Temporal Analysis of the Entire Ethereum Blockchain Network

Lin Zhao
Nanyang Technological University, Singapore
lin018@e.ntu.edu.sg

Sourav Sen Gupta
Nanyang Technological University, Singapore
sg.sourav@ntu.edu.sg

Arijit Khan
Nanyang Technological University, Singapore
arijit.khan@ntu.edu.sg

Robby Luo
Advanced Micro Devices Inc., China
robby.luo@amd.com

ABSTRACT

With over 42 billion USD market capitalization (October 2020), Ethereum is the largest public blockchain that supports smart contracts. Recent works have modeled transactions, tokens, and other interactions in the Ethereum blockchain as static graphs to provide new observations and insights by conducting relevant graph analysis. Surprisingly, there is much less study on the evolution and temporal properties of these networks. In this paper, we investigate the evolutionary nature of Ethereum interaction networks from a temporal graphs perspective. We study the growth rate and model of four Ethereum blockchain networks, active lifespan and update rate of high-degree vertices. We detect anomalies based on temporal changes in global network properties, and forecast the survival of network communities in succeeding months leveraging on the relevant graph features and machine learning models.

CCS CONCEPTS

• **Mathematics of computing** → **Graph algorithms**; **Exploratory data analysis**; • **Applied computing** → **Digital cash**.

KEYWORDS

Blockchain, Ethereum, Smart Contract, Temporal Network Analysis

ACM Reference Format:
Lin Zhao, Sourav Sen Gupta, Arijit Khan, and Robby Luo. 2021. Temporal Analysis of the Entire Ethereum Blockchain Network. In *Proceedings of the Web Conference 2021 (WWW '21), April 19–23, 2021, Ljubljana, Slovenia.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3442381.3449916

1 INTRODUCTION

The emergence of blockchain technology provides new scenarios for transaction data mining. Generally speaking, blockchain is a distributed ledger of transactions or records, stored in a chronological or sequential order. Owing to its decentralized, traceable, immutable, and transparent nature (in most cases), blockchain is expected to be critical in the 'trust economy' of the future [48].

Cryptocurrencies and blockchain are tightly coupled, since the birth of blockchain technology with Bitcoin [55] over a decade ago. Subsequently many cryptocurrencies and business applications laid the foundation for inclusive decentralization and consensus-driven automation through incentives and smart contracts [15]. While Bitcoin and similar cryptocurrency networks deal only with users (wallets) transacting over blockchain, Ethereum [13] operates an automation layer on top of a permissionless blockchain through the smart contracts, which are autonomous agents that can execute complex logic across a decentralized network.

With over USD 42 Billion market capitalization (October 2020), Ethereum is the largest public blockchain that supports smart contracts [18]. Ethereum is a transaction-based state-transition machine, where the state is made up of accounts. Transfer of asset and information between accounts, recorded in the blockchain, cause transitions in the Ethereum 'world state'. There are two types of accounts in Ethereum – users and contracts. Transactions in Ethereum are initiated by user accounts, signed with their private keys, while internal messages in Ethereum can be generated by contract accounts. Ether is the primary asset (currency) for Ethereum blockchain. In addition to ether, Ethereum blockchain allows creation of Tokens, an abstraction of digital assets, via relevant methods implemented through smart contracts. Similar to transacting ether (the base currency), Ethereum accounts may also transact in various Tokens, through mechanisms defined in the respective smart contracts. This allows for a complex asset-transfer-ecosystem of various fungible (e.g., ERC20) and non-fungible (e.g., ERC721) tokens (assets) to flourish on Ethereum blockchain.

Ethereum, and similar public blockchains supporting smart contracts, also bring forth a fascinating ecosystem of humans (users) and autonomous agents (contracts), cohabiting the underlying blockchain fabric. It is neither like online social networks, where the players are all human users, nor like the core financial networks, where all interactions are transfer of value or asset. In essence, a blockchain network like Ethereum is closer to the Internet or Web, where users and programs are allowed to interact with one another, following predefined rules of engagement. In addition to this Web-like architecture, there is also an interaction framework for smart contracts (agents), where they can call, invoke, or kill each other to maintain and advance the 'world state' of the blockchain.

This motivates us to study Ethereum blockchain, as a representative of similar public blockchain networks supporting decentralized automation through smart contracts. We are interested in *all* interactions in the ecosystem: user-to-user, user-to-contract, contract-to-user, and contract-to-contract [44]. Details of our interaction networks are presented in § 3.

Motivation. Recent works [3, 15, 26, 36, 44, 67, 68, 71, 71] have modeled transactions, tokens, and other interactions in the Ethereum blockchain as *static graphs* to provide new observations and insights by conducting relevant graph analysis. Surprisingly, there is

WWW '21, April 19–23, 2021, Ljubljana, Slovenia

ACM ISBN 978-1-4503-8312-7/21/04.
https://doi.org/10.1145/3442381.3449916

much less study on the evolution and temporal properties of these networks. In this paper, we investigate the evolutionary nature of Ethereum interaction networks from a *temporal graph* perspective. Specifically, we aim at addressing three main research questions:

(1) How do Ethereum blockchain networks evolve over time? What growth model do they follow? What is the active lifespan of each vertex? How do the high-degree vertices change over time?

(2) How network properties (e.g., reciprocity, assortativity, clustering coefficient, core decomposition) change over time for Ethereum blockchain networks? Do they indicate anomalies and other external aspects of the network (e.g., popularity, exchanges)? What is the right "time granularity" for such temporal analysis?

(3) Can we detect meaningful communities in Ethereum blockchain networks, and also forecast the 'continuation' (survival) of these communities in succeeding months leveraging on the relevant graph properties (features) and machine learning models?

Contributions. Our main contributions are as follows.

(1) To the best of our knowledge, we are the first to conduct a comprehensive study of the evolutionary, temporal, and predictive aspects of the large-scale Ethereum blockchain network, cohabited by both human users and autonomous smart contracts. We investigate their complex interactions by constructing four temporal networks from the entire Ethereum blockchain data, namely TraceNet, ContractNet, TransactionNet, and TokenNet, at various time granularities. We open source our code and dataset [79].

(2) We study the annual growth rate of four blockchain networks, demonstrating that Ethereum interaction networks are growing at a fast speed and the account information is updated at a fast pace, however all the graphs get sparser and mature over time, and follow the *preferential attachment* growth model. The user accounts remain active much longer than smart contracts on Ethereum (§ 4).

(3) We employ global network properties such as reciprocity, assortativity, clustering coefficient, and core decomposition, to detect significant changes and anomalies over Ethereum blockchain networks. We correlate these anomalies with external aspects of the network, e.g., popularity, exchanges, and systematically drill-down to the appropriate time granularity for our analyses (§ 5).

(4) We forecast the 'continuation' (survival) of certain network communities in succeeding months leveraging on the relevant graph properties (as features) and machine learning models, achieving up to 77% correct predictions for continuation (§ 6).

Our results will be useful for emerging fields such as blockchain intelligence (https://blockchaingroup.io) and blockchain-based social networks [58, 60] that are building blockchain search engines, making use of data mining and analytics skills to help clients avoid transaction risks. We matched our network analysis results with real-world incidents (§ 4, 5, 6). Researchers working in natural language processing and sentiment analysis using tweets and online articles about blockchain [41, 72] can find supporting views and references from our work. Our community longevity prediction method and results (§6) can be utilized by companies to build blockchain ecosystems.

2 RELATED WORK

Graph analyses of cryptocurrency networks. Several works have studied Bitcoin and other cryptocurrency networks based on graph theory and network analyses. These studies have been feasible due to the transparency offered by public permissionless blockchain, which allows anyone to access transactional information on the networks. Graph analyses of blockchain started with the motivation of *de-anonymizing* the "pseudonymous" Bitcoin accounts, via clustering addresses based on transaction behavior [17, 50]. Similar analyses have since been performed on other "anonymous" cryptocurrencies like Monero [54], Zcash [35], and across different cryptocurrency ledgers [77]. BitIodine [69] and Elliptic [1] performed *chain analysis* on the Bitcoin transaction network to extract intelligence. Other works measured network characteristics to predict the market-price of Bitcoin [31, 39, 75], search for influential patterns [24], information propagation [20], stability of Bitcoin P2P network [21], among others.

Network properties of transaction graphs. The large-scale network properties of Bitcoin transaction graph were studied in [32, 64], and the abstraction of any blockchain as a transaction network for analysis has been considered in [3]. Unlike our study of more diverse interaction networks in the Ethereum blockchain, these works were only about the bitcoin (or cryptocurrency) transaction graph, where all interactions are transfer of value. Ferretti and D'Angelo also studied only transactions in the Ethereum blockchain [26] – not all types of network interactions. Somin et al. investigated the entire address graph spanned by ERC20 token trade in Ethereum blockchain [67], and also studied the social signals in the Ethereum ERC20 token trading network [68]. Victor and Lüders recently measured Ethereum-based ERC20 token networks [71].

Measurements on Ethereum blockchain network. To the best of our knowledge, mainly two past works [15, 44] measured the entire Ethereum blockchain network and all interactions therein — via large-scale *static* graph analyses. These approaches closely followed the norms of measuring social networks, Internet, and the Web [2, 12, 25, 45, 53, 66, 74], as the entirety of the Ethereum blockchain network presents itself as an equally complex system. Surprisingly, none of the aforementioned works studied the evolution and temporal properties of these networks. In this paper, we investigate the evolutionary nature of Ethereum interaction networks from *temporal graphs* perspective.

Temporal analyses of Ethereum blockchain network. Very recently and almost concurrent to ours, Bai et al. [7] analyzed the evolutionary behavior of various interactions in Ethereum blockchain from a temporal graph point of view, and correlated them with the average Ether price in a time window. Different from [7], we deeply investigate the growth model of Ethereum blockchain network, active lifespan of different types of vertices, how the network properties evolve at various time granularities and correlate them with anomalies and other external factors of the network, and last but not least, leveraging on the relevant graph properties (features) and machine learning models we forecast the 'continuation' (survival) of network communities in Ethereum blockchain graphs.

Temporal analyses of social networks, Internet, and the Web. Prior studies focused on the evolution and growth of online social networks, including Flickr, LiveJournal, Yahoo! 360, movie-actor, email, and scientific collaboration networks [6, 33, 40, 43, 52, 56, 61]. Their major findings include (a) *preferential attachment* (or its variant) growth model: vertices arrive one by one, and link themselves to a pre-existing vertex with probability proportional to the degree of the latter, (b) proximity bias in link creation between existing users, and (c) density of the network follows rapid growth, decline, and then slow but steady growth.

Kleinberg [37, 38], Watts and Strogatz [73] proposed models to explain the small-world phenomenon and navigability in social networks. Leskovec, Kleinberg, and Faloutsos [46] considered citation graphs and showed that these exhibit densification and shrinking diameters over time; they proposed a *forest-fire* graph model to explain the decreasing diameter phenomenon. The linkage pattern of blogs and the emergence of bursty communities in the blogspace were studied in [42]. Structural properties of different snapshots of the world-wide web graph and online collection growth in Pinterest were investigated in [27, 59] and [47], respectively.

In the context of social network groups, the relationships between structural features of a group and its future growth were analyzed in [6, 34]. Ribeiro et al. studied the recent evolution of the 'Manosphere', web-based misogynistic groups [63]. To the best of our knowledge, we are first to analyze and forecast the 'continuation' (survival) of communities in Ethereum blockchain network (in fact, for any blockchain network) leveraging on the relevant graph properties (features) and machine learning models.

3 DATASETS AND EXPERIMENTAL SETUP

There are two types of 'accounts' on Ethereum: **(1)** Externally Owned Accounts (EOA) or User Accounts, operated with private keys typically owned by human users/wallets, and **(2)** Contract Accounts (CA), governed by the internal contract code acting as an autonomous agent. There are a few *special* accounts in Ethereum, like the Null address `0x0`[1] used for contract creation, and the Burn address[2] used for 'burning' ether. There are four types of interaction between these Ethereum accounts: **(1)** User-to-User (transaction or token transfer), **(2)** User-to-Contract (call or kill), **(3)** Contract-to-User (transaction or token transfer), and **(4)** Contract-to-Contract (create, call, kill, or hard fork). Interactions from a User or Contract **to** the Null address `0x0` denote creation of smart contracts, and transactions to User or Contract accounts without a **from** address denote generation of ether as mining rewards.

Table 1: Ethereum Blockchain Data : Block #0 to #9193265

	Approximate Size of Dataset	Row Count
contracts	21.7 GB	20440014
transactions	265 GB	611647042
traces	702 GB	1290574220
token transfers	97.7 GB	168407170

In this paper, we study temporal evolution of all these interactions between Ethereum accounts, by constructing various *interaction networks* from the Ethereum blockchain data, where vertices are accounts (users or contracts) and arcs represent their interactions. In past literature [15, 26, 44, 67, 68], different *static* interaction networks were constructed, such as money flow graph, smart contract creation graph, smart contract invocation graph, transaction and token networks. Following one of the most recent studies [44], we consider four interaction graphs introduced below, which provide us the most comprehensive view of all interactions in the Ethereum blockchain network. Since we are interested on the temporal analysis, given a **start** and **end** time, we consider the accounts (vertices) and interactions (arcs) present within that duration.

3.1 Data Extraction

We extract all relevant data from the `ethereum_blockchain` dataset under the Google Cloud `bigquery-public-data` repository [19] till `2019-12-31 23:59:45 UTC`, which amounts to all blocks from genesis (#0) up to **#9193265**. Google BigQuery is a data warehouse that handles large-scale data and makes it easy to access via SQL interface. Ethereum blockchain data is available on it and is updated daily. The entire blockchain data is stored in seven different tables, out of which, we extract data from 'contracts', 'token transfers', 'traces', and 'transactions' tables for our temporal analysis (Table 1).

The **traces** table stores executions of all recorded (successful) messages and transactions in the Ethereum blockchain. The from and to addresses recorded in each trace help us create individual arcs in the interaction network of Ethereum accounts (unique 20-byte addresses), and it is also possible to group all traces triggered by a particular transaction. This is the most comprehensive table for analysis. The **transactions** table contains all transaction details such as source and target address, and amount of ether transferred. The transaction table contains all the transactions sent by user accounts. Transactions carry messages containing *value* (ether) or *data* (information) from a user to another user or to a smart contract. Transaction table also contains "User to Null" transactions that result in creation of new smart contracts. The **contracts** table contains all Contract Accounts, their byte code and other properties of byte code such as `block_timestamp`, `block_number`, `token` types (e.g., ERC721, ERC20). The **token transfers** table focuses on all transactions with *tokens* on the blockchain.

3.2 Ethereum Blockchain Networks

We create four interaction networks to perform temporal analysis on Ethereum blockchain, based on the extracted tables, as follows.

TraceNet is built from the 'traces' table. It contains **all** user and smart contract accounts (in the given period) as vertices, and **all** recorded messages and successful transactions as arcs. This is the most comprehensive interaction network for the blockchain.

ContractNet is also constructed from the 'traces' table, since the 'contracts' table only contains information for each smart contract account, and not their transactions. We build ContractNet by considering those transactions in the 'traces' table where both **from** and **to** addresses are smart contracts (verified using the 'contracts' table). Therefore, ContractNet is a sub-graph of TraceNet.

TransactionNet is built based on the 'transactions' table, which contains all transactions **from** user accounts **to** other users, smart contracts, or the Null address `0x0`. The vertices in the graph are addresses (users, contracts, null) and the arcs are the transaction activities (transfer of ether, call, kill, contract creation).

TokenNet is built based on the 'token transfers' table containing the token-based transaction activities from one address to another. The vertices are still addresses (users, contracts), but the arcs only denote movement of tokens (e.g., ERC721, ERC20) of various types.

Importance of interaction networks. Each aforesaid interaction network provides us with a distinctly different perspective

[1] Null : https://etherscan.io/address/0x0000000000000000000000000000000000000000
[2] Burn : https://etherscan.io/address/0x000000000000000000000000000000000000dead

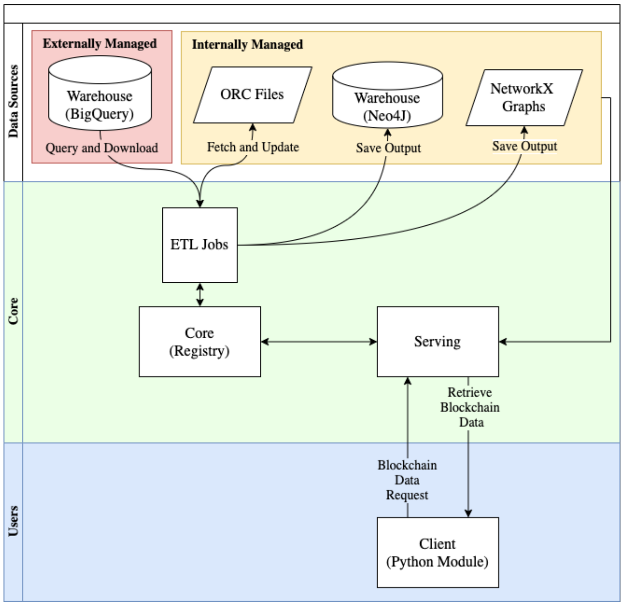


Figure 1: Data extraction pipeline for Ethereum data analysis

on Ethereum blockchain [44] — "While TraceNet presents a global view of interactions between Ethereum accounts, ContractNet focuses only on the automated multi-agent network of contracts, providing us with a functional view of the Ethereum state machine. TransactionNet helps us analyze the base ether transactions in the blockchain, and TokenNet focuses on the rich and diverse token ecosystem built on top of the Ethereum blockchain."

Environment setup. We conduct the experiments on a single core of a 32GB, 2.67GHz server. The code is implemented in Python 3.7. NetworkX and python igraph libraries are used for our analysis. We partially follow the planned data pipeline depicted in Figure 1, where we will extract the tables from Google BigQuery, cache the data in ORC files, create the four interaction networks in Neo4j format, and finally analyze the derived graphs in python.

Open-sourcing data and tool. Our code and data corresponding to the four extracted graphs are available at [79]. We are developing an automatic ETL process (in Python), as described in Figure 1. We plan to publish the automated toolchain in due course of time.

4 EVOLUTION OF THE ETHEREUM BLOCKCHAIN NETWORKS

In this section, we investigate the following questions by analyzing the *annual changes* in the four interaction networks. **(a)** How do Ethereum blockchain networks evolve over time? **(b)** What growth model do they follow? **(c)** What is the active lifespan of each vertex? **(d)** How do the high-degree vertices change over time?

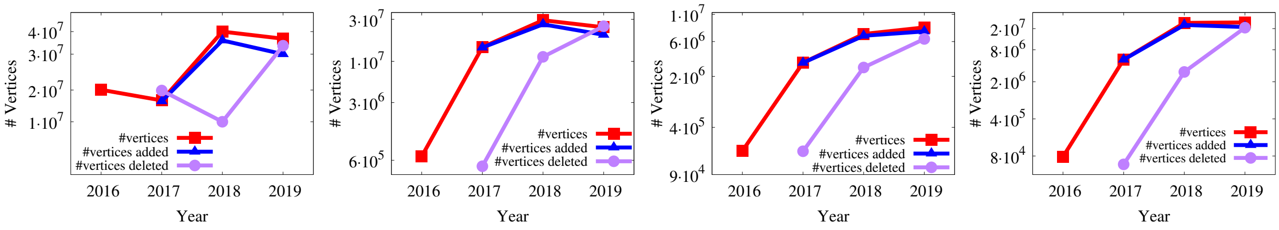
4.1 Evolution of Vertices, Arcs, and Density

We measure the number of vertices, arcs, and their evolution over the years. Ethereum blockchain started in July 2015, and thus, there is only half a year's data for 2015. For consistency of our analysis on a yearly yardstick, we start from the year 2016. We measure the number of new vertices or arcs added, and the number of old vertices or arcs deleted, over each pair of consecutive years. Therefore, these measurements start from 2017. We use simple undirected versions of the four interaction networks, that is, we consider at most one, undirected arc between every pair of vertices, with multiple arcs between vertices counted only once in the simple graph. The density for each network is computed as the number of existing arcs over the number of all possible arcs in the simple undirected version.

Figures 2 and 3 demonstrate the annual growth of vertices and arcs for the four networks. In general, with the graph size expanding, the numbers of vertices and arcs both increased. We notice that the number of new vertices and arcs added into the graphs is almost of the same order of total number of vertices and arcs in the graph at that time, respectively. *This implies that the Ethereum interaction networks are growing at a fast speed and the account information is updated at a fast pace as well.* This is indicative of a highly active network, which implies that any analysis of Ethereum blockchain and any prediction task should be conducted in a short-time period so as to achieve a good-quality consistent result.

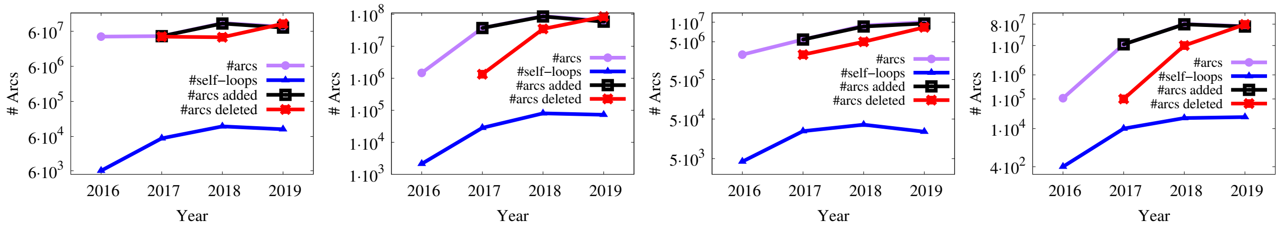
In case of TransactionNet, TraceNet, and TokenNet, the highest number of vertices and arcs were measured in 2018. In these three networks larger number of new vertices and arcs appeared over 2018, and less number of vertices and arcs disappeared. However in 2019, the number of removal exceeded the number added, resulting in the graph sizes to shrink a little. In contrast, we see that ContractNet keeps an upward trend over all years, though in 2019 its increment rate also reduced because a larger number of smart contract accounts were removed. This matches quite closely with reality, as we know that around mid-2019, Ethereum was considered to offer less features compared to the new chains (e.g., EOS, Tron), resulting in a number of contracts moving to the new chains, or to their own chains (like Binance). It is also interesting to note that the cost of deploying contracts on Ethereum blockchain rose sharply after the high-activity period in 2018, demotivating developers to host their applications (contracts) on the chain. The continuing debate on smart contract security and chain scalability issues for Ethereum during early months of 2019 did not help either. During this period, out of the top-50 dApps listed on DappRadar (https://dappradar.com), only three were deployed on Ethereum.

Figure 4 shows the graph density for four networks across five years. Since the density is measured as a *ratio* of the number of arcs over the number of possible arcs, we consider the density values in 2015 even though we only have half a year's data. *With the expansion of networks, the densities drops, indicating that all the graphs get sparser over time.* We notice that the arc increment over two consecutive years is less than vertex increment over the same years for all the networks, resulting in the density to drop. This indicates that the utilization of Ethereum network is low, as the accounts (vertices) interact with only a limited number of other accounts (vertices). We have noticed that the multi-arc, directed graph densities also drop over years, with the trend similar to Figure 4. It seems that all four networks approach saturation in terms of density by 2018 and 2019. TraceNet is the only network which seems to have hit the saturation point in 2016, much earlier than the other three. This depicts the actual underlying trend of sparsity in Ethereum blockchain, as TraceNet is the only graph that contains all vertices and all arcs corresponding to Ethereum interactions. The other networks consider only specific subsets of vertices, thereby underestimating the total number of possible arcs.



(a) traceNet (b) transactionNet (c) contractNet (d) tokenNet

Figure 2: Evolution of #vertices for TraceNet, TransactionNet, ContractNet, and TokenNet



(a) traceNet (b) transactionNet (c) contractNet (d) tokenNet

Figure 3: Evolution of #arcs for TraceNet, TransactionNet, ContractNet, and TokenNet

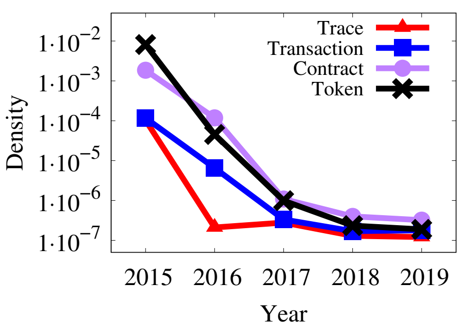


Figure 4: Evolution of density for the four networks

Table 2: TraceNet: New vertices connecting with old vertices

year	# old vertices	# new vertices	# new vertices with arc to old vertices (% of new vertices)	# new vertices without arc to old vertices (% of new vertices)
2017	170805	17531523	6014500 (34.31%)	11517023 (65.69%)
2018	3928217	35850173	17405374 (48.55%)	18444799 (51.45%)
2019	6081076	30502971	21597957 (70.81%)	8905014 (29.19%)

4.2 Network Growth Model

Tables 2-5 show the connection of a new vertex with an old vertex from the previous year. In the four networks, we calculate the number of new vertices, the number (and percentage) of new vertices which have at least one arc in the current year to some old vertex, and the number (and percentage) of new vertices which have no arc in the current year to any old vertex. Clearly, by an old vertex we refer to an account that exists in the current year and also in the previous year. Although the old vertices constitute only a small fraction of total vertices in the current year (Figure 2), the percentage of new vertices which have arcs to old vertices seem to increase every year. In 2018, for example, 41% of new vertices connect to old vertices in ContractNet. This ratio increases to 74% in 2019, which clearly indicates that *as the Ethereum network matures, more accounts remain active and more than half of new vertices participate in interactions with old vertices.*

Table 3: TransactionNet: New vertices connecting with old vertices

year	# old vertices	# new vertices	# new vertices with arc to old vertices (% of new vertices)	# new vertices without arc to old vertices (% of new vertices)
2017	163982	14789934	5646964 (38.18%)	9142970 (61.82%)
2018	3599770	28583252	14279239 (49.96%)	14304013 (50.04%)
2019	5060613	21240780	14807280 (69.71%)	6433500 (30.29%)

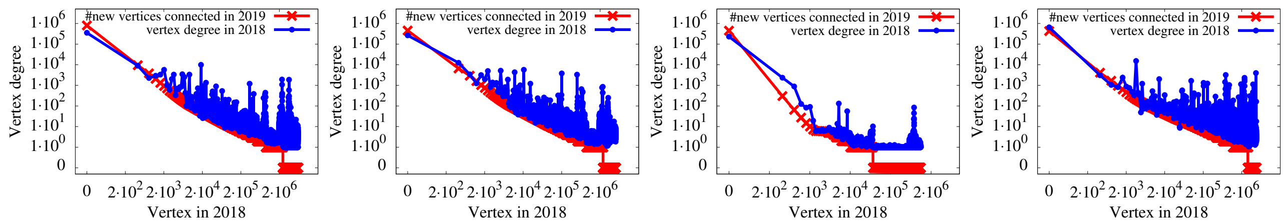
Table 4: ContractNet: New vertices connecting with old vertices

year	# old vertices	# new vertices	# new vertices with arc to old vertices (% of new vertices)	# new vertices without arc to old vertices (% of new vertices)
2017	1859	3070553	182920 (5.96%)	2887633 (94.04%)
2018	426000	7196954	2927928(40.68%)	4269026 (59.32%)
2019	1108567	8266061	6086678(73.63%)	2179383 (26.37%)

Table 5: TokenNet: New vertices connecting with old vertices

year	# old vertices	# new vertices	# new vertices with arc to old vertices (% of new vertices)	# new vertices without arc to old vertices (% of new vertices)
2017	21560	5220566	2045159 (39.18%)	3175407 (60.82%)
2018	2186066	23459461	9120122 (38.88%)	14339339 (61.12%)
2019	4797840	21402631	11922021 (55.70%)	9480610 (44.30%)

Figure 5 shows the correlation between old vertex degree in the previous year (2018) to its number of new connections in the current year (2019). On the X-axis, we sort the old vertices based on their number of connections to new vertices, while Y-axis presents the total degrees of these old vertices in 2018 (blue), as well as their number of connections to new vertices in 2019 (red). We see that if in previous year a vertex had high degree values, it is highly likely that in the current year it would have more new vertex connections. We also verified this observation statistically by measuring the Pearson Correlation Coefficient between the number of new vertices connected in 2019 and the vertices degree in 2018. In case of TransactionNet, TraceNet, TokenNet, and ContractNet, these correlation coefficients are 0.99841, 0.99911, 0.99965, and 0.99995, respectively, revealing the strong positive correlation between the two factors. This observation indicates that the Ethereum blockchain graphs follow the *preferential attachment* growth model: *The more connected a vertex is, the more likely it is to receive new arcs growth* [8].



(a) TraceNet (b) TransactionNet (c) ContractNet (d) TokenNet

Figure 5: Correlation between vertex degree to its number of new connections

Table 6: Top-10 highest degree vertices/accounts in TransactionNet

Top-10 accounts in 2018. Highlighted accounts continue being top-10 in 2019.

Account	Name	Type
0xea674fdde714fd979de3edf0f56aa9716b898ec8	**Ethermine**	**Mining Pool**
0x52bc44d5378309ee2abf1539bf71de1b7d7be3b5	**NanoPool**	**Mining Pool**
0x3f5ce5fbfe3e9af3971dd833d26ba9b5c936f0be	**Binance**	**Exchange**
0x2a0c0dbecc7e4d658f48e01e3fa353f44050c208	**IDEX**	**Dec. Exchange**
0x5a0b54d5dc17e0aadc383d2db43b0a0d3e029c4c	SparkPool	Mining Pool
0x829bd824b016326a401d083b33d092293333a830	F2Pool	Mining Pool
0x8d12a197cb00d4747a1fe03395095ce2a5cc6819	EtherDelta 2	Dec. Exchange
0xfbb1b73c4f0bda4f67dca266ce6ef42f520fbb98	**Bittrex**	**Exchange**
0xa7a7899d944fe658c4b0a1803bab2f490bd3849e	**IDEX 2**	**Dec. Exchange**
0x86fa049857e0209aa7d9e616f7eb3b3b78ecfdb0	EOSToken	Token (dead)

Top-10 accounts in 2019. Highlighted accounts were also in top-10 of 2018.

Account	Name	Type
0xdac17f958d2ee523a2206206994597c13d831ec7	TetherToken	Token
0x174bfa6600bf90c885c7c01c7031389ed1461ab9	More Gold Coin	Token
0xea674fdde714fd979de3edf0f56aa9716b898ec8	**Ethermine**	**Mining Pool**
0x52bc44d5378309ee2abf1539bf71de1b7d7be3b5	**NanoPool**	**Mining Pool**
0x2a0c0dbecc7e4d658f48e01e3fa353f44050c208	**IDEX**	**Dec. Exchange**
0x3f5ce5fbfe3e9af3971dd833d26ba9b5c936f0be	**Binance**	**Exchange**
0xa7a7899d944fe658c4b0a1803bab2f490bd3849e	**IDEX 2**	**Dec. Exchange**
0xd1ceeeeee83f8bcf3bedad437202b6154e9f5405	Dice2Win	Gambling
0x8e766f57f7d16ca50b4a0b90b88f6468a09b0439	Maximine Coin	Token
0xfbb1b73c4f0bda4f67dca266ce6ef42f520fbb98	**Bittrex**	**Exchange**

Table 7: Top-10 highest degree vertices/contracts in ContractNet

Top-10 contracts in 2018. Highlighted contracts continue being top-10 in 2019.

Contract	Name	Type
0x8e306b005773bee6ba6a6e8972bc79d766cc15c8	MerkleMine	Token Distribution
0xa3c1e324ca1ce40db73ed6026c4a177f099b5770	Bittrex Controller	Exchange Control
0x1f573d6fb3f13d689ff844b4ce37794d79a7ff1c	**Bancor**	**Token**
0xabc1c404424bdf24c19a5cc5ef8f47781d18eb3e	EthermonData	Gaming Contract
0x58b6a8a3302369daec383334672404ee733ab239	LivepeerToken	Token
0x12459c951127e0c374ff9105dda097662a027093	0x Exchange v1	Dec. Exchange
0xf20b9e713a33f61fa38792d2afaf1cd30339126a	BancorNetwork	Token Network
0x06012c8cf97bead5deae237070f9587f8e7a266d	CryptoKitties	Gaming Contract
0x182ebf4c80b28efc45ad992ecbb9f730e31e8c7f	MultiMerkleMine	Token Distribution
0xdf6164efd12678bf6a7d5a1ddf73c831493f6574	Ethermon Battle	Gaming Contract

Top-10 contracts in 2019. Highlighted contracts were also in top-10 of 2018.

Contract	Name	Type
0x06a6a7af298129e3a2ab396c9c06f91d3c54aba8	0xUniverse	Gaming Contract
0xfc30a1a7a650d10b20500bc10b06ff8f4b650ad2	0xUniverse Balance	Gaming Contract
0xf0155486a14539f784739be1c02e93f28eb8e960	No known name	Token Exchange
0x01eacc3ae59ee7fbbc191d63e8e1ccfdac11628c	FairWin	Ponzi Scheme
0x5ec8515d15c758472f3e1a7b9eca3e996e8ba902	UtilFairWin	Ponzi Scheme
0x0777f76d195795268388789343068e4fcd286919	GU : Rare Pack	Gaming Contract
0x448a5065aebb8e423f0896e6c5d525c040f59af3	Maker Contract	DAO Contract
0x6ebeaf8e8e946f0716e6533a6f2cefc83f60e8ab	GU : GODS	Gaming Contract
0x1f573d6fb3f13d689ff844b4ce37794d79a7ff1c	**Bancor**	**Token**
0x89d24a6b4ccb1b6faa2625fe562bdd9a23260359	Sai Stablecoin	Token

4.3 Evolution of High-Degree Vertices

Table 6 lists the top-10 most frequently used accounts (highest degree vertices) in TransactionNet. It is no surprise that the most active vertices are large mining pools (Ethermine), trusted exchanges (Binance) and decentralized exchanges (IDEX), popular token contracts (EOS, Tether), and a contract for gambling (Dice2Win). It is interesting however, to note that most of the top-10 vertices in 2018 continue to be top-10 in 2019 as well. The ones staying in top-10 across the years are mostly exchanges and mining pools, with the exception of SparkPool and F2Pool, which did not slide down too far. Amongst the vertices dropping out of the top-10 list of 2018, EOSToken is the most interesting one, as it moved to the EOS Chain, *killing* its Ethereum token contract in June 2019[3]. The top vertex in 2019 thus turned out to be the token contract for Tether USD (USDT), a stable digital token pegged at 1 USD.

In contrast, the top-10 contracts (in terms of activity) in case of ContractNet change quite rapidly. Table 7 lists the top-10 most frequently used contracts (highest degree vertices) in ContractNet. It is clear that Bancor, one of the most popular tokens on Ethereum, is the only contract that persists in the top-10 list across the two years. Popularity of remaining contracts, mostly for gaming (CryptoKitties, Ethermon, Gods Unchained), token exchange (Bittrex, 0x) and distribution (MerkleMine), change quite fast over time. The most interesting contracts in the lists are FairWin and UtilFairWin, pertaining to one of the fastest-growing Ponzi schemes on Ethereum. In late September 2019, the contract was drained of USD 10M worth of ether, resulting in one of the biggest-ever scams.

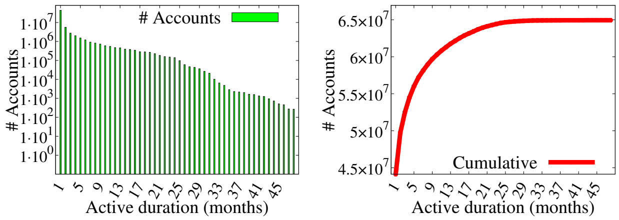
The difference in the rate of change in popularity of top vertices in TransactionNet and ContractNet prompted us to measure the average lifespan of accounts and contracts on these networks.

4.4 Average Activity Period of Vertices

Figures 6 and 7 present the distributions of *active period* for vertices in TransactionNet and ContractNet, respectively. While the vertices measured in Figure 6 are only smart contracts, the vertices measured in Figure 7 include both user accounts and contracts. In case of contracts, we only consider 'activity' as transactions/messages involving the contract account (to or from), and not the initial transaction for contract creation. The X-axis in case of both the figures refers to the amount of *active period* from 1 month to 48 months, and the Y-axis shows the number of accounts (vertices) that remain active for each period. We define the *active period* of a vertex as the duration from its first transaction activity to the last transaction activity between Jan 2016 and Dec 2019.

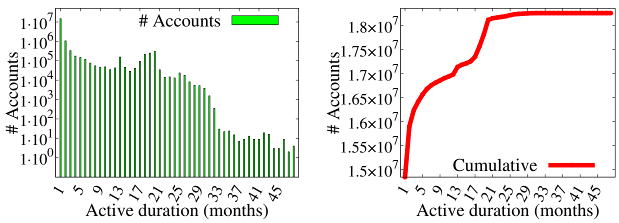
We observe that in general, 80% of smart contract accounts (in ContractNet) have the active period of 1 month, and 91% of smart

[3]Owner invoked `SELF DESTRUCT` on Jun-08-2019 06:13:43 PM +UTC



(a) TransactionNet account active duration (b) TransactionNet account active duration CDF

Figure 6: User accounts' active period (including user to null transaction for smart contract creation)



(a) ContractNet accounts active duration (b) ContractNet account active duration CDF

Figure 7: Smart contracts' active period (active period for transaction only, not including contract creation)

contract accounts have less than or equal to 6 months of active period (Figure 7). However, the same measurements on TransactionNet shows longer active periods per vertex (Figure 6). Thus, it is clear that user accounts are active much longer than smart contracts on Ethereum. Still, our measurements show that 88% of Ethereum accounts have an active period of no more than 6 months, and up to 68% of accounts are only active within a month. This substantiates our previous observations regarding the highly-active vertices (Tables 6 and 7), which remain popular for a longer time in case of accounts in TransactionNet, but not for ContractNet.

5 TEMPORAL EVOLUTION OF NETWORK PROPERTIES

After analyzing the evolution of Ethereum networks, we investigate their network properties from temporal graphs perspective. We study global network properties such as reciprocity, assortativity, connected components, core decomposition, and global clustering coefficient. By analyzing the temporal changes of these properties, we not only understand how the network is connected, but also realize how these connections change over time. The temporal study can reveal any *anomaly* occurred in a specific time duration, that is, whether the property value during a time period is larger/smaller (out of scope) than the average value of this property in the neighboring time periods. As a result, it is possible to shrink the analysis period and locate a more accurate time duration.

5.1 Definitions of Global Properties

We start with definitions of the following global network properties.

Reciprocity represents the probability of reciprocal relationship [4, 11, 28, 29, 51, 78, 80, 81]. It is defined as the ratio of the number of arcs that points to both direction, to the total number of arcs in the graph. We use simple, directed graph, i.e., we consider at most one, directed arc between a pair of source and target vertices. Reciprocity provides a measure of the simplest feedback process, e.g., the tendency of a vertex to respond to another vertex stimulus in a communication network. It also estimates the error introduced when a directed graph is simplified as an undirected graph.

Assortativity measures the preference of vertices getting attached to other vertices that are similar in some way (e.g., vertex degree). Following [57], we employ the following equation that computes the degree assortativity ρ of an observed network.

$$\rho = \frac{|E|^{-1}\sum_i j_i k_i - [|E|^{-1}\sum_i \frac{j_i+k_i}{2}]^2}{|E|^{-1}\sum_i (j_i^2+k_i^2) - [|E|^{-1}\sum_i \frac{j_i+k_i}{2}]^2}$$

j_i, k_i are the degrees of the vertices at the ends of the i-th arc, with $i \in [1, |E|]$, $|E|$ is total number of arcs. Assortativity, ρ, lies in the range: $-1 \leq \rho \leq 1$. A network is assortative (i.e., ρ tends to 1) when high-degree vertices are, on average, linked to other vertices with high degree, and low-degree vertices are, on average, linked to other vertices with low degree. A network is disassortative (i.e., ρ tends to -1) when, on average, high-degree vertices are linked to vertices with lower degree, and vice versa. Assortativity is measured on simple, undirected graph.

Connected components. A connected component is a maximal subgraph in which every pair of vertices is connected. In a directed network, the component is called "weakly connected" if replacing the directed arcs with undirected arcs lead to a connected component. Finding connected components is useful in network clustering, summarization, community detection, and entity resolution.

Core decomposition. The k*-core* of a (simple, undirected) graph is a maximal subgraph in which every vertex is connected to at least k other vertices within that subgraph. The set of all k-cores of a graph, for each k, forms its *core decomposition* [65]. Core decomposition can be computed in linear time by iteratively removing the smallest-degree vertex and setting its core number as its degree at the time of removal [9, 16, 49]. Core decomposition is related to many definitions of a dense subgraph, and it can be used to speed-up or approximate their computation [5, 14, 23]. Specifically, having larger core number for vertices in the innermost core indicates higher density of the innermost core.

Global clustering coefficient. The number of triangles of vertex v is defined as $\Delta(v) = |\{\{u, w\} \in E : \{v, u\} \in E \cap \{v, w\} \in E\}|$. Here, E denotes the set of arcs in the network G. A *triple* γ at a vertex v is a path of length two for which v is the center vertex. The number of triples of vertex v, having degree $d(v)$, is then defined as $\gamma(v) = \binom{d(v)}{2}$. The *local clustering coefficient* of a vertex is the likelihood that its neighbours are also linked. The computation of this score involves triangle counting. The *global clustering coefficient* $C(G)$ is the normalized sum of those local clustering coefficients. Formally, $C(G) = \sum_{v \in V} \frac{\Delta(v)}{\gamma(v)}$. Here, V denotes the set of vertices in G. Triangle count and clustering coefficient are used as features for classifying a website as spam/ non-spam; to find community structure of a social network [22, 70].

5.2 Finding Appropriate Time Granularity

What is the right "time granularity" for temporal analysis? We address this question empirically in this section. In particular, we consider a good time granularity as the shortest time duration by which we can detect an anomaly (i.e., a property value during a time period is larger/smaller than the average value of this property in the neighboring time periods). Moreover, an anomaly detected at

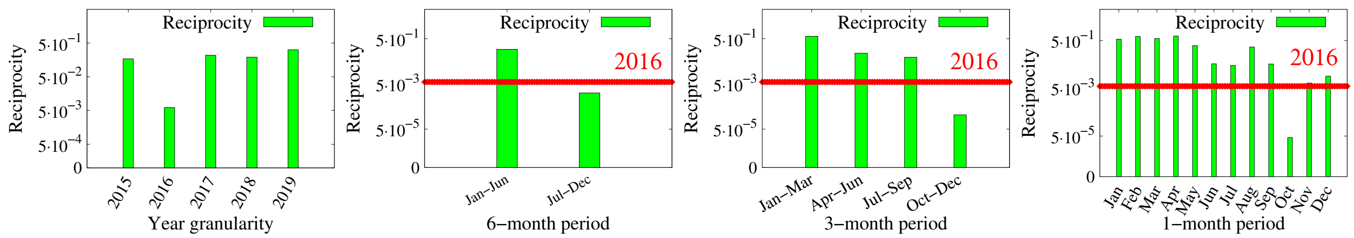


Figure 8: Time granularity analysis for reciprocity; ContractNet 2016

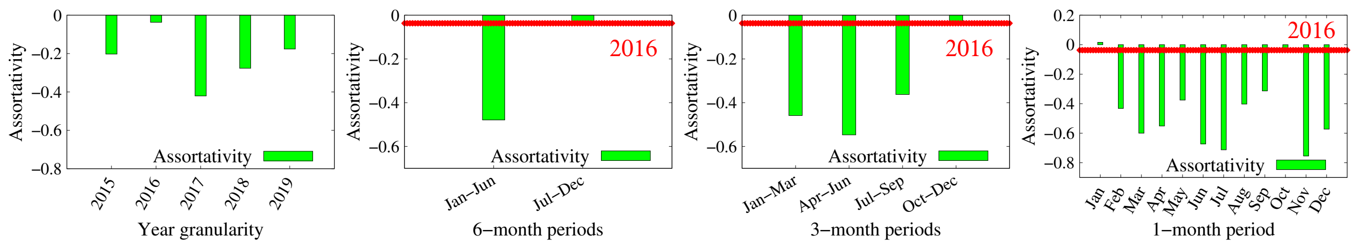


Figure 9: Time granularity analysis for assortativity; ContractNet 2016

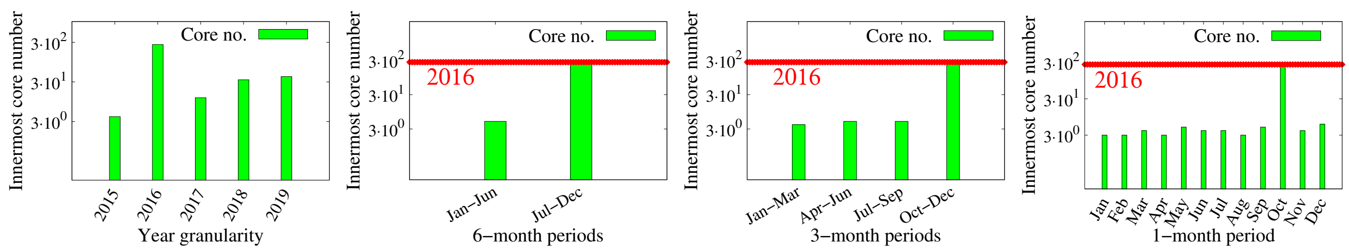


Figure 10: Time granularity analysis for core number in the innermost core; ContractNet 2016

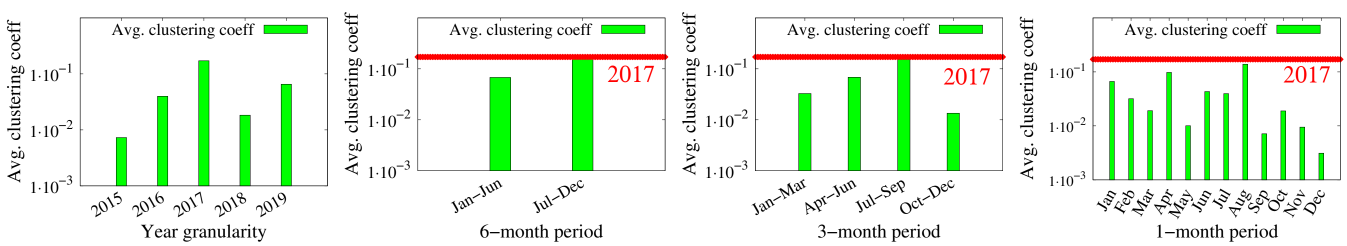
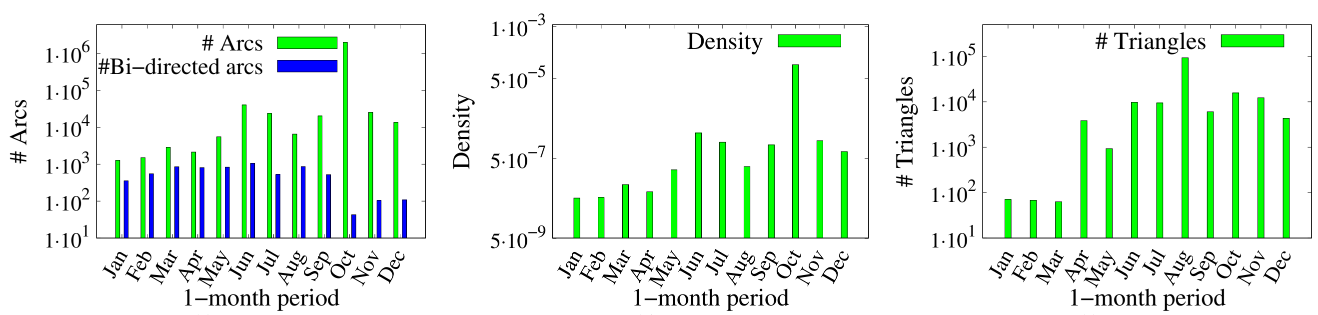


Figure 11: Time granularity analysis for average clustering coefficient (computed over the largest weakly connected component); ContractNet 2017



(a) 2016 (b) 2016 (c) 2017

Figure 12: Statistics on # bi-directional arcs, density, and # triangles for individual months in ContractNet (2016, 2017)

a shorter time granularity is significant if the impact of the anomaly is shown even at a higher granularity covering that shorter duration. We, therefore, introduce a systematic, drill-down approach to find the right granularity. We employ four possible time granularities: annual, half-yearly, quarterly, and monthly, for analyzing the properties of Ethereum network. First, when a property value in the annual granularity data is found to be out of scope than the average value of this property in the neighboring time periods, we divide the anomalous year-level data into two six-month periods data and conduct the analysis. If one of the value is found to be still much smaller (or larger) than that of the other period, we can further reduce the granularity to three-month periods or even one-month periods. In this way, we can empirically identify the right granularity, which is the shortest time duration by which we can detect the anomaly. We present our experimental results on ContractNet.

Empirical Findings. We observe anomalies in ContractNet 2016, based on annual data and with three global network properties: reciprocity, assortativity, and the core number for vertices in the innermost core. In Figure 8(a), the reciprocity in year 2016 is two orders of magnitude smaller than the average of other years. In Figure 9(a), the assortativity in year 2016 is one order of magnitude higher than its average over other years [4]. As shown in Figure 10(a), once again the core number for the innermost core in 2016 is two orders of magnitude higher than the average of that over other years. Based on the abnormal values observed in annual granularity data, we drill-down to smaller time scale, which is 6-months. The results are shown in Figures 8(b), 9(b), and 10(b). The horizontal red line indicates the annual result in 2016 as a reference. The difference in property values between the first and second half of 2016 is prominent in these figures. In the next step, we further split year 2016's data into 3-months basis (Figures 8(c), 9(c), 10(c)). In the quarterly granularity data, we can identify a specific 3-months period where the property value is larger (or smaller) than the average of other quarters in the same year. If we further split the data into monthly basis, we can locate the specific month (October 2016) when the anomaly occurred. These results indicate that the *monthly data* probably has the most suitable time granularity for anomaly detection and for our subsequent temporal analysis.

From ContractNet 2017, we identify that the average clustering coefficient is abnormal compared to the other years. For efficiency reason, we compute the average clustering coefficient over the largest weakly connected component. Figure 11(a) reveals the higher value of this property. Further drilling-down, we detect that the anomaly happened in the third quarter (Jul-Sep) of 2017, and in particular during August 2017.

Correlation of anomalies with real incidents. In this section, we correlate the aforementioned anomalies with *real incidents* that occurred around those periods in Ethereum blockchain, and further affirm them by conducting more temporal analyses over the monthly granularity data (Figure 12).

[4]Notice that Y-axis is negative in Figure 9, since ContractNet is disassortative indicating that there exist generic smart contracts that are used by many other smart contracts. For instance, smart contracts for "decentralized exchanges" perform cryptocurrency exchanges in a decentralized manner, and tend to have many other smart contracts using their services. Thus, the disparity in the degree of such smart contracts and the other smart contracts transacting with it creates higher disassortativity of ContractNet.

The first prominence of Ethereum in the media was in October 2016, such as in Wall Street Journal and Reuters. There were plenty of positive news; Vitalik Buterin (creator of Ethereum) was placed on Fortune's 40 under 40, Wall Street Journal reported on the International Blockchain Week, IBM said that Banks were adopting blockchain "dramatically faster" than expected, rumour spread that Microsoft and Bank of America were working together on a private Ethereum chain project[5]. As a result, a lot of Tokens were deployed on the network, which increased the number of one-directional arcs to the Token Contracts. This is evident in Figure 12(a), where we notice a lot more one-directional arcs in October 2016 compared to other months in the same year. Analogously, the overall activity in the network increased in October 2016. Since plenty of contracts and tokens were deployed on Ethereum, a lot more interactions and transactions happened among all kinds of accounts, including high-degree with high-degree and low-degree with low-degree vertices. As a result, the assortativity also increased in October 2016. Moreover, the density of ContractNet increased in this period (Figure 12(b)) due to a higher number of interacting contracts, such as Tokens, Token Distributions, Decentralized Exchanges and Prediction Markets, where a number of contracts started interacting with one another. Higher density resulted in a larger core number for the innermost core in October 2016.

At a monthly level, in August 2017, we find 93334 triangles in ContractNet, out of which 77013 triangles has two vertices in common – the Bittrex Controller[6] and DefaultSweeper[7] contracts. These two contracts were created in August 2017 by Bittrex cryptocurrency exchange, one of the most popular exchanges at that time. We observe that the third vertex in almost all the triangles are User Wallets exchanging tokens through the Controller and Sweeper. Thus, the triangles involving Bittrex proliferated in August 2017, resulting in an increase in the global clustering coefficient during that month (Figure 12(c)).

6 DETECTION AND SURVIVAL PREDICTION OF CONTRACTNET COMMUNITIES

Our aforementioned analyses were based on temporal properties looking at *historical* Ethereum blockchain network. To better understand how these graph properties help predict the future of the network, we next formulate and address the third key question of our study: *Can we forecast the 'continuation' (survival) of network communities in succeeding months leveraging on the relevant graph properties (features) and machine learning models*?

A community, with respect to graphs, can be defined as a subset of vertices that are densely connected to each other and loosely connected to the vertices outside the community. In Ethereum blockchain network, small communities can be formed through (a) frequent transactions and token transfers between user accounts and contract accounts, and (b) via create, call, kill, and hard fork actions between smart contracts. We first identify interesting communities in Ethereum ContractNet using an efficient community detection algorithm (§6.1) and then study what settings and graph

[5]https://weekinethereumnews.com/

[6]https://etherscan.io/address/0xa3c1e324ca1ce40db73ed6026c4a177f099b5770

[7]https://etherscan.io/address/0xb2233fcec42c588ee71a594d9a25aa695345426c

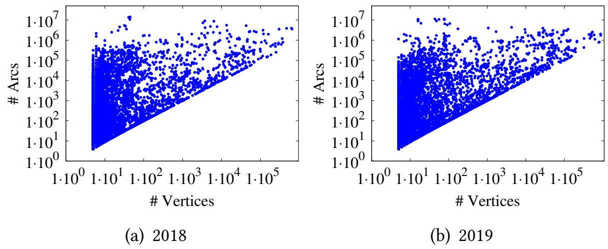


Figure 13: The number of vertices and arcs in communities obtained from ContractNet 2018 and 2019

properties are appropriate for the prediction of survival of these communities (§6.2).

6.1 Community Detection

We consider a number of state-of-the-art community detection methods [76], including Multilevel [10], Arc-betweenness [30], and Walktrap [62]. Their algorithms are implemented in Python "igraph" package. We find that Multilevel scales well over our large-scale datasets and also produces good-quality communities [76], therefore we present our results with this method. In particular, Multilevel is a *greedy* approach that first assigns a different community to each vertex, then a vertex is greedily moved to the community of one of its neighbours with which it achieves the highest positive contribution to modularity. The above step is repeated for all vertices until no further improvement can be achieved. The computational complexity of Multilevel is $O(N \log N)$, where N is the number of vertices in the network.

Figures 13(a) and 13(b) show the number of vertices and arcs in each (non-overlapping) community obtained using Multilevel algorithm over ContractNet 2018 and 2019 networks. We consider multi-arc, undirected versions of these graphs by retaining multiple interactions between each pair of smart contracts (vertices). In the scatter plot, the X-axis shows the number of vertices present in communities in descending order and the Y-axis reports the number of arcs in the corresponding community.

We observe that *the size of the communities follows power-law: a few large communities followed by a long-tail of remaining small communities.* In ContractNet 2019, only 0.7% communities have both # vertices and # arcs > 10^5, whereas we find about 42% communities with # vertices < 10. In general, # arcs is linear to # vertices for larger communities. Smaller communities can be much denser, we identify 7.5% communities having # vertices < 10, however # arcs > 10^3. We note that such dense communities are generally related to tokens, as in the case with CoTrader (COT)[8] and Power Candy (POC)[9], where the other vertices in the communities are token related contracts created by the same user accounts, or generic coin distribution contracts like MerkleProofAirdrop. Such communities with extremely high internal activity is quite common with tokens.

6.2 Community Continuation Prediction

Due to fast changing pace in Ethereum network, we focus on short-term predictions. We detect communities from three consecutive months' data and predict the survivability of these communities in the next (i.e., fourth) month. Specifically, we split the annual dataset into 3-month periods based on a sliding window technique. We employ window size of 3 months and slide stride of 1 month, thereby creating nine 3-month blocks from the annual data (Figures 14, 15).

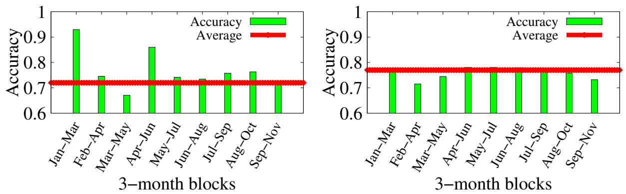
The next step is to extract communities from each 3-month blocks dataset using Multilevel algorithm. As a result, we obtain total 7 915 communities from ContractNet 2018 and 6 986 communities from ContractNet 2019. Multilevel algorithm is also applied to detect communities in the next month's data following each 3-month block. Since communities having too small number of vertices are not interesting for our prediction task, we consider those communities with a minimum of five vertices extracted from 3-month blocks and a minimum of three vertices for those detected from 1-month data.

Our final step in data preparation is to match the communities in a 3-month block to those in the following 1-month block, and subsequently label the communities in 3-month blocks. A community in a 3-month block is labelled as class 1 ("surviving" class) if we find a match of this community in the following 1-month block. A community C_1 in a 3-month block and another community C_2 in the following 1-month block are matched when C_2 has at least half of the vertices that are present in C_1. On the other hand, a community in a 3-month block is labelled as class 0 ("not surviving" class) if we do not find a match of this community in the following 1-month block. Since we have more communities in class 0 than that in class 1, balancing class sizes is necessary to avoid the training biased towards class 0. Therefore, for each run we uniformly at random downsample class 0. After that, 80% of dataset is used for training and 20% of dataset is used for testing. We employ 20-fold cross validation, the testing accuracy is an average of these 20 runs.

For each community (which is a subgraph consisting of vertices and arcs), we consider the following thirteen network properties [44]: #vertices, #arcs, density, #triangle, global clustering coefficient, reciprocity, assortativity, transitivity, #articulation points, adhesion, cohesion, diameter, and radius. We have discussed #vertices, #arcs, density, #triangle, global clustering coefficient, reciprocity, and assortativity in previous sections (§4-§5). Transitivity is calculated as a ratio of the number of closed triplets over the number of connected triples of vertices in a network. It measures the probability that the adjacent vertices of a vertex are connected. An articulation point or cut vertex is a vertex whose removal (along with all its incident arcs) increases the number of connected components of a graph. Adhesion measures the minimum number of arcs that need to be removed to disconnect the graph. Cohesion refers to the minimum number of vertices that must be deleted to disconnect the network. The eccentricity of a vertex v is the maximal shortest path distance between v and any other vertex. The radius of a network is defined as the minimum eccentricity across all vertices, and the diameter is the maximum eccentricity across all vertices. We notice that all these features are numeric, and except reciprocity, adhesion, # arcs, and assortativity, the rest are computed on simple, undirected version of the community graph. For reciprocity, we additionally consider whether an arc exists in both directions. For adhesion, # arcs, and assortativity, we consider multi-arc, undirected graphs, since they retain more information than simple, undirected graphs.

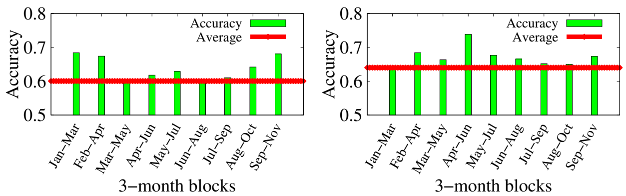
Prediction results. We employ both random forest (RF) and logistic regression (LR) for training. RF and LR are widely adopted due to their simplicity, robustness, and ability to produce good-quality

[8] https://etherscan.io/address/0x5c872500c00565505f3624ab435c222e558e9ff8
[9] https://etherscan.io/address/0xc9c4d9ec2b44b241361707679d3db0876ac10ca6



(a) Prediction accuracy for ContractNet 2018 (b) Prediction accuracy for ContractNet 2019

Figure 14: Random forest prediction accuracy



(a) Prediction accuracy for ContractNet 2018 (b) Prediction accuracy for ContractNet 2019

Figure 15: Logistic regression prediction accuracy

Table 8: Top-3 important features based on ContractNet 2019 community survivability prediction using random forest

Feature	Importance	Effect
Density	0.2028	Higher-density communities survive
#arcs	0.1848	More-arc communities survive
Assortativity	0.1485	Higher-assortativity communities survive

results. They are suitable for our prediction task since we intend to obtain both prediction result and feature importance.

Figures 14 and 15 show the prediction accuracy. Each histogram demonstrates the accuracy for the corresponding 3-month period. The horizontal red line represents the accuracy when we combine all 3-months datasets as one dataset. Using RF the average accuracy is 0.72 for ContractNet 2018 and 0.77 for ContractNet 2019. With LR the average accuracy is 0.60 for ContractNet 2018 and 0.64 for ContractNet 2019. Combining all 3-months data together, although the accuracy does not improve, we find that the accuracy for each run is more stable compared to predicting for each quarterly data individually. Moreover, logistic regression performs worse than random forest in this community prediction task. This is because random forest is able to train if the variables are not linearly separable. The ensemble of multiple decision trees also reduces overfitting. In contrast to RF, logistic regression is a simple model with a linear decision boundary. Our accuracy results prove this point.

In Table 8, we list the top-3 important features extracted from RF prediction using ContractNet 2019 dataset. Density feature achieves the highest importance score, followed by # arcs, and assortativity. Further analyses in these features reveal the fact that higher values of density, # arcs, and assortativity tend to be class 1. Both density and # arcs positively influence the graph size. Recall that assortativity measures the preference of vertices getting attached to other vertices that are similar according to vertex degree. The higher is the assortativity, more arcs happen between vertices with higher degree which imply that the network formed by them is denser. Therefore, it can be concluded that if the community is dense, having more arcs, or with higher assortativity, it is likely that the community will survive in the next month.

7 DISCUSSION AND CONCLUSIONS

In this work we investigated the evolutionary, temporal, and predictive aspects of four Ethereum blockchain interaction networks (TraceNet, ContractNet, TransactionNet, and TokenNet), and conducted a comprehensive empirical evaluation on the graphs.

We find that Ethereum interaction networks are growing at a fast speed, and the account information is updated at a fast pace. However, all the graphs get sparser and mature over time, thereby more old accounts remain active, and more than half of new vertices participate in interactions with old vertices. The networks follow the preferential attachment growth model. The user accounts remain active much longer than smart contracts on Ethereum. As a consequence, the high-degree user accounts from the previous year continue to be high-degree in the following year. In contrast, high-degree contract accounts change rapidly over successive years.

By analyzing the temporal changes of global network properties, e.g., reciprocity, assortativity, core decomposition, and clustering coefficient, we realize how their connections change over time, and reveal anomalies occurred in a specific time duration. We correlate these anomalies with external 'real-life' aspects of the network, e.g., popularity, exchanges, and systematically drill-down to the appropriate time granularity for our analyses.

Last but not least, we detect meaningful communities in Ethereum blockchain networks. We also forecast the continuation (survival) of network communities in succeeding months leveraging on the relevant graph properties (as features) and machine learning models, achieving up to 77% correct predictions for continuation.

To facilitate research, we open source our blockchain network dataset [79]. In future, we shall conduct temporal analyses over other public blockchain platforms to mine interesting time-evolving and predictive phenomena across the Web of blockchain networks.

ACKNOWLEDGMENT

The research of Arijit Khan is supported by MOE Tier-1 and Tier-2 grants RG117/19 and MOE2019-T2-2-042.

REFERENCES

[1] 2013. Elliptic. https://www.elliptic.co/.
[2] Y.-Y. Ahn, S. Han, H. Kwak, S. Moon, and H. Jeong. 2007. Analysis of Topological Characteristics of Huge Online Social Networking Services. In *WWW*.
[3] C. G. Akcora, Y. R. Gel, and M. Kantarcioglu. 2017. Blockchain: A Graph Primer. *CoRR* abs/1708.08749 (2017).
[4] L. Akoglu, P. O. S. V. d. Melo, and C. Faloutsos. 2012. Quantifying Reciprocity in Large Weighted Communication Networks. In *PAKDD*.
[5] R. Andersen and K. Chellapilla. 2009. Finding Dense Subgraphs with Size Bounds. In *Algorithms and Models for the Web-Graph, 6th International Workshop, WAW*.
[6] L. Backstrom, D. Huttenlocher, J. Kleinberg, and X. Lan. 2006. Group Formation in Large Social Networks: Membership, Growth, and Evolution. In *KDD*.
[7] Q. Bai, C. Zhang, Y. Xu, X. Chen, and X. Wang. 2020. Poster: Evolution of Ethereum: A Temporal Graph Perspective. In *IFIP Networking Conference*.
[8] A.-L. Barabasi and R. Albert. 1999. Emergence of Scaling in Random Networks. *Science* 286, 5439 (1999), 509–512.
[9] V. Batagelj and M. Zaveršnik. 2011. Fast Algorithms for Determining (Generalized) Core Groups in Social Networks. *Adv. Data Anal. Classif.* 5, 2 (2011), 129–145.
[10] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. 2008. Fast Unfolding of Communities in Large Networks. *Journal of Statistical Mechanics: Theory and Experiment* 2008, 10 (2008), 10008.
[11] M. Boguna and M. A. Serrano. 2005. Generalized Percolation in Random Directed Networks. *Phys. Rev. E* 72 (2005), 016106. Issue 1.
[12] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener. 2000. Graph Structure in the Web. In *WWW*.
[13] V. Buterin. 2013. Ethereum Whitepaper. https://ethereum.org/en/whitepaper/.
[14] M. Charikar. 2000. Greedy Approximation Algorithms for Finding Dense Components in a Graph. In *Approximation Algorithms for Combinatorial Optimization, Third International Workshop, APPROX*.

[15] T. Chen, Y. Zhu, Z. Li, J. Chen, X. Li, X. Luo, X. Lin, and X. Zhang. 2018. Understanding Ethereum via Graph Analysis. In *INFOCOM*. IEEE, 1484–1492.
[16] J. Cheng, Y. Ke, S. Chu, and M. T. Özsu. 2011. Efficient Core Decomposition in Massive Networks. In *ICDE*.
[17] N. Christin. 2013. Traveling the Silk Road: A Measurement Analysis of a Large Anonymous Online Marketplace. In *WWW*.
[18] Coinmarketcap. 2020. About Ethereum. https://coinmarketcap.com/currencies/ethereum/.
[19] A. Day and E. Medvedev. 2018. Ethereum in BigQuery: a Public Dataset for Smart Contract Analytics. https://cloud.google.com/blog/products/data-analytics/ethereum-bigquery-public-dataset-smart-contract-analytics. Accessed: 2020-10-10.
[20] C. Decker and R. Wattenhofer. 2013. Information Propagation in the Bitcoin Network. In *P2P*.
[21] J. A. D. Donet, C. Pérez-Solà, and J. Herrera-Joancomartí. 2014. The Bitcoin P2P Network. In *Financial Cryptography Workshops*.
[22] J.-P. Eckmann and E. Moses. 2002. Curvature of Co-links Uncovers Hidden Thematic Layers in the World Wide Web. *Proceedings of the National Academy of Sciences* 99, 9 (2002), 5825–5829.
[23] D. Eppstein, M. Löffler, and D. Strash. 2010. Listing All Maximal Cliques in Sparse Graphs in Near-Optimal Time. In *Algorithms and Computation*.
[24] L. Ermann, K. M. Frahm, and D. L. Shepelyansky. 2018. Google Matrix of Bitcoin Network. *The European Physical Journal B* (2018), 91–127.
[25] M. Faloutsos, P. Faloutsos, and C. Faloutsos. 1999. On Power-law Relationships of the Internet Topology. *SIGCOMM Comput. Commun. Rev.* 29, 4 (1999), 251–262.
[26] S. Ferretti and G. D'Angelo. 2019. On the Ethereum Blockchain Structure: A Complex Networks Theory Perspective. *Concurrency and Computation: Practice and Experience* (2019), e5493.
[27] D. Fetterly, M. S. Manasse, M. Najork, and J. L. Wiener. 2004. A Large-scale Study of the Evolution of Web Pages. *Softw. Pract. Exp.* 34, 2 (2004), 213–237.
[28] D. Garlaschelli and M. I. Loffredo. 2005. Structure and Evolution of the World Trade Network. *Physica A: Statistical Mechanics and its Applications* 355, 1 (2005), 138–144.
[29] D. Garlaschelli, F. Ruzzenenti, and R. Basosi. 2010. Complex Networks and Symmetry I: A Review. *Symmetry* 2 (2010), 1683–1709. Issue 3.
[30] M. Girvan and M. E. J. Newman. 2002. Community Structure in Social and Biological Networks. *Proceedings of the National Academy of Sciences* 99, 12 (2002), 7821–7826.
[31] A. Greaves and B. Au. 2015. *Using the Bitcoin Transaction Graph to Predict the Price of Bitcoin*. Technical Report. Stanford.
[32] B. Haslhofer, R. Karl, and E. Filtz. 2016. O Bitcoin Where Art Thou? Insight into Large-Scale Transaction Graphs. In *SEMANTiCS*.
[33] H. Jeong, Z. Néda, and A. L. Barabási. 2003. Measuring Preferential Attachment in Evolving Networks. *Europhysics Letters (EPL)* 61, 4 (2003), 567–572.
[34] S. R. Kairam, D. J. Wang, and J. Leskovec. 2012. The Life and Death of Online Groups: Predicting Group Growth and Longevity. In *WSDM*.
[35] G. Kappos, H. Yousaf, M. Maller, and S. Meiklejohn. 2018. An Empirical Analysis of Anonymity in Zcash. In *USENIX Security Symposium*.
[36] L. Kiffer, D. Levin, and A. Mislove. 2018. Analyzing Ethereum's Contract Topology. In *Internet Measurement Conference*.
[37] J. M. Kleinberg. 2000. Navigation in a Small World. *Nature* 406, 845 (2000).
[38] Jon M. Kleinberg. 2000. The Small-world Phenomenon: An Algorithmic Perspective. In *STOC*.
[39] D. Kondor, I. Csabai, J. Szüle, M. Pósfai, and G. Vattay. 2014. Inferring the Interplay between Network Structure and Market Effects in Bitcoin. *New Journal of Physics* 16 (2014).
[40] G. Kossinets and D. J. Watts. 2006. Empirical Analysis of an Evolving Social Network. *Science* 311, 5757 (2006), 88–90.
[41] O. Kraaijeveld and J. D. Smedt. 2020. The Predictive Power of Public Twitter Sentiment for Forecasting Cryptocurrency Prices. *Journal of International Financial Markets, Institutions and Money* 65, C (2020), S104244312030072X.
[42] R. Kumar, J. Novak, P. Raghavan, and A. Tomkins. 2005. On the Bursty Evolution of Blogspace. *World Wide Web* 8, 2 (2005), 159–178.
[43] R. Kumar, J. Novak, and A. Tomkins. 2006. Structure and Evolution of Online Social Networks. In *KDD*.
[44] X. T. Lee, A. Khan, S. S. Gupta, Y. H. Ong, and X. Liu. 2020. Measurements, Analyses, and Insights on the Entire Ethereum Blockchain Network. In *WWW*.
[45] J. Leskovec and E. Horvitz. 2008. Planetary-scale Views on a Large Instant-messaging Network. In *WWW*.
[46] J. Leskovec, J. M. Kleinberg, and C. Faloutsos. 2005. Graphs over Time: Densification Laws, Shrinking Diameters and Possible Explanations. In *KDD*.
[47] C. Lo, J. Cheng, and J. Leskovec. 2017. Understanding Online Collection Growth Over Time: A Case Study of Pinterest. In *WWW*.
[48] C. Ma, X. Kong, Q. Lan, and Z. Zhou. 2019. The Privacy Protection Mechanism of Hyperledger Fabric and its Application in Supply Chain Finance. *Cybersecur.* 2, 1 (2019), 5.
[49] D. W. Matula and L. L. Beck. 1983. Smallest-last Ordering and Clustering and Graph Coloring Algorithms. *J. ACM* 30, 3 (1983), 417–427.
[50] S. Meiklejohn, M. Pomarole, G. Jordan, K. Levchenko, D. McCoy, G. M. Voelker, and S. Savage. 2016. A Fistful of Bitcoins: Characterizing Payments among Men with no Names. *Commun. ACM* 59, 4 (2016), 86–93.
[51] L. A. Meyers, M. E. J. Newman, and B. Pourbohloul. 2006. Predicting Epidemics on Directed Contact Networks. *Journal of Theoretical Biology* 240, 3 (2006), 400 – 418.
[52] A. Mislove, H. S. Koppula, K. P. Gummadi, P. Druschel, and B. Bhattacharjee. 2008. Growth of the Flickr Social Network. In *WOSN*.
[53] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee. 2007. Measurement and Analysis of Online Social Networks. In *SIGCOMM IMC*.
[54] M. Möser, K. Soska, E. Heilman, K. Lee, H. Heffan, S. Srivastava, K. Hogan, J. Hennessey, A. Miller, A. Narayanan, and N. Christin. 2018. An Empirical Analysis of Traceability in the Monero Blockchain. *PoPETs* 2018, 3 (2018), 143–163.
[55] S. Nakamoto. 2008. Bitcoin: A Peer-to-Peer Electronic Cash System. https://bitcoin.org/bitcoin.pdf.
[56] M. E. J. Newman. 2001. Clustering and Preferential Attachment in Growing Networks. *Phys. Rev. E* 64 (2001), 025102. Issue 2.
[57] M. E. J. Newman. 2002. Assortative Mixing in Networks. *Physical Review Letters* 89, 20 (2002), 208701.
[58] GLOBE NEWSWIRE. 2020. CoinLinked Debuts as First-Ever Blockchain-Based Social Network and Crypto-Commerce Platform. https://www.globenewswire.com/news-release/2020/05/07/2029366/0/en/CoinLinked-Debuts-as-First-Ever-Blockchain-Based-Social-Network-and-Crypto-Commerce-Platform.html.
[59] A. Ntoulas, J. Cho, and C. Olston. 2004. What's New on the Web?: The Evolution of the Web from a Search Engine Perspective. In *WWW*.
[60] F. E. Oggier, A. Datta, and S. Phetsouvanh. 2020. An Ego Network Analysis of Sextortionists. *Soc. Netw. Anal. Min.* 10, 1 (2020), 44.
[61] M. Peltomäki and M. Alava. 2006. Correlations in Bipartite Collaboration Networks. *Journal of Statistical Mechanics: Theory and Experiment* 2006, 01 (2006), P01010–P01010.
[62] P. Pons and M. Latapy. 2005. Computing Communities in Large Networks Using Random Walks. In *Computer and Information Sciences*.
[63] M. H. Ribeiro, J. Blackburn, B. Bradlyn, E. D. Cristofaro, G. Stringhini, S. Long, S. Greenberg, and S. Zannettou. 2021. The Evolution of the Manosphere Across the Web. In *ICWSM*.
[64] D. Ron and A. Shamir. 2013. Quantitative Analysis of the Full Bitcoin Transaction Graph. In *Financial Cryptography and Data Security*.
[65] S. B. Seidman. 1983. Network Structure and Minimum Degree. *Social Networks* 5, 3 (1983), 269–287.
[66] G. Siganos, S. L. Tauro, and M. Faloutsos. 2006. Jellyfish: A Conceptual Model for the AS Internet Topology. *Journal of Communications and Networks* 8, 3 (2006), 339–350.
[67] S. Somin, G. Gordon, and Y. Altshuler. 2018. Network Analysis of ERC20 Tokens Trading on Ethereum Blockchain. In *Complex Systems*.
[68] S. Somin, G. Gordon, and Y. Altshuler. 2018. Social Signals in the Ethereum Trading Network. *CoRR* abs/1805.12097 (2018). arXiv:1805.12097 http://arxiv.org/abs/1805.12097
[69] M. Spagnuolo, F. Maggi, and S. Zanero. 2014. BitIodine: Extracting Intelligence from the Bitcoin Network. In *Financial Cryptography and Data Security*.
[70] J. Ugander, B. Karrer, L. Backstrom, and C. Marlow. 2011. The Anatomy of the Facebook Social Graph. *CoRR* abs/1111.4503 (2011).
[71] F. Victor and B. K. Lüders. 2019. Measuring Ethereum-based ERC20 Token Networks. In *Financial Cryptography and Data Security*.
[72] A.-D. Vo, Q.-P. Nguyen, and C.-Y. Ock. 2019. Sentiment Analysis of News for Effective Cryptocurrency Price Prediction. *International Journal of Knowledge Engineering* 5, 2 (2019), 47–52.
[73] D. J. Watts and S. H. Strogatz. 1998. Collective Dynamics of 'Small-World' Networks. *Nature* 393 (1998), 440–442.
[74] C. Wilson, B. Boe, A. Sala, K. P. N. Puttaswamy, and B. Y. Zhao. 2009. User Interactions in Social Networks and Their Implications. In *EuroSys*.
[75] S. Y. Yang and J. Kim. 2015. Bitcoin Market Return and Volatility Forecasting Using Transaction Network Flow Properties. In *SSCI*.
[76] Z. Yang, R. Algesheimer, and C. J. Tessone. 2016. A Comparative Analysis of Community Detection Algorithms on Artificial Networks. *Scientific Reports* 6, 30750 (2016).
[77] H. Yousaf, G. Kappos, and S. Meiklejohn. 2019. Tracing Transactions Across Cryptocurrency Ledgers. In *USENIX Security Symposium*.
[78] G. Zamora-Lopez, V. Zlatic, C. Zhou, H. Stefancic, and J. Kurths. 2008. Reciprocity of Networks with Degree Correlations and Arbitrary Degree Sequences. *Phys. Rev. E* 77 (2008), 016106. Issue 1.
[79] L. Zhao, S. Sen Gupta, A. Khan, and R. Luo. 2021. Temporal Analysis of the Entire Ethereum Blockchain Network (Code and Dataset). https://github.com/LinZhao89/Ethereum-analysis.
[80] V. Zlatic and H. Stefancic. 2009. Influence of Reciprocal Edges on Degree Distribution and Degree Correlations. *Phys. Rev. E* 80 (2009), 016117. Issue 1.
[81] V. Zlatic and H. Stefancic. 2011. Model of Wikipedia Growth Based on Information Exchange via Reciprocal Arcs. *EPL (Europhysics Letters)* 93, 5 (2011), 58005.