The Hong Kong Polytechnic University

Department of Computing

Evolution Analysis of Ethereum Transactions

LAM Chi Kong

A thesis
submitted in partial fulfilment of the requirements
for the degree of

Master of Science in Information Technology

December 2020

Abstract

Ethereum is one of the most popular decentralized platforms for executing computer programs, they often refer to smart contracts which provide extensive applications on top of transferring value in Ether. The records of Ethereum transactions have grown into large datasets since the launch of Ethereum in 2015, there are over 9 million transactions occurred by the end of 2019 and each transaction contains multiple attribute values of different data types. It causes difficulty to understand the evolution of Ethereum transactions related to Ether value transfer, smart contract creation and smart contract invocation, this thesis analyzes these activities with descriptive statistics, regression model and static plots for the entire time period. In addition, this thesis develops an application which allows users to select parties and time intervals for dynamic graph visualization of these activities. The significance of this thesis is to provide a macro-view on distribution and correlation for statistics, micro-view By developing an open-source program for large graph visualization and large graph layout, this paper enables easy access to evolution analysis of Ethereum transactions while reserves customizable functionalities for further development. To understand major activities of Ethereum transactions including money transfer, contract creation and contract call, this paper conducts evolution analysis to gain new observations.

Acknowledgement

First of all, I would like to thank my supervisor, Dr LUO Xiapu Daniel, for guiding me to complete this thesis under a difficult time. Daniel is knowledgeable and helpful in leading me to choose an interesting topic, then provides me valuable suggestions on learning the most from this research work.

Before starting this thesis, I had no idea about blockchain and I was not an experienced programmer. It was a huge challenge for me to decide a research topic of analyzing transactions on Ethereum blockchain. Daniel is patient enough to clear all my questions and make me comfortable and confident to take the challenge.

Finally, I would like to thank all my friends who have supported me in completing my thesis. With all of your help, I keep improving myself and solve problems more efficiently.

Contents

1 Introduction 1

1.1 Background 1

1.2 Proposed Approach 3

1.3 Thesis Structure 4

2 Related Work 5

2.1 Ethereum Transactions 5

2.2 Evolution Analysis 6

2.3 New Perspective 7

2.4 Research Significance 8

3 Methodology 11

3.1 Data Collection 11

3.1.1 Statistical Data 12

3.1.2 Graph Data 13

3.2 Statistical Analysis 14

3.2.1 Descriptive Statistics 14

3.2.2 Regression Model 15

3.3 Graph Visualization 17

3.3.1 Data View 18

3.3.2 Graph View 19

3.3.3 Node View 20

3.3.4 Control View 21

4 Applications 23

4.1 Abnormality Detection 23

4.2 Impact Analysis 23

4.3 Evolutionary Study 24

4.4 Machine Learning 24

5 Evaluation 25

5.1 Abnormality Detection 25

5.1.1 Creation Statistics 25
5.1.2 Decreation Statistics 27
5.1.3 Significant Event 29
5.2 Impact Analysis 30
5.3 Evolutionary Study 31

6 Conclusion **33**
6.1 Major Contributions 33
6.2 Future Improvements 33

References **35**

List of Figures

2.1 Example of an External Transaction 6

3.1 Datasets of Ethereum Transactions 11
3.2 Data fields for contract creation 12
3.3 Data fields for contract decreation 13
3.4 Example of a time plot 15
3.5 Graph visualization is divided into four components including (A) data view, (B) graph view, (C) node view and (D) control view. 18
3.6 Data view consists of (a) options to select data source, (b) options to select exchange names, (c) options to select starting date, (d) options to select ending date, (e) options to select graph type and (f) retrieval of data under selection criteria. . . 18
3.7 Graph view shows the effects of nodes and edges (a) before clicking a node and (b) after clicking a node. 20
3.8 Node view displays attribute values and names of nodes (a) before clicking a node and (b) after clicking a node. 20
3.9 Control view includes (a) options to filter the selected exchanges, (b) time brush to filter transactions within a specific period, (c) value slider to filter transactions within a specified range, (d) options to select a centrality measure, (e) options to select a community measure, and (f) "Reset" and "Apply" buttons. 21

5.1 Daily number of created contracts 26
5.2 Daily number of contract creators 27
5.3 Daily number of refund records 28
5.4 Daily number of refunders 29
5.5 CIG of decentralized exchanges in selected interval 30
5.6 CIG of decentralized exchanges in incremental intervals 32

List of Tables

3.1 Example of descriptive statistics 14
3.2 Correlation coefficient matrix 15
3.3 Regression Statistics 16
3.4 ANOVA Table 16
3.5 Development Challenges 17

5.1 Daily number of created contracts 25
5.2 Daily number of contract creators 26
5.3 Daily number of refund records 27
5.4 Daily number of refunders 28

Introduction

1

1.1 Background

Ethereum is one of the largest blockchains in terms of market capitalization, the cryptocurrency used in Ethereum transactions is called Ether (ETH). Ethereum is a public blockchain and Ether is distributed based on proof-of-work system, it means miners compete for Ether by consuming resources to solve computationally intensive problems [1]. Similar to Bitcoin blockchain, Ethereum blockchain provides a decentralized platform for transactions to be globally accessible, low-cost, transparent, difficult-to-alter. Different from Bitcoin blockchain, Ethereum blockchain focuses on more than direct value transfer, it provides a global virtual machine for running computer programs as known as smart contracts to cater for complex transactions. For example, a popular application of smart contracts is designing financial products as they often involve complex conditions. To reveal the evolution or growth of blockchain transactions, the graph analysis is conducted in many research studies.

One approach of graph analysis is graph analytics. It constructs graphs with transactions as edges and the participating parties as nodes, then it computes analytics or statistics for the attributes of nodes and edges. For example, centrality measures the importance or significance of individual nodes. Spagnuolo et al. [18] created a modular framework namely "BitIodine", it was applied to multiple real-world cases. In a case related to ransomware, it analyzed the change of significance for known ransomware addresses by investigating the payments from victims' addresses.

Another approach of graph analysis is community detection. It divides nodes into different clusters to minimize the distances for nodes in the same cluster as well as to maximize the distances for nodes not in the same cluster. Nick [14] summarized several heuristic methods that can be used for clustering in forming communities. His approach included the components "Grapher" and "Classifier", "Grapher" was used for graph construction from transaction data and user data while "Classifier" was used for clustering of nodes by adding tag information as attributes.

The significance for the emergence of Ethereum blockchain is the potential of creating a new form of transactions in terms of the diversity of participants, types of activities, security of information, efficiency and cost of establishing contracts, time and place of execution, logical and technical capacity for real-world applications. For the diversity of participants, the Ethereum blockchain has decreasing entry barrier and expanding access channels with its development, it attracts participants from individuals and organizations, different countries, different devices. For the types of activities, currently the major applications on Ethereum blockchain are financial applications which traditionally have standardized services and regulatory requirements. These are typical features of a centralized application which is managed by authorized organizations, they have much weaker role in decentralized applications on Ethereum blockchain because the ownership is largely diluted, it is difficult for a single party to take control of the application even if it is the owner. For the security of information, it is difficult to recognize the identity of an address without additional information provided by its owner, also the address is not uniquely and permanently linked to its owner, therefore the privacy of the owner is protected in some extent. Regarding the public transaction information, it is recorded and shared by all Ethereum clients, the change of data needs consent from over half of the Ethereum clients which constitutes high cost of manipulating transaction information. For the efficiency and cost of establishing contracts, smart contracts are computer programs run on Ethereum Virtual Machine supported by global network of computers, and the transaction fee determined by Ether price is market-driven. For the time and place of transaction execution, currently the Ethereum platform is open all the time and accessible all over the world, it has low geographical restrictions and time cost to establish transactions among different parties. For the logical and technical capacity of real-world applications, the number and diversity of applications built on top the basic utility of moeney transfer continue to growth, some examples include the financial applications, games, social media, productivity software and so on.

To understand the behavioral patterns of Ethereum transactions, this paper raises the following research questions for investigation: 1. How do Ehtereum transactions react to abnormal changes caused by market events? 2. How does the distribution of Ethereum transactions change over time? 3. How do the importance and relationship with neighbors change over time for a specific account?

Based on evolution analysis of Ethereum transactions, this paper obtains the observations below: 1. There was abnormal change in activities of Ethereum transactions in 2018, a relevant significant event was the launch of a stable cryptocurrency Dai. 2. The decentralized exchange, "Allbit" had the most contract invocation transactions associated among all studied decentralized exchanges in 2018, it introduced the cryptocurrency Dai into its products. 3. In 2018, "Allbit" beated its predecessor "DEx.top" in terms of centrality and community measures of contract invocation.

1.2 Proposed Approach

This thesis proposes a approach considering the macro view and micro view of Ethereum transactions, which serves for the applications such as abnormality detection, impact analysis, evolutionary study.

From the macro view, the growth of smart contracts is analyzed by descriptive statistics and visualized by time plots. It can be applied to abnormality detection to inspect unusual changes in descriptive statistics, which can indicate significant events. In section 5.1, it was found that in 2018 the activities of contract creation and decreation have abnormal changes, and the issuance of stable currency Dai was a significant event in that year.

From the micro view, a program is developed to visualize three major transaction activities including money transfer, smart contract creation and smart contract invocation. It can be applied to impact analysis to visualize the distribution of transaction activities on individual accounts. In section 5.2, it was found that in 2018 the sharp increase in contract invocation was contributed mostly by the decentralized exchange "Allbit". Moreover, it can be applied to evolutionary study to illustrate the change in importance of individual accounts and their relationships with others. In section 5.3, it was found that in 2018 the "Allbit" grew sharply in contract invocation and exceeded another decentralized exchange "DEx.top".

This thesis makes the following contributions. From the macro view, transactions over the entire study period are fully imported without down sampling, they are grouped by day to reduce computational complexity, this approach can preserve the population data and extract relevant details. From the micro view, it enables user interaction to select specific accounts and time intervals for dynamic graph generation. Also, it is open-source to non-developers for direct execution and for developers for further customization. To the best of our knowledge, this thesis is the first research paper proposing a approach on fulfilling all of above requirements.

1.3 Thesis Structure

The remaining chapters of this thesis are organized as below.

Chapter 2: Related Work

This chapter includes sections Ethereum Transactions, Evolution Analysis, New Perspective.

Chapter 3: Methodology

This chapter includes sections Data Collection, Statistical Analysis, Graph Visualization. Data Collection includes subsections Statistical Data and Graph Data. Statistical Analysis includes subsections Descriptive Statistics and Regression Model. Graph Visualization includes subsections Data View, Graph View, Node View, Control View.

Chapter 4: Applications

This chapter includes sections Abnormality Detection, Impact Analysis, Evolutionary Study.

Chapter 5: Evaluation

This chapter includes sections Abnormality Detection, Impact Analysis, Evolutionary Study. Abnormality includes subsections Creation Statistics, Decreation Statistics, Significant Event.

Chapter 6: Conclusion

This chapter includes sections Major Contributions and Future Improvements.

Related Work 2

2.1 Ethereum Transactions

In the simplest terms, blockchain represents an unchangeable digital ledger system. A distinctive feature of blockchain technology is its distributed implementation, further discussions on blockchain can be found in many publications [17]. Bitcoin blockchain is one of the best known blockchains for speculation in Bitcoin cryptocurrency and payments with Bitcoin wallets. Ethereum blockchain is another popular blockchain for executing complex transactions on top of transferring value in its cryptocurrency Ether. It allows to create smart contracts and enables users to create crypto-assets. For example, Ethereum tokens are sold in the event of Initial Coin Offering (ICO) [19]. Such token ICOs have attracted many startups and organizations to raise funds by selling digital coins that allow recipients to use the promised service as much as possible. In the years of 2018 and 2019, Ethereum ICO raised billions of dollars. While fluctuations in Ether prices are not Ethereum transactions, ICO is a process involving many Ehtereum transactions.

Ethereum is a state machine based on transactions. In other words, a transaction occurred between two different accounts converts Ethereum globally from one state to another [7]. A transaction is executed in a block, its two accounts are the source and target. These accounts can be Externally Owned Accounts (EOAs) or smart contracts, EOAs are serialized and then submitted to the blockchain. There are two types of transactions including external transactions and internal transactions. In an external transaction, the sender (source) account is an EOA, the recipient (target) account is an EOA if the transaction is money transfer or it is a contract account if the transaction is contract creation or contract invocation [10]. The data fields of an external transaction include account addresses, timestamp, value and others [21]. The Fig. 2.1 shows typical elements of an external transactions. For an internal transaction, the source account is smart contract instead of EOA, its corresponding activities are transferring money to EOA, creating smart contract and invoking smart contract which are the same for an external transaction. Every account has a nonce keeping track of the transactions

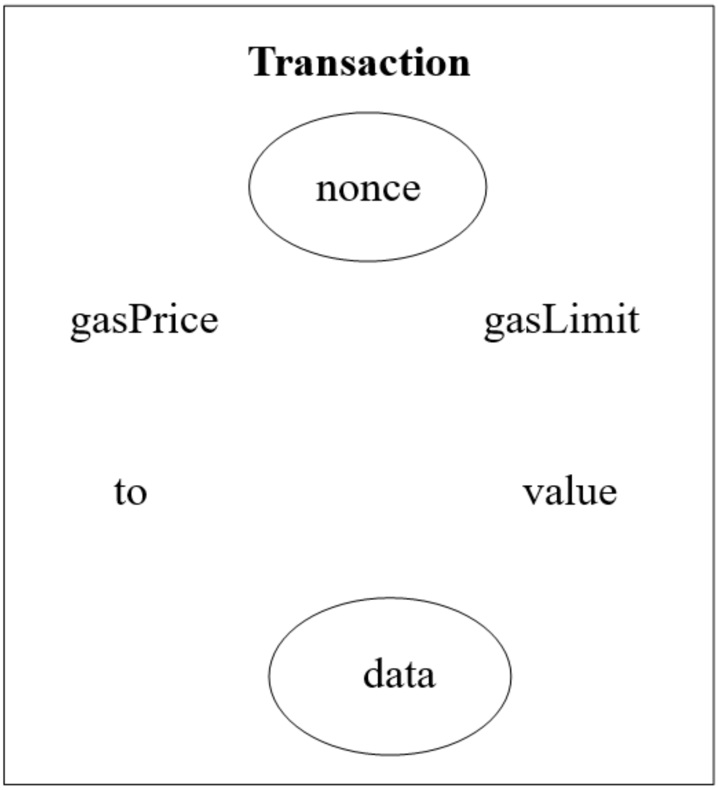


Fig. 2.1: Example of an External Transaction

they perform, it can be used to prevent replay attacks. A transaction is a list of operations to be executed, gas for an operation is relatively fixed and required to perform this operation. The gasPrice field represents the price per unit of gas you are willing to pay while the gasLimit field determines how many units of gas you are willing to pay for.

2.2 Evolution Analysis

Regarding transaction analysis, Chen et al. analyze all the Ethereum transactions from 30 Jul 2015 to 10 Jun 2017 by constructing Money Flow Graph (MFG) for money transfer, Contract Creation Graph (CCG) for smart contract creation and Contract Invocation Graph (CIG) for smart contract invocation [2]. They also provide graph analytics as measures of degree, clustering, correlation, importance, assortativity and connected components. Yue et al. develop a web application called BitExTract for interactive visualization of Bitcoin exchanges, it provides views including exchanges list panel for selecting exchanges, massive sequence view for related news, connection view for inter-exchange relationships grouped by geographical region and comparison view for transaction amounts over time [22]. It is worth noting that it enables users to select exchanges and time interval, which gives users flexibility to extract information about evolution of Bitcoin exchanges.

Regarding graph visualization, Lanum suggests Gephi which is a Graphical User Interface (GUI) written in Java for building graph visualization, it provides many options for configuring its built-in UI [11]. Meanwhile, there is a considerable learning cost of these Gephi-specific configurations, and the manual process of data handling is not as ideal as automated scripts for large datasets. Meeks recommends D3.js which is a JavaScript library for web-based interactive visualization to visualize graph data, it enables data filtering, graph construction and graph visualization all in scripts [13]. There is also a learning cost of configuring graph visualization in JavaScript, and more importantly an efficient algorithm and data representation are required to process and visualize large scale of graph data given the limited capacity of browser.

2.3 New Perspective

Despite the popularity of Ethereum blockchain, there is a lack of in-depth analysis on the evolution of Ethereum transactions. This thesis proposes the macro view and micro view to visualize Ethereum transactions. With the focus on evolution analysis, the macro view covers all the Ethereum transactions from 30 Jul 2015 to 25 Nov 2019 without sampling, and groups the transactions by day instead of the entire period. Moreover, the micro view is provided by an application developed in this thesis for graph visualization over time, this application allows users to interact with graphs for selected accounts and time intervals, it is open-source to developers and non-developers for direct implementation or further customization. To the best of our knowledge, this thesis is the first research work proposing the evolution analysis of Ethereum transactions satisfying all of the above criteria.

For the macro view, there are different metrics in daily transactions for activities related to smart contract as an important feature of Ethereum blockchain. These activities include contract creation which is positive for the growth of smart contracts, and contract decreation which is negative for the growth of smart contracts. To understand the evolution of contract activities, this thesis computes the descriptive statistics of the metrics and visualizes them with time plots. Besides, this thesis constructs a regression model to investigate contract suicide and its correlated factors.

For the micro view, this thesis develops an application to visualize node relationships and interact with graphs on top of constructing MFG, CCG and CIG based on the existing work. To visualize node relationships, the application can highlight the neighbors of a specific node and their edges to observe their relationships, it introduces community measures of Louvain community, label propagation community and union find community upon centrality measures to illustrate relationships among groups of nodes. To interact with graphs, the application provides flexibility for users to select accounts, time intervals and value ranges to generate graphs dynamically.

2.4 Research Significance

The existing approaches of analyzing Ethereum transactions are mainly based on static analysis or static data. The static analysis usually has fixed configuration and parameters, and is not publicly available as the open-source projects. The static data is usually the processed form, and the program to collect and transform the data is missing, therefore the data cannot be retrieved upon user requests and stays updated. These problems lead to the lack of flexibility to reuse and customize the analytical model.

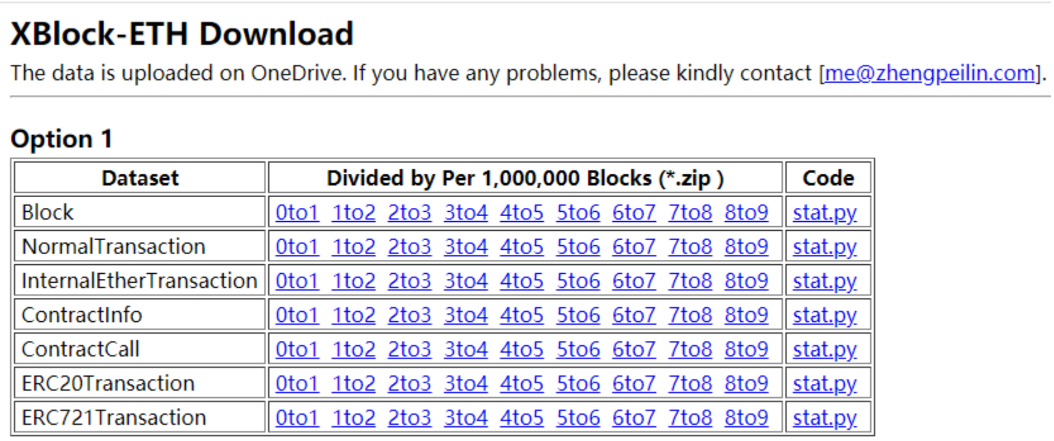
The model is not reusable when it cannot be applied to different datasets to verify its validity and accuracy. The programs for this type of models usually have restrictions on the choices of analysis, variability of time period and interaction with users. For the case in analyzing Ethereum transactions, many research studies adopt fixed time period and provide cumulative statistics, where the behavioral patterns of Ethereum transactions over time are omitted. Some of them are not open-source or provide solely the static analysis result that does not allow user interaction during data visuaization, the users cannot reuse the model for other applications, or even cannot reproduce the analytical results published by the research papers.

The model specifically designed for the investigated dataset usually lacks the capacity for customization. If the data model can only process a predefined dataset, then it has little value for users to study the model. For the case of graph visualization of Ethereum transactions, Users may want the customization on the selection of nodes (e.g. different types of exchange addresses), start date and end date, type of activity (money transfer, contract creation, contract invocation), graph analytics (centrality, community), information of neighbor nodes, selection of time range to illustrate the evolution of Ethereum transactions.

With regard to the above problems, this paper develops a program for interactive graph visualization of Ethereum transactions with the following contributions: 1. Developing a reusable program to visualize graph datasets 2. Developing an open-source program which can be customized to enhance interactive graph visualization 3. Provide use cases of this prgram including the abnormality detection, impact analysis, evolutionary study explained in details at Section 4.

Methodology

3.1 Data Collection



XBlock-ETH Download

The data is uploaded on OneDrive. If you have any problems, please kindly contact [me@zhengpeilin.com].

Option 1

Dataset	Divided by Per 1,000,000 Blocks (*.zip)	Code
Block	0to1 1to2 2to3 3to4 4to5 5to6 6to7 7to8 8to9	stat.py
NormalTransaction	0to1 1to2 2to3 3to4 4to5 5to6 6to7 7to8 8to9	stat.py
InternalEtherTransaction	0to1 1to2 2to3 3to4 4to5 5to6 6to7 7to8 8to9	stat.py
ContractInfo	0to1 1to2 2to3 3to4 4to5 5to6 6to7 7to8 8to9	stat.py
ContractCall	0to1 1to2 2to3 3to4 4to5 5to6 6to7 7to8 8to9	stat.py
ERC20Transaction	0to1 1to2 2to3 3to4 4to5 5to6 6to7 7to8 8to9	stat.py
ERC721Transaction	0to1 1to2 2to3 3to4 4to5 5to6 6to7 7to8 8to9	stat.py

Fig. 3.1: Datasets of Ethereum Transactions

Google BigQuery provides a public dataset which contains updated information of Ethereum transactions. However, it adds complexity in configuring Google account to retrieve data greater than 1 GB, and data size in this thesis is above 100 GB. As an alternative, this thesis adopts data that comes from Xblock website [23]. The datasets of Ethereum transactions are shown in Fig. 3.1.

For dataset "Block", it records information related to blocks and its data fields include "blockNumber", "timestamp", "gasUsed", "miner", "reward" and others. For dataset "NormalTransaction", it records information related to external transactions and its data fields include "blockNumber", "timestamp", "from", "to", "creates", "value" and others. For dataset "InternalEtherTransaction", it records information related to internal transactions and its data fields include "blockNumber", "timestamp", "from", "to", "fromIsContract", "toIsContract", "value" and others. For dataset "ContractInfo", it records information related to contract creation and decreation and its data fields include "address", "createdTimestamp", "creator", "decreatedTimestamp", "refunder", "refundValue" and others. When a smart contract is decreated (or deleted), it costs gas to compensate miners for executing this transaction, then some refund value is sent from the refunder to specific accounts. For dataset "ContractCall", it records information related to contract invocation (or call) and its data fields include "blockNumber", "timestamp", "from", "to", "fromIsContract", "toIsContract", "value" and others. When a smart contract is called, it is executed to perform operations defined in contract code. For dataset "ERC20Transaction", it records information related to ERC20 tokens and its data fields include "blockNumber", "timestamp", "tokenAddress", "from", "to", "value" and others. For dataset "ERC721Transaction", it records information related to ERC721 tokens and its data fields include "blockNumber", "timestamp", "tokenAddress", "from", "to", "tokenId" and others. ERC20 and ERC721 are two design standards for Ethereum tokens, they facilitate the development of smart contracts for token issuance and comparison among different tokens. Tokens are frequently used for fundraising and play a similar role as shares of listed companies.

These datasets cover the Ethereum transactions from block 0 on 30 Jul 2015 to block 8,999,999 on 25 Nov 2019. The dataset "ContractInfo" forms the statistical data to compute descriptive statistics and to formulate regression model for investigating the growth of contract activities. Datasets "NormalTransaction", "InternalEtherTransaction", "ContractInfo" and "ContractCall" form the graph data to construct MFG, CCG and CIG for graph visualization of evolving transaction activities.

3.1.1 Statistical Data

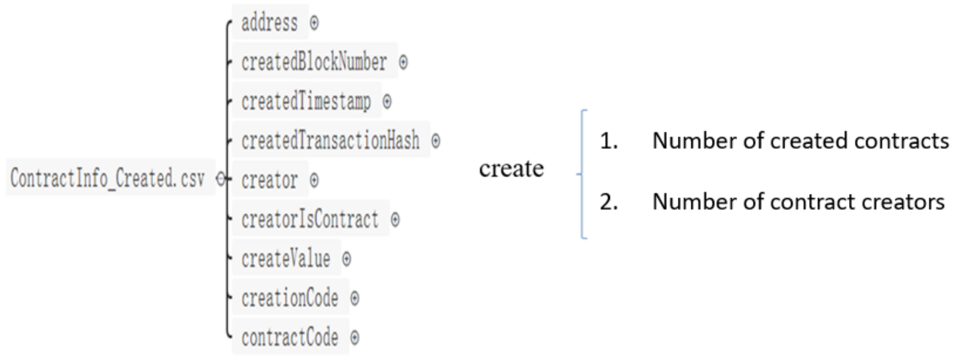


Fig. 3.2: Data fields for contract creation

The dataset "ContractInfo" consists of transaction information related to smart contract creation and decreation.

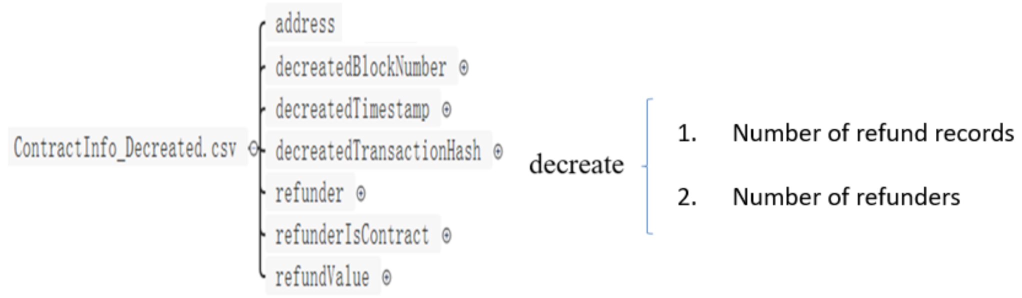


Fig. 3.3: Data fields for contract decreation

Transactions for smart contract creation are recorded with data fields shown in Fig. 3.2, these transactions are grouped by day to compute the metrics including daily number of created contracts and daily number of contract creators. Then, the metrics are used to compute descriptive statistics and to draw time plots.

Transactions for smart contract decreation are recorded with data fields shown in Fig. 3.3, these transactions are grouped by day to compute the metrics including daily number of refund records and daily number of refunders. Then, the metrics are used to compute descriptive statistics and to draw time plots. They are also used to formulate a regression model that describes the correlated factors of refund.

3.1.2 Graph Data

The relevant datasets are processed in the following order:

1. The dataset "ContractInfo" is used to obtain all "ccg_nodes" and "ccg_edges" of CCG data. For "ccg_nodes", its "node_name" and "node_type" are determined by "address" in "ContractInfo". For "ccg_edges", its "from_name", "to_name", "time_stamp" are determined by "creator", "address", "createdTimestamp" in "ContractInfo".

2. The dataset "ContractCall" is used to obtain all "cig_nodes" and "cig_edges" of CIG data. For "cig_nodes", its "node_name" and "node_type" are determined by "from", "fromIsContract", "to", "toIsContract" in "ContractCall". For "cig_edges", its "from_name", "to_name", "time_stamp", "number_of_calls" are determined by "from", "to", "timestamp" in "ContractCall".

3. The dataset "InternalEtherTransaction" is used to obtain partial "mfg_nodes" and "mfg_edges" of MFG data. For "mfg_nodes", its "node_name" and "node_type" are determined by "from", "fromIsContract", "to", "toIsContract" in "InternalEtherTransaction". For "mfg_edges", its "from_name", "to_name", "time_stamp", "value_in_ether" are determined by "from", "to", "timestamp", "value" in "InternalEtherTransaction".

4. The dataset "NormalTransaction" is used to obtain remaining "mfg_nodes" and "mfg_edges" of MFG data. For "mfg_nodes", its "node_name" and "node_type" are determined by "from", "fromIsContract", "to", "toIsContract" in "NormalTransaction". For "mfg_edges", its "from_name", "to_name", "time_stamp", "value_in_ether" are determined by "from", "to", "timestamp", "value" in "NormalTransaction".

During the extraction of graph data, the addresses of exchanges are converted to their names for identification of exchanges later in graph visualization, the address-name pairs of exchanges are found in exchange directory on EtherScan [6]. Finally, the extracted nodes and edges are stored as collections in a database called "ethereum_tx".

3.2 Statistical Analysis

3.2.1 Descriptive Statistics

Table 3.1: Example of descriptive statistics

Daily number of created contracts	
Average	12267.14442
Standard Error	366.8075104
Median	7510.5
Mode	7267
Standard Deviation	11089.48345
Variance	122976643.2
Kurtosis	2.305335977
Skewness	1.789698857
Minimum	650
Maximum	52959
Summation	11212170
Observation	914

Statistical data representing contract creation and contract decreation is used to compute descriptive statistics and to draw time plots. An example of

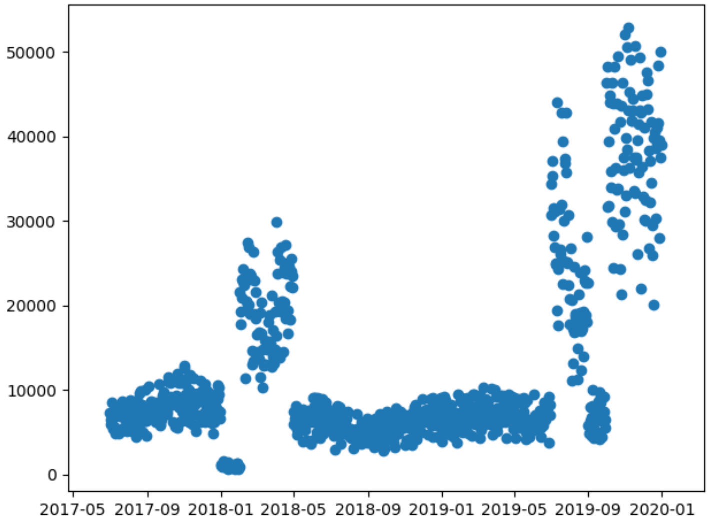


Fig. 3.4: Example of a time plot

descriptive statistics is shown in Table 3.1, these statistics describe the distribution of metric for all observation days with "Average", "Standard Error", "Median", "Mode, "Standard Deviation", "Variance", "Kurtosis", "Skewness", "Minimum", "Maximum", "Summation" and "Observation". Moreover, the change of metric is illustrated by a time plot to visualize its trend, an example is shown in Fig. 5.1.

3.2.2 Regression Model

Table 3.2: Correlation coefficient matrix

Correlation coefficient matrix						
		Created Contracts	Contract Creators	Refund Records	Refunders	Refund Amount
Created Contracts	Related	1	-0.038	.169**	.253**	-.148**
	significant		0.256	0.000	0.000	0.000
Contract Creators	Related	-0.038	1	.349**	.153**	-0.017
	significant	0.256		0.000	0.000	0.605
Refund Records	Related	.169**	.349**	1	.426**	-0.005
	significant	0.000	0.000		0.000	0.878
Refunders	Related	.253**	.153**	.426**	1	-.148**
	significant	0.000	0.000	0.000		0.000
Refund Amount	Related	-.148**	-0.017	-0.005	-.148**	1
	significant	0.000	0.605	0.878	0.000	
*. At level 0.05, the correlation was significant						
**. At level 0.01, the correlation was significant						

Regarding the daily refund amount, the potentially correlated variables including daily number of created contracts as "Created Contracts", daily number of contract creators as "Contract Creators", daily number of refund records as "Refund Records", daily number of refunders as "Refunders", daily refund amount as "Refund Amount" are investigated with the correlation coefficient matrix in Table 3.2. The correlation coefficients in the matrix are Pearson correlation coefficients. In Table 3.2, the rows "Related" and "significance" contain the test statistics of correlation coefficients for each pair of variables and their significance under two-tailed hypothesis testing respectively. If the significance is below a level, 0.05 or 0.01, it is said to be statistically significant to conclude that the pair of variables are correlated.

From the results of correlation coefficient matrix, the daily refund amount is statistically correlated to the daily number of created contracts and daily number of refunders both at significance level 0.01. Therefore, the daily number of created contracts and daily number of refunders are fitted to linear regression model to predict the daily refund amount. The regression statistics are shown in Table 3.3.

Table 3.3: Regression Statistics

Regression Statistics	
Multiple R	.839a
R Square	.704
Adjusted R Square	.653
Standard Error	320.299643300000000
Observations	914

From the regression statistics, the R Square is 70.4%, it means that 70.4% of variation in daily refund amount can be explained by the daily number of created contracts and daily number of refunders. In addition, the Analysis Of Variance (ANOVA) is performed to further investigate the relations between these variables, the results are provided in Table 3.4.

Table 3.4: ANOVA Table

ANOVA Table				
	Coefficients	Standard Error	t Stat	P-value
Intercept	183.198	25.191	7.273	.000
Created Contracts	-0.005	.001	-4.953	.000
Refunders	.081	.026	3.102	.002

From ANOVA table, the P-values of intercept, daily number of created contracts and daily number of refunders are all below 0.01, it implies they are statistically significant to be included in the linear regression model. By considering the coefficients of these explanatory variables, the resultant regression model can be written as:

$$\text{Refund Amount} = 183.198 - 0.005 * \text{Created Contracts} + 0.081 * \text{Refunders}$$

3.3 Graph Visualization

Table 3.5: Development Challenges

Challenge	Module
Interactive UI	PyQt5
Large graph visualization	Vispy
Large graph layout	PyGraphviz
Portable program	PyInstaller

The major development challenges for graph visualization of Ethereum transactions are listed in Table 3.5, they include interactive User Interface (UI), large graph visualization, large graph layout, portable executable program. These challenges are handled by the following Python modules.

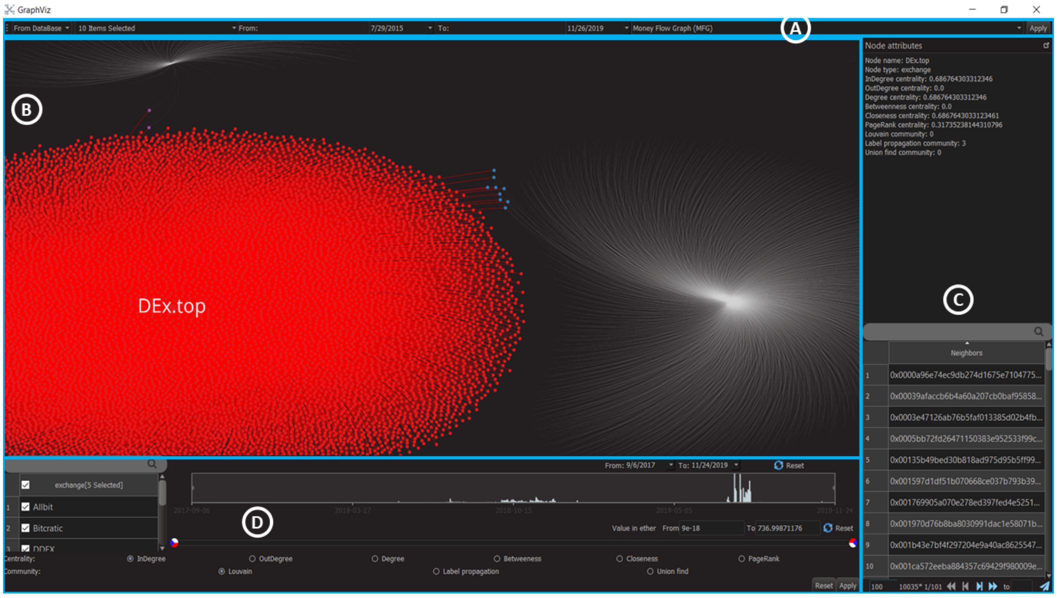
PyQt5 represents UI elements as Python classes, their style can be modified by changing the property values while their behavior can be altered by linking to a function [3]. The built-in Qt designer can be used to create UI elements and convert them into code, then only the functions need to be programmed.

Vispy leverages the computing power of GPU to achieve high performance in large data visualization [20]. It is mainly applied to filtering out the subgraph and computing the graph analytics (i.e. centrality and community measures).

PyGraphviz provides fast algorithms of generating large graph layout which avoids overlapping edges for clear visualization [9].

PyInstaller is a light-weight packaging manager to convert Python program to executable, in this paper it is used to create portable version of the program [16].

The program for graph visualization is shown in Fig. 3.5. It is composed of four core components, namely data view (Fig. 3.5A), graph view (Fig. 3.5B), node view (Fig. 3.5C) and control view (Fig. 3.5D).



From DataBase | 10 Items Selected | From: 7/29/2015 | To: 11/26/2019 | Money Flow Graph (MFG)

Node attributes

Node name: DEx.top
Node type: exchange
InDegree centrality: 0.686764303312346
OutDegree centrality: 0.0
Degree centrality: 0.686764303312346
Betweenness centrality: 0.0
Closeness centrality: 0.6867643033123461
PageRank centrality: 0.31735238144310796
Louvain community: 0
Label propagation community: 3
Union find community: 0

	Neighbors
1	0x0000a96e74ec9db274d1675e7104775...
2	0x00039afaccb6b4a60a207cb0baf95858...
3	0x0003e47126ab76b5faf013385d02b4fb...
4	0x0005bb72fd26471150383e952533f99c...
5	0x00135b49bed30b818ad975d95b5ff99...
6	0x001597d1df51b070668ce037b793b39...
7	0x001769905a070e278ed397fed4e5251...
8	0x001970d76b8ba8030991dac1e58071b...
9	0x001b43e7bf4f297204e9a40ac8625547...
10	0x001ca572eeba884357c69429f980009e...

100 10035* 1/101 to

☑ exchange[5 Selected]
1 ☑ Allbit
2 ☑ Bitcratic
3 ☑ DDEX

From: 9/6/2017 To: 11/24/2019

Value in ether From 9e-18 To 736.99871176

Centrality: ● InDegree ○ OutDegree ○ Degree ○ Betweeness ○ Closeness ○ PageRank
Community: ● Louvain ○ Label propagation ○ Union find

Fig. 3.5: Graph visualization is divided into four components including (A) data view, (B) graph view, (C) node view and (D) control view.

3.3.1 Data View

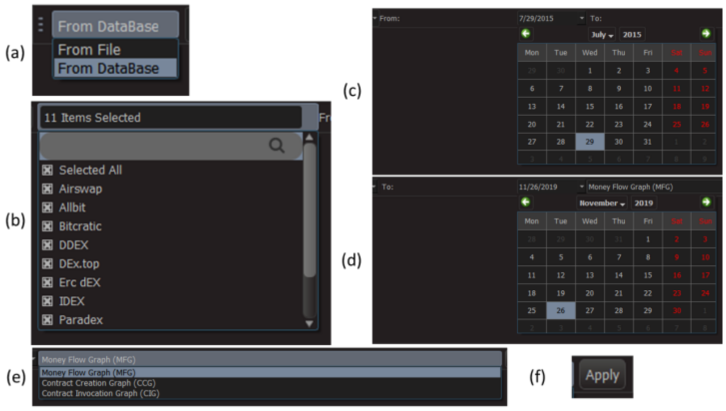


Fig. 3.6: Data view consists of (a) options to select data source, (b) options to select exchange names, (c) options to select starting date, (d) options to select ending date, (e) options to select graph type and (f) retrieval of data under selection criteria.

The data view acts as a menu bar for users to specify the selection criteria to retrieve data from the data source. The available options are in the groups of data source (Fig. 3.6a), exchange names (Fig. 3.6b), starting date (Fig. 3.6c), ending date (Fig. 3.6d) and graph type (Fig. 3.6e). The "Apply" button (Fig. 3.6f) combines these selection options to retrieve data for graph construction.

The selection of data source is to retrieve transactions from either a GEXF file or the MongoDB database. GEXF format is the default format in Gephi representing data structure of graph in the form of nodes and edges.

The selection of exchange names is to retrieve transactions where each transaction has its from address or to address in the list of selected exchanges. The exchange names are grouped by exchange type and sorted alphabetically.

The selection of starting date and ending date is to retrieve transactions occurred between these two dates. The default starting date is one day before the first transaction while the default ending date is one day after the last collected transaction.

The selection of graph types is to retrieve transactions based on three major activities, namely MFG for money transfer, CCG for contract creation, CIG for contract invocation.

Moreover, the program supports the opening of multiple windows which facilitates the comparison analysis. For examples, transactions for the same exchanges in different time periods can be compared in different windows, or transactions for different categories such as crypto exchanges and decentralized exchanges can be compared in different windows.

3.3.2 Graph View

The graph view is used to display the nodes and edges after graph construction. Before clicking a node (Fig. 3.7a), nodes of different sizes and colors are connected by light gray edges. After clicking a node (Fig. 3.7b), the clicked node and all its neighbor nodes are highlighted while the other nodes become transparent. Besides, the name of clicked node is popped up.

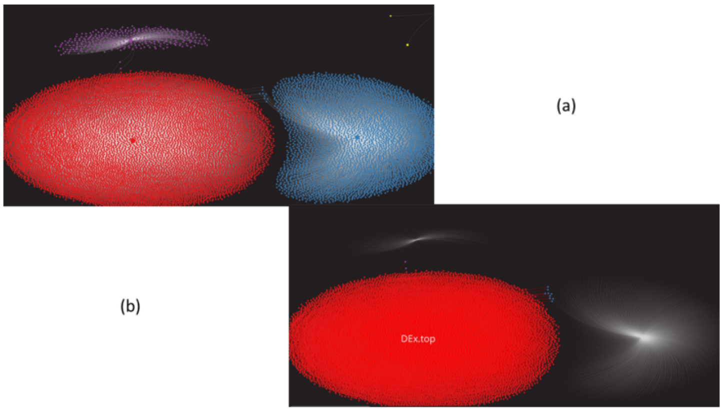
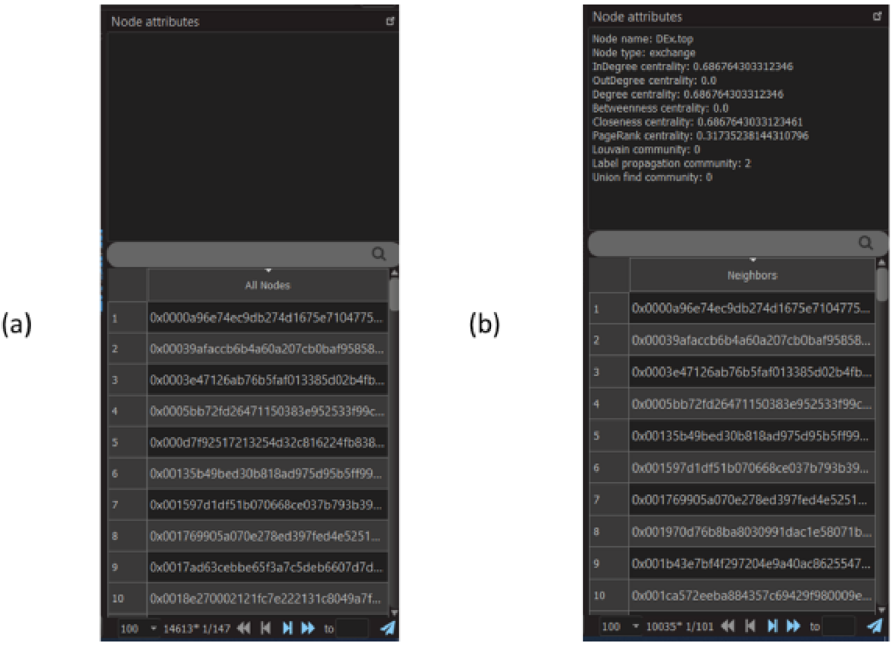


Fig. 3.7: Graph view shows the effects of nodes and edges (a) before clicking a node and (b) after clicking a node.

3.3.3 Node View



(a)

Node attributes

	All Nodes
1	0x0000a96e74ec9db274d1675e7104775...
2	0x00039afaccb6b4a60a207cb0baf95858...
3	0x0003e47126ab76b5faf013385d02b4fb...
4	0x0005bb72fd26471150383e952533f99c...
5	0x000d7f92517213254d32c816224fb838...
6	0x00135b49bed30b818ad975d95b5ff99...
7	0x001597d1df51b070668ce037b793b39...
8	0x001769905a070e278ed397fed4e5251...
9	0x0017ad63cebbe65f3a7c5deb6607d7d...
10	0x0018e270002121fc7e222131c8049a7f...

(b)

Node attributes

Node name: DEx.top
Node type: exchange
InDegree centrality: 0.686764303312346
OutDegree centrality: 0.0
Degree centrality: 0.686764303312346
Betweenness centrality: 0.0
Closeness centrality: 0.6867643033123461
PageRank centrality: 0.31735238144310796
Louvain community: 0
Label propagation community: 2
Union find community: 0

	Neighbors
1	0x0000a96e74ec9db274d1675e7104775...
2	0x00039afaccb6b4a60a207cb0baf95858...
3	0x0003e47126ab76b5faf013385d02b4fb...
4	0x0005bb72fd26471150383e952533f99c...
5	0x00135b49bed30b818ad975d95b5ff99...
6	0x001597d1df51b070668ce037b793b39...
7	0x001769905a070e278ed397fed4e5251...
8	0x001970d76b8ba8030991dac1e58071b...
9	0x001b43e7bf4f297204e9a40ac8625547...
10	0x001ca572eeba884357c69429f980009e...

Fig. 3.8: Node view displays attribute values and names of nodes (a) before clicking a node and (b) after clicking a node.

The node view is to list out the information of a clicked node. Before clicking a node (Fig. 3.8a), the values of node attributes are blank, and a list of all nodes is shown below the node attributes. After clicking a node (Fig. 3.8b), the values of node attributes are displayed, and its neighbor nodes are shown below the node attributes.

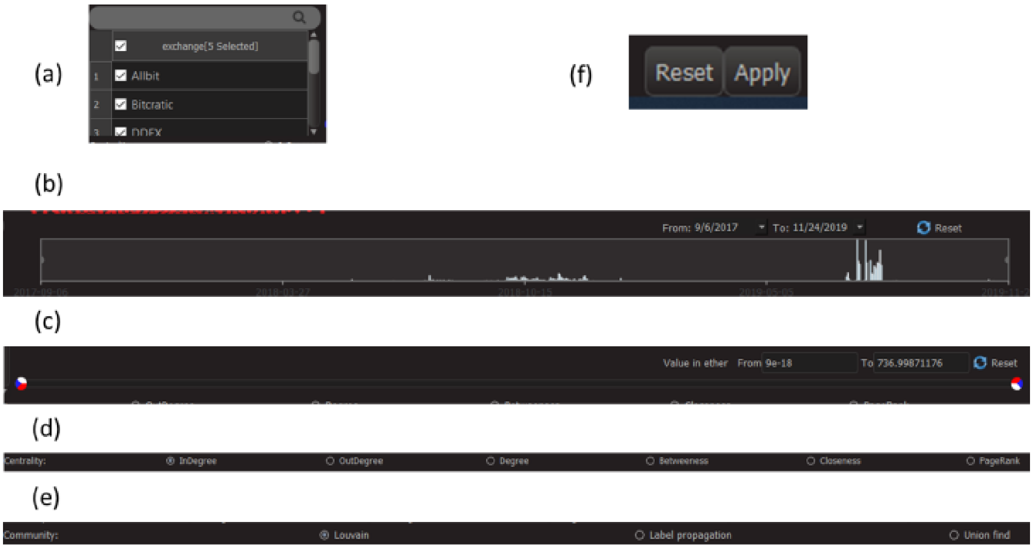


Fig. 3.9: Control view includes (a) options to filter the selected exchanges, (b) time brush to filter transactions within a specific period, (c) value slider to filter transactions within a specified range, (d) options to select a centrality measure, (e) options to select a community measure, and (f) "Reset" and "Apply" buttons.

3.3.4 Control View

The control view contains filters of transactions for selected exchanges (Fig. 3.9a), selected time interval (Fig. 3.9b), and corresponding value range (Fig. 3.9c), and selection options of centrality measure (Fig. 3.9d) and community measure (Fig. 3.9e). The "Reset" button sets the filters and selections to default values while the "Apply" button sets the filters and selections effective.

After filtering the transactions by exchange, a subgraph contains only those nodes and edges to represent the transactions with from address or to address as the specified exchanges.

After filtering the transactions by a time brush, a subgraph containing nodes and edges to represent transactions within the specified time interval is present. To observe the evolution of graph structure or the changes of graph over time, the time brush starts from a fixed time point, then it is moved at increments with the "Apply" button being pressed. As a result, the "growth" of graph can be revealed for evolution analysis.

After filtering the transactions by a value slider, the transactions within the specified value range are represented by a subgraph of nodes and edges, this value range refers to value in ether for MFG and number of calls for CIG and is not applicable for CCG. To observe the transactions at increasing weight, the value slider has a fixed end, then it is moved to right gradually with the "Apply" button taking effect. As a result, the graph with high edge weights "floats" on the surface.

The selection options for centrality measures consist of in-degree centrality, out-degree centrality, degree centrality, betweenness centrality, closeness centrality and PageRank centrality. The centrality value is positively correlated to the node size.

The selection options for community measures include Louvain community, label propagation community and union find community. The community cluster which a node is assigned to determines the node color.

4 Applications

The approach of this thesis provides the macro view and micro view for the evolution of Ethereum transactions, it can identify overall trends as well as subset details of transactions for applications such as abnormality detection, impact analysis and evolutionary study.

4.1 Abnormality Detection

In the section 3.2.1, the daily descriptive statistics over the entire period are computed to draw time plots. From the perspective of statistical analysis, the statistics should stay within certain number of standard deviations from their means for most of the time, otherwise they can be considered as abnormal.

The occurrence of abnormality can be the result reflecting existing events or the indication of future events, its detection assists us in identifying the existing events or predicting the future events.

From the macro view, this thesis monitors the daily numbers of created contracts, contract creators, refund records, refunders to detect abnormality in the growth of smart contracts.

4.2 Impact Analysis

There were certain significant events such as the launch of stable currency and cyber attack to steal cryptocurrency that cause positive or negative impact to the Ethereum blockchain. Compared with individual accounts, the exchanges can be stronger indicators to reflect the distribution of impact for these significant events.

In the section 3.3, the exchanges and activities can be selected by users to construct and visualize graphs over time. To assess the impact of a significant event on Ethereum exchanges, users can select the exchanges and activities covering the period before and after the occurrence of the event, and visualize the graphs during the period to observe the distribution of impact on different exchanges for different activities.

4.3 Evolutionary Study

It can be difficult to understand the evolution of individual accounts in terms of their importance and relationships with other accounts without transforming historical transaction data into graphs over time.

In the section 3.3, centrality measures and community measures are computed for each account and they are represented by node sizes and node colors respectively for graph visualization. The program can project the growth of individual accounts by visualizing their history with dynamic graphs.

4.4 Machine Learning

The data and algorithms are two core components of machine learning applications. For the data part, the training data and test data for pattern recognition of graph evolution is difficult to obtain. The fundamental problem is how to define the labels or clusters for certain patterns or groups, the program developed for graph visualization in this paper provides the first step to explore the graph patterns in Ethereum transactions.

From the above applications where decentralized exchanges are studied to observe the abnormal change in transaction activities, distribution of impact to different individual nodes, growth in importance and neighborhood relationships. All of them exhibit certain patterns, some of them may be regular patterns. If the research dataset is scaled up, different types of exchanges are included, or exchanges from different regions are included, or different categories of applications are included. An increasing amount of patterns are expected to be discovered, these can constitute the training dataset or test dataset if processed systematically.

Evaluation

5.1 Abnormality Detection

The growth of smart contracts is analyzed by two types of statistics namely creation statistics and decreation statistics. Creation statistics which are associated with positive growth of smart contracts include daily number of created contracts and contract creators. Decreation statistics which are associated with negative growth of smart contracts include daily number of refund records and refunders. These statistics are visualized with time plots to detect abnormal changes.

5.1.1 Creation Statistics

Table 5.1: Daily number of created contracts

Created Contracts	
Average	12267.14442
Standard Error	366.8075104
Median	7510.5
Mode	7267
Standard Deviation	11089.48345
Variance	122976643.2
Kurtosis	2.305335977
Skewness	1.789698857
Minimum	650
Maximum	52959
Summation	11212170
Observation	914

From the descriptive statistics of created contracts in 914 days in Table 5.1, the daily numbers of created contracts ranging from 650 to 52959 had a mean at 12267 and a standard deviation at 11089, the high ratio of standard deviation to mean indicated a large variation in the distribution. With reference to a normal distribution which has skewness and relative kurtosis both at 0, skewness and kurtosis measure the deviation from normal distribution in terms of degree of asymmetry and portion of outliers respectively. The daily numbers of created contracts had skewness at 1.79 and kurtosis at 2.31, its distribution was right-skewed and had low portion of outliers.

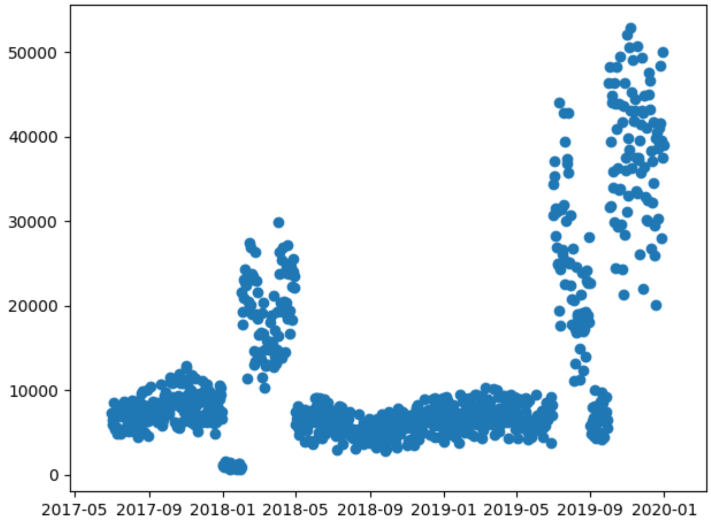


Fig. 5.1: Daily number of created contracts

From the time plot of created contracts in Fig. 5.1, the daily numbers of created contracts had different activity levels in four time intervals. The first interval was before January 2018, the activity level was low with a flat trend. The second interval was from January to May 2018, the activity level rose sharply and reached a peak. The third interval is from June 2018 to June 2019, the activity level dropped sharply and remained a flat trend. The fourth interval was after June 2019, the activity level rose sharply and reached another peak.

Table 5.2: Daily number of contract creators

Contract Creators	
Average	5343.724
Standard Error	100.7515
Median	5278
Mode	1474
Standard Deviation	3045.963
Variance	9277890
Kurtosis	-1.05016
Skewness	0.139566
Range	12489
Minimum	250
Maximum	12739
Summation	4884164
Observation	914

From the descriptive statistics of contract creators in Table 5.2, the daily numbers of contract creators ranging from 250 to 12739 had a mean at 5344 and a standard deviation at 3046, the high ratio of standard deviation to mean indicated a large variation in the distribution. The daily numbers of contract creators had skewness at 0.14 and kurtosis at -1.05, its distribution was slightly right-skewed and had moderately high portion of outliers.

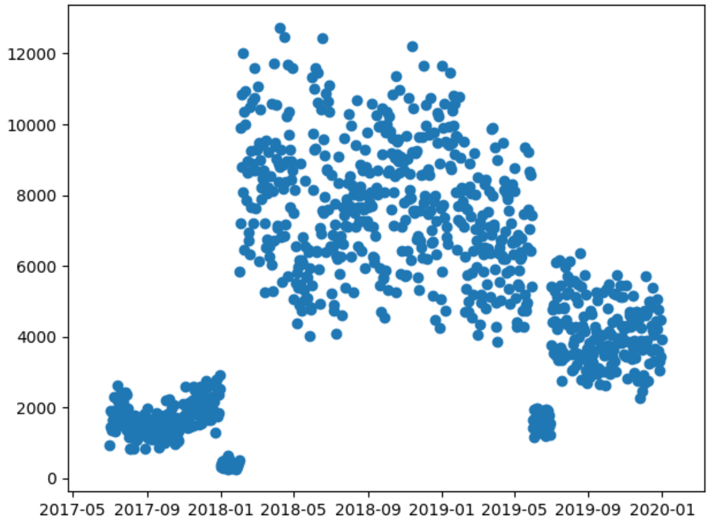


Fig. 5.2: Daily number of contract creators

From the time plot of contract creators in Fig. 5.2, the daily numbers of contract creators had different activity levels in three time intervals. The first interval was before February 2018, the activity level was low with a flat trend. The second interval was from February 2018 to May 2019, the activity level reached a peak and fluctuated with direction. The third interval is after May 2019, the activity level dropped sharply and fluctuated without direction.

5.1.2 Decreation Statistics

Table 5.3: Daily number of refund records

Refund Records	
Average	246067.8654
Standard Error	9534.322429
Median	87599.5
Mode	67
Standard Deviation	288245.7632
Variance	83085620024
Kurtosis	-0.193791222
Skewness	0.953646715
Range	1093572
Minimum	52
Maximum	1093624
Summation	224906029
Observation	914

From the descriptive statistics of refund records in Table 5.3, the daily numbers of refund records ranging from 52 to 1093624 had a mean at 246068 and a standard deviation at 288246, the high ratio of standard deviation to mean indicated a large variation in the distribution. The daily numbers of refund records had skewness at 0.95 and kurtosis at -0.19, its distribution was moderately right-skewed and had slightly high portion of outliers.

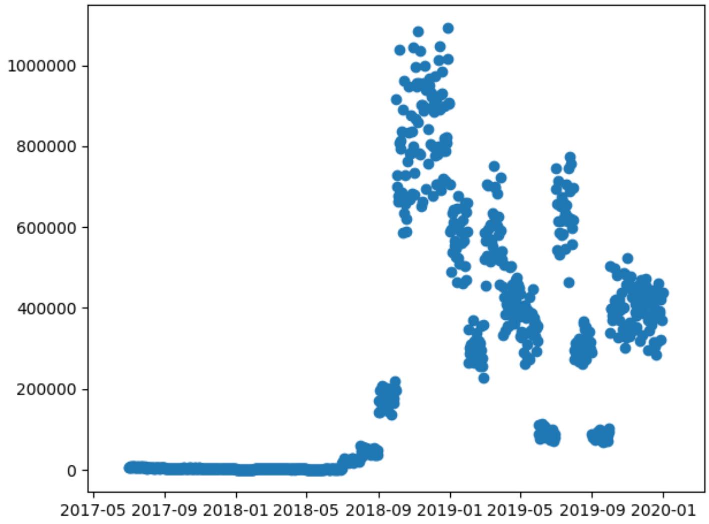


Fig. 5.3: Daily number of refund records

From the time plot of refund records in Fig. 5.3, the daily numbers of refund records had different activity levels in two time intervals. The first interval was before September 2018, the activity level was low with a flat trend. The second interval was after September 2018, the activity level reached a peak and fluctuated without direction.

Table 5.4: Daily number of refunders

Refunders	
Average	276.2144
Standard Error	15.22506
Median	111
Mode	81
Standard Deviation	460.2905
Variance	211867.4
Kurtosis	15.55016
Skewness	3.759675
Range	3169
Minimum	5
Maximum	3174
Summation	252460
Observation	914

From the descriptive statistics of refunders in Table 5.4, the daily numbers of refunders ranging from 5 to 3174 had a mean at 276 and a standard deviation at 460, the high ratio of standard deviation to mean indicated a large variation in the distribution. The daily numbers of refunders had skewness at 3.76 and kurtosis at 15.55, its distribution was right-skewed and had low portion of outliers.

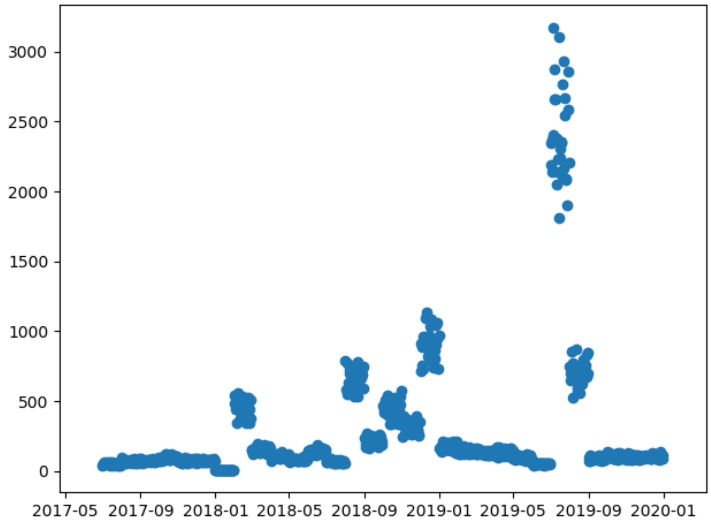


Fig. 5.4: Daily number of refunders

From the time plot of refunders in Fig. 5.4, the daily numbers of refunders had different activity levels in three time intervals. The first interval was before July 2019, the activity level was low and flucuated without direction. The second interval was from July to September 2019, the activity level rose sharply and reached a peak. The third interval was after September 2019, the activity level dropped sharply and remained a flat trend.

5.1.3 Significant Event

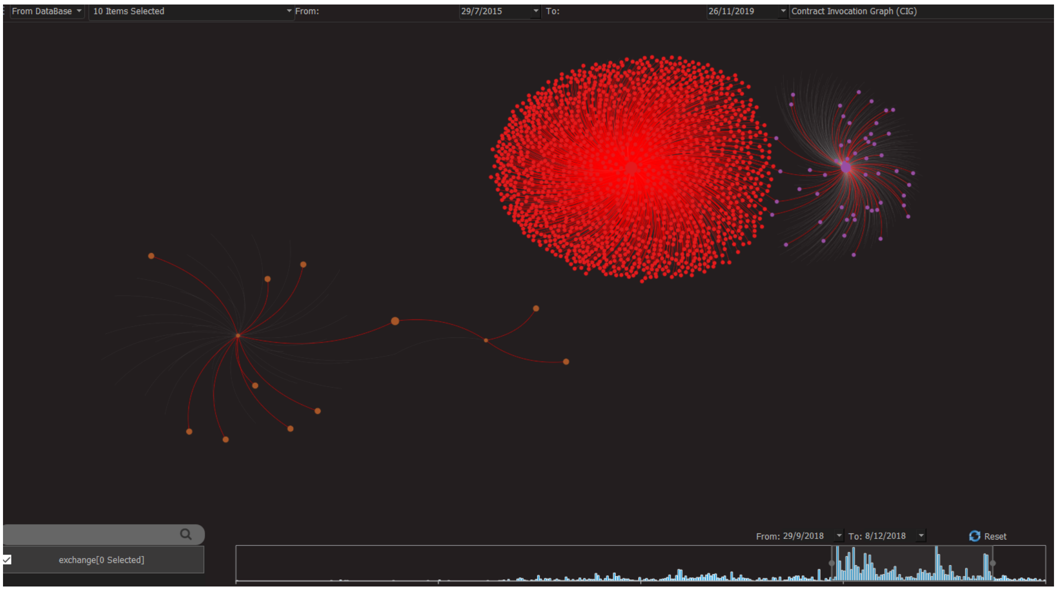
From the descriptive statistics of smart contract growth in section 5.1.1 and section 5.1.2, 2018 was a special year when abnormal changes in activity level took place. Around this year, there was a significant event in Ethereum blockchain, the issuance of a stable currency called Dai.

At the end of 2017, Dai built by Maker Foundation was issued as an ERC-20 token. Unlike Bitcoin which has volatile value and acts more like an investment than cash, Dai is marketed as a digital version of real money and its value consistently tracks the USD. To maintain stable value relative to

USD, it is backed by assets which value is locked in publicly viewable smart contracts and adjusted automatically by algorithm. By removing the volatility of a typical cryptocurrency, Dai can store and exchange value like traditional money to serve many financial applications such as spending and lending, this can be a reason that activity level of smart contracts increased sharply for 2018.

5.2 Impact Analysis

In 2018 after the stable coin Dai was issued, the activities of exchanges including money transfer, smart contract creation, smart contract invocation could partially reflect the distribution of its impact. For example, the CIG representing contract invocation activities of decentralized exchanges can be further analyzed as decentralized exchanges had much less activities in money transfer and smart contract creation in 2018. To be clear, the exchanges are classified as decentralized exchanges in the exchange directory on EtherScan [6].



From DataBase
10 Items Selected
From: 29/7/2015
To: 26/11/2019
Contract Invocation Graph (CIG)

From: 29/9/2018
To: 8/12/2018

☑ exchange[0 Selected]

Fig. 5.5: CIG of decentralized exchanges in selected interval

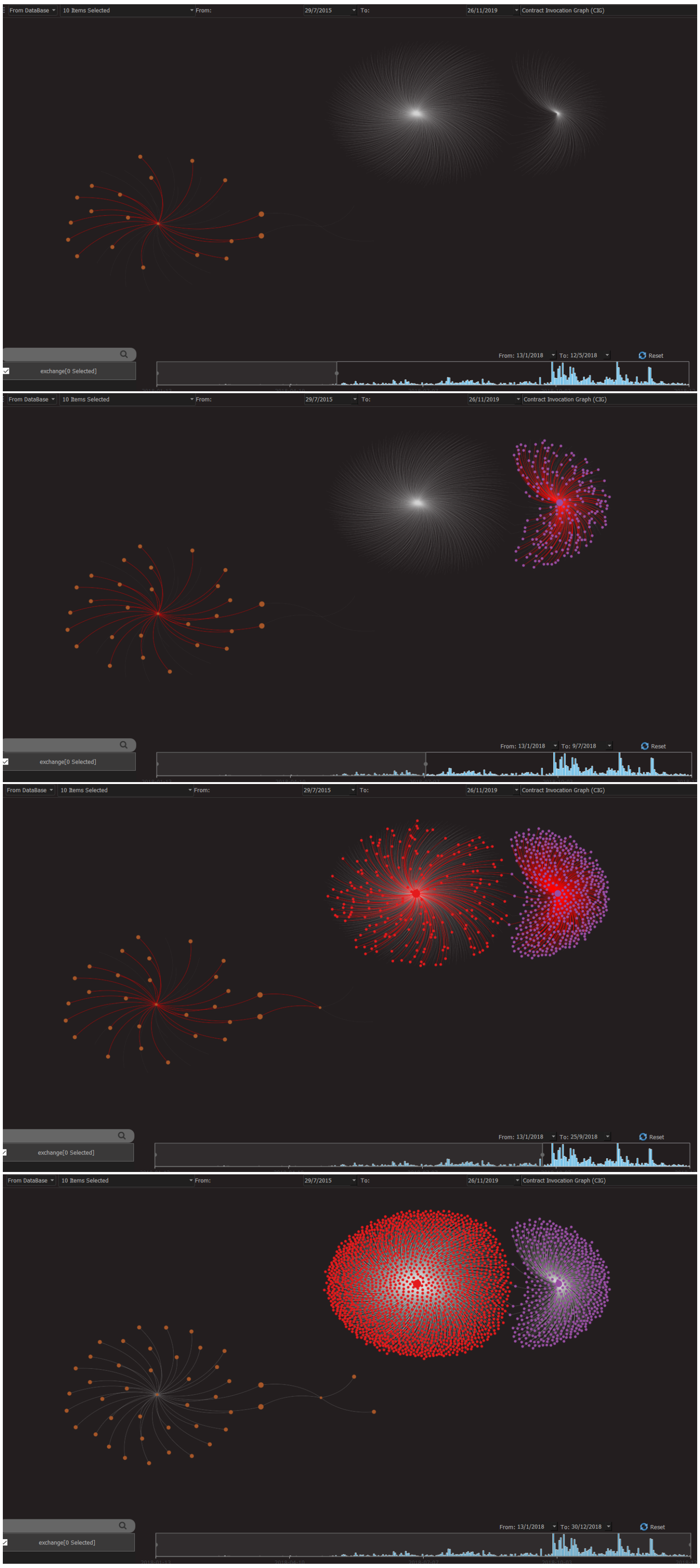
In the selected time interval covering highest activity levels in smart contract invocation by decentralized exchanges in 2018, Fig. 5.5 illustrates the distribution of contribution to these activity levels for different exchanges. Among them, the decentralized exchange "Allbit" as a contract had most transactions associated with it. During this period from Oct 2018 to Dec

2018, "Allbit" introduced Dai coin into its products on 12 Oct [5]. It was probably one of the reasons to attract high contract invocation activity in this period.

5.3 Evolutionary Study

One focus of evolutionary study is to analyze the growth of an individual in terms of its importance and relationships with other parties. In Fig. 5.5 it shows that "Allbit" contributed the highest transaction volume in smart contract invocation among decentralized exchanges in 2018, it is worth noting that it was also launched in 2018. Therefore, the CIG in four incremental time intervals are visualized to analyze the changes of its importance and relationships with others.

From Fig. 5.6, "Allbit" gained the highest degree centrality and formed the largest Louvain community among decentralized exchanges in 2018. "Allbit" was launched after the launch of "DEx.top" when "DEx.top" had certain degree of market share at that moment, then "Allbit" grew at a much higher speed than "Dex.top", eventually "Allbit" exceeded its competitor "DEx.top" in centrality and community measures after adding Dai currency as its product. In Fig. 5.5, it can also be observed that when "Allbit" reached peak transaction volume, "DEx.top" only had little growth [4].



From DataBase 10 Items Selected From: 29/7/2015 To: 26/11/2019 Contract Invocation Graph (CIG)

From: 13/1/2018 To: 12/5/2018 Reset

exchange[0 Selected]

From DataBase 10 Items Selected From: 29/7/2015 To: 26/11/2019 Contract Invocation Graph (CIG)

From: 13/1/2018 To: 9/7/2018 Reset

exchange[0 Selected]

From DataBase 10 Items Selected From: 29/7/2015 To: 26/11/2019 Contract Invocation Graph (CIG)

From: 13/1/2018 To: 25/9/2018 Reset

exchange[0 Selected]

From DataBase 10 Items Selected From: 29/7/2015 To: 26/11/2019 Contract Invocation Graph (CIG)

From: 13/1/2018 To: 30/12/2018 Reset

exchange[0 Selected]

Fig. 5.6: CIG of decentralized exchanges in incremental intervals

Conclusion 6

6.1 Major Contributions

The aim of this thesis is to provide the approach and tools for analyzing the evolution of Ethereum transactions. The approach consists of macroview and microview supplemented with appropriate tools.

For the macroview, it covers all the Ethereum transactions from 30 Jul 2015 to 25 Nov 2019 without downsampling, and groups the transactions by day instead of the entire period to increase accuracy.

For the microview, it is visualized by an application developed in this thesis in the form of graphs over time, this application allows users to select accounts and time intervals which enable more flexible graph generation, it is open-source to non-developers for direct implementation or to developers for further customization.

6.2 Future Improvements

Regarding the application program for graph visualization, its computing performance should be improved in the future to handle larger datasets in shorter time. For example, it can include more parallel computing and pre-calculated data.

For parallel computing, the independent components under computation are identified as much as possible, then they are processed concurrently by techniques such as MapReduce or matrix operation.

For pre-calculated data, it contains the smallest units which can be calculated independently, they remain relatively constant and are usually the results of repetitive calculations. They are identified as much as possible to reduce the time for repetitive calculations.

References

[1] Xing Bo and Marwala Tshilidzi. „Blockchain and Artificial Intelligence“. In: *Ssrn Electronic Journal* (2018) (cit. on p. 1).

[2] Ting Chen, Yuxiao Zhu, Zihao Li, Jiachi Chen, and Xiaosong Zhange. „Understanding Ethereum via Graph Analysis“. In: *ACM transactions on Internet technology* 20.2 (2020), pp. 1–32 (cit. on p. 6).

[3] Riverbank Computing. *PyQt5 Download*. `https://riverbankcomputing.com/software/pyqt/download5`, Year = 2020 (cit. on p. 17).

[4] Cryptogeek. *Compare Allbit vs DEx.top*. `https://cryptogeek.info/en/compare-exchanges/allbit-vs-dextop`, Year = 2019 (cit. on p. 31).

[5] Allbit Exchange. *Allbit Lists Dai (DAI)*. `https://medium.com/@Allbit/allbit-lists-dai-dai-ed2904f295c0`. 2018 (cit. on p. 31).

[6] *Exchanges Directory*. `https://etherscan.io/directory/Exchanges` (cit. on pp. 14, 30).

[7] Stefano Ferretti and Gabriele D’Angelo. „On the Ethereum Blockchain Structure: a Complex Networks Theory Perspective“. In: *Concurrency and Computation: Practice and Experience* (2019) (cit. on p. 5).

[8] M. Fleder, M. S. Kester, and Pillai S. „Bitcoin Transaction Graph Analysis“. In: *arXiv preprint* (2015), arXiv:1502.01657.

[9] A. Hagberg and D. Schult. *PyGraphviz*. `https://github.com/pygraphviz/pygraphviz`. 2020 (cit. on p. 17).

[10] Ethereum Homestead. *Account Types, Gas, and Transactions*. `http://ethdocs.org/en/latest/contracts-and-transactions/account-types-gas-and-transactions.html`. 2020 (cit. on p. 5).

[11] C. L. Lanum. *Visualizing graph data*. Shelter Island, NY: Manning, 2017 (cit. on p. 7).

[12] Yitao Li, Umar Islambekov, Cuneyt Akcora, et al. „Dissecting Ethereum Blockchain Analytics: What We Learn from Topology and Geometry of Ethereum Graph“. In: (2019).

[13]E. Meeks. *D3.js in action*. Shelter Island, NY: Manning, 2015 (cit. on p. 7).

[14]Jonas David Nick. „Data-Driven De-Anonymization in Bitcoin“. In: 2015 (cit. on p. 1).

[15]*Normal transactions VS. Internal transactions in etherscan*. `https://ethereum.stackexchange.com/questions/6429/nor-mal-transactions-vs-internal-transactions-in-etherscan`.

[16]PyInstaller. *PyInstaller Overview*. `https://github.com/pyinstaller/pyinstaller`, Year = 2020 (cit. on p. 18).

[17]Fergal Reid and Martin Harrigan. „An Analysis of Anonymity in the Bitcoin System“. In: *Security and Privacy in Social Networks*. New York, Springer, 2013 (cit. on p. 5).

[18]Michele Spagnuolo, Federico Maggi, and Stefano Zanero. „BitIodine: Extracting Intelligence from the Bitcoin Network“. In: vol. 8437. Mar. 2014, pp. 457–468 (cit. on p. 1).

[19]Melanie Swan. *Blockchain: Blueprint for a New Economy*. O'Reilly, 2015 (cit. on p. 5).

[20]VisPy. *VisPy: interactive scientific visualization in Python*. `https://github.com/vispy/vispy`. 2020 (cit. on p. 17).

[21]W Yougxin. *Exploring Ethereum Nodes and Transactions*. 2017 (cit. on p. 5).

[22]Xuanwu et al. Yue. „BitExTract: Interactive Visualization for Extracting Bitcoin Exchange Intelligence“. In: *IEEE transactions on visualization and computer graphics* 1 (2019), pp. 162–171 (cit. on p. 7).

[23]Peilin Zheng, Zibin Zheng, Jiajing Wu, and Hong-Ning Dai. „XBlock-ETH: Extracting and Exploring Blockchain Data From Ethereum“. In: *IEEE Open Journal of the Computer Society* (2020), pp. 95–106 (cit. on p. 11).