# Graph Based Visualisation Techniques for Analysis of Blockchain Transactions

Jeyakumar Samantha Tharani*, Eugene Yougarajah Andrew Charles†, Zhé Hóu*, Marimuthu Palaniswami‡ and Vallipuram Muthukkumarasamy*
*School of Information and Communication Technology, Griffith University
†Department of Computer Science, University of Jaffna
‡Department of Electrical and Electronic Engineering, University of Melbourne
jeyakumar.samanthatharani@griffithuni.edu.au

***Abstract*—Blockchain is a digital technology built on three main pillars: decentralization, transparency and immutability. Bitcoin and Ethereum are two prevalent Blockchain platforms, where the participants are globally connected in a peer-to-peer manner and anonymously perform trade electronically. The vast number of decentralized transactions and the pseudo-anonymity of participants open the door for scams, cyber frauds, hacks, money laundering and fraudulent transactions. It is challenging to detect such fraudulent activities using traditional auditing techniques, since they need sophisticated algorithms, more processing power and memory for complex queries to join combinations of tables. This paper proposes several algorithms to extract the transaction-related features from the Bitcoin and Ethereum networks and to represent the features as graphs. Moreover, the paper discusses how visualisation of graphs can reflect the anomalies and patterns of fraudulent activities.**

***Index Terms*—Bitcoin, Ethereum, smart contracts, graph models, anomaly detection, blockchain.**

## I. Introduction

Blockchain is a sub-set of distributed ledger technology that allows participants to perform various peer-to-peer network transactions. Bitcoin and Ethereum are two leading blockchain applications that facilitate participants to perform tasks such as online payment and storing digital currencies and other forms of digital assets, and executing smart contracts. The transactions between the participants are decentralised, anonymised and publicly visible to all participants who can access. This has significant business benefits, including greater transparency, enhanced security, improved traceability and efficiency, and faster transactions at reduced costs. On the one hand, the pseudonymous nature of the blockchain brings business transformation and a large number of investors. On the other hand, it opens a back door for bad actors to perform various malicious activities like scams, hacks, transactions related to illicit markets and money laundering [3]. Hence there is an urgent need to develop appropriate tools to analyse blockchain transactions for different applications.

The analysis of transaction data provides various opportunities for data mining researchers to explore the trading behaviours, wealth distribution, generative mechanism of transactions, and identify fluctuations in the financial market of cryptocurrencies. Representing blockchain transaction records as a flat table cannot accurately characterise the temporal and multiplex features of the transactions. Moreover, it will meet various challenges when performing data analysis in terms of processing power, time and memory. Due to these reasons, there is an increased interest in the graph-based visualisation and graph-based analysis [7] of blockchain transactions. The graph-based representation enriches the data by incorporating the relations between transactions and facilitate navigation through the transactions without the need for complex queries to join combinations of tables together as in the relational model. In a Bitcoin network, the graph-based visualisation supports data analysis related to link prediction, node classification, forensic investigation of the transaction and the relationship between entities [11], [5], [8]. Likewise, in the Ethereum network, graph-based visualisation is essential for characterising different transaction activities and investigating security issues such as smart contract commit fraud, vulnerability contracts that are deliberately developed to undermine the fairness, duplicate contracts, etc. [6], [7].

This paper proposes algorithms for extracting transaction-related features from Bitcoin-core and Ethereum, and presents the data as graphs. Moreover, this paper highlights various graph patterns related with the anomaly transactions such as ransomware, and illicit market to support the graph based analysis to identifying abnormal behaviours on the blockchain network, link prediction for coin flow and market trend of the cryptocurrencies.

The remaining sections of this paper are structured as follows: Section II states the methods for modelling different types of graphs for the transaction data available at the blockchain network, section III describes the experimental setup for feature extraction and graph modelling, section IV describes case studies related with the graph-based anomaly transaction pattern. Finally, the contribution of the research work is concluded in Section V.

## II. Graph Models of Blockchain Transactions

Blockchain transaction records contain much information. The addresses available at the input and output elements of Bitcoin transactions indicate a participant or an organisation (Eg: Bitcoin miners) related to the digital trades. Likewise, the from and to addresses at the Ethereum indicate the wallet address of the participant or an address of the smart contract.

There are several graph construction algorithms designed based on the various details of the transaction records. This research work propose a graph model $G(V, E, R)$ which contains the vertex $V(S, T)$ , and the directed edge $E \subseteq S \times T$ connecting source ($S$) and target ($T$) with the relation $R$. The label *BLOCK* indicates the blocks in the constructed graph, *TRANSACTION* representing the transactions stored in the block, *INPUT* define the input entries in the bitcoin transaction, *OUTPUT* define the output entries in the bitcoin transaction and the *COINBASE* is indicates coinbase transaction (transaction with no input). Likewise, the labels *EOATRANS* indicates the money flow transactions within the EOA accounts.

### A. Graph Models for UTXO Transactions

*1) **Transaction Graph**:* The research work [8], [9], [10] describes the need of transaction graph of Bitcoin transaction. The transaction graph $G_{trans}$ = ($S_{transId}$,$T_{transId}$,w) is constructed using the input and output fields of the transaction data. In this graph, each node represents the transaction hash of the source($S$) and target ($T$) transactions. Each directed edge E=$\{(s_i,t_i)$ where $s_i \in S_{transId}$, $t_i \in T_{transId})\}$ contains an amount (w: E $\rightarrow$ R) spent by $S$ and received by $T$. The transaction graph can be constructed only for the UTXO model transactions. If the value of *spent-transaction-hash* of the transaction $T$ and the *hash* value of the transaction $S$ are identical, then there is a directed edge, drawn from S to T with the details of the transferred amount, timestamp and the corresponding relation $R$. The Algorithm 1 gives a systematic approach to extract necessary features from the Bitcoin core dump for constructing a transaction graph.

**Input**: a list tnxList of transactions
**Output**: a transaction graph
`constructTransactionGraph(`*tnxList*`)`
**for** *i ∈ [1,size(tnxList)]* **do**
  $tnx_i \leftarrow$ tnxList[i]
  $hash_i \leftarrow$ hash of the $tnx_i$
  **for** *j ∈ [i+1, size(tnxList)]* **do**
    $tnx_j \leftarrow$ tnxList[j]
    $inputs_j \leftarrow$ list of inputs in $tnx_j$
    **for** *input* $\in inputs_j$ **do**
      $s_{hash_j} \leftarrow$ spent transaction hash of input
      **if** $s_{hash_j}$ ***equals to*** $hash_i$ **then**
        sNode $\leftarrow s_{hash_j}$
        tNode $\leftarrow hash_i$
        E(i,j) $\leftarrow$ transaction info
        V $\leftarrow$ (sNode, tNode)
        Add (V, E(i,j), $R_t$) to $G_t$
      **end**
    **end**
  **end**
**end**
**return** transaction graph $G_t$

**Algorithm 1: Constructing Transaction Graph.**

*2) **Address Graph**:* A participant in the blockchain network can be uniquely identified by the public key address. The address graph $G_{address} = (S_{pubkey}, T_{pubkey}, w)$ can be constructed for both UTXO and account-based models' transaction data. In this type of graphs the nodes represent the participants. A directed edge $E = \{(s_i, t_i)|s_i \in S_{pubkey}, t_i \in T_{pubkey})\}$ contains the details of the transaction (amount, timestamp, transaction fee) between a source $S$ and target $T$. The Algorithm 2 presents an approach for extracting features from raw data and to construct the address graph for Bitcoin transactions.

**Input**: A list tnxList of transactions
**Output**: An address graph
`constructAddressGraph(`*tnxList*`)`
**for** *i ∈ [1, size(tnxList)]* **do**
  $tnx_i \leftarrow$ tnxList[i]
  $hash_i \leftarrow$ hash of the $tnx_i$
  **for** *j ∈ [i+1, size(tnxList)]* **do**
    $tnx_j \leftarrow$ tnxList[j]
    $inputs_j \leftarrow$ list of inputs in $tnx_j$
    **for** *input* $\in inputs_j$ **do**
      $address_j \leftarrow$ address of the input
      $s_{hash_j} \leftarrow$ spent transaction hash of input
      **if** $s_{hash_j}$ ***equals to*** $hash_i$ **then**
        $address_i \leftarrow$ address of the output in $tnx_i$
        sNode $\leftarrow address_i$
        tNode $\leftarrow address_j$
        E(i,j)$\leftarrow$ transaction info
        V $\leftarrow$ (sNode, tNode)
        Add (V, E, $R_a$) to $G_a$
      **end**
    **end**
  **end**
**end**
**return** address graph $G_a$

**Algorithm 2: Constructing Address Graph.**

### B. Graph Models for Account-based Transactions

**Money Flow Graph (MFG)**

A MFG graph $G_{MFG}(S_{EOA}, T_{EOA}, w)$ is helpful to visualize the money flow between two externally owned accounts. Money flow can be an amount of Ether or tokens transferred between a source account($S$) and a target account($T$). The nodes in this graph are the EOA address of sources($S_{EOA}$) and targets($T_{EOA}$). The directed edge $E = \{(s_i, t_i)|s_i \in S_{EOA}, t_i \in T_{EOA})\}$ states the money flow between $S$ and the $T$. The weight $w$ of an edge indicates an amount of Ether or tokens transferred. An algorithm proposed for constructing the money flow transaction graph is given in the Algorithm 3.

## III. Experimental setup for feature extraction and graph construction

The proposed methods for constructing various graphs were tested with actual Bitcoin and Ethereum network data. For

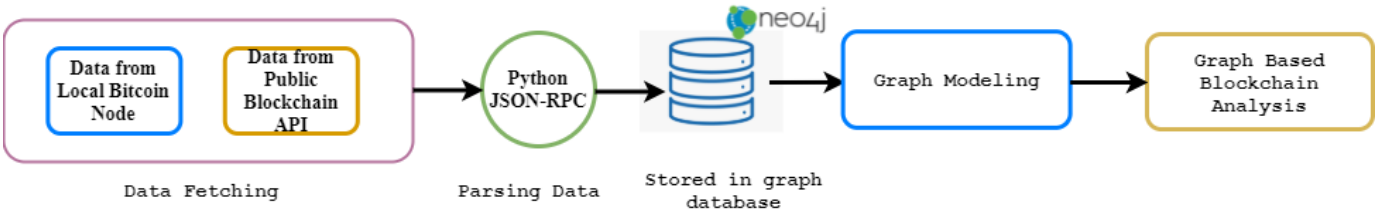


Fig. 1. Experimental Setup of the research work

**Input**: A list tnxList of money flow transactions
**Output**: The money flow transaction graph
`moneyFlowTransactionGraph(`*tnxList*`)`
**for** *tran* $\in$ *tnxList* **do**
  $S_{EOA} \leftarrow$ tran["from"]
  $T_{EOA} \leftarrow$ tran["to"]
  $w \leftarrow$ tran["value"]
  $E \leftarrow w$
  $V \leftarrow (S_{EOA}, T_{EOA})$
  **if** *tran is EOA based* **then**
    Add ($V$, $E$, MFT) to $G_m$
  **else**
    Add($V$, $E$, E721) to $G_m$
  **end**
**end**
**return** money flow transaction graph $G_m$

**Algorithm 3: Constructing MFG.**

the evaluation purposes, the transaction data of the Bitcoin downloaded from Bitcoin core. For this purpose we setup our local Bitcoin node, and it captured around 25.7 MB of data. It included $46,700$ blocks and the transactions corresponding with those blocks. The data related with Ethereum downloaded from the Ethereum public dataset XBlock.pro [12]. This contains the data related with the blocks, normal transactions (EOA), internal ether transactions, contract information, contract calls, ERC20 transactions and ERC721 transaction. Fig 1 shows the experimental setup of the research work. The data downloaded from the Bitcoin core is in binary format (.bat) since it cannot be directly used for graph construction. We use Bitcoin ETL [1], a python based library to convert the binary data into compressed JSON format. The JSON files were loaded and needed features are extracted. Then the graphs were constructed by applying the algorithms stated in section II-A and section II-B, finally the graphs loaded into Neo4j graph database for visualisation. The result for Bitcoin transaction and address graph shown in Fig 2 and the Ethereum money flow graph shown in Fig 3.

## IV. Graph Based Visualization of Anomaly Transactions

The graph models of the different blockchain transaction can help an analyser interpret the hidden relationships among the blocks, transactions, and participants in the blockchain network. Visualising blockchain transactions as graphs by using the algorithms proposed in section II is much appropriate to extract meaningful patterns than the traditional database representation. This will produce an informative dataset for different graph-based analysis [7] like anomaly detection, predict ransomware transaction, identification of illicit market-related trades, cryptocurrency's market trend for future investments, etc.

*1) **Visualizing Ramsomware Transactions**:* Ransomware transactions like CryptoLocker and WannaCry accumulate a large amount of Bitcoins from a specific user or an organisation and deposit to another anomaly address. The address graph of the bitcoin network described in section II-A and the money flow transaction graph of the EOA described in section II-B will help to visualise the coin flow (BTC/Ether) between two or more addresses. These algorithms are able to provide transaction records corresponding with a specific ransomware participant or an EOA. Let us assume that the ransomware participants have a pattern of addresses with the prefix '19oz'. To visualise the transactions corresponding with a specific pattern of addresses, we use the following queries: The query

```
MATCH (tx:Transaction)-[d:Deposite]-(i:Input)
MATCH (i)-[:unlocked]-(da:DepAddress) WHERE
da.address STARTS WITH '19oz'
RETURN tx,i
```

will return the transaction records related to the address with the prefix '19oz' in the Bitcoin network.
Likewise, the query

```
MATCH (b:Block)-[:EOA]-(tx:Transaction)
MATCH (tx)-[:EOATRANS]-(f:Sender) MATCH
(f)-[:MFT]-(r:Receiver)
WHERE r.address STARTS WITH '19oz'
RETURN b, tx, f, r
```

will returns a set of EOA addresses with the prefix '19oz'.

*2) **Visualizing Illicit market transactions**:* Illicit markets facilitate the sale/purchase of an illicit item using Bitcoin. Existing research [4] estimates that around $76 billion worth of illegal activities performed per year involves bitcoins (46% of bitcoin transactions), which is close to the scale of the US and European markets for illegal drugs. Another recent article states that the Bitcoin address "1HQ3Go3ggs8pFnXuHVHRytPCq5fGG8Hbhx" transferred 69,369 BTC to an unknown wallet. This address originates from the illicit market *Silk Road* [2].

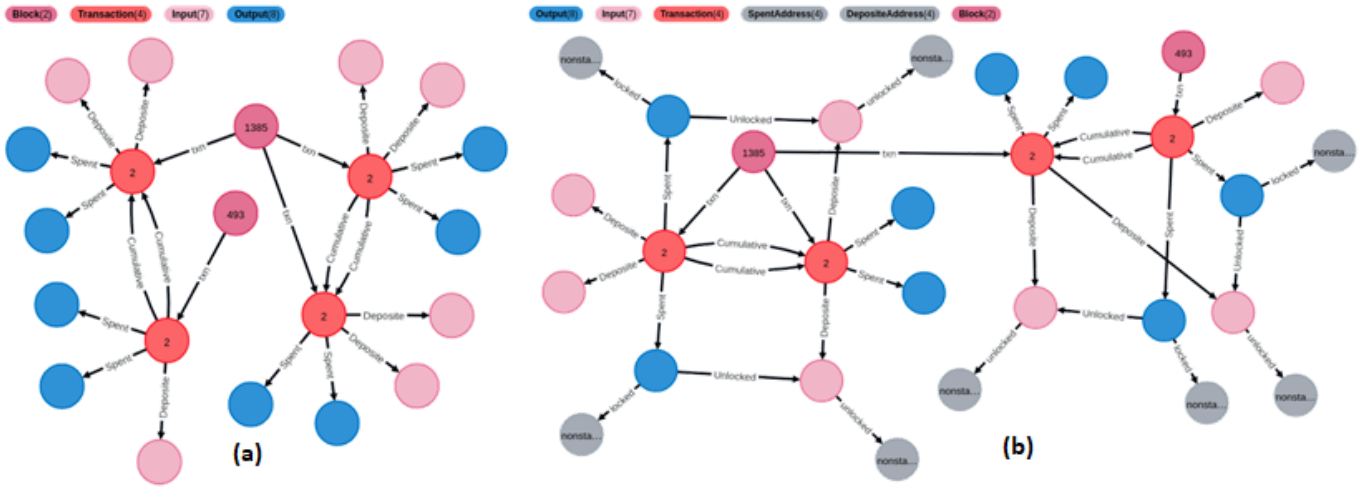


Fig. 2. Graphs for Bitcoin Data: (a) Transaction Graph, (b) Address Graph


Fig. 3. Money Flow Transaction Graph

The address graph of the bitcoin network described in section II-A and the money flow transaction graph of the EOA stated in section II-B facilitate to identify transaction related to specific illicit market addresses. The query `MATCH (tx:Transaction)-[:Spent]-(ou:Output) MATCH (ou)-[:locked]-(sa:SAddress) WHERE sa.address='1HQ3Go3ggs8pFnXu.....'` will return a set of transactions involved with a specific address.

## V. Conclusion

This paper focuses on effective graph representation of blockchain transactions and visualizing anomaly transaction graphs. The paper proposed three algorithms to extract data from the blockchain network and to generate three different type of graphs namely transaction graph, address graph and money flow graph. From the constructed graphs, we demonstrated that the address graph and the transaction graph of the Bitcoin and the money flow transaction graph of the Ethereum are more suitable to visualise the patterns related with the anomalous transactions. Effectiveness of the algorithms were tested by using the actual Bitcoin core data and XBlock (Etherium Public API) datasets. The graph patterns visualize via Neo4j and demonstrated how these patterns reflect the anomaly transactions. Finally, much of this work can be seen as data extraction and feature engineering for a graph-based dataset of blockchain transactions. Such a dataset will be valuable in other tasks such as machine-learning-based analytics, which is our immediate future work.

## References


[1] "BitcoinETL," https://github.com/blockchain-etl/bitcoin-etl, 2019.

[2] "Silk Road Transaction," https://www.coindesk.com/nearly-1b-in-bitcoin-moves-from-wallet-linked-to-silk-road, 2020.

[3] "Cryptocrimes," https://go.chainalysis.com/2021-Crypto-Crime-Report.html, 2021.

[4] S. Foley, J. R. Karlsen, and T. J. Putniņš, "Sex, drugs, and bitcoin: How much illegal activity is financed through cryptocurrencies?" *The Review of Financial Studies*, vol. 32, no. 5, pp. 1798–1853, 2019.

[5] B. Haslhofer, R. Karl, and E. Filtz, "O bitcoin where art thou? insight into large-scale transaction graphs." in *SEMANTiCS (Posters, Demos, SuCCESS)*, 2016.

[6] T. Hu, X. Liu, T. Chen, X. Zhang, X. Huang, W. Niu, J. Lu, K. Zhou, and Y. Liu, "Transaction-based classification and detection approach for ethereum smart contract," *Information Processing & Management*, vol. 58, no. 2, p. 102462, 2021.

[7] V. Patel, L. Pan, and S. Rajasegarar, "Graph deep learning based anomaly detection in ethereum blockchain network," in *International Conference on Network and System Security*. Springer, 2020, pp. 132–148.

[8] F. Reid and M. Harrigan, "An analysis of anonymity in the bitcoin system," in *Security and privacy in social networks*. Springer, 2013, pp. 197–223.

[9] M. Weber, G. Domeniconi, J. Chen, D. K. I. Weidele, C. Bellei, T. Robinson, and C. E. Leiserson, "Anti-money laundering in bitcoin: Experimenting with graph convolutional networks for financial forensics," *arXiv preprint arXiv:1908.02591*, 2019.

[10] J. Wu, J. Liu, Y. Zhao, and Z. Zheng, "Analysis of cryptocurrency transactions from a network perspective: An overview," *arXiv preprint arXiv:2011.09318*, 2020.

[11] C. Zhao and Y. Guan, "A graph-based investigation of bitcoin transactions," in *IFIP International Conference on Digital Forensics*. Springer, 2015, pp. 79–95.

[12] P. Zheng, Z. Zheng, J. Wu, and H.-N. Dai, "Xblock-eth: extracting and exploring blockchain data from ethereum," *IEEE Open Journal of the Computer Society*, vol. 1, pp. 95–106, 2020.