



Analysis and Characterization of ERC-20 Token Network Topologies

Matteo Loporchio^(✉), Damiano Di Francesco Maesa, Anna Bernasconi,
and Laura Ricci

Department of Computer Science, University of Pisa, Largo Bruno Pontecorvo 3,
56127 Pisa, Italy

matteo.loporchio@phd.unipi.it,

{damiano.difrancesco,anna.bernasconi,laura.ricci}@unipi.it

Abstract. The transparent nature of public blockchain systems allows for unprecedented access to economic community data. Examples of such communities are the fungible token networks created by the ERC-20 standard on the Ethereum protocol. In this paper we study ERC-20 token networks, where nodes represent users and edges represent fungible token transfers between them. We focus our analysis on the top 100 largest networks, including a total of about 160 million edges and 60 million nodes. After a global analysis of the size and temporal evolution of such networks, we define and study seven features describing their main topological properties. In an attempt to characterize the networks by their topologies, we use the introduced features to cluster the networks together. To evaluate our results, we manually classify each network depending on the application domain of the corresponding contract and measure the homogeneity of the obtained clusterings. Overall, the results appear to indicate a lack of relationship between the scope of a contract and the topological features of the induced networks.

Keywords: Blockchain · Ethereum · Fungible Token · Network Analysis

1 Introduction

The advent of blockchain technology has disrupted traditional paradigms across multiple sectors, including financial systems, intellectual property, decentralized identity and supply chain management. Indeed, blockchains have the ability to provide secure, transparent, and decentralized record-keeping, eliminating the need for trusted intermediaries in transactions. Within this ever-evolving landscape, Ethereum – ranking as the second largest blockchain by market capitalization – has stood out for its innovations, foremost among them being the capability to store and execute code, in the form of *smart contracts* [13]. A smart contract is a piece of arbitrary code whose execution is validated by consensus, i.e., replicated by all participants of the blockchain network. Smart contracts have enabled the development of a wide range of *decentralized applications*

(DApps) running on the blockchain. Nowadays, DApps serve a variety of purposes, including decentralized finance, gaming, and social networking. Moreover, many DApps utilize the concept of *token*, namely a transferable asset that can be either *fungible* or *non-fungible*. Fungible tokens are interchangeable and identical, like traditional currencies. For instance, in the context of gaming, fungible tokens may represent reputation or player skills, while in the field of finance they can be used to represent assets or fiat currencies. Conversely, non-fungible tokens (NFTs) are unique digital assets with distinct properties, each with a distinct value. NFTs are often used to represent ownership of digital or physical items (e.g., works of art, collectibles, and more).

To enforce interoperability among fungible tokens on Ethereum, the ERC-20 standard was introduced. This standard defines rules for smart contracts implementing such tokens, facilitating token integration and exchange across various decentralized applications. In addition to this, each ERC-20 token creates a unique economy within the Ethereum ecosystem, where participants hold and trade tokens of the same kind. From a more theoretical perspective, we can say that each economy can be modeled as a *token network*, i.e., a graph whose nodes correspond to participants and edges represent token exchanges. Therefore, the analysis of ERC-20 token networks provides useful insights on the corresponding token economies. Indeed, it allows us to understand the evolution of transfers and how users tend to interact within these economies, e.g., whether they form communities, or if certain users hold more central roles with respect to others.

Motivated by these reasons, in this paper we study the properties of the top 100 ERC-20 token networks by total number of transfers. To gather information about transfers, we use data from the first 15 million Ethereum blocks, covering the time period between July 30th, 2015 and June 21st, 2022. Specifically, we exploit Ethereum transaction receipts, which include information about ERC-20 Transfer events. Indeed, such events serve as the main mechanism for notifying participants of token transfers, recording the sender, recipient, and the amount of tokens transferred. Our main contribution is articulated as follows. First, we study the historical evolution of transfer events within the analyzed data set. Then, we analyze the topological properties of token networks by associating each network with a set of seven features describing its connectivity, degree distribution, transitivity, density, diameter, and average shortest path length. Subsequently, we use such features to conduct further analysis based on clustering techniques, aiming at identifying groups of networks sharing similar topological properties. Finally, we classify token networks based on the application domain of the corresponding token. We use this classification to investigate possible connections between the topology of a network and the semantics of the corresponding ERC-20 token.

Related Work. Ethereum token networks have already been studied in the literature. The authors of [9] analyzed the global ERC-20 token network, i.e. the union of all ERC-20 token networks, between February 2016 and February 2018. They found out that the degree distribution follows a power-law and the token popularity among buyers and sellers also follows a power law model. Similarly,

the analysis in [11] revealed that many ERC-20 token networks exhibit either a star or hub-and-spoke topology. Additionally, such networks tend to have low clustering coefficients and are disassortative. Instead, the authors of [4] found out that, despite the high number of ERC-20 tokens, only a few are active and valuable. Moreover, few accounts hold a large number of tokens, while many accounts only hold a small number of tokens. Lastly, the authors discovered that some addresses create a large number of tokens to attack the Ethereum network.

If compared to prior works, our analysis is based on a broader time period and focuses on the top 100 networks with the highest number of token transfers. Moreover, our contribution is not solely focused on analyzing networks but also on comparing them with each other by associating each network with a set of numerical features capturing its topological properties. Lastly, our analysis also introduces a semantic classification of token contracts obtained by manually retrieving information from the internet.

2 Background

Blockchain. A blockchain is a shared, immutable, and decentralized ledger organized in blocks, each containing ledger state updates and managed through a *distributed consensus algorithm*. Ethereum [13] has been the first blockchain project implementing a Turing-complete virtual machine, called *Ethereum Virtual Machine* (EVM). This means that, besides monetary transactions, the Ethereum blockchain is also capable of storing and executing pieces of arbitrarily complex code, called *smart contracts* [10]. Smart contracts are written in a high-level language (e.g., Solidity) and then compiled to bytecode. Their execution is validated by distributed consensus and replicated by all participants. Specifically, each call to a function of a smart contract is executed sequentially in the current block state, and the final state is updated accordingly.

Decentralized Applications and Fungible Tokens. As stated in Sect. 1, smart contracts enable the development of *decentralized applications* (DApps), which may serve a wide range of purposes (e.g., finance, gaming, social networking). Many DApps adopted the concept of fungible token to represent interchangeable assets that can be transferred between participants. The ERC-20 implementation proposal [12] introduces a standard for fungible tokens. Specifically, it defines a consistent set of methods for creating and interacting with tokens. Also, it ensures token interoperability, meaning that all compliant tokens can be easily integrated into different decentralized applications. For the purposes of this paper, we remark that, whenever an ERC-20 contract transfers tokens between two addresses, an *event* must be raised. In Ethereum, events are a mechanism adopted to notify a state update or a particular condition being met during the execution of a smart contract. This facilitates the communication between contracts and off-chain applications. In Solidity, events are identified by a signature specifying the type and number of their parameters. The signatures of the Transfer and Approval events defined by the ERC-20 standard are:

```

event Transfer(address, address, uint256)
event Approval(address, address, uint256)

```

The Transfer event is emitted every time a token transfer occurs between two addresses. Its signature consists of three parameters: the sender address, the recipient address, and the amount of tokens transferred. Conversely, the Approval event is triggered when a user allows another participant to transfer a certain number of tokens on their behalf. We observe that, according to the ERC-20 standard definition, after the issuance of an Approval event, a Transfer event notifying the actual transfer of tokens must necessarily follow. Thus, for the remainder of this paper, we will only consider Transfer events to study token transfers among participants.

3 Transfer Event Graph

Transfer events represent redistributions of tokens between two users. By gathering information about the occurrences of such events, it is therefore possible to analyze the evolution of a token economy. To this aim, in this section we formalize the concept of Transfer event graph, i.e., the graph where nodes represent users and edges represent Transfer event occurrences. In the following, we denote by \mathcal{A} the set of all Ethereum addresses, which are used to identify network participants. An occurrence of a Transfer event can be represented as a tuple $e = (t, from, to, v)$, where $t \in \mathbb{N}$ is a numeric timestamp, $from \in \mathcal{A}$ is the address of the sender, $to \in \mathcal{A}$ is the receiver address and $v \in \mathbb{N}$ is the amount of tokens transferred. In the following, given a contract C , we denote by $\mathcal{T}(C)$ the set of ERC-20 Transfer events triggered by C . We can then define the *Transfer event graph* of C as a simple undirected graph $G_C = (V_C, E_C)$. Here, the set of vertices $V_C = \{a \in \mathcal{A} \mid \exists (t, from, to, v) \in \mathcal{T}(C) \text{ s.t. } a = from \vee a = to\}$ contains all addresses induced by the events in $\mathcal{T}(C)$, while the set of edges $E_C = \{(a, b) \mid \exists (t, from, to, v) \in \mathcal{T}(C) \text{ s.t. } a = from \wedge b = to\}$ includes one edge between two nodes a and b if and only if there exists at least one token transfer between them.

4 Experimental Results

In this section we present the experimental results of our analysis of token networks. First, we study the evolution of Transfer events over time. Then, we compare the topological properties of Transfer event networks and examine possible connections between such properties and the semantics of the corresponding smart contracts. For our experiments, we downloaded the first 15 million blocks of the Ethereum blockchain along with the corresponding transaction receipts, which include all necessary information about triggered events. The time period covered by our data set ranges from July 30th, 2015 03:26:13 PM UTC, to June 21st, 2022 02:28:10 AM UTC. The code for the experiments and data analysis has been written in C++, Java and Python and is publicly available at <https://github.com/mlporchio/EthTokenAnalysis>. In particular, the Transfer event graph analysis was conducted using igraph [6] and WebGraph [1].

4.1 Global Analysis

By analyzing the transaction receipts in our data set, we were able to collect $N_e = 961\,603\,795$ occurrences of the Transfer event, raised by $N_c = 386\,615$ different smart contracts. The plots of Fig. 1 provide further insight into the occurrences of Transfer events. In particular, Fig. 1a illustrates the frequency of ERC-20 Transfer events within the analyzed blocks. It appears that a significant number of blocks (i.e., above 10^6) do not contain any occurrence of such events. Also, we can notice that blocks with a large quantity of transfers are less frequent. Instead, Fig. 1b illustrates the total number of Transfer events on a monthly basis starting from 2015 (i.e., the year of the Ethereum blockchain inception) until June 2022. Using a logarithmic scale on the y -axis, the plot highlights how the number of such events experienced a rapid growth in 2016 and 2017, before stabilizing at around 10^7 transfers per month starting from 2018.

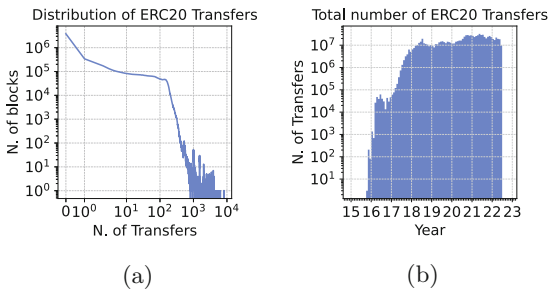


Fig. 1. Frequency distribution of ERC-20 transfers (left) and monthly number of raised Transfer events (right).

4.2 Graph Construction

To gain insight on the trading volume of each token economy, we first ranked the ERC-20 contracts based on the number of raised Transfer events. Table 1 displays the first ten positions of our ranking. As the reader may notice, these contracts alone include approximately 357 million occurrences, thus covering about 37% of the total number of events N_e despite being less than the 0.012% of the number of contracts N_c . Moreover, we can also observe that eight tokens out of ten are related to the field of decentralized finance, as they are associated with stablecoins or wrapped tokens. The only exceptions are represented by the tokens of ChainLink [3], i.e., a decentralized oracle network, and Livepeer, a framework for decentralized video streaming applications.

We selected the top 100 contracts from our ranking and constructed, for each of them, the corresponding Transfer event graph, as discussed in Sect. 3. We then computed the number of nodes and edges of each graph and noticed that, on average, Transfer event graphs have about 759 004 nodes and 1 701 879 edges. We remark that the number of nodes coincides with the number of participants in the

Table 1. Top 10 ERC-20 token contracts by triggered Transfer events.

Contract address	Token name	N. of Transfers	Percentage
dac17f958d2ee523a2206206994597c13d831ec7	Tether USD (USDT)	149 408 698	15.537
c02aaa39b223fe8d0a0e5c4f27ead9083c756cc2	Wrapped Ether (WETH)	104 183 120	10.834
a0b86991c6218b36c1d19d4a2e9eb0ce3606eb48	USD Coin (USDC)	42 601 224	4.430
6b175474e89094c44da98b954eedeac495271d0f	Dai Stablecoin (DAI)	14 387 573	1.496
514910771af9ca656af840dff83e8264ecf986ca	ChainLink Token (LINK)	11 388 177	1.184
174bfa6600bf90c885c7c01c7031389ed1461ab9	More Gold Coin (MGC)	8 947 669	0.930
95ad61b0a150d79219dcf64e1e6cc01f0b64c4ce	SHIBA INU (SHIB)	7 781 424	0.809
990f341946a3fdb507ae7e52d17851b87168017c	Strong (STRONG)	6 964 935	0.724
58b6a8a3302369dac383334672404ee733ab239	Livepeer Token (LPT)	6 025 932	0.627
03cb0021808442ad5efb61197966aef72a1def96	coToken (coToken)	5 370 855	0.559
Total		357 059 607	37.130

corresponding token economy. For a more detailed insight, Fig. 2a summarizes the cumulative frequency of the number of nodes among all graphs. From the plot, it is possible to notice that the majority of all graphs has between 10^4 and 10^6 nodes. Specifically, we can notice that 80 graphs out of 100 have less than 1 million nodes. Similarly, Fig. 2b illustrates the cumulative distribution function for the number of edges, highlighting that approximately 80% of all graphs have less than 1 million edges. Speaking of graph sizes, we observe that the graph with the lowest number of nodes, amounting to 691, corresponds to the “Bancor Network” token, which is related to the field of decentralized finance. Instead, the graph with the highest number of nodes, namely 23 176 194, is that of “Tether USD” token, the stablecoin holding the first position in Table 1. To give a sense of our data set, we note that, if all 100 graphs were combined into a single graph describing all participants and transfer events of the corresponding 100 economies, the resulting graph would comprise 59 120 625 unique nodes and 160 259 567 unique edges.

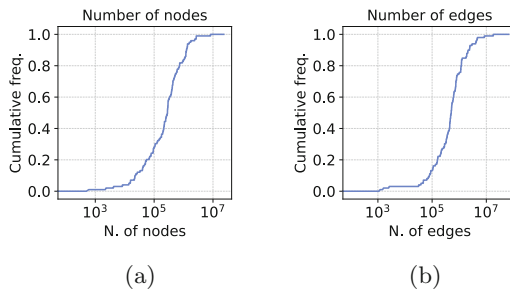


Fig. 2. Cumulative distributions for number of nodes (left) and edges (right) of the considered Transfer event graphs.

4.3 Graph Analysis

We then analyzed the constructed Transfer event graphs. To this aim, we associated each graph with seven numerical features capturing their topological

properties. To deal with disconnected graphs, we have chosen to always compute such measures on the largest connected component for consistency. As such, all features we describe from now on always refer to the subgraph induced by the nodes and edges of the largest component. In particular, given a Transfer event graph G with largest connected component G_{LCC} , we computed the following features. (1) *Coverage*, namely the percentage of nodes of G included in G_{LCC} . (2) *Alpha*, which represents the exponent of the power law distribution best fitting the degree distribution of G_{LCC} . (3) *Fitting error*, which corresponds to the error obtained during the fitting process to obtain the previously described alpha. (4) *Relative diameter*, which represents the ratio between the diameter of G_{LCC} and the natural logarithm of the number of nodes. (5) *Relative average shortest path length*, which is computed as the average shortest path length of G_{LCC} divided by the natural logarithm of the number of nodes. (6) *Transitivity* coincides with the global clustering coefficient of G_{LCC} , namely the ratio between the number of triangles and connected triples in the graph. (7) *Density*, as the ratio between the actual number of edges and the maximum possible number of edges in G_{LCC} . To fit a power law curve on the degree distribution of each graph, we used the procedure detailed in [5]. In accordance with such method, we use the Kolmogorov-Smirnov statistic to quantify the fitting error as the distance between the two distributions. Moreover, we remark that the average shortest path lengths have been computed using the HyperBall algorithm, which provides an approximate but reasonably accurate result [2]. Indeed, due to the sizes of the analyzed graphs, obtaining the exact value for the lengths turned out to be too computationally expensive.

Figure 3 summarizes the distributions of the features among all graphs. In particular, the histogram of Fig. 3a illustrates the coverage distribution and provides information about the connected components of the examined graphs. We can observe that, for 98% of the graphs, the largest connected component covers a percentage of nodes ranging from 90% to 100%. This means that, in most cases, as the token economy evolves, token transfers tend to create a single, large community of users, with only a few nodes remaining isolated. There are, however, two graphs where the coverage percentage falls between 10% and 20%. A further analysis revealed that these two outliers correspond to the “Etheal Promo” and “INS Promo” tokens, whose largest connected components cover around 18% and 14% of all nodes, respectively. Both tokens were launched on the market through *airdropping*, a marketing strategy where tokens are sent to existing users’ wallets, typically as a free giveaway.

Our analysis of node degrees is summarized by Figs. 3b and 3c, which illustrate the distributions of the fitted power law exponents and fitting errors, respectively. More than half of the tokens have a power law exponent between 2.5 and 3.75, while the majority of graphs have a fitting error below 0.05. Indeed, we observed that the mean fitting error over all graphs is 0.02. Interestingly enough, the graph with maximum fitting error (i.e., approximately 0.15) corresponds to the “More Gold Coin” token. As discussed in [7], the associated contract address is known for its spamming campaign, which took place in July 2019. During this

massive campaign, small quantities of tokens were airdropped to many users causing a sudden congestion on the entire Ethereum network.

For what concerns the relative diameter, we observe a mean value of approximately 1.55. Indeed, Fig. 3d shows that, for more than 70% of all graphs, this feature is below 2. So the diameter is within a low linear factor of the logarithm of number of nodes, a classical behaviour in small world networks. Similarly, for the relative average path length, Fig. 3e shows how the values for this feature are concentrated between 0.2 and 0.3 for most graphs, with a mean of 0.28.

The histograms of Figs. 3f and 3g describe the transitivity and density distributions, respectively, using a logarithmic scale on the y -axis. As the reader may notice, in both cases the distributions are positively skewed, with a mean value of about 3.55×10^{-4} for transitivity and 2.07×10^{-4} for density. This suggests that interactions among participants tend to be sparse. Moreover, it leads us to believe that token networks have a weak community structure and participants are not likely to form well-connected groups, in contrast with the small world behavior observed when looking at the diameter.

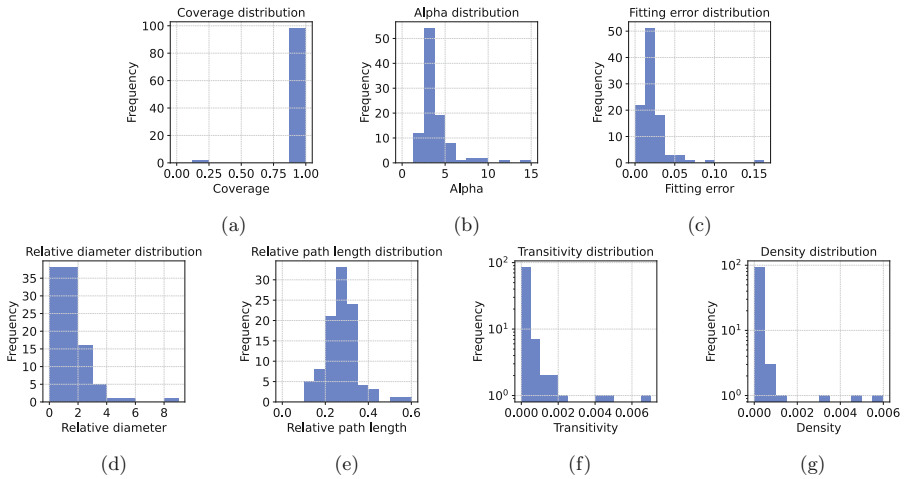


Fig. 3. Distributions of the selected features.

4.4 Clustering

After examining the features of each graph individually, we conducted another analysis employing clustering techniques. The goal of this analysis is to identify groups of contracts with similar topological properties. For our initial experiment, we attempted to identify which subset of the features described in Sect. 4.3 yields the best clustering. To achieve this aim, we employed the K-means algorithm, testing all possible feature subsets while varying the number of clusters k from a minimum of 2 to a maximum of 20. For each subset, we then selected

the value of k maximizing the silhouette coefficient. We note that, with 7 different features, the number of valid subsets is equal to 127. Each subset then generates 19 possibilities, resulting in a total of 2 413 combinations. Figures 4a, 4b and 4c illustrate, respectively, the top three clusterings obtained with this approach, namely those with the highest silhouette scores. As the reader may notice, all three configurations comprise $k = 2$ clusters. The first configuration, with a silhouette of 0.945, was obtained using only the coverage feature. The second configuration, which returned a score of 0.834, was obtained using only the density feature. Finally, the third configuration was obtained by combining both features together, yielding a silhouette of 0.785. We observe that, in all three cases, the obtained clusterings are highly imbalanced. Indeed, we can always find a small cluster, containing no more than 20 elements, and a large cluster, with more than 80 elements.

To attempt a different clustering approach, we also conducted further analysis based on dimensionality reduction. In particular, we used principal component analysis to reduce the number of features and then executed the K-means algorithm on this reduced data set. Before applying the dimensionality reduction, however, we used the explained variance ratio method to determine the optimal number of components. More precisely, we set a threshold of 0.8 (to keep 80% of the total variance of the original data) and selected the minimum number of principal components such that the explained variance ratio is above the threshold. In this regard, the plot of Fig. 4d illustrates the total explained variance ratio as the number of components varies. As the reader may notice, it appears that the optimal number of features is equal to 4. We then applied the K-means algorithm again to the reduced data set, trying values of k ranging from 2 to 20. As before, among the 19 configurations tested, we chose the one that maximized the silhouette score. As shown in Fig. 4e, the maximum silhouette value (slightly above 0.7) is achieved, once again, for $k = 2$ clusters. The corresponding clustering for this configuration is described by the plot of Fig. 4f: it can be observed that this partitioning is highly unbalanced, with 97 contracts assigned to the first cluster and only 3 elements to the second one.

Considering the difficulty encountered in separating contracts according to the associated features, we introduced a new classification based on contract semantics. Specifically, we manually assigned to each contract a categorical label describing its main application domain. The ultimate goal of this analysis was to study the composition of the obtained clusters, in order to determine whether similar graphs correspond to contracts with similar purposes. In this regard, we identified nine token categories: (1) *defi* comprises all tokens related to decentralized finance (e.g., stablecoins, wrapped tokens, tokens issued by exchanges and automated market makers, etc.); (2) *games* includes all token related to games; (3) *blockchain* denotes all tokens related to independent blockchain projects; (4) *layer-2* contains tokens related to layer-2 solutions aimed at improving the scalability of Ethereum; (5) *content* includes reward tokens related to content creation platforms; (6) *storage* represents all tokens related to decentralized storage solutions; (7) *mining* indicates tokens associated with cryptocurrency mining

services; (8) *multimedia* comprises all tokens related to multimedia content (e.g., music, video streaming services, etc.); (9) *other* comprises all tokens whose application domain is not included into any of the previous categories. Table 2 illustrates the number of contracts for each application domain. We can notice that the most numerous category is that of tokens related to decentralized finance, comprising 54 contracts out of 100. Furthermore, 15 contracts did not fall into any of the application domains and were therefore labeled as “other”.

Table 2. Contract classification based on their application domain.

Category	Count
defi	54
other	15
games	9
blockchain	5
layer-2	4
content	4
storage	4
mining	3
multimedia	2
Total	100

We then used this labeling to measure clustering *homogeneity*. Homogeneity quantifies, on a scale from 0 to 1, how much each cluster predominantly contains elements belonging to a certain category of contracts [8]. We assigned a score to each clustering by comparing the labels returned by the K-means algorithm with our manually-assigned categories. To better understand how the clustering reflects such categories, we have focused on clustering results with $k = 8$, i.e., one cluster per category excluding the heterogeneous “other” category. In this regard, Fig. 4g reports the clustering result with $k = 8$, colored by category, yielding the maximum silhouette among all possible combinations of features. Moreover, to also illustrate the best possible division of the categories among clusters, we show in Fig. 4h the result with maximum homogeneity. Finally, in Fig. 4i we report the coloring for $k = 8$ considering the principal component analysis clustering. In all cases we can see how the semantic categories are spread among different clusters. Indeed, in Fig. 4g, despite the high silhouette score indicating a good level of cohesion among the elements within each cluster, the homogeneity of the clusters is rather low. Conversely, the configuration of Fig. 4h exhibits a higher homogeneity, but a lower silhouette score. This suggests that, while the graphs have similar topological properties, their similarity does not reflect on the application domain of the respective contracts. In other words, the topology of Transfer event graphs is not a good indicator of the semantics of the corresponding contracts.

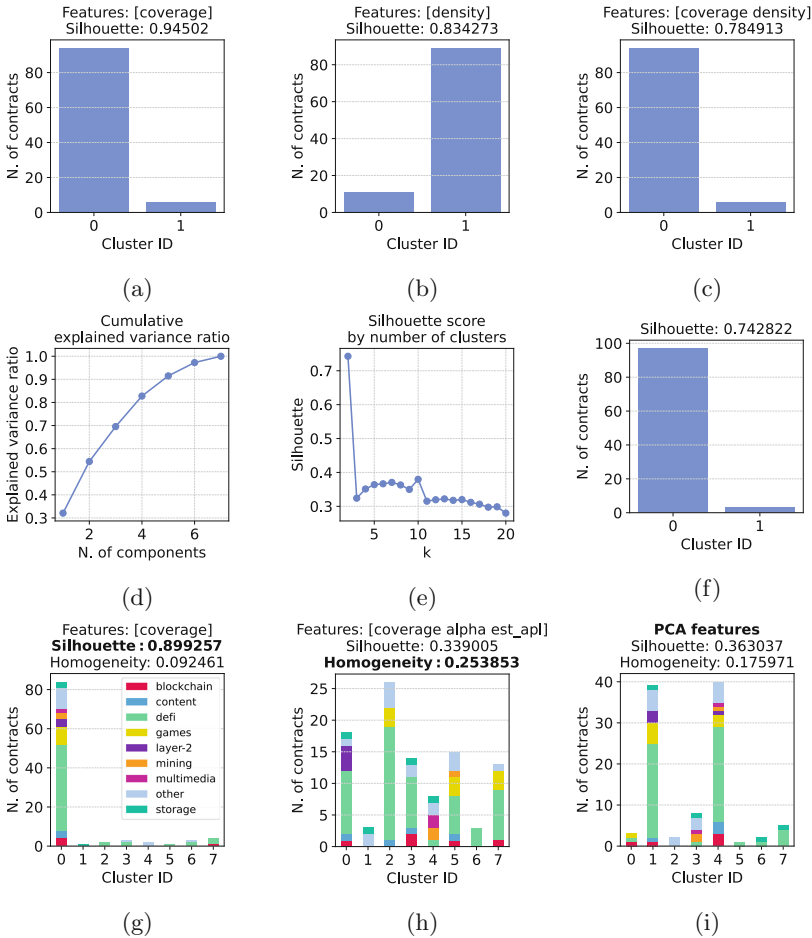


Fig. 4. Clustering analysis results (*est_apl* in figure (h) represents the relative average shortest path feature).

5 Conclusions and Future Work

In this paper we have analyzed the top 100 ERC-20 token networks by number of transfers. The study of the topological properties has revealed that – despite their diameter being of the order of the logarithm of the number of nodes – all networks exhibit a low clustering coefficient. This leads us to believe that such graphs are not small-world networks. Moreover, by analyzing the structure of the largest connected components and their degree distributions, we identified three networks that are associated with promotional tokens. Such tokens were launched through airdropping campaigns and one of them is regarded as an attempt at spamming the Ethereum network by the user community. To identify groups of networks with similar topological characteristics, we conducted a clustering

analysis and compared the results with manually-assigned labels describing the application domains of the contracts. Results suggest that a token network topology does not effectively reflect the semantics of the associated contract, meaning that contracts with similar applications can induce different network structures, and vice versa. Concerning future work, we plan to further explore the relation between contract semantics and network topology by considering additional features and different clustering methods. It could also be possible to enrich the graph with edge weights (e.g., transfer timestamp or amount). The data set might also be expanded by considering more contracts, including non-fungible ones.

References

1. Boldi, P., Vigna, S.: The webgraph framework I: compression techniques. In: Proceedings of the 13th International Conference on World Wide Web, WWW 2004, New York, NY, USA, May 17-20, 2004, pp. 595–602. ACM (2004)
2. Boldi, P., Vigna, S.: In-core computation of geometric centralities with hyperball: a hundred billion nodes and beyond. In: 2013 IEEE 13th International Conference on Data Mining Workshops, pp. 621–628. IEEE (2013)
3. Breidenbach, L., et al.: Chainlink 2.0: next steps in the evolution of decentralized oracle networks. Chainlink Labs 1, 1–136 (2021)
4. Chen, W., Zhang, T., Chen, Z., Zheng, Z., Lu, Y.: Traveling the token world: a graph analysis of Ethereum ERC20 token ecosystem. In: WWW '20: The Web Conference 2020, Taipei, Taiwan, pp. 1411–1421. ACM / IW3C2 (2020)
5. Clauset, A., Shalizi, C.R., Newman, M.E.: Power-law distributions in empirical data. *SIAM Rev.* **51**(4), 661–703 (2009)
6. Csardi, G., Nepusz, T., et al.: The Igraph software package for complex network research. *Int. J. complex syst.* **1695**(5), 1–9 (2006)
7. Lucian, A.: Scam ‘More Gold Coin’ Clogs Ethereum Network Causing Gas Price to Spike. <https://beincrypto.com/scam-more-gold-coin-clogs-ethereum-network-causing-gas-price-to-spike>, Accessed 2 Sep 2023
8. Rosenberg, A., Hirschberg, J.: V-measure: a conditional entropy-based external cluster evaluation measure. In: Proceedings of the 2007 joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), pp. 410–420 (2007)
9. Somin, S., Gordon, G., Altshuler, Y.: Network analysis of ERC20 tokens trading on Ethereum blockchain. In: Unifying Themes in Complex Systems IX: Proceedings of the Ninth International Conference on Complex Systems 9, pp. 439–450 (2018)
10. Szabo, N.: Smart contracts: building blocks for digital markets. *EXTROPY: The J. Transhumanist Thought*,(16) **18**(2), 28 (1996)
11. Victor, F., Lüders, B.K.: Measuring Ethereum-Based ERC20 Token Networks. In: Goldberg, I., Moore, T. (eds.) FC 2019. LNCS, vol. 11598, pp. 113–129. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-32101-7_8
12. Vogelsteller, F., Buterin, V.: ERC-20: Token Standard (2015)
13. Wood, G.: Ethereum: a secure decentralised generalised transaction ledger (2014)