IEEE ICBC'24 1570980590

Exploring collaboration in Ethereum scams

***Abstract*—Similar to all other cryptocurrency platforms, Ethereum is constantly confronted with malicious activities. In recent years, research efforts have targeted detection and mitigation of malicious activities and the associated accounts within the Ethereum ecosystem. Yet, the malicious accounts represent only a small visible part of substantial collaborative network enabling these activities. In this work, we offer the first analysis of this collaborative network and the corresponding affiliate accounts that often remain hidden from detection. We propose a clustering approach for the a characterization of these affiliate accounts. Our research findings lay the foundation for detection of the collaborative networks enabling Ethereum scams.**

***Index Terms*—Ethereum, scams, blockchain, security**

I. Introduction

With rapid increase in popularity, Ethereum has faced a similarly unprecedented amount of malicious activities. Some of these activities exploit vulnerabilities in the infrastructure, while others rely on social engineering techniques to deceive the victims. With a significant financial impact, these scams threaten the viability of blockchain platforms and hinder their adoption in industry. To mitigate this malicious presence, numerous research studies have focused on detection of malicious activities. In recent years, research studies focused on detection of either generic malicious transactions [1]–[3], or known illicit activities. For example, honeypot smart contracts [4], phishing attacks [5]–[9], and Ponzi scheme activities [10]–[14]. The overwhelming majority of these research studies focus solely on identifying a specific type of malicious behaviour, thus limiting their applicability in real environment.

A more few solutions were developed to detect suspicious transactions through instrumentation of Ethereum nodes or contracts [15]–[18] or replaying historic transactions (e.g., HORUS [19], EthScope [9]).

Despite more than a decade of research, the detection of malicious activities on the blockchain remains challenging. In practice, detection is hindered by the hidden network of adversarial accounts that facilitate malicious activities yet do not directly involve in them [20], [21]. Despite their active role, there has not been comprehensive research on understanding how collaboration between malicious and affiliate accounts on Ethereum works.

The identification of accounts affiliated with malicious activities has numerous benefits. Understanding collaboration among accounts can help detect malicious behaviour, making it possible to identify and flag accounts engaging in malicious activities, consequently preventing potential losses.

In this work, we address this gap and present our findings from a large-scale analysis of collaboration between malicious and affiliate accounts on Ethereum. For this analysis, we collected a large set of 7915 unique Ethereum addresses previously flagged as malicious. This list of known malicious accounts as ground truth, we gather all other accounts that ever had any interactions with our malicious set. We examine the interactions between malicious and other accounts and use clustering to identify accounts engaging in coordinated malicious activity. Finally, we identify intermediary accounts that are not directly involved in malicious activities, but form a hidden network of adversarial accounts that connect with visible malicious accounts. These intermediary accounts facilitate transfer of funds. They are used to obfuscate the origin of the transferred funds and, consequently, make it more difficult to trace these funds' movement.

To summarize, our contributions are:

- We offer a first large-scale analysis of collaboration between known malicious Ethereum accounts and their hidden network of affiliates. We characterize the collaboration patterns and shed light on the intricate network of fund transfers that underline various types of scams and fraudulent activities within the Ethereum space. Through our comprehensive analysis, we offer insights into the behavior of malicious actors and the network dynamics that shape their often hidden activities.
- We propose a clustering approach to investigate affiliate account network. We design and evaluate a set of features that contribute the most in affiliate account analysis.
- We report our findings and release the dataset of malicious and affiliate accounts collected in this study to facilitate further research in this area[1].

II. Related work

The continued growth of cryptocurrencies has attracted significant research attention. A few studies focused specifically on characterization of malicious activity on the blockchain, e.g., characterization of Bitcoin scams [22], [23], analysis of victim's activity in Bitcoin Ponzi schemes [24]. However, the majority aimed to detect malicious accounts. The earlier studies were primarily focused on detecting vulnerabilities in smart contracts based on the patterns of known insecure behaviour in code (e.g., Slither [25], Vandal [26], Oyente [27], Securify [28], Solc-Verify [29], ContractWard [30], and Mythril [31]). More recent approaches offered detection of malicious accounts already deployed on the chain. Graph learning methods have emerged as popular approaches for identifying specific categories of malicious activities within Ethereum. For instance, Wen et al. [7] proposed a method based on analyzing transaction discrepancies involving phishing accounts and their immediate neighbors. Similar adjacency-based approaches have been developed by other researchers for the detection

[1] anonymous

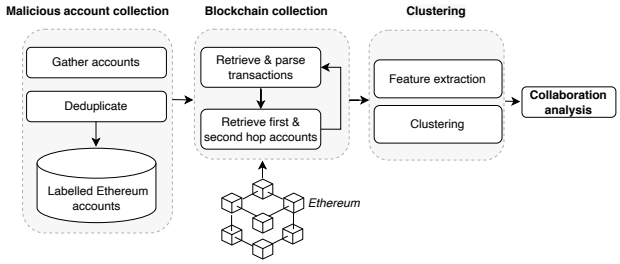

Fig. 1: The overview of the analysis pipeline

of phishing [5], [6], [8], [9], [32] and Ponzi scams [11], [13]. Nonetheless, these techniques are limited in scope, primarily addressing a narrow spectrum of malicious behavior and often lacking a comprehensive analysis of interconnected accounts associated with malicious activities. More broad detection of other various types of malicious activities within the Ethereum was offered based on machine learning analysis, e.g., [1], [4], [33]. Given the significance of pseudo-anonymity within the blockchain ecosystem, numerous studies focused on de-anonymizing the associated entities and their relationships. Address clustering was explored as the primary means in this analysis. To de-anonymize accounts researchers leveraged network traffic information [34]–[36], smart contract source and bytecode [37], transaction histories of addresses [38]–[43], unconfirmed transactions [44]. Similar efforts have been extended to the Ethereum blockchain [45], [46], with Victor et al.'s approach [46], grounded in address clustering, holding the potential to identify recurring attackers within the Ethereum ecosystem.

Typically, these approaches focus solely on the malicious accounts themselves. We, on the other hand, extend this scope and examine malicious accounts beyond those directly involved in malicious behavior. We characterize connections linking malicious accounts with seemingly innocuous entities.

III. Analysis methodology

The overview of our analysis pipeline is presented in Figure 1. It comprises of four main steps: ① Collection of malicious accounts, ② Blockchain data collection & parsing, ③ Clustering and ④ Collaboration analysis.

A. Collection of malicious accounts

Over the years, a large portion of the studies focusing on blockchain security gathered datasets of benign and malicious Ethereum accounts. We analyzed 2,376 papers focused on security of Ethereum[2] and gathered references to all publicly available datasets containing malicious accounts mentioned or used in these studies. We accumulated 36,600 Ethereum accounts associated with malicious activity, mostly related to phishing, from the following sources covering a period from 2017 to 2023: CryptoScamDB[3], PhishingDB[4], EtherScamDB[5], Ethereum repository of malicious accounts[6], Ethereum phishing accounts [7], Forta labelled malicious smart contracts[8] and phishing contracts[9], EtherScan set of phish-hack addresses[10], malicious datasets used in papers [1], [5], [14], [47]–[49]. The majority of these sources contained duplicate accounts. We further confirmed that most of these accounts were also labelled by Etherscan, a block explorer platform for Ethereum blockchain, as malicious or dangerous. These accounts were deduplicated and validated to ensure the structural correctness of each address, the final list contained 7,915 malicious Ethereum addresses. The summary of the collected set is shown in Table I. Accounts that were consistently and uniquely labelled across sets are presented with the corresponding label, all other accounts are considered to be multi-labelled. We further refer to the accounts present in this set as malicious accounts.

TABLE I: The malicious Ethereum account set

Total	Exploit	Heist	Malicious	Phish-hack	Phishing	Ponzi	Scam	Multi label
7,915	194	92	35	3,625	19	289	58	3,603

B. Blockchain data collection & parsing

Our dataset of 7,915 malicious accounts has been used as a reference for further data collection. For each address in our dataset, we queried the Etherscan platform and retrieved the corresponding transaction history. For each normal and internal transaction, we retrieved the timestamp, sender and receiver addresses, amount of Ether transferred, data field, bytecode, and if present, source code. This resulted in the creation of a new data set containing unique transactions directly associated with our list of malicious addresses.

Subsequently, we extracted the Ethereum sender addresses associated with the source of each transaction, identified by the transaction field "from". We then extracted the Ethereum receiver addresses associated with the destination, i.e., the transaction field "to". As such, we created a new *first hop* set encompassing all addresses that directly interacted with malicious addresses in our initial set.

We then repeated this process for the first hop addresses to generate the *second hop* set which included Ethereum addresses with the second-degree relation to our original malicious addresses.

C. Clustering

We observed that the amount of participating addresses across hops grows exponentially, with the number of transactions growing even faster. To facilitate our analysis, we employ clustering and derive 4 groups of features listed in Table II: *General features* represent the scope and amount of incoming and outgoing transaction activity for all collected

[2]The set of the relevant papers was extracted using Google Scholar and the search terms: "Ethereum", "malicious account", "malicious", "phishing", "scam", "ponzi", "exploit".

[3]https://github.com/CryptoScamDB/blacklist

[4]https://github.com/brianleect/etherscan-labels

[5]https://github.com/MrLuit/EtherScamDB

[6]https://github.com/MyEtherWallet/ethereum-lists

[7]https://github.com/yuanqi7/Phishing-Detection-on-Ethereum

[8]https://github.com/forta-network/labelled-datasets

[9]https://raw.githubusercontent.com/forta-network/labelled-datasets/main/labels/1/phishing_scams.csv

[10]https://github.com/dawsbot/evm-labels

TABLE II: Clustering features

GENERAL FEATURES
Number of incoming/outgoing transactions
Number of addresses with incoming/outgoing transactions
Amount of incoming/outgoing Ether
Average number of incoming$^{\#/-}$/outgoing$^{\#/-}$ transactions per month
Average amount of incoming$^{\#/-}$/outgoing$^{\#/-}$ ether per month
Average time between incoming/outgoing transactions
standard deviation time between incoming/outgoing transactions
MALICIOUS FEATURES
Number of incoming$^{*\#/-}$/outgoing$^{\#/\#}$ malicious transactions
Number of addresses with incoming$^{\#/\#}$/outgoing$^{\#/\#}$ malicious transactions
Amount of incoming$^{\#/-}$/outgoing$^{\#/\#}$ malicious ether
Fraction of incoming$^{-/\#}$/outgoing malicious transactions to all incoming transactions
Fraction of malicious addresses with incoming/outgoing transactions to all addresses with incoming/outgoing transactions
Fraction of incoming$^{-/\#}$/outgoing ether from malicious accounts to all incoming/outgoing ether
Average time between incoming$^{*\#/-}$/outgoing$^{-/\#}$ malicious transactions
Standard deviation time between incoming$^{*\#/-}$/outgoing$^{-/\#}$ malicious transactions
ADDITIONAL FEATURES
Number of months account is active$^{*\#/\#}$
Number of transactions associated with malicious$^{\#/\#}$/non-malicious addresses
Number of self-transactions
Number of transactions with other addresses
Number of normal transactions with ether value of 0$^{*\#/\#}$
min* /max$^{*\#/\#}$ /sum$^{*\#/\#}$ of transferred ether value
Number of addresses related to this address via transactions$^{*\#/\#}$
Label description
Whether the address is considered malicious or not at a current hop
Original Etherscan label

* selected features in first-hop analysis

$^{\#/\#}$ selected features in second-hop analysis, (dissemination/concentration affiliates)

addresses. *Malicious features* represent the extent of the scope and amount of transaction activity associated with malicious addresses. *Additional features* provide context information that may help to put account behaviour in context. *Labels* when relevant, represent if the address should be considered malicious at that particular point of view. For example, if a benign address is determined to be a malicious collaborator, then in the following hop it may be flagged as affiliate malicious. Overall, we chose features that encompass the regularity and extent of overall activity and the amount of interactivity with flagged malicious accounts.

The sheer number of addresses and transactions in our analysis proved to be unscalable for many clustering methods, hence, we employ the k-means approach, a computationally efficient unsupervised machine learning algorithm used for partitioning a dataset into k distinct clusters. It can handle large datasets with a moderate number of dimensions. It is widely used in practice due to its speed and scalability. To determine the optimal number of clusters for k-means, we started with the Elbow curve. It is used to determine the optimal number of clusters (k) in k-means clustering or other clustering algorithms. It helps identify the point of diminishing returns, where the addition of more clusters does not significantly improve the clustering performance. It uses inertia as a measure. Inertia refers to the total of squared distances between samples and their nearest cluster centre, and our aim is to minimize this value as much as we can. The apparent "elbow" of the curve signifies the optimal number of clusters. We also used silhouette score, which measures the compactness and separation of clusters, to support our selection of k. The silhouette score is a widely used metric for assessing the quality of clusters. It takes into account both how closely data points are grouped within clusters (compactness) and how well-separated clusters are from each other (separation). For each data point, the silhouette score is computed using the following formula:

$$s_i = \frac{b_i - a_i}{max(a_i, b_i)}$$

where a_i represents the average distance from data point i to the other points within the same cluster, and b_i represents the average distance from data point i to points in the nearest neighbouring cluster. The silhouette score for the entire dataset is computed as an average of all individual silhouette scores.

TABLE III: Collected Ethereum addresses

First hop	
Total addresses	275,636
receiving from malicious addresses	39,343
sending to malicious addresses	220,159
Second hop	
Total addresses	7,878,125
receiving from malicious/affiliate addresses	739,340
sending to malicious/affiliate addresses	7,420,466
first-hop affiliate addresses	16,891

D. Collaboration analysis

In the context of this study, collaboration analysis of malicious accounts refers to analysis of associate accounts involved in interactions and transfer of funds from and to malicious accounts. These associate accounts appear to be benign.

With the incoming transactions, we expect that malicious accounts either acquire funds directly from victims, or participate in the transfer of funds received by other malicious accounts. With the outgoing transactions, they either pass the funds on to other accounts, or invest the funds through crypto-mixers, anonymous exchanges, decentralized exchanges, etc.

While funds can be funnelled through known malicious accounts (e.g., phishing accounts) that also accept transfers from victims, we are primarily interested in accounts that have no direct interactions with victims and thus remain hidden from detection. We call these accounts *affiliate accounts* or affiliates in short. Note that these affiliate accounts are typically not labelled as malicious. With the goal to understand collaboration between malicious and affiliate accounts, we focus primarily on interactions between them.

IV. Experimental results

a) ***Collected data****:* The summary of the collected set is shown in Table III. For our initial set of 7,915 accounts, we collected 275,636 addresses that had some interactions with malicious accounts in the first hop. The vast majority (220,159) of these interacting accounts were sending funds to malicious accounts which is consistent with victim behaviour. 39,343 accounts had incoming transactions from the malicious set, i.e., they were receiving funds transferred by malicious accounts. These are viewed as affiliate malicious accounts. Out of these accounts, we have further identified 14 addresses currently labelled as malicious by the Etherscan platform. The rest of affiliate accounts had no labels indicating their involvement in any malicious activities.

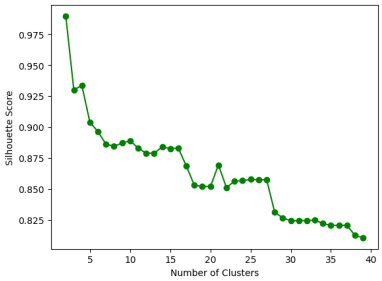

Fig. 2: Silhouette scores (first hop clustering)

The second hop data was collected from the point of view of the first hop affiliate malicious accounts. We compiled a list of all the second hop accounts to which the first hop accounts sent funds. It is important to note that only 34,561 first hop affiliates had outgoing transactions. For these second hop accounts, we collected all incoming transactions. Although we are primarily interested in propagation of funds from these affiliate addresses further down the chain, our collected data shows that 16,891 second hop (effectively, part of our first hop set) addresses have received funds directly from malicious accounts. These addresses illustrate situation in which affiliate malicious accounts interact with each other. Consequently, they exhibit a dual role: on one side, they serve as second hop addresses, and on the other, they directly receive funds from known malicious accounts.

b) ***Experimental setup****:* For our study, we developed scripts in Python3 to collect and analyze data related to known malicious addresses. The graph analysis core was implemented with the NetworkX Python module [50]. The transaction data was collected using the Etherscan API methods [51]. For visualization purposes, we used Gephi version 0.10.1. We implemented the model with `cluster.KMeans` from the Python3 library `sklearn version 1.0` [52]. For the parameters, we used $init = k - means++$, $max\_iter = 500$, $random\_state = 0$, $algorithm = lloyd$. For clustering analysis, we standardized all numerical features using the `sklearn.preprocessing.StandardScaler` which converted all our features to values between -1 and 1.

V. Clustering results

A. First hop clustering

Firstly, we analyzed the first hop addresses, focusing on the flow of funds collected by known malicious accounts. We focused only on affiliate malicious accounts that had at least one incoming transaction from one of the malicious addresses.

To determine optimal number of clusters, we calculated the Elbow curve. The resulting curve did not feature a particularly outstanding point, so we opted to employ silhouette scores, which measure the compactness and separation of clusters. We calculated the silhouette scores for the first hop set for different number of clusters, their values are presented in Figure 2. We ultimately chose $k = 14$ as our number of clusters for the first hop, as it represents a local maximum on the silhouette score graph, while maintaining relatively low inertia.

The resulting silhouette score is 0.8835 (silhouette score ranges from -1 to 1), which suggests a good separation and compactness of clusters, indicating that the algorithm has effectively grouped similar data points together.

We then calculated feature importance, represented by the absolute mean values calculated for each feature across all clusters. This analysis allowed to select features that were the most influential in forming the clusters. These features are indicated in Table II. The analysis was then redone with the selected features.

a) Results: The distribution of addresses in resulting clusters is shown in Table IV. We observe 2 primary patterns: dissemination and aggregation of funds.

Dissemination of funds: The first pattern is formed by two large clusters 0 and 7, where a number of affiliate addresses>malicious addresses. *Addresses in these clusters appear to be points of dissemination for maliciously acquired funds*, i.e., numerous malicious accounts transfer funds to a substantial quantity of affiliate accounts, we thus refer to them as *dissemination affiliates*. For example, cluster 0 appears to be the largest cluster with connections to 59.71% of all the malicious addresses via 41.37% of all transactions. All 14 affiliate addresses labelled as malicious by Etherscan appear in this cluster. Interestingly, it contains affiliate addresses involved in all types of scams present in our set (Table V). *This indicates that affiliate accounts are reused by adversaries across various malicious activities.* Similarly to cluster 0, cluster 7 does not have an exclusive connection to any type of scams.

The affiliates in cluster 0 received primarily ether within a very tight window of malicious activity, while affiliates in cluster 7 received primarily tokens from malicious addresses.

To further examine these clusters, we sampled addresses for manual analysis. From each cluster, we selected several affiliate addresses that had the smallest distance to the cluster centre, calculated via `sklearn.metrics.pairwise_distances`. Significant number of these accounts are no longer active. This is somewhat expected as our initial collection of 7,915 malicious addresses spawned a period of six years.

We observed that the affiliate malicious accounts in clusters 0 and 7 serve as a transfer point and pass funds received from the malicious accounts further to other accounts. We refer to these accounts as *intermediary affiliate accounts*. The next transfer point is either another intermediate affiliate account, a crypto exchange or a mixer account. This forms a chain of affiliate accounts that transfer money from malicious accounts. Crypto-mixers are platforms that combine the cryptocurrencies of multiple users, creating a mix of funds to obscure the source and owners of these funds. Such services are commonly employed to make financial transfers between platforms untraceable and typically do not necessitate Know Your Customer (KYC) verifications. Similarly, anonymous crypto exchanges and decentralized exchanges (DEX) are platforms that allow users to buy and sell cryptocurrencies without revealing their identity. These exchanges do not comply with "know your

TABLE IV: Clustering results of affiliate malicious addresses (the first hop set)

Cluster N	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Number of affiliate addresses	36,694	190	4	1	1	238	8	935	87	33	4	12	4	1,132
Number of affiliate addresses labelled as malicious	14	0	0	0	0	0	0	0	0	0	0	0	0	0
Number of malicious addresses with outgoing transactions to affiliate	4,726	594	2	1	538	286	7	683	915	52	213	961	480	1,347
% of malicious addresses	59.71	7.5	0.03	0.01	6.8	3.61	0.09	8.63	11.56	0.66	2.69	12.14	6.06	17.02
Number of transactions from malicious addresses	78,585	5,241	14	1	29,423	635	9	3,191	8,748	73	3,552	7,037	25,085	28,378
% of malicious transactions	41.37	2.76	0.01	0.00	15.49	0.33	0.00	1.68	4.6	0.04	1.87	3.7	13.2	14.94

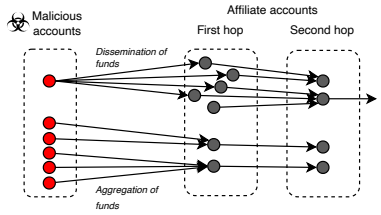

Fig. 3: Example of dissemination and aggregation of funds

customer" (KYC) laws, which means that it can be difficult to identify the parties involved in cryptocurrency transactions. This anonymity and privacy makes these exchanges attractive to adversaries who want to launder the stolen funds. Although we have not traced every intermediate affiliate account until the end, we suspect that many of these accounts end in a crypto exchange, token transaction, or a mixer account to conceal their origin.

Aggregation of funds: Addresses in clusters 1, 4, 5, 8–13 accumulate funds in a few affiliate accounts. Malicious to affiliate ratio differs from 1.2 (e.g., cluster 5) to 538 in cluster 4 where 538 malicious accounts transferred funds to 1 affiliate. *Addresses in these clusters appear to be points of concentration for maliciously acquired funds, we thus refer to them as concentration affiliates*. Addresses in all clusters were mostly receiving token transfers from malicious addresses.

As opposed to dissemination affiliates forming a chain that transfers funds from malicious accounts, we observed that the first hop addresses in concentration clusters are exchange or mixer accounts themselves. Accumulating funds from malicious accounts directly in exchange or mixer platforms, allows to quickly (within 1 hop) hide the stolen funds.

Overall, in the first hop, there are 37,629 dissemination affiliates and 1,701 concentration affiliates.

Outliers: Clusters 2, 3 and 6 are very small in both size and number of connected malicious addresses to make a broader conclusions. However, they are clear outliers and were grouped together mostly due to the amount of funds transferred through them, i.e., the affiliate addresses received the biggest sums of ether with 21-22 orders of magnitude within a small period of time.

These clusters appear to be similar in their behaviour to accounts in clusters 0 and 7. We manually investigated their behavior. They were mostly inactive and represented intermediary affiliate accounts. For example, all addresses in cluster 2 were intermediary affiliate accounts that transferred funds to Upbit Exchange. Similarly, some of affiliates in cluster 6 transferred funds to Cream Finance proxy contract and Bitfinex.

TABLE V: Distribution of malicious addresses with transactions to first hop affiliate addresses

Cluster N	0	1	2	3	4	5	6	7	8	9	10	11	12	13
exploit	99	20	1	1	21	13	2	32	54	3	58	65	19	67
heist	45	7	1	-	11	4	2	16	27	-	22	27	9	23
malicious	-	-	-	-	-	-	-	-	-	-	-	-	-	-
phish_hack	2,300	323	-	-	360	181	1	401	503	33	116	568	326	742
phishing	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ponzi	-	-	-	-	-	-	-	-	-	-	-	-	-	-
scam	29	4	-	-	1	1	1	4	3	-	-	4	2	8
multi-labelled	2,253	240	-	-	145	87	1	230	328	16	17	297	124	507

B. Second hop clustering

Overall, 7,878,125 second-hop addresses interacted with the first-hop affiliates and only 739,340 addresses had incoming transactions, i.e., received funds. This shows the vast scale of the collaboration network within two hops. Following the funds through outgoing transactions of first-hop affiliate addresses, we found that 618,738 second-hop addresses received transactions from 37,629 first-hop dissemination addresses, and 170,357 addresses that received transactions from 1,701 first-hop concentration addresses. Note that these groups are not mutually exclusive. 49,755 of second-hop addresses were found in both groups of second-hop affiliates.

We examined the transactions at large and found that 97% of first-hop concentration affiliates appeared as recipients of transactions from first-hop dissemination affiliates. This indicates that eventually intermediate affiliates appearing as dissemination points transfer funds to crypto mixers and exchange accounts. 35% of first-hop dissemination affiliates also appeared to receive transactions from first-hop dissemination affiliates. *This shows high interactivity between first-hop affiliates, with almost all dissemination affiliates propagating funds towards concentration affiliates.*

We also observed the opposite flow of funds, from concentration affiliates (crypto mixers and exchanges) to 14% of malicious addresses and 14% of first-hop dissemination affiliates. This flow corresponds to a significantly smaller amounts of ether. Typically, crypto mixers shuffle the cryptocurrencies deposited by users. The funds (minus a small service fee) can be later withdrawn to new addresses covering the origin of the funds. This is money laundering behavior that we saw we first-hop concentration affiliates.

However, in this case we also see that malicious addresses (directly or through first-hop affiliates) are leveraging crypto mixers as safe deposit accounts. We suppose they are likely withdrawing funds needed to be used in scam activities, e.g., sending money to other malicious accounts to give an appearance of active account use.

a) Clustering results: To explore the core group of participating addresses, we proceed with the second-hop clustering. For this clustering step, we decided to explore dissemination and concentration affiliates separately. We chose $k = 9$ and $k = 13$ to be the best choice for the numbers of clusters

5

TABLE VI: Clustering of second-hop affiliates (with incoming transactions from first-hop affiliates only)

Dissemination affiliates									
Cluster N	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Total addresses	**598,045**	**3**	**26**	**18**	**20,555**	**1**	**8**	**5**	**77**
Malicious addresses	1,781	0	0	0	81	0	0	1	1
First-hop dissemination affiliates	11,077	0	1	0	2,343	0	0	0	16
First-hop concentration affiliates	464	3	1	0	1,144	0	0	0	48
Second-hop-exclusive addresses	584,723	0	24	18	16,987	1	8	4	12
Transactions from first-hop addresses									
Total incoming transactions	**1,771,462**	**207,742**	**8,592**	**23**	**1,048,556**	**9**	**1,017**	**33,885**	**330,400**
Transactions received by included malicious addresses	11,257	0	0	0	3,705	0	0	2,522	465
Transactions received by included first-hop dissemination addresses	130,975	0	197	0	292,582	0	0	0	61,518
Transactions received by included first-hop concentration addresses	12,674	207,742	35	0	271,177	0	0	0	223,956
Transactions received by second-hop-exclusive addresses	1,616,556	0	8,360	23	481,092	9	1,017	31,363	44,461

with silhouette scores 0.8987 and 0.9286. The features selected for this analysis step are indicated in Table II. The distribution of addresses in resulting clusters is shown in Table VI. In the clustering of dissipation affiliates, most addresses fell into two clusters, 0 and 4; most malicious addresses were in cluster 0; the rest of the clusters had less than 100 addresses. It appears that, at least for cluster 0, funnelling and re-cycling operations move beyond two hops of transactions. The other cases most likely represent masking behaviour of dissemination affiliates that use only a single hop to arrive at exchanges and mixers. Clusters 1 and 8 consisted primarily of first-hop affiliates, while others were comprised mostly of second-hop-exclusive addresses.

We exclude the results for the second-hop clustering of concentration affiliates as provided expected results. The interesting insights as it confirmed our findings. Since the first-hop concentration affiliates represent crypto exchanges, the number of second-hop addresses receiving funds from these exchanges grows drastically.

As with the first-hop clusters, our closer analysis showed that none of the second-hop clusters appear to be exclusively connected to certain types of scams. Surprisingly, 206 malicious addresses which did not have any outgoing transactions, reappeared in the second hop set. Investigating this further, we discovered interesting collaborative behavior. Some malicious accounts transfer funds to first hop affiliate addresses that in turn send funds/tokens to these 206 malicious addresses without outgoing transactions. In essence, these transactions support malicious accounts imitating legitimate addresses transferring funds.

VI. Collaboration analysis

a) Account reuse: Our close analysis of 7915 malicious accounts showed inconsistencies in labelling. 3,603 malicious addresses in our initial set have multiple labels indicating that their malicious activities have been viewed differently by different sources. This may be an indication that the field needs a better understanding of the malicious behaviour on the blockchain. However, we suspect that this may point out at the reuse of malicious accounts by adversaries across various scams. Indeed, the detection and labelling of malicious accounts on blockchain platforms is often delayed consequently opening a room for adversaries to leverage accounts with a successful fund transfer history.

To investigate possible reuse behavior, we explore the changes in labelling over time (Figure 4). These changes are sequential and persistent, they are not sporadic and once made they persist over time.

These changes are sometimes related to spurs in transaction activity which suggests that some malicious accounts are reused for different types of malicious activities.

However, we also observe that in some cases the changes in labels are not drastic, e.g., 'phish-hack' is replaced by 'phishing', or 'malicious' label is replaced by more specific 'phish-hack'. This labelling appears to emphasize a similar nature of activities, and hence could be considered as improved, more precise classification.

In a few cases, however the changes point at different type of behavior, e.g., 'scam' accounts are relabelled as 'phishing' and 'exploit' are changed to 'scam'. These are the accounts that are most likely reused for different activities over time. Hence, although we see this reuse in our set, these instances do not appear often.

b) Malicious accounts behavior: To further understand the existing collaboration, we start by investigating transaction activity of malicious accounts (Figure 5).

1,349 malicious accounts in our set have no incoming transactions, we consider them inactive. Perhaps they were created to facilitate scams but were not successful in attracting victims. Similarly, 1,709 of malicious accounts have only 1-2 incoming transactions, we suspect these are similarly associated with the least successful fraudulent schemes. In other words, 38.6% of malicious accounts in our set can be viewed as unsuccessful in their fraudulent activity.

Most malicious addresses have no transactions with other malicious addresses, however, there are a few malicious addresses that performed more than 50 transactions with other malicious addresses. They appear to be a part of token scam schemes, where the scammers transfer tokens to imitate legitimate behavior and attract users.

c) Collaboration among malicious addresses: The vast majority of benign addresses have 1–2 transactions to malicious addresses, with a small number of addresses having more than 100 such transactions; a sizable amount of addresses have no transactions to malicious addresses, meaning they could not be considered victims. As for malicious addresses receiving transactions from benign addresses, the distribution of transactions does not follow any apparent pattern, with some

Fig. 4: Transaction activity of multi-labelled accounts over time

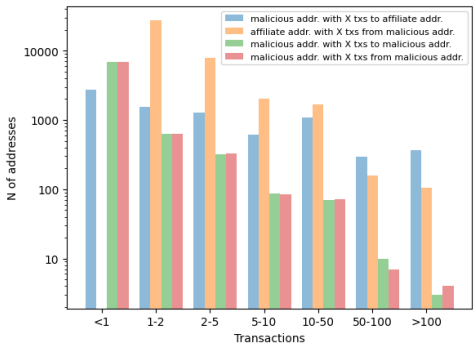

Fig. 5: Transactions between known malicious and first-hop addresses

malicious addresses having no incoming transactions, some having a small number of incoming transactions, and some having received more than 100 benign transactions. There are also more notable cases, such as benign addresses receiving funds from malicious addresses, and malicious addresses sending funds to benign addresses (represented with orange and green bars, respectively). A small amount of the former transactions would not be suspicious, as these could indicate some form of a scam operation, such as a Ponzi scheme, which provide profits for the earlier investors. However, we can see a non-negligible fraction of addresses receiving more than five such transactions. These addresses represent potential affiliates that are used by malicious actors to transfer funds out of the scam operation. The latter group of transactions should also be considered, as they represent the flow of funds out of malicious addresses. The majority of malicious addresses carry out 5–50 outgoing transactions throughout the lifetime of a scam.

In this research, we did not focus on true victims of such scams: our goal was to evaluate the senders and recipients of transactions from malicious addresses and determine yet undetected points of access (affiliates) that scammers set up to reclaim their profits. Additionally, we could observe that addresses transferring funds to malicious accounts received funds from other malicious accounts. This places doubt on presumably benign participant accounts, which may instead be parts of coordinated funnelling operations.

d) Collaboration among 'victims': The vast majority of addresses in the first hop set (220,159) have outgoing transaction to our malicious set. Most of them send 1–2 transactions to malicious addresses (Figure 5). This is consistent with the expected behaviour of victims that would only be deceived once and then realize the mistake.

We also observe a small number of addresses having more than 100 such transactions to malicious addresses. A closer analysis revealed that some of these malicious accounts are presented as investment opportunities. As a result, victims may invest multiple times in these accounts and consequently attract other users to these investment scams. For example, among such scams is an account `0x4ac6307a85d83962503f86457de9c331a6926f48`, now known to be associated with the Westland Storage scam. The scam included several cryptocurrencies and was presented as a legitimate real estate investment platform that promised a return rate of 1-2% of the total investment per day. The account functioned for a few months, receiving small transfers from large number of victims, eventually transferring out all remaining funds through several affiliate accounts.

However, our further investigation showed that 14% (30,822) of accounts that we consider to be victims, i.e., send funds to our original 7,915 malicious accounts, also receive funds from 4,193 malicious accounts.

Since we rely on labelled accounts, it is possible that some of these victim accounts are controlled by adversaries as well. We found that none of these victim accounts received funds from more than one malicious account in our set. It is possible that these victims are compromised and unwilling scam participants.

e) Transfer of funds to affiliates: We observe numerous cases of addresses receiving funds from malicious accounts. We would expect these to represent affiliate accounts, indeed

7

the majority of them receive more than 2 transactions from malicious accounts which indicates an established fund transfer path.

While we cannot confirm whether these subsequent affiliate accounts that received the funds are also malicious, their involvement in only receiving transfers from known malicious accounts raises suspicion. To focus on these suspicious accounts and to simplify the analysis, we exclude all accounts associated with victims, i.e., accounts with transactions transferring funds to the malicious accounts.

In our first hop accounts, the majority (60%) of malicious accounts send funds to only a small number (14%) of affiliate accounts. It appears that *adversaries leverage the same set of affiliate accounts to transfer accumulated funds. None of these transfer points appear in our labelled set of malicious accounts.* We confirmed this behaviour in the clustering step.

Other affiliate accounts appear to be isolated and not connected to many malicious accounts. Some of these isolated affiliate accounts have no further outgoing transactions and we suspect that they are associated with unsuccessful scams.

Overall, the majority of malicious accounts in our set carry out 5–50 outgoing transactions throughout their lifetime.

VII. Lessons and implications

Our analysis of the formed clusters and movement of funds reveals several aspects of malicious activity:

First, the funds from malicious accounts are typically first transferred to seemingly benign accounts. These accounts do not appear in our malicious set and do not have labels associated with suspicious or malicious activity on the Etherscan platform. We noticed only a few instances where the second hop account has also been flagged as malicious.

Second, it appears that certain accounts are being utilized as transfer points by multiple crypto scams, i.e., shared between adversaries for different malicious activities. These accounts are not directly involved in fraudulent activities and serve two purposes: they accumulate funds across multiple scams, and transfer these funds, usually in smaller amounts, to other accounts.

Third, transaction paths transferring funds from malicious accounts typically end in accounts that belong to cryptocurrency exchange markets. Although we have not recursively traversed all transaction flows, we have manually traversed paths of representative affiliate accounts. The paths we observed included a chain of fund transfers between accounts that typically do not appear as malicious and include 1-3 hops ending in cryptocurrency markets. The funds transferred through dissemination affiliates in the first hop ended in exchange or router platforms in the second hop or third hop (manually confirmed). This formed 2-3 hop chain for funds transferred by the malicious accounts. Concentration affiliates typically represented exchange or mixer accounts themselves forming a 1-hop chain.

These hops may accumulate funds from other malicious accounts along the way. On a few occasions, we also noticed that the funds at the second hop are split between several outgoing accounts. Although we do not have evidence that these accounts are controlled by the same attackers, the flow of funds seems to indicate that.

Fourth, we observed a high interactivity between affiliates, with almost all dissemination affiliates propagating funds towards concentration affiliates (usually representing exchange or mixer accounts).

Fifth, malicious accounts are rarely reused over time for different malicious activities. It appear that inconsistency is labelling is likely to be related to our limited understanding of fraudulent activities and the absence of standardized categorization.

Implications: Our findings reveal several important implications for detection and mitigation efforts:

Tracing hidden dissemination affiliates. Our analysis shows that only dissemination affiliates are likely to be accounts under control of malicious users as opposed to concentration affiliates, and our clustering approach allows to identify them.

Monitoring communication between affiliates. Our analysis shows that the collaborative network is stable and interactive. Various malicious activities leverage the same sets of affiliate accounts over a period of time, and these affiliates transfer funds between each other. This provides a unique opportunity for monitoring transfer of funds between malicious affiliates to support law enforcement investigations.

Involvement of benign exchange services. Our analysis showed a considerable role of crypto mixers and exchange platforms in both laundering and storage of funds for malicious accounts. Many of the mixers and exchange are legitimate platforms. This presents an opportunity to thwart the transfer and subsequent laundering of illicitly acquired funds by malicious accounts. Exchange services are in a position to restrict transfers and freeze funds associated malicious accounts.

Centralized blacklisting. Detection and mitigation of malicious activity is challenging partially due to the lack of understanding which accounts are involved. Establishing and maintaining centralized blacklists of accounts known to engage in malicious activity and propagate funds exclusively for these accounts can facilitate user awareness and consequently limit number of victims.

VIII. Conclusion

In this work, we present the first large-scale study of collaboration between malicious accounts in Ethereum blockchain. The implications of our research extend beyond the detection of hidden malicious Ethereum accounts, as the insights of our study can serve as a foundation for building advanced detection and prevention systems. By understanding the mechanisms of fund transfer, affiliate account reuse, and the establishment of intricate chains of transactions, we provide security professionals and researchers with valuable insights to proactively identify and mitigate fraudulent activities. We hope this analysis will facilitate future research aimed at safeguarding the security of blockchain ecosystem.

REFERENCES

[1] S. Farrugia, J. Ellul, and G. Azzopardi, "Detection of illicit accounts over the ethereum blockchain," *Expert Systems with Applications*, vol. 150, p. 113318, 2020.

[2] T. Hu, X. Liu, T. Chen, X. Zhang, X. Huang, W. Niu, J. Lu, K. Zhou, and Y. Liu, "Transaction-based classification and detection approach for ethereum smart contract," *Information Processing Management*, vol. 58, no. 2, p. 102462, 2021.

[3] C. Shi, Y. Xiang, J. Yu, L. Gao, K. Sood, and R. R. M. Doss, "A bytecode-based approach for smart contract classification," in *2022 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*, 2022, pp. 1046–1054.

[4] K. Hara, T. Takahashi, M. Ishimaki, and K. Omote, "Machine-learning approach using solidity bytecode for smart-contract honeypot detection in the ethereum," in *2021 IEEE 21st International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, 2021, pp. 652–659.

[5] 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.

[6] S. Li, F. Xu, R. Wang, and S. Zhong, "Self-supervised incremental deep graph learning for Ethereum phishing scam detection," *arXiv preprint arXiv:2106.10176*, 2021.

[7] H. Wen, J. Fang, J. Wu, and Z. Zheng, "Transaction-based hidden strategies against general phishing detection framework on ethereum," in *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2021, pp. 1–5.

[8] Y. Xia, J. Liu, and J. Wu, "Phishing detection on ethereum via attributed ego-graph embedding," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 69, no. 5, pp. 2538–2542, 2022.

[9] 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*, vol. 52, no. 2, pp. 1156–1166, 2022.

[10] W. Chen, Z. Zheng, J. Cui, E. Ngai, P. Zheng, and Y. Zhou, "Detecting Ponzi schemes on Ethereum: Towards healthier blockchain technology," in *Proceedings of the 2018 world wide web conference*, 2018, pp. 1409–1418.

[11] Y. Lou, Y. Zhang, and S. Chen, "Ponzi contracts detection based on improved convolutional neural network," in *2020 IEEE International Conference on Services Computing (SCC)*, 2020, pp. 353–360.

[12] Y. Zhang, S. Kang, W. Dai, S. Chen, and J. Zhu, "Code will speak: Early detection of ponzi smart contracts on ethereum," in *2021 IEEE International Conference on Services Computing (SCC)*, 2021, pp. 301–308.

[13] A. Aljofey, Q. Jiang, and Q. Qu, "A supervised learning model for detecting ponzi contracts in ethereum blockchain," in *International Conference on Big Data and Security*. Springer, 2021, pp. 657–672.

[14] Y. Zhang, W. Yu, Z. Li, S. Raza, and H. Cao, "Detecting ethereum ponzi schemes based on improved lightgbm algorithm," *IEEE Transactions on Computational Social Systems*, vol. 9, no. 2, pp. 624–637, 2022.

[15] W. Chen, Z. Zheng, E. C.-H. Ngai, P. Zheng, and Y. Zhou, "Exploiting blockchain data to detect smart ponzi schemes on ethereum," *IEEE Access*, vol. 7, pp. 37 575–37 586, 2019.

[16] X. Wang, J. He, Z. Xie, G. Zhao, and S.-C. Cheung, "ContractGuard: Defend ethereum smart contracts with embedded intrusion detection," *IEEE TSC*, pp. 314–328, 2019.

[17] M. Zhang, X. Zhang, Y. Zhang, and Z. Lin, "TXSpector: Uncovering attacks in ethereum from transactions," in *USENIX Security*, 2020.

[18] S. Linoy, S. Ray, and N. Stakhanova, "Etherprov: Provenance-aware detection, analysis, and mitigation of ethereum smart contract security issues," in *2021 IEEE International Conference on Blockchain (Blockchain)*, 2021, pp. 1–10.

[19] C. F. Torres, A. K. Iannillo, A. Gervais, and R. State, "The Eye of Horus: Spotting and Analyzing Attacks on Ethereum Smart Contracts," *arXiv preprint arXiv:2101.06204*, 2021.

[20] P. Xia, H. Wang, B. Gao, W. Su, Z. Yu, X. Luo, C. Zhang, X. Xiao, and G. Xu, "Trade or trick? detecting and characterizing scam tokens on uniswap decentralized exchange," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 5, no. 3, pp. 1–26, 2021.

[21] A. Trozze, T. Davies, and B. Kleinberg, "Of degens and defrauders: Using open-source investigative tools to investigate decentralized finance frauds and money laundering," *Forensic Science International: Digital Investigation*, vol. 46, p. 301575, 2023.

[22] M. Vasek and T. Moore, "There's no free lunch, even using bitcoin: Tracking the popularity and profits of virtual currency scams," in *Financial Cryptography and Data Security*. Springer, 2015, pp. 44–61.

[23] G. Atondo Siu, A. Hutchings, M. Vasek, and T. Moore, ""invest in crypto!": An analysis of investment scam advertisements found in bitcointalk," in *Symposium on Electronic Crime Research*. APEG, 2022.

[24] M. Vasek and T. Moore, "Analyzing the bitcoin ponzi scheme ecosystem," in *Financial Cryptography and Data Security: FC 2018 International Workshops, BITCOIN, VOTING, and WTSC, Nieuwpoort, Curaçao, March 2, 2018, Revised Selected Papers 22*. Springer, 2019, pp. 101–112.

[25] J. Feist, G. Grieco, and A. Groce, "Slither: A static analysis framework for smart contracts," in *2019 IEEE/ACM 2nd International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB)*, 2019, pp. 8–15.

[26] L. Brent, A. Jurisevic, M. Kong, E. Liu, F. Gauthier, V. Gramoli, R. Holz, and B. Scholz, "Vandal: A scalable security analysis framework for smart contracts," *arXiv preprint arXiv:1809.03981*, 2018.

[27] L. Luu, D.-H. Chu, H. Olickel, P. Saxena, and A. Hobor, "Making smart contracts smarter," in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 2016, pp. 254–269.

[28] P. Tsankov *et al.*, "Securify: Practical Security Analysis of Smart Contracts," in *CCS*, 2018.

[29] Á. Hajdu and D. Jovanović, "Solc-verify: A Modular Verifier for Solidity Smart Contracts," in *VSTTE*, 2020.

[30] W. Wang, J. Song, G. Xu, Y. Li, H. Wang, and C. Su, "Contractward: Automated vulnerability detection models for ethereum smart contracts," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 2, pp. 1133–1144, 2021.

[31] "Mythril classic," 2021. [Online]. Available: https://github:com/ConsenSys/mythril-classic

[32] J. Wang, P. Chen, X. Xu, J. Wu, M. Shen, Q. Xuan, and X. Yang, "Tsgn: Transaction subgraph networks assisting phishing detection in ethereum," in *International Conference on Blockchain and Trustworthy Systems*, 2021, pp. 187–200.

[33] R. Camino, C. F. Torres, M. Baden, and R. State, "A data science approach for detecting honeypots in ethereum," in *2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, 2020, pp. 1–9.

[34] A. Biryukov, D. Khovratovich, and I. Pustogarov, "Deanonymisation of clients in bitcoin p2p network," in *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, 2014, pp. 15–29.

[35] P. Koshy, D. Koshy, and P. McDaniel, "An analysis of anonymity in bitcoin using p2p network traffic," in *Financial Cryptography and Data Security: 18th International Conference, FC 2014, Christ Church, Barbados, March 3-7, 2014, Revised Selected Papers 18*. Springer, 2014, pp. 469–485.

[36] T. Neudecker and H. Hartenstein, "Could network information facilitate address clustering in bitcoin?" in *Financial Cryptography and Data Security: FC 2017 International Workshops, WAHC, BITCOIN, VOTING, WTSC, and TA, Sliema, Malta, April 7, 2017, Revised Selected Papers 21*. Springer, 2017, pp. 155–169.

[37] S. Linoy, N. Stakhanova, and S. Ray, "De-anonymizing ethereum blockchain smart contracts through code attribution," *Int. J. Netw. Manag.*, vol. 31, no. 1, jan 2021.

[38] D. Ermilov, M. Panov, and Y. Yanovich, "Automatic bitcoin address clustering," in *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2017, pp. 461–466.

[39] X. He, K. He, S. Lin, J. Yang, and H. Mao, "Bitcoin address clustering method based on multiple heuristic conditions," *IET Blockchain*, vol. 2, no. 2, pp. 44–56, 2022.

[40] Y. Zhang, J. Wang, and J. Luo, "Heuristic-based address clustering in bitcoin," *IEEE Access*, vol. 8, pp. 210 582–210 591, 2020.

[41] F. Liu, Z. Li, K. Jia, P. Xiang, A. Zhou, J. Qi, and Z. Li, "Bitcoin address clustering based on change address improvement," *IEEE Transactions on Computational Social Systems*, 2023.

[42] C. Kang, C. Lee, K. Ko, J. Woo, and J. W.-K. Hong, "De-anonymization of the bitcoin network using address clustering," in *Blockchain and Trustworthy Systems: Second International Conference, BlockSys 2020,*

Dali, China, August 6–7, 2020, Revised Selected Papers 2. Springer, 2020, pp. 489–501.

[43] T.-H. Chang and D. Svetinovic, "Improving bitcoin ownership identification using transaction patterns analysis," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 1, pp. 9–20, 2018.

[44] K. Wang, M. Tong, C. Wu, J. Pang, C. Chen, X. Luo, and W. Han, "Exploring unconfirmed transactions for effective bitcoin address clustering," *arXiv preprint arXiv:2303.01012*, 2023.

[45] H. Sun, N. Ruan, and H. Liu, "Ethereum analysis via node clustering," in *Network and System Security: 13th International Conference, NSS 2019, Sapporo, Japan, December 15–18, 2019, Proceedings 13*. Springer, 2019, pp. 114–129.

[46] F. Victor, "Address clustering heuristics for ethereum," in *Financial Cryptography and Data Security: 24th International Conference, FC 2020, Kota Kinabalu, Malaysia, February 10–14, 2020 Revised Selected Papers 24*. Springer, 2020, pp. 617–633.

[47] M. Bartoletti, S. Carta, T. Cimoli, and R. Saia, "Dissecting ponzi schemes on ethereum: Identification, analysis, and impact," *Future Generation Computer Systems*, vol. 102, pp. 259–277, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X18301407

[48] S. Al-E'mari, M. Anbar, Y. Sanjalawe, and S. Manickam, "A labeled transactions-based dataset on the ethereum network," in *Advances in Cyber Security*, M. Anbar, N. Abdullah, and S. Manickam, Eds. Singapore: Springer Singapore, 2021, pp. 61–79.

[49] L. C. Xuezhi He, Tan Yang, "Ctrf: Ethereum-based ponzi contract identification," *Security and Communication Networks*, vol. 2022, p. 10, 2022.

[50] A. A. Hagberg, D. A. Schult, and P. J. Swart, "Exploring network structure, dynamics, and function using networkx," in *Proceedings of the 7th Python in Science Conference*, G. Varoquaux, T. Vaught, and J. Millman, Eds., Pasadena, CA USA, 2008, pp. 11 – 15.

[51] https://docs.etherscan.io/.

[52] https://scikit-learn.org/1.0/.