

From Innovations to Prospects: What Is Hidden Behind Cryptocurrencies?

Ang Jia
jiaang@stu.xjtu.edu.cn
Xi'an Jiaotong University
School of Cyber Science and
Engineering

Di Cui
cuidi@stu.xjtu.edu.cn
Xi'an Jiaotong University
School of Cyber Science and
Engineering

Kai Ye
kaiye@xjtu.edu.cn
Xi'an Jiaotong University
School of Electronics and Information
Engineering

Ming Fan
mingfan@mail.xjtu.edu.cn
Xi'an Jiaotong University
School of Cyber Science and
Engineering

Wenyong Wei
waving@stu.xjtu.edu.cn
Xi'an Jiaotong University
School of Cyber Science and
Engineering

Ting Liu
tingliu@mail.xjtu.edu.cn
Xi'an Jiaotong University
School of Cyber Science and
Engineering

Xi Xu
xx19960325@stu.xjtu.edu.cn
Xi'an Jiaotong University
School of Cyber Science and
Engineering

Zijiang Yang
zijiang.yang@wmich.edu
Western Michigan University
Department of Computer Science

ABSTRACT

The great influence of Bitcoin has promoted the rapid development of blockchain-based digital currencies, especially the altcoins, since 2013. However, most altcoins share similar source codes, resulting in concerns about code innovations. In this paper, an empirical study on existing altcoins is carried out to offer a thorough understanding of various aspects associated with altcoin innovations. Firstly, we construct the dataset of altcoins, including source code repository, GitHub fork relation, and market capitalization (cap). Then, we analyze the altcoin innovations from the perspective of source code similarities. The results demonstrate that more than 85% of altcoin repositories present high code similarities. Next, a temporal clustering algorithm is proposed to mine the inheritance relationship among various altcoins. The family pedigrees of altcoin are constructed, in which the altcoin presents similar evolution features as biology, such as power-law in family size, variety in family evolution, etc. Finally, we investigate the correlation between code innovations and market capitalization. Although we fail to predict the price of altcoins based on their code similarities, the results show that altcoins with higher innovations reflect better market prospects.

CCS CONCEPTS

• **Software and its engineering** → *Software evolution*.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

MSR '20, October 5–6, 2020, Seoul, Republic of Korea

© 2020 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-1-4503-7517-7/20/05...\$15.00

<https://doi.org/10.1145/3379597.3387439>

KEYWORDS

Altcoins, Innovations, Relation, Prospects

ACM Reference Format:

Ang Jia, Ming Fan, Xi Xu, Di Cui, Wenyong Wei, Zijiang Yang, Kai Ye, and Ting Liu. 2020. From Innovations to Prospects: What Is Hidden Behind Cryptocurrencies?. In *17th International Conference on Mining Software Repositories (MSR '20)*, October 5–6, 2020, Seoul, Republic of Korea. ACM, New York, NY, USA, 12 pages. <https://doi.org/10.1145/3379597.3387439>

1 INTRODUCTION

The great influence of Bitcoin¹ has promoted the rapid development of blockchain-based digital currencies (cryptocurrency) since 2013. There are three critical features for existing cryptocurrencies: 1) Large variety, in August 2019, the number of available cryptocurrencies is over 2300 and it is still growing quickly[3]. 2) Fast delivery, the number of ICOs² (initial coin offering) newly increased per month is more than 60 since 2018. 3) Huge price change, a drastic fluctuation occurred in Bitcoin when it reached a peak of more than 20000\$ on Dec 17, 2017, and fell to 3830.04\$ on Dec 30 2018[5]. These three features have raised concerns about the quality of the cryptocurrencies and the real value of them.

The cryptocurrency world has experienced a complicated evolution since its birth. Bitcoin is the first cryptocurrency and now accounts for nearly 70% of the market[7]. The later cryptocurrencies can be divided into two categories: Alternative Cryptocurrency Coins (Altcoins) and Tokens[4]. Altcoins refer to the coins that are alternatives to Bitcoin. Usually, they have their own chains. However, tokens serve the blockchain and they are often built on top of another blockchain such as the Ethereum blockchain. In this study, we only focus on altcoins. According to the source of

¹<https://bitcoin.org/>

²https://en.wikipedia.org/wiki/Initial_coin_offering

altcoins, they can be divided into two groups: the variant of Bitcoin and the cryptocurrency with self-developed blockchain[4]. The majority of altcoins are variants (forks) of Bitcoin, built using Bitcoin's open-sourced, original protocol with changes to its underlying codes, therefore conceiving an entirely new coin with a different set of features. Other altcoins are not derived from Bitcoin's open-source protocol. Rather, they have created their own blockchain and protocol that supports their native currency.

To determine whether a cryptocurrency should be invested, one key point is its code innovation. According to a report[1] released in September 2018, there exist great similarities between altcoins, i.e., *"the code underpinning more than 90% of the projects shared a similarity score of at least 80%"*. The severity of code clone raises concerns about the innovation of altcoins. However, investors only focus on market trends and ignore the implementation details of the codes. Moreover, existing studies that analyze the cryptocurrency only focus on the technical issues, and they would ignore the code innovation. In addition, what is amazing for technologists is that the value of a technology product seems to have nothing to do with the technology's most direct vehicle, i.e., the code. Therefore, it is non-trivial to get an understanding of the altcoin innovations, so well as their relation with the market prospects.

In this work, we aim to offer a thorough understanding of the various aspects associated with innovations of altcoins. Specifically, we first construct our dataset by crawling three different types of data from GitHub and CoinMarketCap[3], including source code repositories, GitHub fork relations of repositories and market capitalization (market cap) of altcoins. To conduct a thorough analysis, we empirically study the altcoin codes by answering three research questions listed below.

RQ1: *To what extent the codes are shared by altcoins?*

Innovations. Many altcoins describe themselves with identical and innovative features and the ability to serve as better alternatives to Bitcoin. This ability actually lies in innovations of their implementation, i.e., source codes. However, it remains unknown what extent they share codes with other existing cryptocurrencies. Therefore, at the first step, we would like to offer an overview of the duplication in altcoin codes implemented in C++ (the largest part) to answer RQ1.

RQ2: *What is the relation between altcoins?*

Relations. Software developments mostly do not start from scratch. Altcoins may also stem from one or many templates. Therefore, many altcoins may share similar source codes. However, the similarities between them cannot reveal the process of their inheritance. Although existing clustering algorithms offer clustering analysis of similar individuals, they ignore the sequence of input instances and cannot be applied in our work directly. In the second step, we aim to conduct the familial analysis of altcoins to further explore their inner relation by answering RQ2.

RQ3: *What is the correlation between prospects and innovations?*

Prospects. Same with rules of nature that similar organisms will compete for the same resources, altcoins implemented with the same codes are more likely to suffer the same competitions. Therefore, innovations may play an important role in affecting the prospects of altcoins. However, the complicated environment in the cryptocurrency world makes it hard to understand it directly. At the

last step, we are willing to offer a deep insight into the correlation between prospects and innovations by answering RQ3.

Contributions: Our main contributions are list below:

- (i) To the best of our knowledge, our empirical study for the first time investigated the innovations of altcoins from the view of source code, and we proposed a novel temporal clustering algorithm to mine the inner relations between altcoins by constructing family pedigrees.
- (ii) We investigate the correlations between the altcoin innovations and their market capitalization from the view of code similarity. The results show that the altcoins with high innovations reflect better market prospects and, the cryptocurrency world is moving toward diversity.
- (iii) We release the dataset of altcoin including the source code repositories, GitHub fork relations, and market caps. The dataset can be accessed from GitHub³.

Overview of the paper: Section 2 presents an motivating example. Section 3 presents the construction of our dataset and its statistics. Section 4 introduces the study design and results by answering the three research questions. After the discussing of the related work and threats to validity in Section 5 and Section 6, we conclude our paper in Section 7.

2 MOTIVATING EXAMPLE

This section motivates our research using a real-world example. StrikeBitClub⁴ and HollyWoodCoin⁵ are two altcoins, where StrikeBitClub is a hybrid script PoW + PoS based one⁶ and HollyWoodCoin is a PoS-based one⁷. Table 1 shows the collected information of this two cryptocurrencies. They are both released in December 2017, they both suffer a huge price loss from December 2017 to September 2018.

Table 1: StrikeBitClub and HollyWoodCoin

currency	StrikeBitClub	HollyWoodCoin
release time	2017/12/9	2017/12/14
price(release)	0.570882\$	17.62\$
price(2018/4)	0.008867\$	2.52\$
status(2019/8)	dead	dead

The source code repositories of these two altcoins and their GitHub fork information are crawled on 28 March 2018. They do not indicate GitHub fork relation between their repositories or to other repositories. However, using textual code clone detection, the two repositories share about 97% of codes.

The 'coincidences' in their codes, their release times, and their price loss inspire us to carry out studies to further mine the hidden phenomenons. Altcoins are running based on similar techniques, and these two altcoins share highly similar source codes, which inspire us to explore the innovations of other altcoins and the relation between these altcoins. Companies with the same technology in the

³<https://github.com/island255/Cryptocurrencies-Coins-Sourcecode-download>
<https://github.com/island255/GitHub-Fork-relation>
<https://github.com/island255/CrawlPrice>

⁴<https://coinmarketcap.com/currencies/strikebitclub/>

⁵<https://coinmarketcap.com/currencies/hollywoodcoin/>

⁶<https://github.com/sbcoin/sbcoin-source>

⁷<https://github.com/hollywoodcoin-project/HollyWoodCoin>

same industry will compete for the same resources and these two similar altcoins suffer a decline in the price, which raises our exploration about the relation between code innovations and market prospects in the cryptocurrency world. To illustrate our approaches to solve these questions, we introduce our dataset in section 3, and our study in section 4.

3 DATASET

This section presents details of the dataset. We crawl source code repositories, GitHub fork relations, and market caps of all altcoins from GitHub and CoinMarketCap.com to construct our dataset. The process of dataset construction and statistics of the dataset will be illustrated in the following subsections.

3.1 Dataset Construction

Source Code Repositories: By 4 April 2019, more than 3000 cryptocurrencies have emerged. Bitcoin has about 20,000 commits and more than 200 release versions. We collect all versions of the cryptocurrencies (including Coin and Token) to construct our dataset. Source code repositories of these cryptocurrencies were accessed through the following steps. First, we access the website of each cryptocurrency on CoinMarketCap.com and get their GitHub links. Then we access the GitHub links and identify whether it is a repository or repositories. If it is a repository, we will download this repository; otherwise, we will download the pinned repositories or (if the pinned repositories do not exist) the first three repositories.

Market Caps: We obtain the historical closing prices and the market capitalization of all cryptocurrencies by crawling the website CoinMarketCap.com. We use Node.js to crawl the information and store them in JSON format.

GitHub Fork Information: The distinction between GitHub fork and blockchain fork needs to be explained. A fork in the blockchain is when a change is made to the software of a cryptocurrency to create another version of the blockchain, technically[2], while A fork in GitHub is a copy of a repository. Bitcoin hard forks (blockchain fork) leads to the creation of many altcoins, such as Bitcoin Cash, Bitcoin Gold, etc. When an altcoin is a product of Bitcoin hard forks, there may be GitHub fork relation between their repositories. We would like to collect their GitHub fork relation between all altcoins before analyzing their code similarity relation. We query the GitHub API to obtain information of their repositories. Results are structured in JSON format, and the fork relation can be obtained from its "parent" node.

3.2 Statistics of Dataset

By 4 April 2019, there come up 3338 cryptocurrencies (including coin and token). Among them, 1253 have opened their code links and 1230 cryptocurrencies set GitHub repositories as their links. Using the method in Section 3.1, we successfully access 1698 repositories belonging to 1023 cryptocurrencies (some links are invalid or unable to open). Among them, more than 40% (693 in 1698) have only one release and the repository *bitcoin/bitcoin* of Bitcoin has the largest number of releases as 216 times.

We try to crawl the information (created time, language, and fork) of these 1698 repositories using GitHub API, but only access the information of 989 repositories successfully. Among these

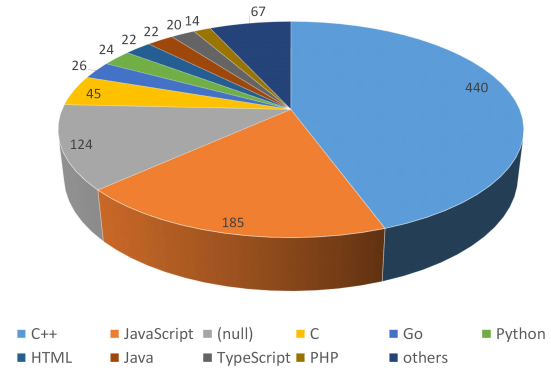


Figure 1: Top 9 language in cryptocurrencies' repositories

989 repositories, 865 repositories are labeled with a specific language. Figure 1 shows the proportion of languages used in these repositories. Among the labeled 865 repositories, 440 repositories are labeled that its primary language is C++, and 185 repositories' main language is JavaScript. Other languages used in more than 20 repositories are C, Go, HTML, Java, Python, and TypeScript.

We also analyze their created times from our obtained statistics. Figure 2 shows the number of repositories written in different languages during different periods. It indicates enthusiasm for creating repositories (including coin projects and token projects) since 2017. While the creating of all repositories arrives at its peak during 2017/11/30 to 2018/6/1, creating of C++ written repositories reaches its peak during 2019/6/1 to 2018/11/30. Moreover, from 2017/11/30 to 2018/6/1, there arise 239 repositories, which means a new repository comes out every 1.5 days.

Figure 3 shows changes in total market capitalization and the number of cryptocurrencies. The market capitalization of cryptocurrency experience a pull-up from 2017/1 to 2018/1 and a plunge after 2018/1. Besides, the number of newly-created repositories indicates a similar trend with a slight delay. Although cryptocurrencies spring up like mushrooms after the rain of market expansion, there may exist some hidden troubles in the course of this explosive development.

4 STUDY DESIGN AND RESULTS

The overview of our study is presented in figure 4. First, we acquire market caps and source code repositories of different cryptocurrencies from CoinMarketCap.com and GitHub. Then, we select repositories written in C++ (the largest part) and then calculate the similarities between them. Next, we propose a novel clustering algorithm combined with temporality to construct families of cryptocurrencies based on code similarities. Finally, we compare the market prospects with their code similarities and relations.

We present the study results by answering three research questions. For each question, we present the motivation behind the research question, the approach, and a discussion of our findings.

4.1 To what extent the codes are shared by altcoins?

Motivation: Altcoins serve for a common purpose—exchange and demonstrate their features in offer, decentralization, immutability,

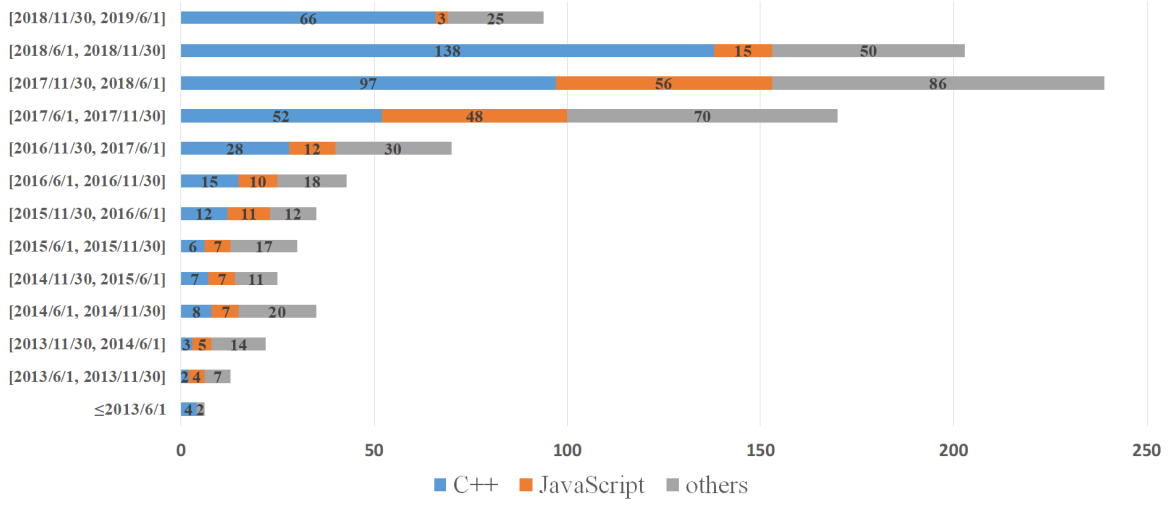


Figure 2: Number of repositories written in different languages during different periods

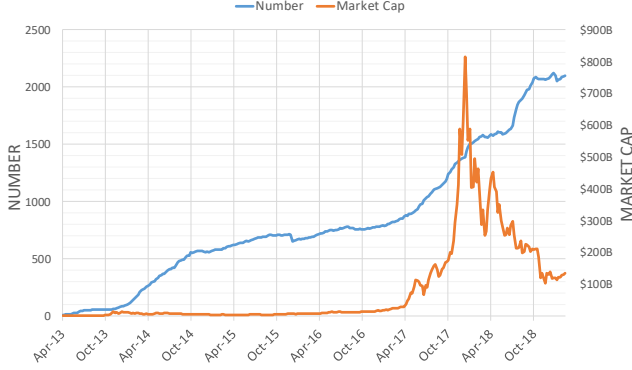


Figure 3: Total market capitalization and number of newly-released cryptocurrencies

anonymity, etc. However, even though an altcoin describes it has many innovative and meaningful creations in its new release, there may be little difference in its source code and others. To verify the true innovations of an altcoin, the analysis of their codes and the similarities with others are needed.

Approach: Our study measures the global situation of cryptocurrencies' innovation from two aspects. First, we obtain their fork relation through the GitHub API. The GitHub API provides end-points for users to consume GitHub data as well as make changes on a user's behalf[6]. We send requests through GitHub API and obtain a thorough description of this repository. With the query results structured in JSON, we are able to access the fork related information of a repository from the 'parent' (directly fork) and 'source' (original source) nodes. We finally crawl the fork information of 1698 repositories belonging to 1023 cryptocurrencies deployed in GitHub.

Second, we implement code clone detection to calculate the similarities between repositories of different cryptocurrencies. It is unnecessary and difficult to calculate the similarities between all repositories of every version belonging to all cryptocurrencies. We

cut two snapshots from our dataset on 28 March 2018 (D1) and 28 September 2018 (D2). In each snapshot, we select repositories of altcoins of their latest version at that time. As comparing repositories written in different languages is out of our scope and the C/C++ written repositories account for the largest (see Figure 1), the repositories whose main language is C/C++ are preserved. Table 2 shows the statistics about these C/C++ written projects of these two snapshots. Time, number of altcoins, and minimum, maximum, and average of LOCs (lines of code) are presented in this table.

Table 2: Statistics for projects written in C++

Dataset	Time	Number	MinLOC	MaxLOC	AvgLOC
D1	Mar	485	15804	392312	60937
D2	Sep	566	13495	2560877	60443

To calculate the similarities between these repositories, we first remove configuration files, readme documents, and files implemented in other languages. Then we treat remaining source code files as text and use RKR-GST (Running-Karp-Rabin Greedy-String-Tiling [26]) algorithm to find the same code snippets in two repositories. This algorithm, first carried out by Wise, uses the GST (greedy string tiling) algorithm, which aims to find similar fragments following the order from long to short and reduce the complexity of string extension matching by KR (Karp Rabin) algorithm. In this process, these similar code snippets are one-to-one mappings in two repositories. Finally, considering that even though a large repository contains codes of a small one, the similarity between them should not be high due to the difference in their size, we use the following formula to calculate the similarity between repository A and B:

$$similarity = \frac{2 \times |A \cap B|}{|A| + |B|} \quad (1)$$

We regard the codes of A and B as two sets. $|A \cap B|$ represents the size of matched code snippets between A and B, $|A|$ and $|B|$

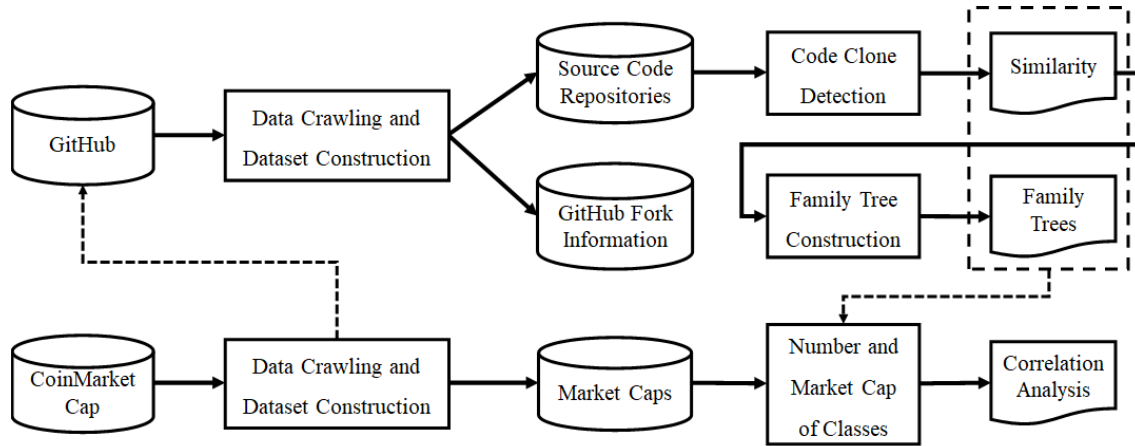


Figure 4: Overview of our approach

respectively represent the size of A and B. The size can be measured by the length of character.

For cryptocurrencies with one repository, we set similarities of this one repository with others as similarities of this altcoin. For cryptocurrencies with more than one repository, we merge the similarities of repositories by selecting the highest similarities.

Findings: About 15% of repositories shows fork relation with other repositories. We crawl the fork information of 1698 repositories belonging to 1023 cryptocurrencies deployed in GitHub and successfully get the information of 989 repositories belonging to 983 cryptocurrencies. In this 989 repositories, 146 repositories have a fork relation with other repositories. Among them, *bitcoin/bitcoin*⁸ is the most forked repository forked by 12 other cryptocurrencies. The second is *PIVX-Project/PIVX*⁹ which is forked by 7 cryptocurrencies. Besides, *bitcoin/bitcoin* is also the most sourced repository. It is the source of 34 repositories, indicating that apart from 12 repositories which directly fork this repository, the other 22 repositories forking other repositories also come from *bitcoin/bitcoin*. From this point, the implementation of Bitcoin does have a great influence on other cryptocurrencies.

More than 85% of the repositories share 80% of codes with the other existing repositories. Figure 5 shows the distribution of maximum similarity of a repository with other existing repositories appearing before it. The similarities of these two experiments have a similar distribution that similarities between 90% and 100% account for the largest part, and the second are similarities between 80% and 90%.

33.92% cryptocurrencies have a similarity of 80% with more than 50 cryptocurrencies. According to the results of D2, five cryptocurrencies have a similarity of more than 90% with Bitcoin and 12 of more than 80%. However, the number of cryptocurrencies whose similarities with PIVX more than 90% and 80% is 11 and 48. Besides, there are 55 cryptocurrencies of which each has a similarity of more than 90% with more than 50 cryptocurrencies. Moreover, there are 192 cryptocurrencies (33.92%) of which each also has a similarity of more than 80% with more than 50 cryptocurrencies.

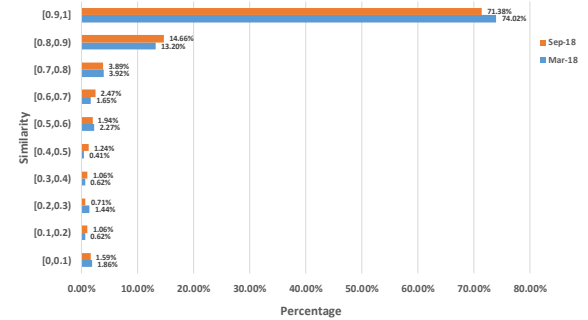


Figure 5: Distribution of maximum similarities with existed repositories

While few repositories show fork relation with others, more than 85% of the repositories share 80% of codes with the other earlier-created repositories.

4.2 What is the relation between altcoins?

Motivation: Conducting a thorough analysis of the relationship between different cryptocurrencies is critical. However, while the statistics show a high degree of duplication in codes of cryptocurrencies, it's hard to get a deeper understanding about their extended connections such as the vulnerability caused by their shared code snippets, the technology trends distributed in different groups and the inheritance of their codes during software development. In our study, we propose a temporal clustering algorithm to construct family pedigrees of cryptocurrencies.

Approach: To gain a deep insight into the relationship between cryptocurrencies, clustering them by similarities is a shortcut to acquiring an analyzable representation. However, existing clustering methods are insufficient to reflect the development of cryptocurrencies as they do not take temporality into account. We propose a novel clustering algorithm to present their similarities in chronological order.

⁸<https://github.com/bitcoin/bitcoin>

⁹<https://github.com/PIVX-Project/PIVX/>

Before the description of the algorithm, we introduce two rules to simplify the relationship between cryptocurrencies:

Rule 1. When the similarity between two cryptocurrencies is high, it is more likely to find that later-released cryptocurrency follows early-released cryptocurrency, which means the similarity is more like a directed relation.

Rule 2. When a cryptocurrency has high similarities with more than one cryptocurrency released early than it, it is more likely to follow the cryptocurrency with maximum similarity.

Based on the aforementioned rules, we only pay attention to pairs of altcoins which is most similar. Besides, we distinguish the different relation of two cryptocurrencies with different occurrence time intervals. With long-time intervals, we consider one cryptocurrency is a follower of the other one. With short time intervals, we consider they are in the same group. Finally, we propose our algorithm, as shown in algorithm 1.

Algorithm 1 The Temporal Clustering Algorithm

Input: altcoin[n], time[n], similarity[n][n]
 {n is the number of altcoins and altcoin[n],time[n] are sorted in chronological order}

Output: Family-Pedigrees

```

1:  $i = 1$ ;
2: while  $i \leq n$  do
3:   put altcoin[i] in Family-Pedigrees;
4:    $MaxSim[i] = \max_{1 \leq x \leq n} similarity[x][i]$ ;
5:   if  $i=1$  or  $MaxSim[i] < \Theta_s$  then
6:     set altcoin[i] as a initial node of a new tree;
7:     continue;
8:   end if
9:    $j = \arg \max_{1 \leq j \leq n} similarity[j][i]$ 
10:  if node[j] exists then
11:    if  $|t(node[i]) - t(node[j])| > \Theta_t$  then
12:      altcoin[i].fathercoin = altcoin[j];
13:    else
14:      altcoin[i].brothercoin = altcoin[j];
15:    end if
16:  end if
17: end while

```

Algorithm 1 illustrates a way to construct family pedigrees for cryptocurrencies. At first, we sort cryptocurrencies by the chronological order of their occurrence time for convenience. Then, in line 2-8 in Algorithm 1, we define that a new tree is created as a coin is selected without high similarities with early-released cryptocurrencies. Next, in line 9-16, we classify the relation between cryptocurrencies with maximum similarity based on the interval of their occurrence time. Finally, a forest composed of many family pedigrees will form.

To have a graphic impression about the relations between hundreds of cryptocurrencies, we propose a way to present them in a tree diagram. The father coin will be the father node of this coin, while its brother coin will be in the same horizontal position. Figure 6 shows part of one family pedigree as our graphic representation of D2 (setting $\Theta_s = 70\%$ and $\Theta_t = 3$ months). A horizontal arrow

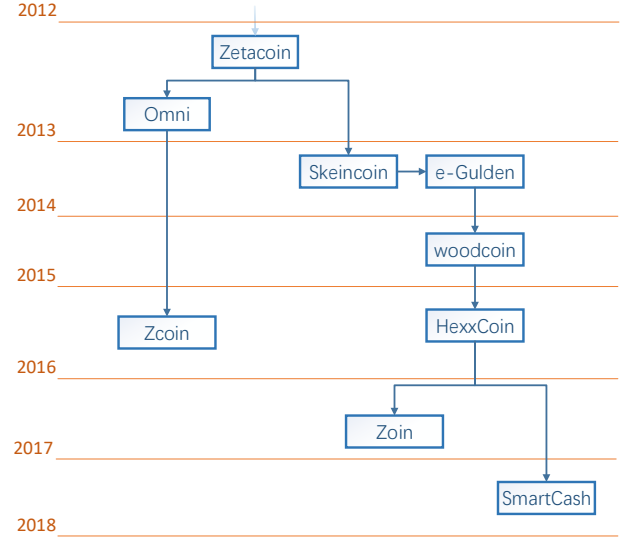


Figure 6: Part of a family pedigree in altcoin pedigrees

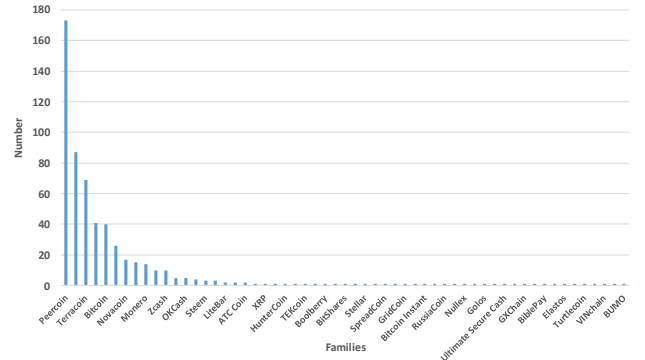


Figure 7: Number of altcoins in different families

represents brotherhood, and an upright arrow indicates a father-son relationship. For example, Zetacoin is the father node of Skeincoin and e-Gulden is the brother node of Skeincoin.

Findings: Most altcoins congregate in few groups. We use our proposed algorithm to construct family pedigrees in D2 by setting $\Theta_s = 70\%$. Figure 7 shows the distribution of the numbers in different families. The distribution has a big head and a long tail, which indicates an unbalanced distribution of altcoins in families. The phenomenon of this power-law distribution in altcoins is actually similar to the species richness in clades of organisms[60]. To be accurate, 72.4% (410 in 566) of altcoins congregate in 8.8% (5 in 57) of family pedigrees. It obeys the 80/20 principle, which indicates a more tight aggregation in the more technically-close groups.

Early-released altcoins are more likely to form large families. Figure 8 shows the time of the first-released altcoin in families. Every node represents a family pedigree. The horizontal axis shows the number of altcoins in this family, while the vertical axis shows the release time of the first altcoin in this family. There are eight

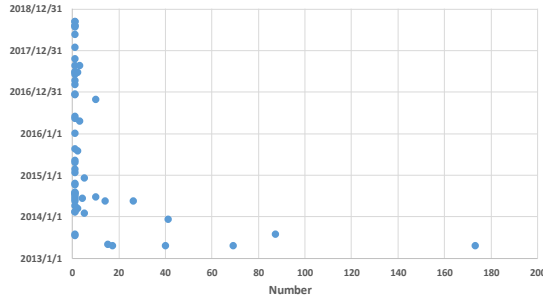


Figure 8: Time of the first altcoin in families trees

families with their first altcoins released before 2014, and 5 of them have more than 40 altcoins. There are 49 families of which altcoins are released after 2014, and 48 in them are families of less than 20 altcoins. There indicates a trend of families with early-released altcoins to own more members likely. We attribute this phenomenon to two reasons. One is that families with early-released altcoins will grow larger by our proposed algorithm. Algorithm 1 illustrates a way to construct family pedigrees from early-released altcoins. As a consequence, early-released altcoins will stay at a higher level and thus have more descendants. Another is that early-released altcoins have more reputations than later-released altcoins. The first several altcoins serve as a prototype for later ones, and Bitcoin, in particular, has a massive impact on the formation of other altcoins.

Most altcoins congregate in few families and early-released altcoins are more likely to form large families.

4.3 What is the correlation between prospects and innovations?

Motivation: Charles Darwin has observed that due to the limited resources, there is a struggle for existence among individuals - often with only a fraction of offspring surviving through each generation to reproduce successfully. In the cryptocurrency world, there is also a rat race between different kinds of cryptocurrencies. Many factors may influence the prospect of a cryptocurrency, including but not limited to overall market trends, government policies, competitors with the same functionality, and its technology implementations. Especially, mining the relation between their technology implementations and their prospects are useful to see the importance of technology in the cryptocurrency market.

In our previous study, we try to predict their price change from their codes (details in section 6.1). However, It is the same as common knowledge that the correlation between code similarity and market cap is weak. In this study, we carry out the following two questions to help further analyze their relation:

1. What is the correlation between prospects of cryptocurrencies and code similarities?
2. What is the correlation between prospects of cryptocurrencies and family pedigrees?

Approach: To solve question 1, we select D1 and D2 used in section 4.1 as our research case. According to figure 3, D1 and D2 are in a shrinking market with more competitive pressures which helps us to mine their correlation easier. To mine the correlation

between prospects of cryptocurrencies and code similarities, we firstly define three classes of cryptocurrencies, which are cryptocurrencies without code link, cryptocurrencies with high similarities, and cryptocurrencies with low similarities. Then, to represents market prospects of these three classes, we define the rate of number change (NCR) and market cap change (MCCR) as follows:

$$NCR(T) = \frac{Number(T_0 + T)}{Number(T_0)} \quad (2)$$

$$MCCR(T) = \frac{MarketCap(T_0 + T)}{MarketCap(T_0)} \quad (3)$$

Prospects are calculated by analyzing their NCRs and MCCRs setting T_0 as half a year and a whole year. For D1, we calculate the change of their numbers and market caps from 28 March 2018 to 28 September 2018 and from 28 March 2018 to 28 March 2019. Similarly, we set the periods of D2 as from 28 September 2018 to 28 March 2019 and 28 September 2018 to 28 September 2019.

From the prospects of different classes, we can discover some differences between cryptocurrencies with different similarities and cryptocurrencies without code links.

To solve question 2, we use family pedigrees (FP) of D1 and D2 generated by the algorithm 1 in section 4.2. First, we match the two sets of family pedigrees using algorithm 2. As the maximum similarities (edges of family pedigrees) may change during this period, we only focus on the attributes of their nodes to establish their matches. In algorithm 2, we match all pairs of FP1 and FP2 as long as there exist nodes in trees of FP1 appearing in trees of FP2. The consequence is that one tree in FP2 may match more than one tree in FP1. To avoid this embarrassment, we then select the match with most nodes in common. We search trees in FP1 (earlier) to match trees in FP2 (later) based on our assumption that one tree in the earlier family pedigree is likely to break up into two trees in the later family pedigree while two trees in the earlier one are not likely to unite into one tree in the later one. Then, same as the aforementioned approach, we calculate number change and prospects of family pedigrees from 28 March 2018 to 28 September 2018. From the statistics we processed, we present our research findings.

Findings: Altcoins with code links are more likely to have better prospects than altcoins without. Table 3 shows changes in the number and market cap of different classes of cryptocurrencies. We classify "with code" and "without code" through their websites on CoinMarketCap.com. When there is a "Source Code" link in its main page, we classify it as "with code". Otherwise, it will be classified as "without code". To be more accurate, we make our best effort to verify the altcoins without code link through searching on Google and GitHub. NCR/MCCR rate (HA/WA) in Table 3 represents NCR and MCCR of this class in half a year or a whole year. From the statistics, we can see that the number change rates of altcoins with code links are all higher than those without, which indicates that altcoins with code implementation as their technical support are more likely to survive in a shrinking market. Besides, in most cases, the market cap rates of altcoins with code are higher than those without. All suggest that altcoins with code links are more likely to have better prospects.

Table 3: Number change and prospects of cryptocurrencies with different similarities

Snapshots	Classes	Number/market cap at occurrence Time	Number/market cap in half A Year(HA)	Number/market cap in one Year(WA)	NCR/MCCR rate (HA)	NCR/MCCR rate (WA)
D1	with code	644/2.26E+11\$	514/1.99E+11\$	453/1.20E+11\$	0.80/0.88	0.70/0.53
	without code	223/5.02E+08\$	143/4.63E+08\$	115/2.66E+08\$	0.64/0.92	0.52/0.53
	≥ 80%	408/1.04E+10\$	313/6.34E+09\$	268/5.24E+09\$	0.77/0.61	0.66/0.50
	< 80%	60/1.80E+11\$	51/1.67E+11\$	46/1.03E+11\$	0.85/0.93	0.77/0.57
D2	with code	676/2.06E+11\$	596/1.26E+11\$	566/1.96E+11\$	0.88/0.61	0.84/0.95
	without code	151/1.15E+09\$	121/5.89E+08\$	108/7.47E+08\$	0.80/0.51	0.72/0.65
	≥ 80%	462/9.19E+09\$	403/7.05E+09\$	382/6.87E+09\$	0.87/0.77	0.83/0.75
	< 80%	74/1.76E+11\$	67/1.08E+11\$	64/1.80E+11\$	0.91/0.62	0.86/1.02

Algorithm 2 Family Pedigrees Matching Algorithm**Input:** Family-Padigrees1 (FP1), Family-Padigrees2 (FP2)

{FP1 is earlier than FP2}

Output: Matches[]

```

1: FP2 = Pruning(FP1, FP2);
   {Remove the nodes not in FP1 from FP2}
2: while trees in FP2 remain unvisited do
3:   Match = [ ];
4:   tree2 = GetTree(FP2);
   {take a family pedigree unvisited from FP2}
5:   SetVisited(tree2);
6:   while trees in FP1 remain unvisited do
7:     tree1 = GetTree(FP1);
8:     SetVisited(tree1);
9:     if tree1 ∩ tree2 ≠ ∅ then
10:      Match ← {tree1, tree2};
11:    end if
12:  end while
13:  Matches ← arg max{tree1, tree2} ∈ match ||tree1 ∩ tree2|| ;
14:  SetUnvisited(FP2);
15: end while

```

Altcoins with low similarities are more likely to have better prospects than altcoins with high similarities. Table 3 presents an example by using 80% as the threshold to distinguish altcoins with high similarities and altcoins with low similarities. The similarity of an altcoin is defined as its maximum similarity with earlier-released altcoins. The changes of number and market cap are also calculated in the same way with classes "code" and "without code". From the statistics, it may arise doubt that altcoins with low similarities have fewer members but own larger market caps. This is because Bitcoin dominates the market and is the first cryptocurrency. In all cases, altcoins with low similarities have higher NCRs than altcoins with high similarities. Also, in most cases, altcoins with low similarities have higher MCCR than altcoins with high similarities. All suggest that altcoins with low similarities are more likely to have better prospects.

Small families are likely to have better prospects than big families and the cryptocurrency world are moving toward diversity. Figure 9 present graphical representation of families in D1 and D2. Table 4 shows details of the families comprised of more

than ten altcoins. Families in the right (D2) indicate matches with families in the left (D1). Numbers and market caps are listed for each family, and the last column shows the number of altcoins shared by two families. Firstly, we found that the evolution of altcoins is similar to the evolution of species, such as one species may emerge, become extinct, or split into two species. For example, family *Bitcoin* and *Terracoin* in D2 all come from family *Bitcoin* in D1, and they share 20 and 38 altcoins with family *Bitcoin* in D1, separately. Besides, families of different size are experiencing different evolution. In the 11 families except for the second, families whose members are less than 40 experience an increase in their numbers, while families whose members are more than 40 experience a decrease both in their numbers and market caps. Although most families experience a loss in their market cap, families comprised of more altcoins seem to suffer more. In this way, small families are likely to have better prospects than big families. Due to the reduction of big families and the growth of small families, all families are moving to medium size. It is the same with biological evolution as this ecosystem is moving toward diversity.

Altcoins with code links, low similarity, and in small families are preferred, and the cryptocurrency world is moving toward diversity.

5 RELATED WORK

We discuss related works in two areas: analyses of cryptocurrencies and code clone detection.

Analyses of Cryptocurrencies. There are many researches that measure properties of cryptocurrencies from several aspects including but not limited to peer-to-peer networks [14, 29, 30, 47], data encryption [17, 40], attack [11, 25, 52] and scam identifications [13, 15]. In this paper, we only focus on papers most related to our own. Reibel et al.[39] carried out research to identify the extent of innovation in the cryptocurrencies landscape using the open-source repositories. They focused on the code reuse of Bitcoin by using hash to calculate the similarities between repositories and analyze their relations through graphs. And they illustrated 8 clusters with directory structure similar relations and file similar relations. Azouvi et al. [12] conducted a study to analyze the centralization in the existing governance structures of Bitcoin and Ethereum. They collected discussions of the contributors and their contributions to their GitHub repositories and found that Ethereum appears more centralized than Bitcoin in terms of improvement proposals, but is

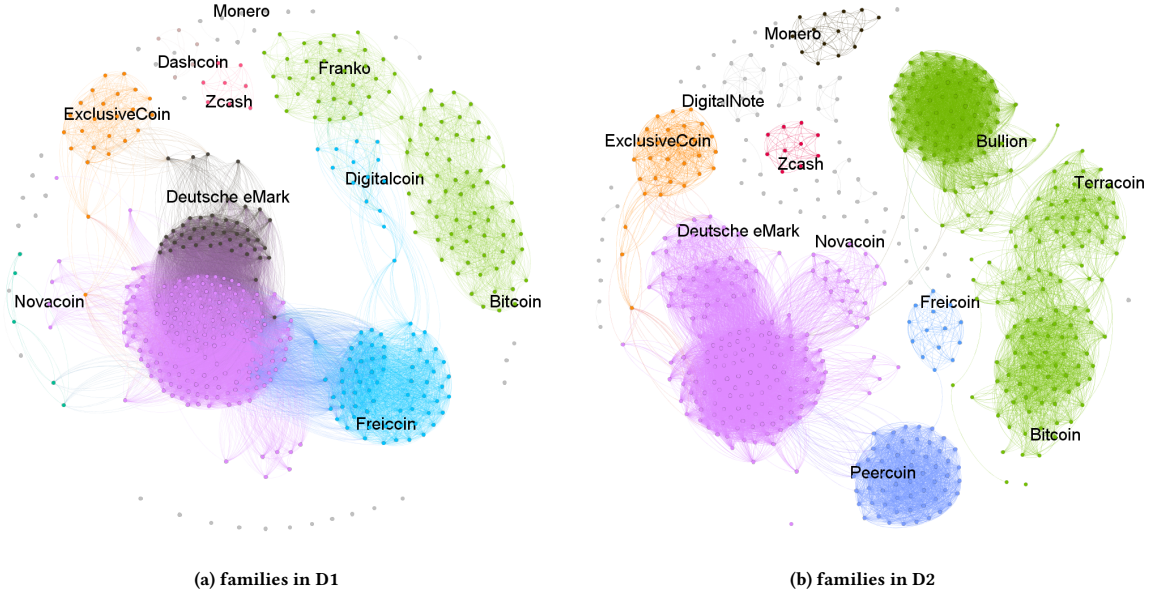
Table 4: Number change and prospects of cryptocurrencies in different families

28 March 2018 (D1)			28 September 2018 (D2)			$ D1 \cap D2 $
family*	number	market cap	family*	number	market cap	
Bitcoin**	63	1.27E+11	Bitcoin	40	1.19E+11	20
			Terracoin**	69	2.85E+09	38
Dashcoin**	5	3.09E+06	DigitalNote**	10	2.85E+07	2
Deutsche eMark	45	1.44E+08	Deutsche eMark	41	3.72E+07	30
Digitalcoin**	14	1.28E+08	Freicon**	15	2.85E+09	10
ExclusiveCoin	22	2.76E+08	ExclusiveCoin	26	1.21E+08	16
Franko**	33	6.77E+08	Bullion**	87	2.54E+08	25
Freicon**	238	1.46E+09	Peercoin**	173	5.54E+08	147
Monero	3	2.79E+09	Monero	14	2.11E+09	5
Novacoin	17	7.74E+07	Novacoin	17	2.45E+07	14
Zcash	8	7.42E+08	Zcash	10	8.06E+08	8
single***	29	5.81E+10	single***	38	3.43E+10	18

* We use the first coin of a family to represent that family. For example, *Bitcoin* represents a family in which the first released coin is Bitcoin.

** As altcoins evolve from D1 to D2, the first altcoin of a certain family may change as some altcoin may die. Also, the code of a certain altcoin may change, too. So the presentation of a certain family and the member of a family will change according to their survival state and code similarity.

*** "single" represents families comprised of only one altcoin.

**Figure 9: Families in D1 and D2**

more decentralized in terms of the discussion around its codebase. In our work, we further analyze the relationship between innovation and prospects and find many significant findings that can provide suggestions for investors and developers.

Code Clone Detection. Tools and techniques of code clone detection can be classified as six categories [8]: textual approaches [10, 23, 27, 34, 37, 51, 56, 58], lexical approaches [35, 44, 45, 50, 55, 59], tree-based approaches [16, 36, 57], metric-based approaches [38, 48, 49], semantic approaches [18, 19, 22, 24, 43, 54] and hybrid approaches [9, 20, 21, 31, 32, 41, 46, 53]. In particular, we introduce several works closely related to our work. Nakamura et al.

[33] introduce an approach to detect inter-language clones for web applications. As web applications contain source code written in co-dependent multiple programming languages, they merged co-dependent code clones detected from each programming language to further detect code clones. Yu et al. [58] propose a multi granularity code clone detection method based on Java bytecode to detect code clones at both method level and block level. Apart from the similarity calculation between instruction sequences, they also calculate the similarity between the method call sequences to improve its effectiveness in detecting semantic code clones. Liu et al. [28] propose VEDFECT, a system to detect vulnerable code clone

based on fingerprint. The fingerprint is constructed by applying hash function to code blocks in the diff. Then the vulnerable code clone is detected by matching the preprocessed code blocks in the target project with the fingerprint. In our work, we further carry out analyses based on the code similarities acquired by code clone detection.

6 DISCUSSION AND THREATS

6.1 Discussion

Vulnerability inheritance. Security is one of the most severe issues in cryptocurrencies. In 2018, Tim Ruffing [42] suggests that there was a cryptographic denial-of-spending attack on the original Zerocoin protocol, which is used in several cryptocurrencies including Zcoin, PIVX, SmartCash, Zoin, and HexxCoin. Although these cryptocurrencies handled this vulnerability through their ways, we believe that there may be some connections between their codes. As shown in Figure 6, HexxCoin is the father node of Zoin and SmartCash, and Zcoin is also a member of this family. Although the similarity relation between cryptocurrencies is not equal to the inheritance of vulnerabilities, the family relation, which indicates a highly-similar relation, is useful for discovering hidden vulnerability associations to some extent. We did not carry out a strict process to examine the effectiveness of family pedigrees as it is not the focus of this research.

Price prediction. Interests are always what investors chase for in the cryptocurrency world. As we found high similar family pedigrees of altcoins, we wonder whether there is potential to predict their price change from their code similarities. Then we use the similarities of an altcoin with others as its feature, define the range of its relative price change against Bitcoin (< 0.1 , between 0.1 and 1 , and > 1) as its label, and use multiple machine learning models such as decision tree, Naive Bayes, and multi-layer neural network to predict the label of the newly-released altcoins by the model trained based on earlier-released altcoins. However, the accuracy shows a decline year by year, as the accuracy of predicting altcoins released in 2015 based on models trained using altcoins before 2015 is about 60% while the accuracy of 2017 and 2018 is about 40% and 33%. Though it may be partly due to reasons caused by the change of dataset size, it reveals a hidden truth that the technical implementation of altcoins has less and less impact on price changes.

6.2 Threats

Dataset. Our study focuses on altcoins written in C++. Altcoins implemented in other languages will be analyzed in our subsequent works. We select two snapshots on 28 March 2018 and 28 September 2018 to carry out our research. As the cryptocurrency world experienced a loss of market cap from 28 March 2018 to 28 September 2018, it would be more evident to see the difference of market prospects between different kinds of altcoins. But these two snapshots are limited, and we plan to give continuous analyses across several periods.

Code clone detection. We treat source code as text and use RKR-GST algorithm to calculate their similarities. It is simple, so it can't resist some obfuscations. But we hold the opinion that obfuscations are modifications to the source code, and we are supposed

to take that difference into account. And we can get a thorough graphic representation of the same snippets to support the similarity results as the ground truth. When applying it to cryptocurrencies whose codes are written in other languages, it is also adequate for similarity calculation.

Threshold. We propose two thresholds as θ_s and θ_t to impose restrictions on family pedigree construction. In our results, we generate family pedigrees by setting $\Theta_s = 70\%$ and $\Theta_t = 3months$. And we use these results to analyze the relationship between prospects and families further. We also set 80% as the threshold to distinguish cryptocurrencies with high similarities and low similarities. As we set the threshold by a certain value, the results may be a little different from the results obtained by other thresholds. However, the relation between innovations and prospects remains unchanged.

Correlation between prospects and innovations. We carry out a correlation analysis between codes and market caps in Section 4.3 and price prediction in Section 6.1, to investigate whether the code innovation of an altcoin is an essential factor of its market caps. It is the same as common knowledge that the correlation between code similarity and market cap is weak. However, our works show altcoins with high innovation tend to have better prospects, emphasizing the importance of innovation.

7 CONCLUSION

In this paper, an empirical study on existing altcoins is carried out to offer a thorough understanding of various aspects associated with altcoin innovations. Firstly, we construct the dataset of altcoins, including source code repositories, GitHub fork relations, and market capitalization (cap). Then, we analyze the altcoin innovations from the perspective of source code similarities. The results demonstrate that more than 85% of altcoin repositories present high code similarities. Next, a temporal clustering algorithm is proposed to mine the inheritance relationship among various altcoins. The family pedigrees of altcoin are constructed, in which the altcoin presents similar evolution features as biology, such as power-law in family size, variety in family evolution, etc. Finally, we investigate the correlation between code innovations and market capitalization. Although we fail to predict the price of altcoins based on their code similarities, the results show that altcoins with higher innovations reflect better market prospects.

Our results emphasize the importance of code innovation and we suggest that newly-released cryptocurrencies pay more attention to technological innovations. Our work can be expended in several directions, such as the inheritance of code libraries, innovation analysis of tokens, relation between releases of highly-similar altcoins, and evolution of codes in the cryptocurrency world.

ACKNOWLEDGMENT

This work was supported by National Key R&D Program of China (2016YFB1000903), National Natural Science Foundation of China (61902306, 61632015, 61772408, U1766215, 61721002, 61532015, 61833015), Ministry of Education Innovation Research Team (IRT_17R86), China postdoctoral science special foundation (No. 2019TQ0251), and Project of China Knowledge Centre for Engineering Science and Technology. Ting Liu is the corresponding author.

REFERENCES

- [1] 2018. Chinese report finds 9 in 10 altcoins to have stolen 80% of the code. <https://news.bitcoin.com/chinese-report-finds-9-in-10-altcoins-stolen-80-code>. [Online; accessed 2-December-2019].
- [2] 2019. A Full List Of Bitcoin Hard Forks (UPDATED 2019). <https://www.investinblockchain.com/a-full-list-of-bitcoin-hard-forks/>. [Online; accessed 3-September-2019].
- [3] 2019. All Cryptocurrencies | CoinMarketCap. <https://coinmarketcap.com/all/views/all/>. [Online; accessed 19-August-2019].
- [4] 2019. Altcoins vs. Tokens: What's the difference? <https://masterthecrypto.com/differences-between-cryptocurrency-coins-and-tokens/>. [Online; accessed 30-August-2019].
- [5] 2019. Bitcoin price, charts, market cap and other metrics | CoinMarketCap. <https://coinmarketcap.com/currencies/bitcoin/>. [Online; accessed 19-August-2019].
- [6] 2019. github-api-GitHub Topics. <https://github.com/topics/github-api>. [Online; accessed 9-September-2019].
- [7] 2019. Global Charts | CoinMarketCap. <https://coinmarketcap.com/charts/>. [Online; accessed 19-August-2019].
- [8] Qurat Ul Ain, Wasi Haider Butt, Muhammad Waseem Anwar, Farooque Azam, and Bilal Maqbool. 2019. A Systematic Review on Code Clone Detection. *IEEE Access* 7 (2019), 86121–86144.
- [9] Junaid Akram, Zhendong Shi, Majid Mumtaz, and Ping Luo. 2018. DroidCC: A Scalable Clone Detection Approach for Android Applications to Detect Similarity at Source Code Level. In *2018 IEEE 42nd Annual Computer Software and Applications Conference, COMPSAC 2018, Tokyo, Japan, 23-27 July 2018, Volume 1*. 100–105.
- [10] Omar H Alhazmi and Yashwant K Malaiya. 2005. Quantitative vulnerability assessment of systems software. (2005), 615–620.
- [11] Maria Apostolaki, Aviv Zohar, and Laurent Vanbever. 2017. Hijacking Bitcoin: Routing Attacks on Cryptocurrencies. (2017), 375–392.
- [12] Sarah Azouvi, Mary Maller, and Sarah Meiklejohn. 2018. Egalitarian Society or Benevolent Dictatorship: The State of Cryptocurrency Governance. In *Financial Cryptography and Data Security - FC 2018 International Workshops, BITCOIN, VOTING, and WTSC, Nieuwpoort, Curaçao, March 2, 2018, Revised Selected Papers*. 127–143.
- [13] Massimo Bartoletti, Salvatore Carta, Tiziana Cimoli, and Roberto Saia. 2017. Dissecting Ponzi schemes on Ethereum: identification, analysis, and impact. *arXiv: Cryptography and Security* (2017).
- [14] Alex Biryukov, Dmitry Khovratovich, and Ivan Pustogarov. 2014. Deanonymisation of Clients in Bitcoin P2P Network. (2014), 15–29.
- [15] Weili Chen, Zibin Zheng, Jiahui Cui, Edith C. H. Ngai, Peilin Zheng, and Yuren Zhou. 2018. Detecting Ponzi Schemes on Ethereum: Towards Healthier Blockchain Technology. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, April 23-27, 2018*. 1409–1418.
- [16] Sergej Chodarev, Emilia Pietrikova, and Jan Kollar. 2015. Haskell clone detection using pattern comparing algorithm. (2015), 1–4.
- [17] Nicolas Christin and Reihaneh Safavi-Naini (Eds.). 2014. *Financial Cryptography and Data Security - 18th International Conference, FC 2014, Christ Church, Barbados, March 3-7, 2014, Revised Selected Papers*. Lecture Notes in Computer Science, Vol. 8437. Springer.
- [18] Jonathan Crussell, Clint Giber, and Hao Chen. 2015. AnDarwin: Scalable Detection of Android Application Clones Based on Semantics. *IEEE Trans. Mob. Comput.* 14, 10 (2015), 2007–2019.
- [19] Ming Fan, Jun Liu, Xiapu Luo, Kai Chen, Zhenzhou Tian, Qinghua Zheng, and Ting Liu. 2018. Android malware familial classification and representative sample selection via frequent subgraph analysis. *IEEE TIFS* 13, 8 (2018), 1890–1905.
- [20] Ming Fan, Xiapu Luo, Jun Liu, Chunyin Nong, Qinghua Zheng, and Ting Liu. 2019. CTDroid: leveraging a corpus of technical blogs for android malware analysis. *IEEE Transactions on Reliability* (2019).
- [21] Ming Fan, Xiapu Luo, Jun Liu, Meng Wang, Chunyin Nong, Qinghua Zheng, and Ting Liu. 2019. Graph embedding based familial analysis of Android malware using unsupervised learning. In *Proc.ICSE*. IEEE, 771–782.
- [22] Yikun Hu, Yuanzhang Zhang, Juanru Li, and Dawu Gu. 2017. Binary code clone detection across architectures and compiling configurations. In *Proceedings of the 25th International Conference on Program Comprehension, ICPC 2017, Buenos Aires, Argentina, May 22-23, 2017*. 88–98.
- [23] Shruti Jadon. 2016. Code clones detection using machine learning technique: Support vector machine. (2016).
- [24] C. M. Kamalpriya and Paramvir Singh. 2017. Enhancing program dependency graph based clone detection using approximate subgraph matching. In *11th IEEE International Workshop on Software Clones, IWSC 2017, Klagenfurt, Austria, February 21, 2017*. 61–67.
- [25] George Kappos, Haaron Yousaf, Mary Maller, and Sarah Meiklejohn. 2018. An Empirical Analysis of Anonymity in Zcash. In *27th USENIX Security Symposium, USENIX Security 2018, Baltimore, MD, USA, August 15-17, 2018*. 463–477.
- [26] Richard M Karp and Michael O Rabin. 1987. Efficient randomized pattern-matching algorithms. *Ibm Journal of Research and Development* 31, 2 (1987), 249–260.
- [27] Seulbae Kim, Seunghoon Woo, Heejo Lee, and Hakjoo Oh. 2017. VUDDY: A Scalable Approach for Vulnerable Code Clone Discovery. (2017), 595–614.
- [28] Zhen Liu, Qiang Wei, and Yan Cao. 2017. Vfdetect: A vulnerable code clone detection system based on vulnerability fingerprint. In *2017 IEEE 3rd Information Technology and Mechatronics Engineering Conference (ITOEC)*. IEEE, 548–553.
- [29] Sarah Meiklejohn, Marjori Pomarole, Grant Jordan, Kirill Levchenko, Damon McCoey, Geoffrey M. Voelker, and Stefan Savage. 2016. A fistful of Bitcoins: characterizing payments among men with no names. *Commun. ACM* 59, 4 (2016), 86–93.
- [30] Andrew Miller, Malte Möser, Kevin Lee, and Arvind Narayanan. 2017. An Empirical Analysis of Linkability in the Monero Blockchain. *CoRR* abs/1704.04299 (2017).
- [31] Md Rakib Hossain Misu and Kazi Sakib. 2017. Interface Driven Code Clone Detection. In *24th Asia-Pacific Software Engineering Conference, APSEC 2017, Nanjing, China, December 4-8, 2017*. 747–748.
- [32] Md Rakib Hossain Misu, Abdus Satter, and Kazi Sakib. 2017. An Exploratory Study on Interface Similarities in Code Clones. In *24th Asia-Pacific Software Engineering Conference Workshops, APSEC Workshops 2017, Nanjing, China, December 4-8, 2017*. 126–133.
- [33] Yuta Nakamura, Eunjong Choi, Norihiro Yoshida, Shusuke Haruna, and Katsuro Inoue. 2016. Towards Detection and Analysis of Interlanguage Clones for Multilingual Web Applications. In *10th International Workshop on Software Clones, IWSC@SANER 2016, Osaka, Japan, March 15, 2016*. 17–18.
- [34] Christian D. Newman, Tessandra Sage, Michael L. Collard, Hakam W. Alomari, and Jonathan I. Maletic. 2016. srcSlice: a tool for efficient static forward slicing. In *Proceedings of the 38th International Conference on Software Engineering, ICSE 2016, Austin, TX, USA, May 14-22, 2016 - Companion Volume*. 621–624.
- [35] Manziba Akanda Nishi and Kostadin Damevski. 2018. Scalable code clone detection and search based on adaptive prefix filtering. *Journal of Systems and Software* 137 (2018), 130–142.
- [36] Jayadeep Pati, Babloo Kumar, Devesh Manjhi, and K K Shukla. 2017. A Comparison Among ARIMA, BP-NN, and MOGA-NN for Software Clone Evolution Prediction. *IEEE Access* 5 (2017), 11841–11851.
- [37] Chaoyong Ragkhitwetsagul and Jens Krinke. 2017. Using compilation/decompilation to enhance clone detection. (2017), 1–7.
- [38] Chaoyong Ragkhitwetsagul, Jens Krinke, and Bruno Marnette. 2018. A picture is worth a thousand words: Code clone detection based on image similarity. In *12th IEEE International Workshop on Software Clones, IWSC 2018, Campobasso, Italy, March 20, 2018*. 44–50.
- [39] Pierre Reibel, Haaron Yousaf, and Sarah Meiklejohn. 2019. Short Paper: An Exploration of Code Diversity in the Cryptocurrency Landscape. In *Financial Cryptography and Data Security - 23rd International Conference, FC 2019, Frigate Bay, St. Kitts and Nevis, February 18-22, 2019, Revised Selected Papers*. 73–83.
- [40] Fergal Reid and Martin Harrigan. 2011. An Analysis of Anonymity in the Bitcoin System. In *PASSAT/SocialCom 2011, Privacy, Security, Risk and Trust (PASSAT), 2011 IEEE Third International Conference on and 2011 IEEE Third International Conference on Social Computing (SocialCom), Boston, MA, USA, 9-11 Oct., 2011*. 1318–1326.
- [41] Roopam and Gurpreet Singh. 2017. To enhance the code clone detection algorithm by using hybrid approach for detection of code clones. (2017).
- [42] Tim Ruffing, Sri Aravinda Krishnan Thyagarajan, Viktor Ronge, and Dominique Schroder. 2018. Burning Zerocoins for Fun and for Profit: A Cryptographic Denial-of-Spending Attack on the Zerocoin Protocol. *IACR Cryptology ePrint Archive* 2018 (2018), 612.
- [43] Yusuke Sabi, Yoshiki Higo, and Shinji Kusumoto. 2017. Rearranging the order of program statements for code clone detection. In *11th IEEE International Workshop on Software Clones, IWSC 2017, Klagenfurt, Austria, February 21, 2017*. 15–21.
- [44] Hitesh Sajani, Vaibhav Saini, Jeffrey Svajlenko, Chanchal K Roy, and Cristina Videira Lopes. 2016. SourcererCC: scaling code clone detection to big-code. *international conference on software engineering* (2016), 1157–1168.
- [45] Yuichi Semura, Norihiro Yoshida, Eunjong Choi, and Katsuro Inoue. 2017. CCFinderSW: Clone Detection Tool with Flexible Multilingual Tokenization. In *24th Asia-Pacific Software Engineering Conference, APSEC 2017, Nanjing, China, December 4-8, 2017*. 654–659.
- [46] Abdullah Sheneamer, Swarup Roy, and Jugal Kalita. 2018. A detection framework for semantic code clones and obfuscated code. *Expert Syst. Appl.* 97 (2018), 405–420.
- [47] Michele Spagnuolo, Federico Maggi, and Stefano Zanero. 2014. BitIodine: Extracting Intelligence from the Bitcoin Network. In *Financial Cryptography and Data Security - 18th International Conference, FC 2014, Christ Church, Barbados, March 3-7, 2014, Revised Selected Papers*. 457–468.
- [48] Jeffrey Svajlenko and Chanchal K. Roy. 2017. CloneWorks: a fast and flexible large-scale near-miss clone detection tool. In *Proceedings of the 39th International Conference on Software Engineering, ICSE 2017, Buenos Aires, Argentina, May 20-28, 2017 - Companion Volume*. 177–179.

- [49] Jeffrey Svajlenko and Chanchal Kumar Roy. 2017. Fast and flexible large-scale clone detection with CloneWorks. In *Proceedings of the 39th International Conference on Software Engineering, ICSE 2017, Buenos Aires, Argentina, May 20-28, 2017 - Companion Volume*. 27–30.
- [50] Rajkumar Tekchandani, Rajesh Kumar Bhatia, and Maninder Singh. 2017. Code clone genealogy detection on e-health system using Hadoop. *Computers & Electrical Engineering* 61 (2017), 15–30.
- [51] Hannes Thaller, Rudolf Ramler, Josef Pichler, and Alexander Egyed. 2017. Exploring code clones in programmable logic controller software. In *22nd IEEE International Conference on Emerging Technologies and Factory Automation, ETFA 2017, Limassol, Cyprus, September 12-15, 2017*. 1–8.
- [52] Marie Vasek, Micah Thornton, and Tyler Moore. 2014. Empirical Analysis of Denial-of-Service Attacks in the Bitcoin Ecosystem. In *Financial Cryptography and Data Security - FC 2014 Workshops, BITCOIN and WAHC 2014, Christ Church, Barbados, March 7, 2014, Revised Selected Papers*. 57–71.
- [53] Tijana Vislavski, Gordana Rakic, Nicolás Cardozo, and Zoran Budimac. 2018. LICCA: A tool for cross-language clone detection. In *25th International Conference on Software Analysis, Evolution and Reengineering, SANER 2018, Campobasso, Italy, March 20-23, 2018*. 512–516.
- [54] Min Wang, Pengcheng Wang, and Yun Xu. 2017. CCSharp: An Efficient Three-Phase Code Clone Detector Using Modified PDGs. In *24th Asia-Pacific Software Engineering Conference, APSEC 2017, Nanjing, China, December 4-8, 2017*. 100–109.
- [55] Pengcheng Wang, Jeffrey Svajlenko, Yanzhao Wu, Yun Xu, and Chanchal K. Roy. 2018. CCAAligner: a token based large-gap clone detector. In *Proceedings of the 40th International Conference on Software Engineering, ICSE 2018, Gothenburg, Sweden, May 27 - June 03, 2018*. 1066–1077.
- [56] Hongfa Xue, Guru Venkataramani, and Tian Lan. 2018. Clone-hunter: accelerated bound checks elimination via binary code clone detection. In *Proceedings of the 2nd ACM SIGPLAN International Workshop on Machine Learning and Programming Languages, MAPL@PLDI 2018, Philadelphia, PA, USA, June 18-22, 2018*. 11–19.
- [57] Yanming Yang, Zhilei Ren, Xin Chen, and He Jiang. 2018. Structural Function Based Code Clone Detection Using a New Hybrid Technique. (2018), 286–291.
- [58] Dongjin Yu, Jie Wang, Qing Wu, Jiazha Yang, Jiaojiao Wang, Wei Yang, and Wei Yan. 2017. Detecting Java Code Clones with Multi-granularities Based on Bytecode. (2017), 317–326.
- [59] Yusuke Yuki, Yoshiki Higo, and Shinji Kusumoto. 2017. A technique to detect multi-grained code clones. (2017), 54–60.
- [60] Oldřich Říčan. 2012. Historical Biogeography of Neotropical Freshwater Fishes. *Systematic Biology* 61, 3 (2012), 543–545.