Airdrop Hunter Bot Sybil Identification Techniques

Dharpawar K Ishan[1], Dange Nandini[2], Dhumal Varun[3], Patil Sakshi[4]
[1,2,3,4]*DY Patil Univrsity*

Abstract—In the realm of cryptocurrency airdrop campaigns, concerns have been raised regarding the fair distribution of tokens due to the increasing prevalence of airdrop hunter bots. A range of Sybil identification techniques has been investigated in this research paper, including blockchain analysis, social network pattern recognition, and machine learning algorithms. The effectiveness of these methodologies in distinguishing genuine participants from Sybil actors is assessed, and ethical considerations related to user privacy are taken into account. The importance of continuous research and development in the field of Sybil identification is emphasized, as it is crucial to safeguard the fairness and integrity of token distributions within cryptocurrency communities, mitigating the disruptive influence of airdrop hunter bots and fostering a more secure and equitable ecosystem.

Keywords—cryptocurrency, airdrop campaigns, token distribution, airdrop hunter bots, Sybil identification techniques, blockchain analysis, social network pattern recognition, machine learning algorithms, genuine participants, ethical considerations, user privacy, research and development, fairness, integrity, security measures, wallet activity, patterns of interaction, anomalies, manipulation, cryptocurrency projects, secure ecosystem

I. INTRODUCTION

Amid the landscape of cryptocurrency airdrop campaigns, there arises a growing concern regarding the fair and just distribution of tokens, exacerbated by the ever-increasing prevalence of airdrop hunter bots. This research paper delves into a comprehensive examination of diverse Sybil identification techniques that have been painstakingly scrutinized as responses to this pressing issue. These techniques encompass blockchain analysis, social network pattern recognition, and the strategic deployment of machine learning algorithms, each seeking to fulfill the vital role of effectively distinguishing between authentic participants and Sybil actors.

The assessment of the effectiveness of these methodologies in discerning genuine participants from Sybil actors is a central facet of this research endeavor. Furthermore, it conscientiously attends to the ethical considerations entwined with the deployment of these techniques, particularly with respect to safeguarding user privacy. This reflective approach acknowledges the imperative need to strike a balance between the security imperative and individual privacy concerns.

Underpinning the entire research narrative is the emphatic significance of ongoing research and development in the realm of Sybil identification. This importance stems from the pivotal role such research plays in upholding the fundamental principles of fairness and integrity within token distributions in cryptocurrency communities. As these efforts gradually curtail the disruptive influence of airdrop hunter bots, they contribute substantively to the establishment of a cryptocurrency ecosystem that is characterized by increased security and fairness, thus inspiring greater confidence among participants in airdrop campaigns.

Furthermore, the very utilization of these Sybil identification techniques underscores the fluid and evolving nature of cryptocurrency security measures. Notably, the research underscores that a combination of these methodologies is frequently adopted, integrating the capabilities of blockchain analysis for the traceability of wallet activity, social network analysis to unveil the intricate web of interaction patterns among participants, and the astute use of machine learning algorithms for the identification of anomalies.

II. BACKGROUND

This section introduces the background knowledge of the study in this paper, including blockchains, decentralized applications, and token economics. Interested readers could refer to recent surveys and books, such as [7]–[10], for a comprehensive overview and details.

A. Blockchain and Transactions

At the heart of cryptocurrency lies the blockchain, a decentralized and immutable ledger that records all

transactions. Blockchains are designed to ensure transparency and security, making them the foundation of most cryptocurrencies. Transactions within a blockchain are pseudonymous, and they are executed by participants across the network. This element of transparency and pseudonymity sets the stage for airdrop hunter bots to exploit airdrop events, as they can easily create numerous fake identities to deceive the system.

B. Decentralized Applications and Smart Contract

The emergence of decentralized applications (DApps) and smart contracts within blockchain ecosystems has introduced new paradigms of automation and trust. Smart contracts, self-executing agreements with code-based rules, enable airdrop campaigns to be executed seamlessly. However, their automated nature also creates an opening for airdrop hunter bots to manipulate these events, demanding more sophisticated Sybil identification techniques to discern human users from automated agents.

C. Tokenomics

Token distributions are a core component of the cryptocurrency ecosystem, often employed to stimulate user participation and engagement. Airdrops, as one such distribution method, rely on the principle of rewarding active users. However, the prevalence of airdrop hunter bots disrupts this ecosystem by concentrating rewards in the hands of a few, undermining the principles of tokenomics that underpin these distributions.

D. Sybil Behavior

Sybil behavior, named after the famous case of a woman with multiple personalities, refers to the creation of multiple fake identities in a network. In cryptocurrency, these fake identities can range from duplicate wallets to fictitious social media profiles. Understanding Sybil behavior is pivotal in developing effective Sybil identification techniques to distinguish between genuine participants and malicious actors, thereby preserving the fairness and integrity of token distributions.

1) DApp System Model

DApps offer graphical user interfaces (GUI) for Dapp users via Web or mobile clients, and they implement smart contracts on public blockchains. It is uncommon for regular DApp users to communicate directly through code with these smart contracts. One unique aspect of DApps is that they employ blockchains to preserve user states and interactions as transactions.

Consider Uniswap3, the most widely used decentralized finance (Defi) protocol. With Uniswap, users can exchange one type of cryptocurrency token for another.

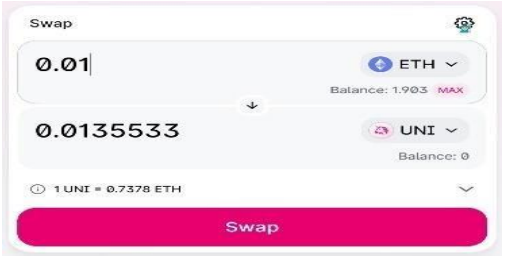

Figure 1. Uniswap GUI

with the use of the matching DApp4, as seen in Fig. 1.)Acollection of smart contracts that are implemented on blockchains like Ethereum form the basis of the exchanging procedure. The automatic market maker (AMM) smart contract is one of the most important of them. AMMs use liquidity pools (LP) in place of a conventional market of buyers and sellers to enable the automatic and permissionless trading of digital assets. Users' swaps are transactions between users' accounts and the liquidity pools' smart contracts as seen through the lens of the blockchain. The user in Figure 1 wishes to exchange 0.01 Ethereum for UNI on Ethereum.

DApps are growing rapidly. There are likely more complex DApps in the game, media, social, and other domains that combine decentralized and centralized components. Nevertheless, user interactions with these DApps, which alter their states, may always be linked to blockchain transactions. This allows for the analysis of Sybil's activity within various DApps through the examination of the specifics of related transactions.

2) Sybil's Attack Model

The actions of the several accounts that Sybils manage and employ to communicate with the targeted DApp are largely the same as those of the regular user account. Typically,each account activates a few DApp features. Assuming Uniswap is the intended DApp, an engagement can involve exchanging tokens or offering liquidity. But a regular user often has a small number of accounts, whereas a Sybil manipulates numerous accounts. A Sybil typically uses a specially created

computer program, known as a bot, to carry out the interactions from these accounts automatically in order to control many accounts. Naturally, there are also industrious Sybils who carry out conversations by hand. In this scenario, we regard them to be manual bots, and thefollowing. We treat them as manual bots in this instance, and the analysis that follows is still valid. These accounts may be referred to as Sybil's accounts or bot's accounts in the sections that follow in this paper.

3) An Examination of Sybil's Conduct

We will examine Sybil's actions in depth together with a Defi application, Hop Protocol6, in the sections that follow. With Hop Protocol, users can transmit tokens from one rollup or side chain to another practically instantly without waiting for the network's challenge time. Hop Protocol is a scalable rollup-to-rollup universal bridge. Hop Protocol offers a number of features, including "send," "add liquidity," "convert," and "stake." Users of the Defi program may readily comprehend these features; curious readers who are not familiar with Defi can consult Hop Protocol's documentation to learn more about these features' specific significance. It is important to note that neither the DApp's domains nor the specific definitions of its functionality have any bearing on the study of Sybil's behaviors that follows.

Finding accounts managed by the same Sybil or bot requires Sybil detection. Remember that these exchanges are documented in transactions on blockchains, and that transactions represented by signed structure texts are directives to change states that transfer money or tokens between accounts. We contend that patterns in the transactions resulting from the activities within the DApps' interactions can be used to deduce the identities of bots.

It is true that the account interacts with the smart contracts that the DApp has deployed when a bot initiates DApp actions from a supervised account. A blockchain network's nodes receive the corresponding transactions from the address, which are then added to a block. An example is already shown in Fig. 2.We concentrate on two aspects of the transaction information. One is shown in Fig. 3 by the field of transaction receipt event logs. The kind of DApp activities that the account initiates are disclosed in these event logs. Three event logs released by the smart contract are displayed in Fig. 3 by the red boxes: 1) There is a token transfer; 2) The liquidity pool's reserves are updated; and 3) There is a token swap. The account's token swapping activity is then shown by this transaction.

The transaction charge field is the other. In order for a transaction to be included in a blockchain block, the account submitting the transaction needs to have sufficient funds to compensate miners and states with petrol fees. The execution instructions in the code are used to determine the total amount of petrol fees if the account communicates with a smart contract. Petrol fees are frequently paid for using the native token of the blockchain, such as Ethereum's ether (ETH). The field tokens that have been transmitted may already be seen in Fig. 2. It is necessary for accounts to have some initial tokens since several interactions between accounts and DApps entail token transfers, such "swap" in Uniswap or "Send" in Hop protocol.

With the above observation, we propose to study Sybil’s behaviors, specifically their account activities, by exploring the above activity and transaction patterns.

4) DApp Activities

Different DApps may include different user engagements. However, practically all of these exchanges involve the transfer of specific tokens, which are available from the associated transaction event logs. To use Hop Protocol, for example, a bot manages several accounts. The functions that are offered include "send," "add liquidity," "convert," and "stake." Token transfers to Ethereum smart contracts or layer 2 blockchains are involved in these operations when an account initiates them to engage with Hop Protocol.

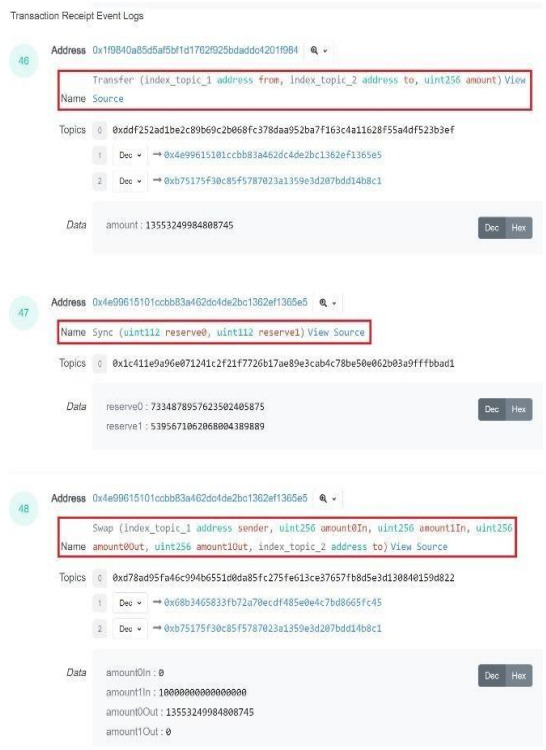

Figure 3. The transaction receipt event logs of a swap

If we take an example DApp account and call it B, we have a triplet B = (t, a, p), where t is the timestamp of the activity, an is the type of activity that indicates the functionalities of the DApp, and p is a parameter set that could be anything from the quantity of tokens transferred to other input data for smartcontracts. Then, a series of these activities, with c = B1, B2,..., Bk, might represent all of the actions from account c on a DApp. When determining the similarity of interaction sequences, we in this paper simply take into account the activity type a and the parameter set p.

5) Token Transfers

As previously stated, Sybil's accounts have comparable activity sequences, but it would be arbitrary to claim that Sybil controls the accounts only because of these similarities. Coincidences occur. It is possible for multiple users to consult the same DApp online tutorial. They might have comparable account activity sequences in this instance. We require improvement.

These accounts should have funds available for interaction or for covering transaction gas fees when a bot manipulates them to interact with DApps' smart contracts.

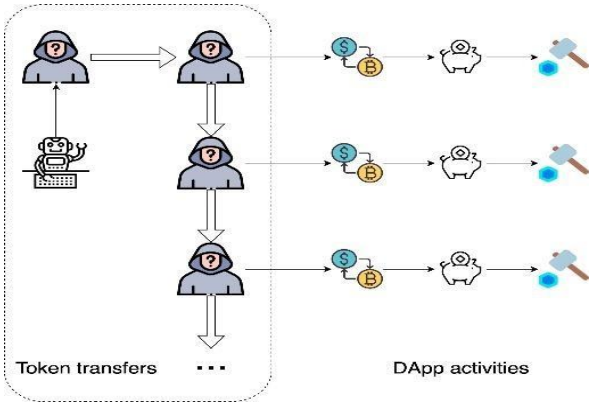

Figure 4. Token transfer pattern: sequential

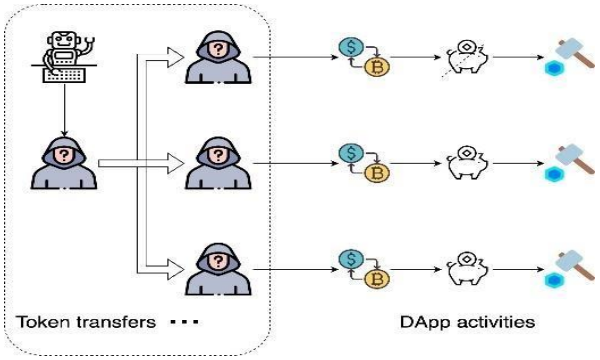

Figure 5. Token transfer pattern: radial

They cannot, by accounts, fall from the sky. These accounts are managed by bots, thus for security purposes, they never keep tokens for very long. Typically, bots are able to send tokens to these addresses prior to manipulating them and retrieve any remaining tokens at the end of all interactions. Token transfers follow two basic and consistent patterns.

Sequential pattern: Tokens from one Treasury account are sent to the first account by a bot. The next address that the bot controls receives these tokens after they have been transferred from the last account that received them, as indicated by the dotted box in Figure 3

Radial pattern: As indicated by the dotted box in Figure 5, a bot transfers tokens directly to each account it controls from one treasure account. Keep in mind that these transfers don't have to occur simultaneously.

It is easy to see how these two basic token transfer patterns could be combined to create patterns that are more intricate. Figures 6 and 7 depict two potential combinations in the order of the fundamental patterns. Fig. 6 shows a radial first, sequential later pattern. Tokens are initially sent by the bot from one treasure account to a group of accounts, and then

each account in the group sends tokens to additional accounts. Fig. 7 shows a sequential pattern that becomes radial later. Every account following the logical sequence

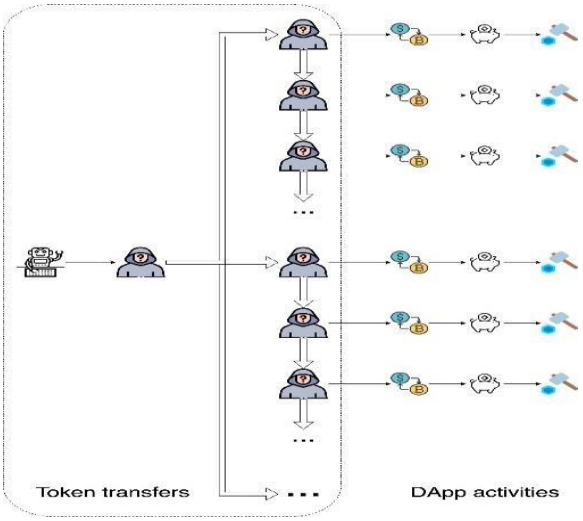

Figure 6. Complex token transfer pattern: radial first, sequential later

first takes tokens from the bot's treasury, then distributes those tokens radially to several accounts. Even though adding more fundamental patterns could result in more complex token transfer patterns, capturing only the fundamental patterns is sufficient because they already illustrate the relationships between these accounts. In Section 5, we provide an example of a complex pattern from the experimental evaluation.

6) Sybil's Behaviour Model

As illustrated in Figs. 4 and 5, we present two typical Sybil's behaviour models that correlate to the two basic token transfer patterns mentioned above. Figures 4 and 5's "right out of the box" depict comparable actions that were sparked by the bot's accounts. A bot in Fig. 4 transfers tokens from one Treasury to the first account by using the sequential pattern. Subsequently, the bot initiates communication between the first account and a specific DApp. Once all interactions have been completed, leftover tokens are moved from the first account to the second, and the process is repeated in the second account.

Using the radial pattern, the bot in Fig. 5 sends tokens from one treasure account to every address under its control, causing each account to interact with a specific DApp. After all interactions are complete in both scenarios, the remaining tokens are typically returned to the Treasury, though this is not required. To give the impression that they are regular users, some Sybils will leave these tokens in these accounts.

*E.*Detecting Sybil's Accounts

The framework for detecting Sybil accounts in airdrop also campaigns leverages cutting-edge bot detection technologies and novel approaches to ensure the integrity and fairness of distribution. By employing advanced machine learning algorithms and behavior analysis, the system identifies suspicious patterns indicative of Sybil accounts, such as unusually high transaction frequencies or repetitive actions. Furthermore, it incorporates realtime monitoring and geolocation data to distinguish between legitimate and fraudulent users, ensuring that the airdrop reaches its intended recipients while minimizing the impact of Sybil attacks. This innovative solution represents a significant advancement in safeguarding the integrity of airdrop campaigns and bolstering trust within the blockchain and cryptocurrency community.

1. Locating Analogous DApp Tasks

To qualify the similarity between DApp activity sequences, which may vary in length and contain similar activities with nearly the same temporal orders, a refined approach is needed. The goal is to account for activity noise and consider the temporal order of activities. In this context, the Jaccard similarity coefficient is proposed as a suitable measure for qualifying sequence similarity. This measure is based on representing activity sequences as activity pairs. Given an activity sequence 'A', all possible activity pairs are extracted while preserving the temporal order. For example, if 'A = B1, B2, B3, B4', the corresponding activity pair set is 'Pairs(A) = (B1, B2), (B1, B3), (B1, B4), (B2, B3), (B2, B4), (B3, B4)'. With this representation, the Jaccard similarity coefficient is defined as:

$$SeqSim(Bm, Bn) = \frac{|Pairs(Bm) \cap Pairs(Bn)|}{|Pairs(Bm) \cup Pairs(Bn)|}$$

This similarity measure takes into account both the common activities and their temporal order in two sequences. With this properly defined activity sequence similarity, popular clustering algorithms like DBSCAN can be applied to group similar sequences into cohesive clusters. DBSCAN is chosen for this purpose as it is a density-based clustering non-parametric algorithm, which doesn't require specifying the number of clusters and can identify clusters of arbitrary shapes. This approach is instrumental in

detecting subtle similarities in DApp activity sequences while accounting for potential noise introduced by malicious actors, making it a robust method for identifying Sybil accounts.

2. Searching Token Transfer Patterns

This section will introduce the approach to discovering token transfer patterns among accounts with similar activity sequences in a cluster.

To identify Sybil accounts with similar activity sequences, it's essential to analyze token transfer patterns within the transactions. We achieve this by constructing a transaction graph derived from the blockchain's transaction history. A snapshot captures the state of the entire decentralized ledger, encompassing all addresses and their associated data, such as transactions, fees, balances, and metadata. When a DApp plans an airdrop, its development team creates a snapshot, typically taken from the moment the DApp is active up to a specified point just before the airdrop event. Qualified airdrop recipients are chosen from the snapshot based on predefined criteria.

3. Searching Sequential Patterns

We will now talk about looking for sequential patterns in accounts that belong to a cluster and have comparable activity sequences. Finding a path on that can pass through each of the vertices that correspond to these accounts is the main task. Two observations are made: 1) A simple path is not necessary because locating simple paths can greatly increase computation complexity; 2) The path should have the fewest vertices outside of the cluster as feasible.

3. Overall Framework

This section outlines the general structure for identifying Sybil's accounts in Algorithm 3 by combining the information from Sections 4.1 and 4.2. Line 1 extracts the Sybil-related transactions, and line 2 builds the transaction graph. Since Algorithms 1 and 2 don't need to be run over the entire graph, we can find every connected component [22] of in line 3. The algorithm creates activity sequences for each vertex in each loop from line 4 to line 12, then uses DBSCAN to identify clusters of cohesive activity sequences. The algorithm then calls Algorithms 1 and 2 to search for two basic token transfer patterns, returning the results.

1. Clustering Results of DApp Activities

We display the DApp activity clustering results in this section. As previously stated, Hop Protocol builds a transaction graph for every blockchain. Table 1 displays the DBSCAN, eps, and min pts parameters, which were found using grid search based on cluster quality. The clustering results are also displayed in Table 1, along with the number of clusters, the quality of the clusters, and the number of noise points excluded from the clusters. For clustering results, the silhouette coefficient is a widely used quality metric.the silhouette coefficient is a widely used quality metric.

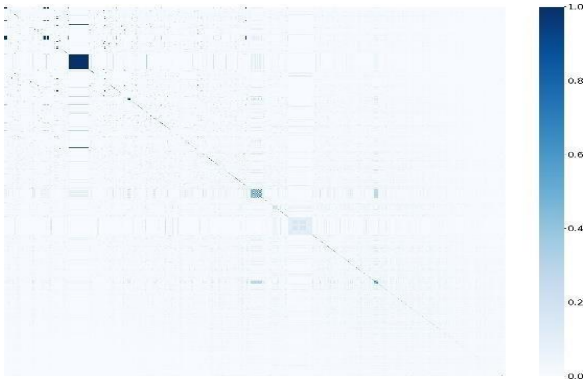

Figure 12. Jaccard similarity coefficients on Arbitrum

The similarity matrix is not the raw similar matrix, where we group rows and columns together in the same cluster according to the DBSCAN results. It is evident from these two figures that activity sequences from vertices within the same cluster resemble each other considerably more than those from vertices within different clusters.

The statistics of the clusters on Arbitrum with Hop Protocol activity sequences are displayed in Fig. 14. The average Jaccard similarity coefficient of each cluster is shown on the right y-axis, while the left yaxis shows the total number of accounts in each cluster. The statistics of the Gnosis clusters with Hop Protocol activity sequences are displayed in Fig. 15. The Jaccard similarity coefficient values in Figures 14 and Fig. 15 are close to 1, which means the corresponding cluster is cohesive.

2. Token Transfer Patterns

In this section, we showcase the identification of basic token transfer patterns through a number of case studies. A radial pattern's transaction subgraph is displayed in Figure 16. Optimism is the source of all transactions. The transaction graph designed for optimism does not have duplicate edges; nonetheless, duplicate edges are permitted in order to make the transactions between these accounts easy to see. There are two edges between two vertices if there are two token transfer transactions between them.

Figures 16 through 18 show some initial findings. The transaction subgraph of a star transfer pattern is displayed in Fig. 16. Optimism is the source of all transactions.

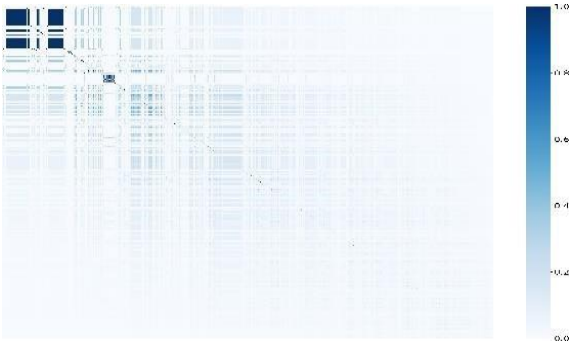

Figure 13. Jaccard similarity coefficients on Gnosis

Reacting Table 2 displays the hop transactions for these addresses, which are fairly comparable. Table 2 displays all of the accounts in Figure 16's transactions on the Hop Protocol. These accounts have very similar, and in some cases identical, Hop activity sequences. Table 2 also shows the quantity of transferred tokens. Almost all of the transactions have a transferred token amount of approximately 1. The treasury account is address 0x4dD1cb26, which is the centre of the radial pattern and has similar activity sequences but different token amounts.

The transaction subgraph of a sequence transfer pattern, in which every transaction originates from Arbi- trum, is depicted in Figure 17. A more intricate pattern that corresponds to Fig. 7 is shown in Fig. 18. Every transaction originates from Gnosis. We find the complex pattern by tweaking Algorithm 3.

Assume that the radial later, sequential first pattern is what we are looking for. Using Algorithm 1, we search for radial patterns first. Afterwards, we use Algorithm 2 to search for sequential patterns, feeding it the centre vertices of the returned set.

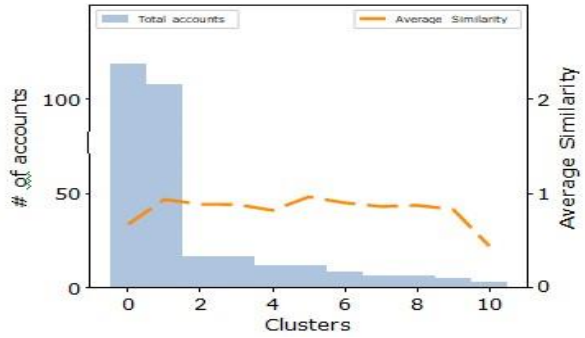

Figure 15. Clusters of similar DApp activities on Gnosis

Figure 16. Transaction subgraph of a Sybil: radial transfer pattern

4.DISCUSSION

In the section, we highlight the implications, constraints, and possible directions for future research.

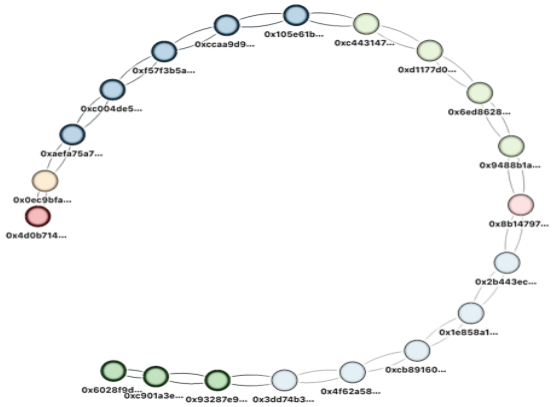

Figure 17. Transaction subgraph of a Sybil: sequential transfer pattern

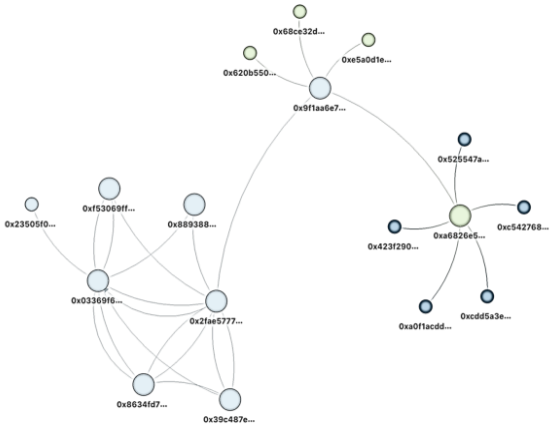

Figure 18. Transaction subgraph of a Sybil: complex transfer pattern

1. Implications for DApps

It is crucial to identify Sybil's accounts from the airdrop qualification list because more and more DApps are using airdrops as a marketing tactic. Finding Sybils is challenging because of the ground truth and benchmark. The framework and analysis presented in this paper may be useful in the development of a detection system. Hop Protocol organised a Sybil hunt. Participants in the event are anyone who is interested in finding Sybils. It is known that some later projects disqualified Sybil's Hop Protocol addresses after directly stealing them. Particularly for the manual bots covered in this paper, this is inappropriate. It is difficult to determine with certainty whether an account belongs to Sybil without considering similar activity sequences and consistent token transfer patterns.

5.LIMITATIONS

Numerous accounts on the extensive transaction graph acquire native tokens from Centralised Exchange (CEX) hot wallets, creating an ideal star token transfer pattern. Up to hundreds of neighbouring vertices may be present in the hot wallets of some well-known CEX. If the number of different types of interactions is relatively small, users with a large number of neighbours may occasionally engage in similar DApp interactions. In this instance, accounts belonging to regular users may inadvertently be identified as Sybil's account. Multi-source interactions, or interactions with other DApps, could help prevent this error. The likelihood of this kind of error occurring decreases with the number of interaction types.

6.FUTURE WORK

The suggested detection framework has two interesting extensions that could help DApps even more during airdrops.

What happens if the eligible prerequisites aren't predefined? The strategy put forth in this paper requires pre-defining prerequisites and requires knowledge of eligible airdrop addresses. Can we still identify Sybil's accounts if there aren't any predetermined eligible prerequisites? Sure, is the response. On massive datasets, the DBSCAN clustering algorithm performs well. Faster algorithms are required to search token transfer patterns because the bottleneck is at the transfer pattern. Setting up prerequisites based on a thorough examination of Sybil's current behaviours in the DApp will also be beneficial.

7.RELATED WORK

In this section, we discuss previous work related to Sybil detection. There are research works on Sybil detection or attacks in several domains. Although, generally speaking, Sybil is a synonym referring to hackers, the concrete defi- nition and behaviors of Sybils are different.

In social networks, there are research works about find- ing malicious accounts based on their activities on social networks [14]–[18]. Sybils in social networks are similar to Sybils in airdrops since Sybils on social networks pro- ducebehavior sequences, such as click streams. Based on the learning method, the Sybil detection methods in social network could be categorized into two classes: supervised learning [19]–[23] or unsupervised learning [23]–[27].

Research work such as [39] and [40] employ machine learning model on constructed features number of follow- ings/followers to identify malicious users and analyze the spammers' behavior on social networks. Yang et al. [20] made an empirical analysis of the evasion tactics utilized by Twitter spammers and then designed robust features to detect Twitter spammers. Ghosh et al. [21] are the first to investigate link farming in the Twitter network and then explore mechanisms to discourage the activity. Galan-Garcia et al. [22] proposed an approach to detect and associate fake accounts on Twitter that are employed for defamatory activities to a real account within the same network by analyzing the content of comments generated by both real and fake accounts. Wang et al. [23] detect fake identities by using Support Vector Machine (SVM) based on server-side clickstream models.

8.CONCLUSION

This paper is the first that investigates Sybils in DApp airdrops, as far as we know. We provide a Sybil detection technique to identify accounts that are controlled by bots. Based on the specifics of the transactions made on blockchains when Sybils alter controlled accounts to communicate with DApps, we closely examined Sybil's actions. Using a similarity

measure defined on the sets of activity pairs, a popular cluster algorithm is applied in the proposed detection framework to find cohesive groups of similar DApp activities.

9.ACKNOWLEDGMENT

We extend our heartfelt gratitude to the research community, colleagues, and organizations that have contributed to the development of the Airdrop Hunter Bot Sybil Identification Techniques presented in this research paper. This work would not have been possible without their dedication and expertise.
Feel free to personalize and expand on this acknowledgment as per your specific situation and the contributions you'd like to recognize.

REFERENCE

[1] K. Malinova and A. Park, "Tokenomics: when tokens beat equity," *Available at SSRN 3286825*, 2018.

[2] E. Lyandres, B. Palazzo, and D. Rabetti, "Initial coin offering (ico) success and postico performance," *Management Science*. [Online]. Available:https://doi.org/10.1287/mnsc.2022.4312

[3] C. R. Goforth, "It's raining crypto: The need for regulatory clarification when it comes to airdrops," *Indian Journal of Law and Technology*, vol. 15, pp. 321–344, 2019. [Online]. Available: https://heinonline.org/HOL/Page?handle=hein.journal s/indiajoula15&div=11

[4] A. Hafid, A. S. Hafid, and M. Samih, "A tractable probabilistic approach to analyze sybil attacks in sharding-based blockchain pro- tocols," *IEEE Transactions on Emerging Topics in Computing*, pp. 1–1, 2022.

[5] B. Barani Sundaram, T. Kedir, M. K. Mishra, S. H. Yesuf, S. M. Tiwari, and P. Karthika, "Security analysis for sybil attack in sensor network using compare and match-position verification method," in *Mobile Computing and Sustainable Informatics*, S. Shakya, R. Bestak, R. Palanisamy, and K. A. Kamel, Eds. Singapore: Springer Singapore, 2022, pp. 55–64.

[6] C. Pu and K.-K. R. Choo, "Lightweight sybil attack detection in iot based on bloom filter and physical unclonable function," *Computers & Security*, vol. 113, p. 102541, 2022. [Online]. Available:https://www.sciencedirect.com/science/article/pii/S0167404821003655

[7] B. A. Sassani Sarrafpour, A. Alomirah, S. Pang, and S. Sarrafpour, "Coding observer nodes for sybil attacks detection in mobile wireless sensor networks," in *2021 IEEE 19th International Conference on Embedded and Ubiquitous Computing (EUC)*, 2021, pp. 87– 94.

[8] J. Newsome, E. Shi, D. Song, and A. Perrig, "The sybil attack in sensor networks: analysis &defenses," in *Third International Symposium on Information Processing in Sensor Networks, 2004. IPSN 2004*, 2004, pp. 259–268.

[9] D. Gupta, J. Saia, and M. Young, "Bankrupting sybil despite churn," in *2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS)*, 2021, pp. 425–437.