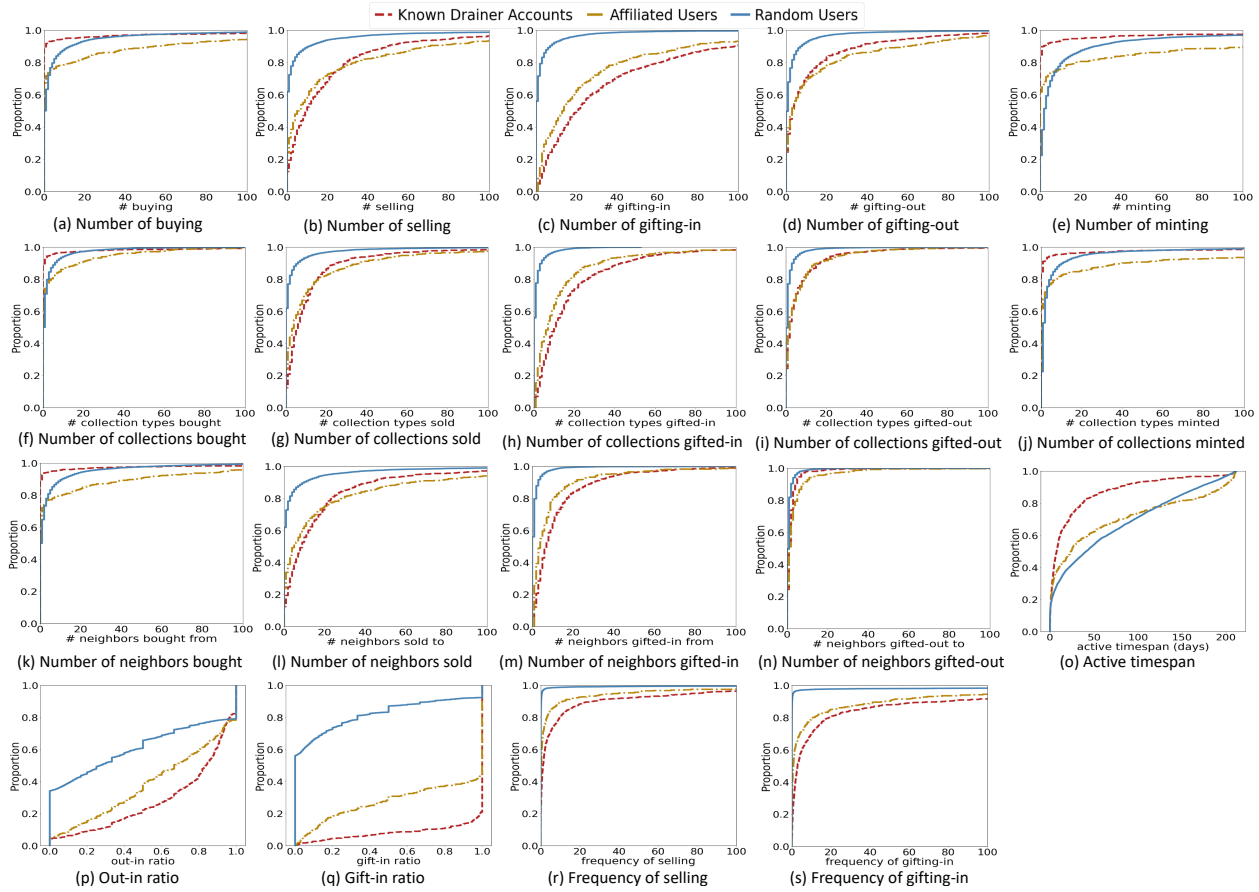


Fig. 9: CDFs of 19 behavioral features according to the user types.

APPENDIX E
EVASION ATTACK RESULTS WITH VARIED PARAMETER X ON D_1 & D_3

TABLE XI: The results of evasion attacks on D_1 and D_3 datasets averaged over 5 runs.

	Dataset (ratio)		D_1 (1:10)			D_1' (1:10)			D_3 (1:1000)			D_3' (1:1000)		
	L	X	Pre.	Rec.	F1	Pre.	Rec.	F1	Pre.	Rec.	F1	Pre.	Rec.	F1
Attack3	10	1	0.872	0.116	0.205	0.969	0.597	0.738	0.084	0.120	0.099	0.226	0.610	0.329
		10	0.868	0.112	0.199	0.966	0.582	0.725	0.082	0.117	0.096	0.222	0.595	0.323
		60	0.866	0.110	0.195	0.966	0.574	0.719	0.081	0.115	0.095	0.220	0.591	0.320
	30	1	0.881	0.126	0.221	0.970	0.659	0.784	0.088	0.127	0.104	0.236	0.681	0.350
		10	0.868	0.112	0.199	0.968	0.634	0.766	0.081	0.116	0.096	0.231	0.653	0.341
		60	0.852	0.098	0.176	0.965	0.625	0.758	0.074	0.104	0.086	0.222	0.635	0.328
	50	1	0.903	0.155	0.265	0.972	0.684	0.802	0.107	0.157	0.127	0.265	0.704	0.384
		10	0.889	0.133	0.231	0.973	0.669	0.793	0.093	0.133	0.110	0.263	0.677	0.378
		60	0.873	0.114	0.202	0.970	0.644	0.774	0.082	0.118	0.097	0.264	0.648	0.374
Attack4	10	1	0.525	0.018	0.035	0.953	0.421	0.583	0.015	0.020	0.017	0.170	0.477	0.251
		10	0.525	0.018	0.035	0.950	0.437	0.598	0.016	0.021	0.018	0.172	0.482	0.254
		60	0.551	0.020	0.039	0.952	0.425	0.587	0.017	0.023	0.019	0.171	0.474	0.251
	30	1	0.525	0.018	0.035	0.958	0.601	0.738	0.016	0.021	0.018	0.199	0.630	0.302
		10	0.525	0.018	0.035	0.955	0.576	0.718	0.016	0.021	0.018	0.192	0.610	0.292
		60	0.426	0.012	0.024	0.956	0.563	0.709	0.010	0.013	0.012	0.183	0.587	0.278
	50	1	0.430	0.012	0.024	0.964	0.663	0.786	0.011	0.014	0.012	0.222	0.691	0.335
		10	0.468	0.014	0.028	0.961	0.642	0.770	0.012	0.016	0.014	0.211	0.658	0.319
		60	0.430	0.012	0.024	0.961	0.634	0.764	0.011	0.014	0.012	0.207	0.651	0.314
	DRAINCLoG		**0.989**	**0.622**	**0.763**	**0.989**	**0.622**	**0.763**	**0.448**	**0.628**	**0.523**	**0.448**	**0.628**	**0.523**

To evaluate *DRAINCLoG*'s detection capabilities against evasion tactics, we adjusted previous evaluation datasets. These modifications were guided by the specific attack strategy and its attack level (L), where $L \in \{10, 30, 50\}$. For Attack 3, we changed $L\%$ of gifting-in transactions to buying transactions by sending X% of the average sale price of each NFT to those victims, where $X \in \{1, 10, 60\}$. For Attack 4, we integrated the tactics of Attack 1 and Attack 2, both at level 50, with the methods of Attack 3. We re-trained the classifier layer of *DRAINCLoG* (SVM) with 3% of evasion attackers and evaluated the remaining evasion attackers on datasets D_1' and D_3'.