Journal of Parallel and Distributed Computing 180 (2023) 104730

Contents lists available at ScienceDirect

Journal of Parallel and Distributed Computing

journal homepage: www.elsevier.com/locate/jpdc

Why blockchain needs graph: A survey on studies, scenarios, and solutions

Jie Song [a,*], Pengyi Zhang [a], Qiang Qu [b,c], Yongjie Bai [c], Yu Gu [d], Ge Yu [d]

[a] *Software College, Northeastern University, Shenyang, 110819, China*
[b] *Shenzhen Institute of Advanced Technology, Chinese Academy of Science, ShenZhen, 518055, China*
[c] *Blockchain Lab, Huawei Cloud Tech Co., Ltd, ShenZhen, 518101, China*
[d] *School of Computer Science and Engineering, Northeastern University, Shenyang, 110819, China*

ARTICLE INFO

Article history:
Received 23 December 2022
Received in revised form 17 April 2023
Accepted 12 June 2023
Available online 19 June 2023

Keywords:
Blockchain
Blockchain applications
Blockchain platform
Graph algorithm
Graph computing
Graph-integrated blockchain platform

ABSTRACT

The popularity of blockchain platforms and their applications in industry and academia keeps rising. The multifarious requirements stimulate another technique, graph data and algorithms, to join the blockchains; thus, studies, scenarios, and solutions about graph-related blockchains have emerged. This paper aims to see whether the state-of-the-art studies satisfy the applications through a comprehensive survey on graph-related blockchains. To answer why a blockchain needs graphs in general, we analyze literature about blockchain and graph, as well as use cases on the application-oriented and graph-related scenarios collected from practical blockchain projects. The paper summarizes three graph-related blockchain studies: graph algorithms for blockchains, graph data in blockchains, and graph applications on blockchains. Based on these summarization, it figures out the gaps between the studies and the applications, that is, few of studies natively integrate graph computing into a blockchain. Here, the "graph integration" means processing on-chain graph data, which contains blockchain information, in a real-time, distributed, and consensual manner. We propose the prospect of the Graph-integrated Blockchain Platform (GiBP for short), and explain why a GiBP is inevitable, the challenges of a GiBP, and the GiBPs' main functions and features people expect for future research.

1. Introduction

A blockchain is a distributed ledger where data and transactions are not under the control of any third party [64]. In recent years, blockchains have been developing rapidly with an exponential increment of the data volume. The global blockchain market has been valued at USD 4.67 billion in 2021, and it is predicted to be USD 163.83 billion in 2029 [11]. With this trend, industries and academies have started developing new blockchain systems, including public and consortium chains. In particular, big companies are interested in consortium blockchain development, such as the Hyperledger Fabric of IBM [32] and the AWS of Amazon [1]. In comparison, researchers focus on both the public and the consortium chains, such as the well-known public chain Bitcoin and Ethereum, as well as the consortium chain Tether and Steem [23,27].

As a data structure, graphs have significant advantages in describing the correlation between data [57]. The massive explosion in data volumes has made graph computing increasingly popular as a tool for processing large-scale network data over the past decades [50], including the blockchain. Therefore, industries and academies have also developed many graph computing frameworks and platforms. For example, Huawei [31] and Ali [2] concentrate on the study of graph computing service platforms, and typical graph computing frameworks are Pregel [49], GraphLab [46], and PowerGraph [26].

"Blockchain + Graph" grows efficiently. State-of-the-art studies have reported on graph-related blockchain scenarios, such as graph algorithms for blockchains, graph data in blockchains, and graph applications on blockchains. For examples, Lv et al. [47] designed a variant of PageRank algorithm to make the blockchain consensus more credible. Abay et al. [3] and Kumar et al. [36] abstracted the Bitcoin network as a blockchain graph data structure for predicting bitcoin's price and understanding how its evolution is affected by social and anti-social tendencies. Guidi et al. [27] created the Steemit blockchain application for evaluating sub-graphs capturing social and monetary aspects. However, why blockchain needs

* Corresponding author.
E-mail addresses: songjie@mail.neu.edu.cn (J. Song), 2201353@stu.neu.edu.cn (P. Zhang), qiang.qu@siat.ac.cn (Q. Qu), baiyongjie3@huawei.com (Y. Bai), guyu@mail.neu.edu.cn (Y. Gu), yuge@mail.neu.edu.cn (G. Yu).

https://doi.org/10.1016/j.jpdc.2023.104730
0743-7315/

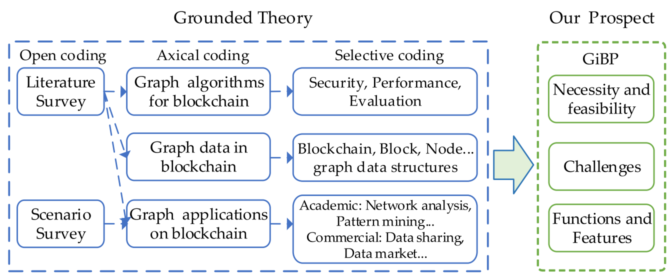


Fig. 1. Research method and our prospect as the road map of our study.

graph? Concretely, what features exist in state-of-the-art studies on graph-related blockchains and the graph applications running on blockchain platforms, are there the gaps between them? what are the expected solutions for the gaps? These questions are interesting and have not been well-addressed yet.

This paper firstly aims to answer the above questions. We offer a new perspective on the graph-supported blockchain platform, as the target system, by analyzing literature and real-world application scenarios. To understand why and how to support graph in a blockchain, we use the grounded theory [24] to make the survey. The grounded theory contains open coding, axical coding, and selective coding part. In the open coding part, we confirm the properties of the concept: literature survey and scenarios survey. In the axical coding part, we focus on the dimensions of the concept and the properties, including the graph algorithms for blockchains, the graph data in blockchains, and the graph applications on blockchains. In the selective coding, we focus on the detailed categories and summarize the whole concept to propose the prospect on GiBP. The "graph algorithms for blockchains" concludes the different functions of blockchains enhanced by graph algorithms, such as security, performance and evaluation; the "graph data in blockchains" defines the graph data hidden within blockchains, such as blockchain graph and node graph; the "graph applications on blockchains" summarizes the blockchain applications conducted through graph computing, such as network analysis in academic studies and data sharing in commercial cases. The left part of Fig. 1 shows the details about the research methods.

On the one hand, our literature analysis focuses on 51 papers on both blockchain and graph. We report the data and algorithms of graph-related blockchains. Also, we summarize, classify and refine them to make readers clearly know the prior works. For example, we classify the graph data in blockchains and refine them to seven graphs. On the other hand, our application scenarios analysis focuses on the application-oriented and graph-related scenarios collected from practical blockchain projects. We report data, participators, conditions, processes, and expected results of these scenarios. Also, what are the system features if it supports the above scenarios? The literature and scenarios analysis help clarify the research questions (RQs) mentioned in Section 2.1.

Based on the surveys, we draw the gaps between the state-of-the-art studies and application scenarios. We argue that applications prefer to run on a Graph-integrated Blockchain Platform (GiBP for short). Most of academic and all the commercial applications need GiBP for processing on-chain graph data in a real-time, distributed, and consensual manner. However, few academic studies focus on such platform well. For example, no one proposes a "graph integrated Hyperledger" natively supporting graph computing in a consortium chain. To propose the perspective of a GiBP, we make a definition, integrating the gaps and features of state-of-the-arts; then, we draw its necessity, feasibility, and challenges from the three dimensions; finally, we suggest the functions and features of the GiBP from the aspects of graph creation, graph storage, and graph processing in transaction and query. Right part of Fig. 1 shows our prospect on GiBP.

To this end, the overall contributions of this research are as follows:

(1) Summarizing and classifying prior works, such as the existing purposes of graph algorithms for blockchains, and academic graph applications on blockchains;
(2) Summarizing and refining prior works, such as refined definitions of graph data in blockchains, and paradigms abstracted from commercial graph applications on blockchains.
(3) Analyzing and drawing the gaps between the studies and the applications, that is, lack of processing on-chain graph data in a real-time, distributed, and consensual manner.
(4) Proposing the prospect of the Graph-integrated Blockchain Platform (GiBP for short), explaining the GiBP's features, challenges, and expected operations, which are potential research topics, from the system perspective.

The rest of this paper is as follows. Section 2 introduces our survey methods, including research questions (RQs) and how to collect the literature and application scenarios. Sections 3, 4, and 5 analyze and summary the state-of-the-art studies from the aspects of the graph algorithm for blockchains, the graph data in blockchains, and graph applications on blockchains, respectively. Section 6 defines GiBP, and then discusses indispensable features, challenges, and expected operations of GiBP as the research prospect. Section 7 concludes the paper.

2. Survey methods

This section introduces the main research questions of the survey, and explains our survey methods: survey on literature and application scenarios, and methods we collect these literature and scenarios.

2.1. Research questions

According to the design science [61], the paper starts with a general question of "Why blockchain needs graph", which is subsequently divided into four Research Questions (RQs). The paper answers the RQs through literature and scenario surveys, i.e., analyzes evidences and makes summaries. Research questions include:

RQ1: What are the studies on graph-related blockchains (answered in Section 3.4 and Section 4.8)?
RQ2: What are the features of blockchain platforms that the graph-related applications require (answered in Section 5.3)?
RQ3: What are the gaps between graph-related studies on blockchains and graph-related applications on blockchains (answered in Section 5.3)?
RQ4: What are the expected solutions for the gap, and the main focuses of the solutions from a research perspective (answered in Section 6.4)?

2.2. Literature

We conduct a preliminary search about blockchain and graphs using the Google Scholar and Web of Science with the keywords "blockchain AND graph." In Google Scholar, the keywords are chiefly matched in the body of papers instead of the abstract, and the search results contain the patents and research reports. Also, many IoT studies mentioned graph technologies and blockchain platforms. They are not our main focus. Therefore, the search keywords are "blockchain AND graph NOT IoT." In Web of Science, we

Table 1
All related works and their catalogs.

Dimensions	Approaches	Count	References
Graph algorithms for blockchain	Security	5	[43,47,67,69,70]
	Performance	2	[19,59]
	Evaluation	2	[25,55]
Graph data in blockchain	Blockchain Graph	7	[3,7,22,36,44,56,58]
	Block Graph	3	[20,62,63]
	Node Graph	7	[10,18,27,28,52,60,73]
	Transaction Graph	13	[8,12,13,15,17,23,35,38,41,45,53,60,75]
	Smart Contract Graph	5	[10,15,21,38,39]
	Token Graph	2	[16,38]
	Channel Graph	2	[9,34]
Graph application on blockchain	Network analysis	21	[5,7,10,15–18,23,27,28,30,36,38,44,45,52–54,60,68,74]
	Patten mining	3	[41,56,68]
	Feature Prediction	2	[3,72]
	Transaction Semantics Analysis	2	[13,65]
	Optimality	2	[4,9]

search the same keywords in the title, abstract, and keywords. The search scope is "Article AND Meetings."

The initial number of retrieved documents amounted to around 430 publications. We selected 51 papers as state-of-the-art works which contain interesting graph-related blockchains, according to the following rules:

- Including English-written peer-reviewed papers published in journals and proceedings of conferences in recent seven years;
- Excluding studies that blockchain is not the major research technology;
- Excluding studies that graph is not a technology but a noun;
- Excluding studies that focus on researching DAG (Directed Acyclic Graph), which contain the same goal and the same graph structure to improve the blockchain's scalability;

In the end, we obtained 51 related papers and grouped them into three dimensions and corresponding approaches, as shown in Table 1.

2.3. Scenarios

The published commercial applications of graph-related blockchains are almost news and lack of details. Hence, we manage to collect the application scenarios from the industry. We take the scenarios in China, collected by Huawei Cloud Ltd., as examples. The scenarios in other markets collected by other corporations, even we cannot access them entirely, may be similar with our selection. We adopt the method of "semi-structured expert interview" to collect the scenarios. We also invited eight experts to participant our interview. Among them, three are consultants, three are Product Managers (PMs), and two are marketing staffs. All participants have at least three years' work experience on blockchains.

To obtain the experienced scenarios, we conducted a guideline of the interview, developed from the literature survey on graph-related blockchain. The guideline consists of three parts: start-up, trigger, and follow-up questions. The start-up introduced the purpose of the interview to the experts and helps us understand their job position, background, and related experience. In the trigger part, we provided these experts with a description of identified indicators and asked them to provide us with at least three scenarios of graph-related blockchain, including relevant application fields, suitable application scenarios, and specific application cases. The follow-up questions part collected new ideas from the experts and further details on the use cases.

After interviewing the experts, we collect their description on the application scenarios. The selected scenarios are mostly from government/enterprise blockchains, as a kind of consortium blockchain initiated by the governments/enterprises. They have three major applications: 1) Enterprises services, such as electronic signatures and licenses, one-stop service, and remote bidding; 2) Government regulations such as judicial evidence storage, traceability regulation, funds supervision, hazardous chemicals regulation, project management, and Open Economic Zone administration; 3) People's Livelihood protection such as Fair and open license-plate lottery, time banking, health records, and electronic medical records. The selected scenarios cover data sharing, data market, administrative decisions, and funds supervision cases.

3. Graph algorithms for blockchains

Graph algorithms that enhance blockchain's functions or performance have two types: structure-oriented and analysis-oriented algorithms, defined as follows.

Definition 1 *(Structure-oriented algorithm).* Structure-oriented graph algorithms for blockchains either introduce graph structures and operations, or apply graph features and rules, to blockchains. The graph structure could be explicit or implicit. If the graph structure is explicitly contained in the blockchain data, users may query or traverse the structure to retrieve the results directly. If the graph structure is explicitly contained in the blockchain data, users may define and build the structure first.

For example, the "cross-closed undirected graph for blockchain authentication" in [70] focuses on the graph query algorithm, and the latter is the "graph isomorphism for zero-knowledge proof algorithm for blockchains" in [69] focuses on the graph creation algorithm.

Definition 2 *(Analysis-oriented algorithm).* Analysis-oriented graph algorithms for blockchains employ the known graph data analysis algorithms to optimize the blockchains' performance.

For example, the mentioned graph data analysis algorithms are PageRank in [47] and the centrality algorithm in [55].

Algorithms, but not hardware, guarantee the security and performance of blockchains. So that researchers may leverage graph algorithms to achieve better security and performance. Besides, graph algorithms also help to evaluate security and performance. Hence, in state-of-the-art studies, security, performance, and evaluation are three purposes of graph algorithms for blockchains. This section introduces the existing graph algorithms for blockchains according to the above definitions, their purposes, and RQ1's answers concluded from them.

3.1. Security

Blockchain security requirements vary, including identity management, attack prevention, and privacy protection. State-of-the-art studies enhance blockchains' security through structure-oriented and analysis-oriented graph algorithms.

For the structure-based graph algorithms, the studies first proposed novel graph structures, explained how the structures contribute to blockchain security, and then designed the corresponding operations on these structures. Lin et al. [43] design a Transitively Closed Undirected Graph (TCUG) structure to bind a digital signature to a physical identity safely. TCUG's vertexes are blockchain nodes and edges are signatures. It can identify a node only through its signatures. They combine three Probabilistic Polynomial Time (PPT) algorithms with TCUG. Compared with other undirected graphs, TCUG can support both dynamic adding and edge deletion, which can improve the security of the blockchain. Besides, Wang et al. [70] construct multiple topological authentications by topological graph structures to enhance the blockchain's identity validation. They design three construction algorithms based on the topological graph structure, in order to build up the multiple topological authentications. Moreover, encryption techniques such as zero-knowledge proof based on graph isomorphism are adopted to improve privacy protection in [69]. The common grounds of these studies are that the graph structures consist of an existing graph structure with several customized or well-known algorithms. Besides, their graph structures hold the additional information but not the blockchain data.

For the analysis-based graph algorithms, the studies changed the critical mechanism of blockchains, especially the consensus protocol, to improve its security. Thakur and Breslin [67] optimized the graph edge coloring algorithm into a PBT protocol; as a result, the PBT protocol can prevent the DoS and Eclipse attack. Lv et al. [47] designed a variant of PageRank to calculate the credit matrix, and the matrix can find the node with the highest credit as the delegate node to make the consensus more credible.

3.2. Performance

Only a few studies improve the performance of blockchains through graph algorithms. Their main focus is network performance but not computing performance due to the network being modeled as a graph easily.

For the structure-based graph algorithms, Shahsavari et al. [59] propose a random graph model to solve the impact of crucial blockchain parameters on Bitcoin's overall performance. They model the Bitcoin overlay network using an Erdos-Renyi model to generate connected random graphs. Then, they implement their theoretical model with the network simulator OMNet++. The model improves the block propagation delay and traffic overhead.

For the analysis-based graph algorithms, Essaid et al. [19] design a customized version of the PageRank algorithm for deeper graph analysis processing to get higher network stability for Bitcoin nodes. They collect Bitcoin data and assemble data into a topological graph structure to gain knowledge on the Bitcoin network size, and the network stability in terms of well-connected Bitcoin nodes.

3.3. Evaluation

In a blockchain, security and performance could be evaluated through specific indicators. For example, the blockchain's decentrality is for its security, and blockchain users' centrality is for the performance of the blockchain network. Therefore, the evaluation of blockchain adopts analysis-oriented algorithms. Gochhayat et al. [25] employ the Betweenness and Closeness centrality algorithm to identify and quantify decentrality in blockchain-based systems for blockchain security. Pontiveros et al. [55] propose a "mint centrality" algorithm to measure the centrality of the Bitcoin transaction graph, which can show the load performance of the Bitcoin blockchain.

3.4. Summary (RQ1)

This subsection explains our summaries according to the survey on "graph algorithms for blockchains." Currently, graph algorithms are helpful for improving blockchains' security, performance, and evaluation. The studies in this direction are not numerous; however, novel ideas and new objectives may emerge along the studying. Whether graph algorithms are introduced to blockchains in a structure-oriented or analysis-oriented manner, the graph algorithms are not "in" blockchains, on the contrary, they are executed outside the blockchain environment. Their results are applied to blockchains, and their graphs contain no on-chain blockchain data. Whether the studies of "graph algorithms for blockchains" are for security, performance, or evaluation, none targets the runtime mechanism and business logic of blockchains.

4. Graph data in blockchains

This section summarizes the state-of-art research on graph data in blockchains to see whether blockchains need graphs in this direction. We give symbols that occur in this section before we discuss categories of graph data in blockchains. Given a blockchain system, then:

(1) **Symbol** V_U. V_U is a set of users who own an account and join in at least one transaction in the blockchain. For example, V_U represents a set of addresses in the Bitcoin network, and a set of Externally Owned Accounts (EOAs) in the Ethereum network.
(2) **Symbol** V_{SC}. V_{SC} is a set of smart contracts being called or destroyed by elements of V_U.
(3) **Symbol** V_T. V_T is a set of validated transactions in the blockchain.
(4) **Symbol** V_B. V_B is a set of blocks ordered by the timestamps in the blockchain.

Besides, all studies involved graph data apply on-chain or off-chain data processing, such as creation, storage, and analysis. A graph is a data structure. In this paper, we provide a uniform classification of these studies based on their data structure. On-chain data structure and off-chain data structure are defined as follows:

Definition 3 *(On-chain data structure).* The graph data are stored in the blockchain instead of external storage, or the graph data are processing on the blockchain instead of external computing engines. For example, the block graph data are created and stored in the block, and the channel graph data are processed in the blockchain.

Definition 4 *(Off-chain data structure).* The graph data are constructed from the blockchain dataset, or the graph data are analyzed out of the blockchain. For example, the node graph data are constructed and analyzed out of the blockchain.

According to the literature analysis, we refine the graph data in blockchains into seven types. Table 2 shows these types, the supporting literature, and the original name and application of the graph data in each literature.

Table 2
Seven categories of graph data in blockchains and original names in literature.

Graph type	On-chain/ Off-chain	Graph and application in literature
Blockchain graph	On-chain	Chain graph is for fighting against fake news in [58]. The nodes are news consumers, social media and press media. Edges are verified news.
	Off-chain	Blockchain graph is for predicting Bitcoin's price in [3]. Nodes are addresses. Edges are transactions. Ethereum blockchain graph is for understanding how sharding would affect Ethereum in [22]. Nodes are accounts or contracts. Edges are interactions involving nodes. Bitcoin network graph is for understanding how the social and anti-social tendencies in the user base of Bitcoin affect its evolution in [36]. Nodes are wallet addresses. Edges are exchanges of bitcoins between wallet addresses. Ethereum blockchain graph provides basic quantitative insights related to the activity on the Ethereum public network in [7]. Nodes are addresses. Edges are transactions. Blockchain graph is for detecting illicit nodes on blockchain networks in [56]. Nodes are user nodes. Edges are transactions. Directed K-order subgraph is for accurately characterizing the temporal and multiplex features of the edges in [44]. Nodes are nodes in the blockchain. Edges are transaction information that contains weights and timestamps.
Block graph	On-chain	Block graph is for improving blockchain's throughput in [62]. Nodes are atomic units(AUs) of blocks. Edges are the order of execution of AUs. Block graph is for improving blockchain's throughput by sharding transactions in [20]. Nodes are atomic units(AUs) of blocks. Edges are the order of execution of AUs. Block graph is for improving blockchain's throughput by adding concurrency to the execution of AUs in [63]. Nodes are atomic units(AUs) of blocks. Edges are the order of execution of AUs.
Node graph	Off-chain	Cluster graph and address graph are for noting various ways to analyze the Bitcoin network in [60]. Nodes are addresses in the network. Edges are transactions from source address to target address. Bitcoin user graph is for exploring the local topology and geometry of the Bitcoin network in [52]. Nodes are users' wallet addresses. Edges are exchanges of bitcoins between users' wallets. User-user graph and user-contract graph are for studying the evolutionary behavior of Ethereum transactions in [10]. Nodes are Externally Owned Accounts(EOAs) or contracts. Edges are the transfer of Ether in EOAs-EOAs or EOAs-Contracts. User interaction graph is for unveiling crucial knowledge concerning blockchain users in [27]. Nodes are users. Edges are facts that two connected users interacted at least once. User graph is for identifying user information and unpredictable characteristics by utilizing user behavior and association analysis in [73]. Nodes are users. Edges are interactions of users in exchanging bitcoins. Follower-Following graph is for understand how the social and the economic aspects of BOSMs intertwine and influence each other in [28]. Nodes are users. Edges are token "follow" relationships between nodes. User graph is for discovering the evolution of users' wealth in [18]. Nodes are addresses. Edges are transactions between nodes.
Transaction graph	On-chain	Cross-validate transaction graph is for reducing the transaction delays in [12]. Nodes are transactions. Edges are the verification between two transactions. Transaction directed acyclic graph is for providing a semantic transaction function in blockchain in [13]. Nodes are transactions' states and witness. Edges are the relation between a state and a witness.
	Off-chain	Ethereum transaction graph is for behavior analysis on Ethereum from the perspective of graph mining in [8]. Nodes are accounts. Edges are temporal transactions. Tether transaction graph is for identifying the main actors that play a leading role in the network in [23]. Nodes are token holders. Edges are transactions in which token changes from one participant to another. Haloed second-order subgraph is for exploring Ethereum transactions' evolution in [45]. Nodes are addresses. Edges are transactions. TraceNet and TransactionNet are for comparing social networks with blockchain networks in [38]. Nodes are all possible users and smart contracts. Edges are successful tracings of all addresses from/to. K-order subgraph is for mining the abundant patterns encoded in bitcoin transactions in [41]. Nodes are addresses. Edges are transactions. Transaction graph is for noting various ways to analyze Bitcoin's network [60]. Nodes are transactions and addresses. Each edge is an output that connects two transactions. Large transaction graphs of energy trading are for enhancing the security and privacy of the EV user's information in [17]. Nodes are energy nodes. Edges are transactions. ATGraph is for detecting the Ethereum phishing account in [35]. Nodes are user nodes. Edges are transactions Edge Aggregated Graph is for detecting Ethereum phishing accounts in [75]. Nodes are accounts. Edges are flows of transaction funds. Money flow graph is for identifying what methods, transaction signatures, and data can be adopted to understand the transactions in [53]. Nodes are users. Edges are money flows between users. Money flow graph is for characterizing Ethereum via graph analysis in [15]. Nodes are every externally owned accounts(EOAs). Edges are transactions between EOAs.
Smart contract graph	Off-chain	Contract-Contract graph for studying the evolutionary behavior of Ethereum transactions in [10]. Nodes are smart contracts. Edges are the functions "create", "call", "suicide" between pair-wise contracts. Call graph of smart contracts is for quantifying the smart contract risks in [21]. Nodes are smart contracts. Edges are the callings between smart contracts. ContractNet is for comparing social networks with blockchain networks in [38]. Nodes are smart contracts. Edges are the callings between smart contracts. Dependency graph is for detecting re-entrancy vulnerabilities through fuzzy testing in [39]. Nodes are smart contracts. Edges are the callings between smart contracts. Contract creation graph and contract invoke graph are for characterizing Ethereum via graph analysis in [15]. Nodes are the set of EOAs and smart contracts. Edges are the callings between smart contracts.

(*continued on next page*)

Table 2 (*continued*)

Graph type	On-chain/ Off-chain	Graph and application in literature
Token graph	Off-chain	TokenNet is for comparing social networks with blockchain networks in [38]. Nodes are users. Edges are the token transactions. Token creation graph and token holder graph are for characterizing the token creator, holder, and transfer activity in [16]. Nodes are accounts. Edges are token transactions between accounts.
Channel graph	On-chain	Off-chain topology graph is for improving the payment ability in [34]. Nodes are users. Edges are payment channels. Channel network graph is for improving the payment ability by a framework in [9]. Nodes are users. Edges are payment channels.

4.1. Blockchain graph

The blockchain graph contains all the transactions or smart contract calling records. It is commonly for offering a specific analysis sight of a blockchain. As an overview of blockchain, the blockchain graph is generally for machine learning [3].

Definition 5 *(Blockchain graph).* A blockchain graph is a directed graph $G_{BC} = (V, E)$, where $V = V_U \cup V_{SC}$. E has two conditions.

- If $E = \{(v_i, v_j)|v_i, v_j \in V_U\}$, E is a directed transaction between v_i and v_j. Nodes V and edges E could have extra attributes, such as labels for nodes, also amount, weight, and timestamp of transaction for edges.
- If $E = \{(v_i, v_j)|v_i \in V_U, v_j \in V_{SC}\}$, E is an interaction from a caller v_i to a smart contract v_j. Edges E can also have a weight W to show the total number of one or more transactions called along the edge.

The blockchain graph is a “leader” of other graphs because it contains the complete information compared with the others. Besides, most of the blockchain graph is an off-chain data structure whose data is downloaded from the blockchain and analyzed in off-chain processing by a third party.

4.2. Block graph

Nowadays, the block graph is for improving the scalability of blockchain. In a concurrent environment, the “concurrent miners” create a block graph that helps validators re-execute the same smart contract transaction concurrently and deterministically [62].

Definition 6 *(Block graph).* A block graph is a directed graph $G_B = (V, E)$ in the blockchain, where $V = V_B$, $E = \{(v_i, v_j)|v_i, v_j \in V_B\}$. Edge E represents that the child block of v_i is v_j. Nodes V could have extra attributes, such as timestamps and identities.

The block graph is an on-chain data structure. It is generated from the block data and accessed by miners during the mining process.

4.3. Node graph

The node graph is a graph data of blockchains with tokens. It is commonly for researchers to discover the Bitcoin and Ethereum transaction patterns among Bitcoin and Ethereum addresses. Generally, it shows the “follow” relationship between users.

Definition 7 *(Node graph).* A node graph is commonly a directed graph $G_N = (V, E)$ in the blockchain, where $V = V_U$, and E represents token (such as bitcoin and ETH) exchange between users’ or wallets’ addresses. The timestamp and amount of the transaction are two attributes of E.

All the node graphs are off-chain data structures whose data are retrieved from online blockchain datasets. Pattern mining on the graph is also off-chain processing. Inspired by the node graph, researchers started to design the token graph for focusing on the transition of tokens.

4.4. Transaction graph

The transaction graph is for transaction analysis through big data approaches. It highlights the relationship between transactions.

Definition 8 *(Transaction graph).* A transaction graph is a directed graph $G_{TR} = (V, E)$, where $V = V_U \cup V_T$. E has two conditions.

- If $E = \{(v_i, v_j)|v_i, v_j \in V_U\}$, E is an interaction from a caller v_i to a smart contract v_j. Edges E can also have a weight W to show the total number of one or more transactions called along the edge.
- If $E = \{(v_i, v_j)|v_i, v_j \in V_T\}$, E represents the order in which smart contracts are invoked from v_i to v_j. Nodes are annotated with the address and block number created by the smart contract. Each E can also represent the token flow instead of the token graph.

The transaction graph is an on/off-chain hybrid data structure. As an on-chain data structure, it supports on-chain transaction validation instead of block validation. As an off-chain data structure, it has no information about the smart contract. Moreover, transactions are created in channels, so the transaction graph contains the channel information.

4.5. Smart contract graph

The smart contract graph is for analyzing the creation of transactions and tokens, as well as the calling relationship between smart contracts.

Definition 9 *(Smart contract graph).* A smart graph is a directed graph $G_{SC} = (V, E)$, where $V = V_U \cup V_{SC}$. E has two conditions.

- If $E = \{(v_i, v_j)|v_i \in V_U, v_j \in V_{SC}\}$, E is an interaction from a caller v_i to a smart contract v_j. Each E also shows how the token flows from the smart contract to the v_i’s account address.
- If $E = \{(v_i, v_j)|v_i, v_j \in V_{SC}\}$, E represents the “call” relationship from v_i to v_j. Nodes are annotated with the address and block number created by the smart contract, and the token flow is replaced by the smart contract’s functions, such as “call,” “create,” and “kill.”

The smart contract graph only contains information about smart contracts compared to the blockchain graph. As an analytical graph, the smart contract graph is an off-chain data structure.

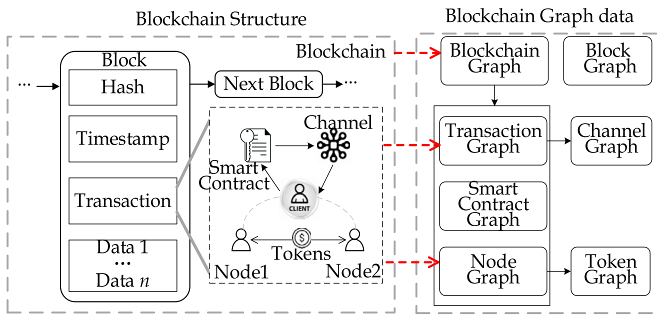


Fig. 2. Relationships between blockchain structure and graph data; and those among the graph types.

The data are downloaded from the blockchain and analyzed in off-chain processing. Moreover, the smart contract connects with the token graph because it also represents the token flow.

4.6. Token graph

The token graph has a similar structure to the node graph, but it is designed for token analysis. It reveals the generation and circulation characteristics of tokens.

Definition 10 *(Token graph).* A token graph is a directed graph $G_{TO} = (V, E)$, where $V = V_U$, highlighted the token of a user, and $E = \{(v_i, v_j)|v_i, v_j \in V_U\}$. Each E represents the order of the creative relationship. E also has a weight W to indicate the node holds W shares of the token.

The token graph is a special version of the node graph, and is generated later than the node graph. It is also created and accessed in off-chain processing. In a token graph, the node only holds the user's token, while in a node graph, it holds more user information than the token. In other words, token graphs are the specialization of node graphs.

4.7. Channel graph

The channel graph evolves from the transaction graph. It only contains the blockchain's channel information, and supports the computation and analysis of the user's payment process.

Definition 11 *(Channel graph).* A channel graph is a directed graph $G_C = (V, E)$, where $V = V_U$, and $E = \{(v_i, v_j)|v_i, v_j \in V_U\}$. Each E represents a payment channel of the blockchain.

Channel graphs are on-chain data structures because the payment process occurs online.

4.8. Summary (RQ1)

This subsection explains our summaries according to the survey on "graph data in blockchains." Seven types of graph data are summarized in the studies of "graph data in blockchains". Based on the basic graph definition like $G = (V, E)$, each graph data in blockchains has a distinct definition of its V and E. Moreover, these graphs are all "connected" and generated from a blockchain structure. Among these graphs, the blockchain graph contains complete information, while the transaction graph, the node graph, and the smart contract graph have partial information. That is, the blockchain graph covers these three graphs. In addition, the channel and token graphs are refined transaction graphs and the node graphs. Fig. 2 shows their relationships.

Table 2 shows that the applications on graph data vary. It is not only for the academics, such as network analysis and prediction, but also for business, such as fake news detection, payment analysis, and energy trading analysis. With further studies, more and more graph data and applications will emerge.

According to Table 2, the graph data in blockchains can be accessed in two manners: on-chain and off-chain processing. On the one hand, the conditions of on-chain processing for the graphs are: **1)** applying to channel graphs or block graphs; **2)** participating in the distributed computing process, consensus process, or the validating process in the blockchain; **3)** retrieving blockchain online. On the other hand, the conditions of off-chain processing for the graphs are: **1)** generating graphs from a blockchain dataset which are off-line data of a blockchain and primarily for data mining and machine learning; **2)** using graph data to analyze or predict through big data processing approaches. As a comparison, the studies of "graph algorithm for blockchains" all take the off-chain processing by graph algorithms.

5. Graph application on blockchains

Graph applications on the blockchain are two-fold: academic applications and commercial ones. These applications apply graph computing to implement the main purpose, such as pattern mining and feature prediction, or the main business logic, such as data sharing and funds supervision. This section summarizes the state-of-the-art graph applications and their computing paradigm.

5.1. Academic

In the academic field, graph applications are mainly for Bitcoin and Ethereum. They are closely related to the graph data mentioned in Section 4 and Table 2 because graph algorithms always deal with graph data. We catalog them into network analysis, pattern mining, feature prediction, transaction semantics analysis, and optimality according to their purposes.

5.1.1. Network analysis

Treating a blockchain as a network, some studies try to retrieve interesting information from the network. We roughly group such applications as network analysis. It is the most frequent graph application on blockchains. Table 3 lists the details of each study. These studies propose the graph data representing the Bitcoin or Ethereum network, then perform graph algorithms on the network for specific information. In Table 3, several studies focus on Bitcoin's and Ethereum's network evolution through their customized graph data, such as [10], [45] and [74]. Several studies treat a blockchain network as a social network and compare Bitcoin with Ethereum to analyze the blockchain's social features, such as [38] and [23]; Several studies explore the roles and activities over blockchain networks, such as [18] and [15]; The rest studies perform novel graph operations, for instance, the traverser and visualization on a blockchain network, such as [16] and [68].

5.1.2. Pattern mining

Pattern mining is a further study with graph technologies for novel blockchain patterns than the network analysis's. Li et al. [41] propose a k-order graph model for mining the rich coding patterns in Bitcoin transactions. These patterns help researchers analyze the coding rules of Bitcoin, and make Bitcoin prediction easier. Poursafaei et al. [56] propose a SIGTRAN-Feature extraction algorithm for mining the patterns of illegal activity in a blockchain transaction network. These patterns help researchers find illegal activities in blockchains easily. Tharani et al. [68] design a transaction graph and define a higher-order proximity algorithm between two EOAs for mining the patterns of the communities in Ethereum. These

Table 3
Academic applications as network analysis.

Platform	Ref	Graph algorithm/computing	Objective and conclusion of analysis
Bitcoin	[30]	PageRank and Closeness	Analyze the trend of Lightning Network's efficiency and anti-attack performance. It shows that the success rate of payment routing is low and is decreasing over time for a large amount of transactions on routing analysis, and .the current LN relies heavily on some important nodes.
	[60]	Computational Graph Analytics and Graph Pattern Matching	Summarize the various applications and technologies that perform data analysis on the Bitcoin blockchain, including Average Fees per Transaction and High Value Transactions over time.
	[36]	Network measurement tools	Explore the local topology and geometry of the Bitcoin network during the first decade of its existence. It shows that The edge density is low in both the directed graph and the undirected graph for the period 2009-2020 compared to social networks. Unlike social networks, it has no giant LSCC but follows properties of "scale-free" networks.
	[53]	Key reuse, graph connections, and graph overlap	Link relevant information while parsing the blockchain and show the relationship between lightning network channels identified on the blockchain. It shows that at least 75% of all P2WSH transactions are Lightning transactions, and some of the channels can be deduced from the blockchain analysis.
	[54]	Open source graph-aware cryptocurrency analytics platform	Estimate the potential income of spammers and analyze the patterns of currency flows in the Bitcoin. It shows that sextortion spamming is a lucrative business and spammers will likely continue to send bulk emails that try to extort money through cryptocurrencies.
	[5]	Calculate connectivity and degree assortativity	Analyze the evolution of the entire Bitcoin transaction graph and quantify the evolution of the key structural properties of the Bitcoin. The network exhibits a two-orders-of-magnitude larger diameter, sparse treelike communities, and an overwhelming majority of transitional or intermediate accounts with incoming and outgoing edges but zero cumulative balances.
	[18]	Clustering algorithm and other mathematical analysis	Time Evolution analysis of the Bitcoin network, verification of the "rich get richer" hypothesis, and detection of nodes that are critical to network connectivity. It shows that the diameter of the BITCOIN network is much larger than the one of social networks and that the degree distributions have some spikes for some specific values.
Bitcoin& Ethereum	[68]	Algorithms to extract the transaction-related features	Analyze how the visualization of graphs can reflect the anomalies and patterns of fraudulent activities. It proves that the ransomware participants have a pattern of addresses with the prefix '19oz'.
Ethereum	[52]	Data analysis tools	Make topological data analysis and functional data deep blockchain data analysis by data analysis tools. It concludes that the edge density is low in both the directed graph and the undirected graph for the period 2009–2020 compared to social networks.
	[74]	Temporal graph measurement	Investigate the evolutionary nature of Ethereum interaction networks from a temporal graphs perspective. Including SPLIT pattern, CHAIN pattern and MERGE pattern.
	[10]	Gini index of networks	Analyze the Gini indexes of the transaction graphs and the user wealth of the Ethereum. It concludes that Ethereum is found to be very unfair since the very beginning, "the rich is already rich".
	[38]	Traditional and customized graph algorithms	Discuss four graph data's similarities and differences with social networks and the Web. It shows that blockchain networks are disassortative, having very low transitivity.
	[45]	Link prediction algorithm	Explore the evolution mechanism of Ethereum transactions. The local and microscopic structure of Ethereum networks is starshaped, and the transaction frequency of addresses has a great impact on the evolution of Ethereum transaction relationships.
	[44]	Several flexible temporal walk strategies for random-walk	Characterize the temporal and multiplex features of the edges by analyzing the directed k-order sub-graph of the Ethereum. It shows that temporal information and multiplicity characteristic of edges are indispensable for accurate modeling and understanding of Ethereum transaction networks.
	[7]	Centrality algorithms such as Betweenness	Study the robustness of the network and the distribution of activities within users from the view of graphs. It shows that the creation of the Ethereum Alliance consortium has been a game changer in the use of the technology.
	[16]	Finding entity algorithm	Propose an algorithm to discover potential relationships between tokens and other accounts. It founds that the users 0x3e and 0x1a trade with 0x5b 15,557 times; User 0xcc and user 0xb3 trade with each other 16,554 times; User 0xe7 trades with himself 3,012 times, including various tokens.
	[15]	Centrality and degree distribution algorithms	Analyze the main activities in the Ethereum. For examples, it can reveal the identity of all 15,416 accounts belonging to the WCC via graph-based deanonymization. It also observes that users purchase many tokens without burning, thus lots of created contracts are not self-destructed.

Table 3 (*continued*)

Platform	Ref	Graph algorithm/computing	Objective and conclusion of analysis
Steemit Blockchain	[27]	In-degree and out-degree distribution algorithms	Evaluate three subgraphs capturing social and monetary aspects. It shows that half of the users have a passive social behavior and that 80% of the users tend to accrue economic value.
	[28]	Graph analysis tools	Evaluate the characteristics of the Steemit graph to understand how aspects of BOSMs intertwine and influence each other. It shows that Steemit is an ultrasmall world, and the richest users are not the most socially active.
Tether blockchain	[23]	Centrality and assortativity algorithms	Conduct the social network analysis to identify the main actors playing a leading role in the network. It concludes that the Tether transaction network does not enjoy the Smallworld property, and the cryptocurrency exchanges are the nodes with the greatest centrality.
Permissioned blockchain	[17]	Energy Trading Rank (ETR) algorithm	Use ETR to analyze the large transaction graphs of energy trading. It concludes that the overall incentive gain for EVs with maximum, truthful and committed participation increases and thus promotes active participation in the system.

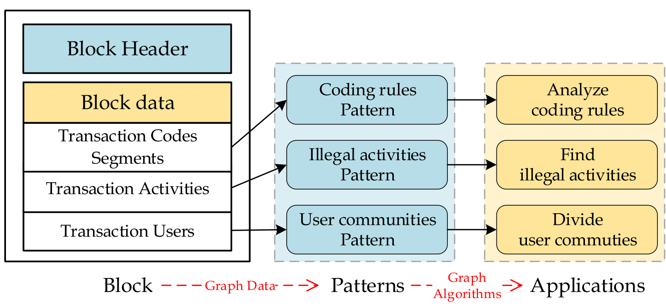


Fig. 3. State-of-the-art patterns in blockchains.

patterns help researchers clearly know the user community structure in Ethereum. Fig. 3 schematically shows the above patterns in blockchains. In summary, pattern mining, as a graph application on blockchains, summarizes features through novel graph algorithms and graph data visualization.

5.1.3. Feature prediction

Besides analysis and mining, prediction is also a typical application. Researchers adopt graph technologies to predict the development trend and the risk probability of blockchains. Yu et al. [72] propose an attack graph of blockchains and make predictions for preventing the DoS attack. Abay et al. [3] propose a Bitcoin graph to build a machine-learning model for predicting Bitcoin's price. The token price prediction may be widely off due to the influence of subjective factors. Researchers may also focus on predicting the other perspective, such as smart contracts and channels.

5.1.4. Transaction semantics analysis

In the blockchain, each transaction contains full details. The semantic information resides implicitly or explicitly in these details. Several studies focus on transaction semantics analysis to increase the blockchain's trustlessness and security. Sopek et al. [65] present the technological foundations on which an ecosystem of semantic data objects can be implemented on the latest blockchain platforms; and design an efficient hashing of the graph data structures to support the semantic analysis. As a result, the semantic output makes the blockchain more credible; it demonstrates that building a Knowledge Representation system on the blockchain is feasible. Cachin et al. [13] point out the blank of guaranteed blockchain semantic transaction models, which has no formal model of the transaction semantics that a blockchain is supposed to guarantee. They propose a Transaction-Directed Acyclic Graph (TDAG) to fix the blank. Without the blank, the blockchain becomes more credible from the transaction validation aspect.

5.1.5. Optimality

Optimality is a classical question in software systems. Researchers adopt graph technologies in blockchains to optimize blockchain resource allocation and service payment price policy. Abdellatif et al. [4] propose a graph-based matching model to allocate the optimal resource between the servers' provisions and the blockchain miners' demands. Avarikioti et al. [9] design a channel network graph and examine the optimal graph structure and expense allocation to maximize the Payment Service Provider (PSP) profit. They introduce the centrality graph algorithms to the blockchain. As a result, they maximize the blockchain's social optimality from the payment price aspect.

5.2. Commercial

This subsection discusses the commercial applications with three aspects: scenario, paradigm, and use case. As mentioned in Section 2.3, application scenarios are collected through semi-structured interviewing on experts. The proposed paradigm is a group of common elements and their interaction for each scenario. It involves five elements: *graph, participators, conditions, algorithm,* and *transactions*. They work together in blockchain transaction processing, following the same interaction pattern. For a scenario described as the paradigm, it has specific graph type and structure. The participators create, maintain, and access the graph. When it runs as a transaction, the algorithm is executed on the graph to see whether the conditions are satisfied. If they are, the transaction is confirmed; otherwise, the transaction is aborted. Fig. 4 shows the paradigm in a generalized scenario and how it works with transaction processing in blockchains. The use case is an instantiated paradigm of a scenario. Under the paradigm, this section briefly introduces four commercial scenarios and the corresponding use case.

5.2.1. Data sharing

Data sharing scenario commonly appears in blockchains of government affairs.

Scenario: The data for sharing is restrained and authorized so a blockchain can manage the data items and their metadata fairly and openly. When data is shared, the graph application examines the authority of executing data sharing and records the data sharing information on the block if the authority examination passes.

Paradigm:

- **Transaction**: the data sharing record.

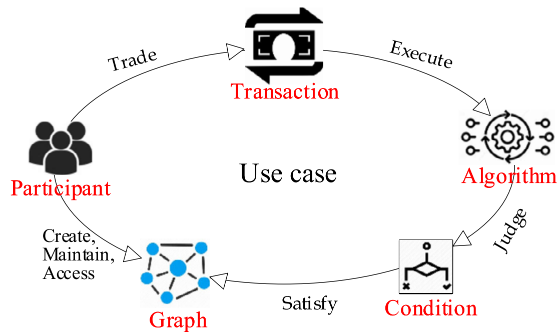


Fig. 4. The paradigm in a generalized case and how it works with the transaction processing in blockchains.

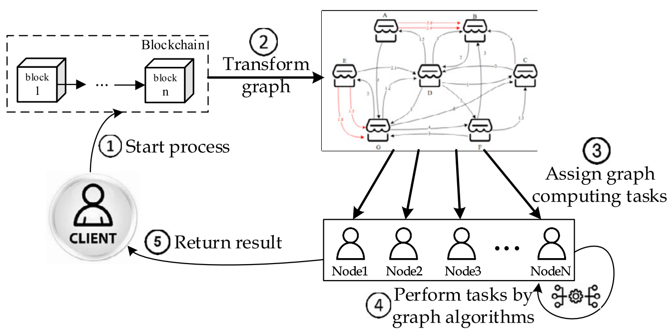


Fig. 5. Data sharing use case.

- **Graph**: department data dependency graph whose nodes are the departments and the edges are the data dependency features between two nodes. The government data items can explicitly or implicitly contain dependency relationships between departments. The dependency relationships are sharing, referring, describing, expanding, and governing dependency.
- **Participator**: the departments of the government.
- **Condition**: the sharing initiator's and the sharing receiver's Closeness centrality satisfies a predefined comparative relationship.
- **Algorithm**: the Closeness centrality algorithm.

Use case: This case aims to offer an authenticated governmental data-sharing function. The government blockchain obtains the above dependency relationship from the block data and generates a "Department data dependency graph." When the department applies for data dependence, the government blockchain can use the Closeness algorithm to determine whether there is a conflict in data sharing; and decide whether to authorize the application. For example, the platform would reject the data citing requests from a department because of a risk of authorization extension dependency, if the department cites data whose referring dependency Closeness is greater than the governing dependency Closeness. Fig. 5 shows the data-sharing process.

5.2.2. Data market

The data market has been a rising commercial scenario in recent years. It commonly appears in commercial blockchains.

Scenario: Commercialization and marketization of data elements is an essential application in the commercial blockchain. Enterprises, companies, and institutions are data suppliers who hold data and data demanders who need data. The blockchain stores the metadata of data in markets and transactions between suppliers and demanders.

Paradigm:

- **Transaction**: the data trading record.
- **Graph**: the data element transaction graph whose nodes are data suppliers and demanders; edges are the information of data trading transactions between nodes.
- **Participator**: the data supplier and data demander.
- **Condition**: the data demanders bidding with the data supplier.
- **Algorithm**: the Betweenness algorithm.

Use case: This case aims to offer a bidding function to the data demanders. Commercial blockchains can obtain information such as the data supplier, the data demander, the transaction cost, and the change of data ownership from the block data; and then generate the "data element transaction graph" based on the block data. When demanders bid for data, the smart contract for mining the value of data elements adopts the Betweenness algorithm to pricing data elements. In the data market, data trading is available for everybody. The betweenness computes the graph centrality and the significance of the nodes, which is one of the dominant factors in pricing the data element. It retrieves the most active and valuable data suppliers whose data element price is deserved to increase. The blockchain could retrieve bidding relationships among data suppliers and demanders on time from the "data element transaction graph", and eventually price data elements.

5.2.3. Administrative decisions

The administrative decision scenario commonly appears in a market platform or a business union stimulated by the platform or union manager.

Scenario: The platform manager aims to create a "business-to-business belt" and push all the companies' development through the service collaboration blockchain. The companies provide marketing services to each other through the blockchain. The manager can make an incentive mechanism to boost the market growth.

Paradigm:

- **Transaction**: service collaboration information.
- **Graph**: the collaboration value graph, whose nodes are companies and edges are service values between nodes.
- **Participator**: companies.
- **Condition**: the company's rank value is above the threshold.
- **Algorithm**: the PageRank algorithm.

Use case: This case aims to help the manager regularly find valuable companies and reward them. The service collaboration blockchain can obtain service collaboration information such as service object, service quality, service scale, service frequency, and service profit from the block data; and then generate a "collaboration value graph" based on the block data. The blockchain can analyze the essential and isolated nodes in the "collaboration value graph" through graph analysis algorithms, such as the PageRank algorithm. The manager may reward the company for serving others better if its rank value exceeds the threshold.

5.2.4. Fund supervision

Fund supervision scenario commonly appears between traders and regulators.

Scenario: It is the most significant goal for traders to use financial funds in a compliant, orderly, diversified, and market-oriented manner. This scenario helps the regulators discover the traders' illegal or improper use of funds.

Paradigm:

- **Transaction**: the fund transferring record.
- **Graph**: the fund usage graph, whose nodes are the traders and edges are the fund flow information between traders.
- **Participator**: traders.
- **Condition**: the subgraph of the "fund usage graph" matches with a predefined specific graph structure.
- **Algorithm**: the Motif algorithm.

Use case: This case aims to detect illegal or improper capital flow. The investor regularly issues funds to traders, and traders use the funds on a blockchain under the supervision of regulators. The blockchain records all capital flows between traders and generates a "fund usage graph" from the block data. It reports illegal or improper transferring as soon as the transferring request is submitted as a transaction. Suppose the involved subgraph matches a specific graph structure, such as frequent fund exchanges between two traders. It implies that the traders may perform fake trading for embezzlement or tax evasion.

5.3. Summary (RQ2&RQ3)

This subsection summarizes the survey on "graph applications on blockchains", including academic and commercial applications, and answers RQ2 and RQ3. It analyzes the features of blockchain platforms that the applications require, and the gap between state-of-the-art studies and such application requirements.

5.3.1. For RQ2

For academic applications, a few academic applications required a blockchain tightly integrated with graph computing capability; most of them required loosely attached. The integrated one means graph data and graph algorithm instances have the same lifestyle as the block data and transaction processing. They follow the same blockchain computing paradigm. In comparison, the loosely attached one does the opposite. The key reason why an integrated one is unnecessary is the skipped consensus; that is, the graph computing results are assumed to be credible. For example, network analysis, feature prediction, and optimality applications involve more system-level logic, so the credibility of results is not vital. In contrast, pattern mining and transaction semantics analysis applications involve more business logic, so the credibility of results is critical.

For commercial applications, the paradigm in commercial applications requires a blockchain platform that support on-chain graph analysis capability and provides interfaces for developing graph-based smart contact. To make on-chain graph analysis, the graph analysis of commercial application should process the on-chain graph data, which should be processed in real-time. For example, the betweenness calculation with consensus process in the smart contract of data market scenario. It should obtain the latest transaction graph data and pricing the data element in the real-time.

5.3.2. For RQ3

According to the survey, studies and applications have gaps on lacking of processing on-chain graph data in a real-time, distributed, and consensual manner. It is explained as the following points:

- All the studies of "graph algorithms for blockchain" and most of studies of "graph data in blockchain" are off-chain graph processing; however, the graph-related applications requires on-chain one. The off-chain graph processing has higher scalability, but reduce the graph data freshness and can not guarantee the real-time processing, which reduce the accuracy of graph computing.
- Other on-chain processing studies of "graph data in blockchain" need to store the graph data in the block. However, they require the extra on-chain storage and extra computing resource. With these additional resources, the scalability of the current blockchain platform is challengeable to satisfy the graph-related applications.
- The applications need a suitable consensus mechanism but the studies on consensual graph processing on the blockchain is still blank, especially in the decentralized environment. The trustlessness of graph computing can not be guaranteed. The decentralized consensus approach on graph is a challenge.
- The customization of graph-based smart contracts is also challengeable to develop graph data and algorithm in a blockchain platform.

6. Graph-integrated blockchain platform

Sections 3, 4, and 5 answer the former three research questions proposed in Section 2.1, and summarize the graph algorithms, graph data, and graph application in blockchains, respectively. However, a blockchain platform can not natively support these data, algorithms, and applications. In other words, it requires third-party components to retrieve data from blockchains, run the algorithms, and implement the applications. Nevertheless, so far now, none of the platforms, such as HyperLedger, can support such a manner. A GiBP is expected to support the graph-related application.

Definition 12 *(Graph integrated blockchain platform).* A GiBP is an expected platform for solving gaps between existing blockchain-related studies and applications. It has indispensable features, challenges, and expected operations. Currently, no platform can satisfy the features, solve the challenges, and implement the operations of GiBP. Therefore, GiBP represent a research prospect.

This section discusses GiBPs as research prospects. We explain why integrating graphs into a blockchain platform is inevitable, what are the challenges of GiBPs, and what are GiBPs' major functions and features people expect. As a prospect, GiBP has not been developed yet. The challenges and the expected functions of GiBP are hardly partitioned into an exact architecture. Therefore, the GiBP's architecture are not designed in this section, but Fig. 6 in Section 6.3 shows the sketch of GiBP including layers, key functions, and techniques. The sketch can explain the GiBP from the commercial application perspective.

6.1. Indispensable

There are sufficient studies and applications to show that blockchain needs graphs. On the one hand, the "graph + blockchain" solution can satisfy part of the requirements. Despite the sophisticated graph algorithm and data structure, the solution is as simple as "retrieving graph data from blockchains to off-chain storage" and "running graph algorithms outside the blockchain." Conversely, the rest requirements, especially those commercial applications following the paradigm in Fig. 4, show that a GiBP keeps three indispensable features of blockchain and graph technology: real-time processing, fresh data, and trustlessness.

6.1.1. Real-time processing

Real-time processing on the graph means results should be calculated immediately so that the transaction processing would not be interrupted, due to the results being the critical condition of the transaction. For example, in the commercial use cases mentioned in Section 5.2.1, the data sharing departments of the government should be processed in the real-time when the department request is acquired by the blockchain. It requires real-time processing by Closeness algorithm with the historical data dependencies. The necessity of real-time processing in a GiBP should satisfy Assumption 1.

Assumption 1. the use case requires a real-time smart contract, and the real-time smart contract involves results of graph algorithms.

On the one hand, as a data-based transaction system, a blockchain is a dynamic network with constantly increasing data. So, real-time data processing, such as the consensus and the validating processes, could reflect the up-to-date situation [37]. On the other hand, according to our literature review and commercial survey, graph algorithms meet both the tuning requirements and business logic of blockchains. The blockchain also contains various graph data implicitly and explicitly [38]. Therefore, the smart contracts satisfy Assumption 1.

In a GiBP, graph algorithms are tightly integrated into the platform so that smart contracts can invoke them efficiently without accessing the third part component. However, an alternative way is that smart contracts access the results pre-computed by non-integrated algorithms. This approach possibly ensures real-time processing but with the considerable cost of data pre-computation.

6.1.2. Fresh data

Fresh data for the graph means only results calculated on the up-to-date data are valid and adoptable. For example, in the commercial use cases mentioned in Section 5, their smart contracts require a condition reflecting both historical and current data. The necessity of fresh data in a GiBP should satisfy Assumption 2.

Assumption 2. the graph algorithms rely on the up-to-date blockchain states or block data. That is, the results of the algorithms can not be pre-computed or enumerated.

The graph data mentioned in Section 4 are mostly on-chain data. It dynamically changes with the blockchain and block data. So that the results based on these data are unpredictable. When analyzing the dynamic graph, we must consider both the feature recognition of the graph at a particular time and how the graph has evolved over time. If real-time updates do not accommodate graph computing, inferences become less informative, and graph computing becomes valueless [33]. Therefore, dynamic graph computing also requires real-time processing, which makes Assumption 2 true.

According to the former two assumptions, it is better to integrate both graph data and algorithms into blockchain platforms for performance reasons. The performance of an external graph computing system may not be comparable if it extracts data from and feeds the result back to the blockchain. Even if it were comparable, its results are non-consensual.

6.1.3. Trustlessness

Trustlessness is an indispensable feature of the blockchain. Trustless computing of the blockchain deliberately breaks with centralized methods and promotes decentralized solutions for correctly executing transactions. Unfortunately, the graph-supported blockchain platform is a centralized solution because the external graph computing system can hardly be decentralized [71].

Trustlessness does not imply lacking of trust, but changing a form in how trust is managed, including the consensus and validation process of the blockchain [66]. Thus, a GiBP should handle graph computing results through the consensus and validation process. The necessity of trustlessness in a GiBP should satisfy Assumption 3.

Assumption 3. graph computing is in a trustlessness environment. Thus, consensus and validation are essential processes before the computing results are applied.

Any computation and results should be consented and validated in transactions and smart contracts [42], as does graph computing. It makes Assumption 2 true. It is costly even if we employ an external graph computing system and develop its own consensus and validation process, while tightly integrating into the blockchain platform is a far better solution.

6.2. Challengeable

Even though the requirements of graph-integrated block-chain emerge, a mature GiBP is still far behind. We believe decentralization, scalability, and customization are three challenges through our literature review and commercial survey, due to the lacking of convinced solutions in state-of-the-art studies.

6.2.1. Decentralization

Trustlessness is the primary reason why blockchain is decentralized. When integrating graph computing into a blockchain, it has to abandon the centralized and stand-alone computing environment; and switch to the decentralized and distributed one [29]. On the one hand, any existing experiences in sophisticated graph computing frameworks, such as distributed graph databases (e.g., Neo4j-APOC [51], GraphX [48], GraphLab [46], Pregel [49]), and graph oriented state DBMSs (e.g., SPARQL-based) are not adaptable anymore. They all have efficient graph storage and computing capabilities, but can not guarantee the decentralized environment due to the existence of malicious nodes. On the other hand, most known solutions of graph algorithms for blockchains, as discussed in Section 3, are off-chain algorithms with a centralized or stand-alone environment; their non-consensual results are for performance but not the business transaction. Most graph data in blockchains, as discussed in Section 4, are off-chain data; Several on-chain graphs, such as transaction graphs and block graphs, are processed at clients in a stand-alone manner. Therefore, how to support graph computing in the decentralized blockchain environment is a new topic.

6.2.2. Scalability

Scalability optimization is a hot topic of blockchain research, and existing studies adopt Transactions Per Second (TPS) to measure the scalability of blockchain [14]. Horizontal scaling refers to expanding the scale of the system by adding more peers to ensure or improve TPS [40]; vertical scaling refers to upgrading peers (scale-up) to increase TPS accordingly [6]. In a non-GiBP, each peer deals with the same transaction data and reaches a consensus; commonly, the consensus algorithm is the PoS for the public blockchain and the PBFT for the consortium blockchain. In a GiBP, distributed graph computing with consensus brings new difficulties to the scalability.

On the one hand, graph data storage, such as on-chain storage or off-chain storage, distributed storage or centralized storage, unique storage or replicated storage, persistent storage or in-memory storage, affect horizontal and vertical scalability. On the other hand, decentralization and consensus on graph computing also bring new scalability issues. Decentralized graph computing is a complex process that has multiple steps. For example, the basic idea of decentralized graph computing is to divide the graph into several parts, calculate the intermediate results in each part,

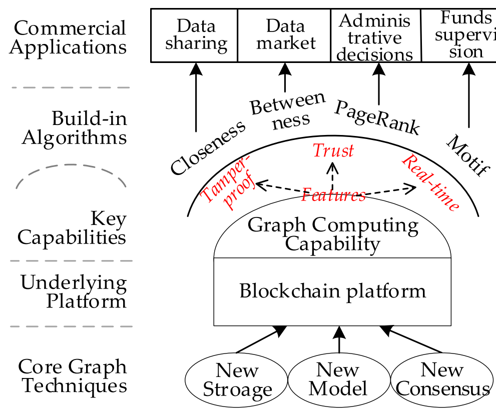


Fig. 6. A sketch of GiBP designed for supporting Commercial Applications in Section 5.2.

exchange the intermediate results, and merge the intermediate results into the final one. Both intermediate and final results require a consensus. Designing a highly scalable graph computing algorithm suitable for blockchain is challenging.

6.2.3. Customization

Currently, users develop smart contracts through the interfaces provided by the blockchain platform. Customization targets a broad set of interfaces, allowing users to create, extend, redefine, and maintain the development of smart contracts through the interface. In a non-GiBP, the interface is relatively simple because the operations on block data are only a few, such as query and write ledgers. When graph computing is integrated into the blockchain, the interfaces become more complicated because the graph query and analysis algorithms are quite a lot. Therefore, customization through these interfaces becomes a problematic issue. As a new topic, it has two branches: data customization and algorithm customization.

Data customization allows developers to define new graph structures, control the creation of graph data, and maintain the data lifecycle in a GiBP. Researchers may study the sophisticated data structure and storage. So that GiBP could offer enough flexibility to customize graph data. Algorithm customization allows developers to introduce new graph algorithms in the GiBP. However, graph algorithms in the GiBP run in a parallel and distributed manner and reach a consensus. Researchers may study how to keep user-defined graph algorithms running like built-in ones. In other words, the GiBP has the transparency of decentralization and consensus to the user-defined algorithms. Decentralization and consensus is the major challenge for algorithm customization. We may study their common processes on various graph algorithms, integrate the common into the algorithm execution engine, and let the rest be customizable.

6.3. Expectable

As shown in Fig. 6, we expect that a GiBP novelly introduces storage for graph data, a computing model for graph algorithms, and a consensus mechanism for graph analysis results. We integrate graph query and analysis capabilities into a well-known blockchain platform based on these core graph techniques. We expect the capabilities have features of tamper-proofing, trust, real-time, and efficiency. A GiBP contains built-in graph algorithms, such as Closeness, Betweenness, PageRank, and Motif, for commercial applications, for example, those defined in Section 5.2. This section discusses the expected functions of a GiBP. They are in four groups: graph creation, graph storage, graph queries, and graph analysis.

6.3.1. Creation

As discussed in Section 4, a GiBP may employ two ways to create graph data: the platform-defined program automatically extracts graph data from the blockchain according to system-level purposes; otherwise, the user-defined program retrieves graph data from block data according to business logic. The former is generally for academic applications, and the latter is for commercial ones.

6.3.2. Storage

Data in a blockchain is time-variant, and the same as graph data. The graph storage should hold both the current and the historical states. Therefore, a GiBP should store temporal graphs, the special kind of labeled graphs that change with time. The time element is modeled, maintained, and indexed in the storage implicitly or explicitly. A GiBP could design a particular data structure for the temporal graph, and transfer it to the block data structure for storing, for example, records of the ledger table. The temporal graphs have four possible storage locations: the world-state database, the block data, the both, or other new forms of storage.

6.3.3. Query

In a GiBP, graph query employs algorithms such as Motif and K-core to search and filter important features or subgraphs in the original graph. The output result is a subgraph or a set of subgraph nodes/edges of the original graph. In the data management field, graph query is the core operation of OLTP (On-Line Transaction Processing). The query process should be executed in parallel, and the query result should be validated to reach a consensus.

6.3.4. Analysis

In a GiBP, graph analysis is for information that cannot be intuitively obtained from the graph, such as the authority of nodes. Graph analysis algorithms like PageRank can get this information. In the data management field, graph analysis is the core operation of OLAP (On-Line Analytical Processing). The analysis process should be executed in parallel, and the result should be validated to reach a consensus.

6.4. Summary (RQ4)

In summary, GiBP is an expected platform for solving gaps between existing blockchain-related studies and applications. It should support real-time and trustless graph computing through fresh data. It could adopt expected storage for storing and obtaining fresh data, design expected models and consensus mechanisms for real-time and trustless graph processing, and offer customized interfaces for developing smart contracts. Currently, no blockchain platforms can support all the features together. In future, the main focus of GiBP will on its challenges: decentralization, scalability, and customization.

Besides, the expected functions of a GiBP are also connected with these challenges. For example, the graph creation dominates the decentralization degree because the graph may be built in a centralized or distributed manager; the storage and query mechanisms ensure the scalability if a scalable data and computing framework is applied; and the rich-featured graph analysis are developed through customizable smart contracts.

7. Conclusion

This paper surveys state-of-the-art studies and applications on graph-related blockchains. We aim to answer why a blockchain needs graphs, and what are the gaps between studies and applications, as well as why GiBP is necessary and challengeable. Our survey starts with two survey methods and four research questions, continue with the description of prior works through three dimensions, and ends with summary for each dimension and answers for each research question.

(1) The graph algorithms for blockchains. Blockchains' security, performance, and evaluation are three purposes for applying graph algorithms. The "graph + blockchain" solution but not the GiBP solution is enough for the purposes because the graph contains neither on-chain blockchain data nor business logic.
(2) The graph data in blockchains. The seven types of graphs are all "connected" and generated from a blockchain data structure. They are operated in on-chain processing and off-chain processing manners. They supported broad blockchain applications.
(3) The graph applications on blockchains. Half of the academic applications need tightly integrated with graph computing. And all the commercial applications should follow the same computing paradigm.

Based on the above surveys, we conclude the gaps of lacking a real-time and trustless computing environment with fresh data. We then propose GiBP, an expected platform for solving gaps between existing blockchain-related studies and applications, has indispensable features, challenges, and expected operations. We explain why integrating graphs into a blockchain platform is inevitable, what are the challenges of GiBPs, and what are GiBPs' main functions and features people expect. Additionally, we also discuss the proposed opportunities for GiBPs. They could be referred to as research prospects. We emphasize the following four directions: research on the graph data structure and graph storage in GiBPs; research on the decentralized graph computing with a consensus mechanism in GiBPs; research on the scalable and extendable GiBP; and case study on GiBP.

CRediT authorship contribution statement

Jie Song: Conceptualization, Methodology, Writing – original draft.
Pengyi Zhang: Investigation, Formal analysis, Writing – revised draft.
Qiang Qu: Resources, Methodology.
Yongjie Bai: Resources, Data Curation, Project administration.
Yu Gu: Validation, Writing – reviewing and editing.
Ge Yu: Supervision, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgments

This paper is supported by the Research Foundation of HuaWei Cloud Co. Ltd., China, under Grant No. TC20220104482.

References

[1] Amazon blockchain, https://aws.amazon.com/blockchain/?nc1=h_ls, December 2022.
[2] Ali Graph Computing Service, https://help.aliyun.com/product/121021.html, December 2022.
[3] Nazmiye Ceren Abay, Cuneyt Gurcan Akcora, Yulia R. Gel, Murat Kantarcioglu, Umar D. Islambekov, Yahui Tian, Bhavani Thuraisingham, ChainNet: learning on blockchain graphs with topological features, in: 2019 IEEE International Conference on Data Mining (ICDM), IEEE, Beijing, China, November 2019, pp. 946–951.
[4] Kobbane Abdellatif, Charkaoui Abdelmouttalib, Graph-based computing resource allocation for mobile blockchain, in: 2018 6th International Conference on Wireless Networks and Mobile Communications (WINCOM), IEEE, Marrakesh, Morocco, October 2018, pp. 1–4.
[5] Israa Alqassem, Iyad Rahwan, Davor Svetinovic, The anti-social system properties: bitcoin network data analysis, IEEE Trans. Syst. Man Cybern. Syst. 50 (1) (January 2020) 21–31.
[6] Elli Androulaki, Christian Cachin, Angelo De Caro, Eleftherios Kokoris-Kogias Channels, Horizontal scaling and confidentiality on permissioned blockchains, in: Javier Lopez, Jianying Zhou, Miguel Soriano (Eds.), Computer Security, in: Lecture Notes in Computer Science, vol. 11098, Springer International Publishing, Cham, 2018, pp. 111–131.
[7] Andra Anoaica, Hugo Levard, Quantitative description of internal activity on the Ethereum public blockchain, in: 2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS), IEEE, Paris, February 2018, pp. 1–5.
[8] Xiang Ao, Yang Liu, Zidi Qin, Yi Sun, Qing He, Temporal high-order proximity aware behavior analysis on Ethereum, World Wide Web 24 (5) (September 2021) 1565–1585.
[9] Georgia Avarikioti, Gerrit Janssen, Yuyi Wang, Roger Wattenhofer, Payment network design with fees, in: Joaquin Garcia-Alfaro, Jordi Herrera-Joancomartí, Giovanni Livraga, Ruben Rios (Eds.), Data Privacy Management, Cryptocurrencies and Blockchain Technology, in: Lecture Notes in Computer Science, vol. 11025, Springer International Publishing, Cham, 2018, pp. 76–84.
[10] Qianlan Bai, Chao Zhang, Nianyi Liu, Xiaowei Chen, Yuedong Xu, Xin Wang, Evolution of transaction pattern in Ethereum: a temporal graph perspective, IEEE Trans. Comput. Soc. Syst. 9 (3) (June 2022) 851–866.
[11] Blockchain market size, https://www.fortunebusinessinsights.com/industry-reports/blockchain-market-100072, December 2022.
[12] Xavier Boyen, Christopher Carr, Thomas Haines, Graphchain: a blockchain-free scalable decentralised ledger, in: Proceedings of the 2nd ACM Workshop on Blockchains, Cryptocurrencies, and Contracts - BCC '18, ACM Press, Incheon, Republic of Korea, 2018, pp. 21–33.
[13] Christian Cachin, Angelo De Caro, Pedro Moreno-Sanchez, Björn Tackmann, Marko Vukolic, The Transaction Graph for Modeling Blockchain Semantics, Cryptoeconomic Systems, December 2020.
[14] Huan Chen, Yijie Wang Minichain, A lightweight protocol to combat the utxo growth in public blockchain, J. Parallel Distrib. Comput. 143 (2020) 67–76.
[15] Ting Chen, Zihao Li, Yuxiao Zhu, Jiachi Chen, Xiapu Luo, John Chi-Shing Lui, Xiaodong Lin, Xiaosong Zhang, Understanding Ethereum via graph analysis, ACM Trans. Internet Technol. 20 (2) (May 2020) 1–32.
[16] Weili Chen, Tuo Zhang, Zhiguang Chen, Zibin Zheng, Yutong Lu, Traveling the token world: a graph analysis of Ethereum ERC20 token ecosystem, in: Proceedings of the Web Conference 2020, ACM, Taipei Taiwan, April 2020, pp. 1411–1421.
[17] Anurag Choubey, Sourajit Behera, Yashwant Singh Patel, Karanam Mahidhar, Rajiv Misra, EnergyTradingRank algorithm for truthful auctions among EVs via blockchain analytics of large scale transaction graphs, in: 2019 11th International Conference on Communication Systems & Networks (COMSNETS), IEEE, Bengaluru, India, January 2019, pp. 1–6.
[18] Damiano Di Francesco Maesa, Andrea Marino, Laura Ricci, Uncovering the bitcoin blockchain: an analysis of the full users graph, in: 2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA), IEEE, Montreal, QC, Canada, October 2016, pp. 537–546.
[19] Meryam Essaid, Sejin Park, Hongteak Ju, Visualising bitcoin's dynamic P2P network topology and performance, in: 2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), IEEE, Seoul, Korea (South), May 2019, pp. 141–145.
[20] Sébastien Forestier, Damir Vodenicarevic, Adrien Laversanne-Finot, Blockclique: scaling blockchains through transaction sharding in a multithreaded block graph, arXiv:1803.09029 [cs], February 2020.
[21] Michael Fröwis, Rainer Böhme, In code we trust?: measuring the control flow immutability of all smart contracts deployed on Ethereum, in: Joaquin Garcia-Alfaro, Guillermo Navarro-Arribas, Hannes Hartenstein, Jordi Herrera-Joancomartí (Eds.), Data Privacy Management, Cryptocurrencies and Blockchain Technology, in: Lecture Notes in Computer Science, vol. 10436, Springer International Publishing, Cham, 2017, pp. 357–372.
[22] Enrique Fynn, Fernando Pedone, Challenges and pitfalls of partitioning blockchains, in: 2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W), IEEE, Luxembourg, June 2018, pp. 128–133.
[23] Rosa Giovanni, Remo Pareschi Tether, A study on bubble-networks, Front. Blockchain 4 (August 2021) 686484.

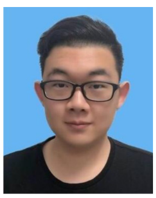
[24] Barney G. Glaser, Anselm L. Strauss, The Discovery of Grounded Theory: Strategies for Qualitative Research, Aldine, 1967.
[25] Sarada Prasad Gochhayat, Sachin Shetty, Ravi Mukkamala, Peter Foytik, Georges A. Kamhoua, Laurent Njilla, Measuring decentrality in blockchain based systems, IEEE Access 8 (2020) 178372–178390.
[26] Joseph Gonzalez, Yucheng Low, Haijie Gu, Danny Bickson, Carlos Guestrin, PowerGraph: distributed graph-parallel computation on natural graphs, in: OSDI, October 2012, pp. 17–30.
[27] Barbara Guidi, Andrea Michienzi, Laura Ricci, Steem blockchain: mining the inner structure of the graph, IEEE Access 8 (2020) 210251–210266.
[28] Barbara Guidi, Andrea Michienzi, Laura Ricci, A graph-based socioeconomic analysis of steemit, IEEE Trans. Comput. Soc. Syst. 8 (2) (2021) 365–376.
[29] Shaoyong Guo, Xing Hu, Song Guo, Xuesong Qiu, Feng Qi, Blockchain meets edge computing: a distributed and trusted authentication system, IEEE Trans. Ind. Inform. 16 (3) (March 2020) 1972–1983.
[30] Yuwei Guo, Jinfeng Tong, Chen Feng, A measurement study of bitcoin lightning network, in: 2019 IEEE International Conference on Blockchain (Blockchain), IEEE, Atlanta, GA, USA, July 2019, pp. 202–211.
[31] Huawei Graph Computing Engine, https://www.huaweicloud.com/product/ges.html, December 2022.
[32] Hyperledger Fabric of IBM, https://www.hyperledger.org/use/fabric, December 2022.
[33] Sachini Jayasekara, Shanika Karunasekera, Aaron Harwood, A multi-stage hierarchical window model with application to real-time graph analysis, in: 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS), IEEE, Atlanta, GA, USA, June 2017, pp. 2561–2564.
[34] Julia Khamis, Ori Rottenstreich, Demand-aware channel topologies for off-chain payments, in: 2021 International Conference on COMmunication Systems & NETworkS (COMSNETS), IEEE, Bangalore, India, January 2021, pp. 272–280.
[35] Jaehyeon Kim, Sejong Lee, Yushin Kim, Seyoung Ahn, Sunghyun Cho, Graph learning-based blockchain phishing account detection with a heterogeneous transaction graph, Sensors 23 (1) (2023) 463.
[36] Ajay Kumar, Abhishek Kumar, Pranav Nerurkar, Muhammad Rukunuddin Ghalib, Achyut Shankar, Zheng Wen, Xin Qi, Empirical analysis of bitcoin network (2016-2020), in: 2020 IEEE/CIC International Conference on Communications in China (ICCC Workshops), IEEE, Chongqing, China, August 2020, pp. 96–101.
[37] Sungho Lee, Minsu Kim, Jemin Lee, Ruei-Hau Hsu, Tony Q.S. Quek, Is blockchain suitable for data freshness? An age-of-information perspective, IEEE Netw. 35 (2) (March 2021) 96–103.
[38] Xi Tong Lee, Arijit Khan, Sourav Sen Gupta, Yu Hann Ong, Xuan Liu, Measurements, analyses, and insights on the entire Ethereum blockchain network, in: Proceedings of the Web Conference 2020, ACM, Taipei Taiwan, April 2020, pp. 155–166.
[39] Bixin Li, Zhenyu Pan, Tianyuan Hu. ReDefender, Detecting reentrancy vulnerabilities in smart contracts automatically, IEEE Trans. Reliab. 71 (2) (June 2022) 984–999.
[40] Songze Li, Mingchao Yu, Chien-Sheng Yang, Amir Salman Avestimehr, Sreeram Kannan, Pramod Viswanath, PolyShard: coded sharding achieves linearly scaling efficiency and security simultaneously, IEEE Trans. Inf. Forensics Secur. 16 (2021) 249–261.
[41] Xiao Li, Weili Wu, A blockchain transaction graph based machine learning method for bitcoin price prediction, arXiv:2008.09667 [cs, q-fin, stat], August 2020.
[42] Xiaoyun Li, Zibin Zheng, Hong-Ning Dai, When services computing meets blockchain: challenges and opportunities, J. Parallel Distrib. Comput. 150 (2021) 1–14.
[43] Chao Lin, Debiao He, Xinyi Huang, Muhammad Khurram Khan, Kim-Kwang Raymond Choo, A new transitively closed undirected graph authentication scheme for blockchain-based identity management systems, IEEE Access 6 (2018) 28203–28212.
[44] Dan Lin, Jiajing Wu, Qi Yuan, Zibin Zheng, Modeling and understanding Ethereum transaction records via a complex network approach, IEEE Trans. Circuits Syst. II, Express Briefs 67 (11) (November 2020) 2737–2741.
[45] Dan Lin, Jialan Chen, Jiajing Wu, Zibin Zheng, Evolution of Ethereum transaction relationships: toward understanding global driving factors from microscopic patterns, IEEE Trans. Comput. Soc. Syst. 9 (2) (April 2022) 559–570.
[46] Yucheng Low, Danny Bickson, Joseph Gonzalez, Carlos Guestrin, Aapo Kyrola, Joseph M. Hellerstein, Distributed GraphLab: a framework for machine learning and data mining in the cloud, Proc. VLDB Endow. 5 (8) (April 2012) 716–727.
[47] Sai Lv, Hui Li, Han Wang, Xiangui Wang, CoT: a secure consensus of trust with delegation mechanism in blockchains, in: Xueming Si, Hai Jin, Yi Sun, Jianming Zhu, Liehuang Zhu, Xianhua Song, Zeguang Lu (Eds.), Blockchain Technology and Application, in: Communications in Computer and Information Science, vol. 1176, Springer Singapore, Singapore, 2020, pp. 104–120.
[48] Michael Malak, Robin East, Spark GraphX in Action, Simon and Schuster, 2016.
[49] Grzegorz Malewicz, Matthew H. Austern, Aart J.C. Bik, James C. Dehnert, Ilan Horn, Naty Leiser, Grzegorz Czajkowski, Pregel: a system for large-scale graph processing, in: Proceedings of the 28th ACM Symposium on Principles of Distributed Computing - PODC '09, ACM Press, Calgary, AB, Canada, 2009, p. 6.
[50] Lifeng Nai, Yinglong Xia, Ilie G. Tanase, Hyesoon Kim, Exploring big graph computing — an empirical study from architectural perspective, in: Special Issue on Scalable Computing Systems for Big Data Applications, J. Parallel Distrib. Comput. 108 (2017) 122–137.
[51] Neo4j-APOC, https://neo4j.com/labs/apoc, December 2022.
[52] Pranav Nerurkar, Dhiren Patel, Yann Busnel, Romaric Ludinard, Saru Kumari, Muhammad Khurram Khan, Dissecting bitcoin blockchain: empirical analysis of bitcoin network (2009–2020), J. Netw. Comput. Appl. 177 (March 2021) 102940.
[53] Mariusz Nowostawski, Jardar Tøn, Evaluating methods for the identification of off-chain transactions in the lightning network, Appl. Sci. 9 (12) (June 2019) 2519.
[54] Masarah Paquet-Clouston, Matteo Romiti, Bernhard Haslhofer, Thomas Charvat, Spams meet cryptocurrencies: sextortion in the bitcoin ecosystem, in: Proceedings of the 1st ACM Conference on Advances in Financial Technologies, ACM, Zurich Switzerland, October 2019, pp. 76–88.
[55] Beltran Borja Fiz Pontiveros, Mathis Steichen, Radu State, Mint centrality: a centrality measure for the bitcoin transaction graph, in: 2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), IEEE, Seoul, Korea (South), May 2019, pp. 159–162.
[56] Farimah Poursafaei, Reihaneh Rabbany, Zeljko Zilic, SigTran: signature vectors for detecting illicit activities in blockchain transaction networks, in: Kamal Karlapalem, Hong Cheng, Naren Ramakrishnan, R.K. Agrawal, P. Krishna Reddy, Jaideep Srivastava, Tanmoy Chakraborty (Eds.), Advances in Knowledge Discovery and Data Mining, in: Lecture Notes in Computer Science, vol. 12712, Springer International Publishing, Cham, 2021, pp. 27–39.
[57] Chayma Sakouhi, Abir Khaldi, Henda Ben Ghezala, Hammer lightweight graph partitioner based on graph data volumes, J. Parallel Distrib. Comput. 158 (2021) 16–28.
[58] Zonyin Shae, Jeffrey Tsai, AI blockchain platform for trusting news, in: 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS), IEEE, Dallas, TX, USA, July 2019, pp. 1610–1619.
[59] Yahya Shahsavari, Kaiwen Zhang, Chamseddine Talhi, Performance modeling and analysis of the bitcoin inventory protocol, in: 2019 IEEE International Conference on Decentralized Applications and Infrastructures (DAPPCON), IEEE, Newark, CA, USA, April 2019, pp. 79–88.
[60] Aman Sharma, Ashutosh Bhatia, Bitcoin's blockchain data analytics: a graph theoretic perspective, arXiv:2002.06403 [cs], February 2020.
[61] Herbert A. Simon, The Sciences of the Artificial, MIT Press, 1969.
[62] Parwat Singh Anjana, Sweta Kumari, Sathya Peri, Sachin Rathor, Archit Somani, An efficient framework for optimistic concurrent execution of smart contracts, in: 2019 27th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP), IEEE, Pavia, Italy, February 2019, pp. 83–92.
[63] Parwat Singh Anjana, Sweta Kumari, Sathya Peri, Sachin Rathor, Archit Somani, OptSmart: a Space Efficient Optimistic Concurrent Execution of Smart Contracts, Distributed and Parallel Databases, May 2022.
[64] Jie Song, Pengyi Zhang, Mohammed Alkubati, Yubin Bao, Ge Yu, Research advances on blockchain-as-a-service: architectures, applications and challenges, Digit. Commun. Netw. 8 (4) (August 2022) 466–475.
[65] Mirek Sopek, Dominik Tomaszuk, Szymon Glab, Filip Turobos, Ivo Zielinski, Dominik Kuzinski, Ryszard Olejnik, Piotr Luniewski, Przemyslaw Gradzki, Technological foundations of ontological ecosystems on the 3rd generation blockchains, IEEE Access 10 (2022) 12487–12502.
[66] Stefan Tai, Continuous, trustless, and fair: changing priorities in services computing, in: Alexander Lazovik, Stefan Schulte (Eds.), Advances in Service-Oriented and Cloud Computing, in: Communications in Computer and Information Science, vol. 707, Springer International Publishing, Cham, 2018, pp. 205–210.
[67] Subhasis Thakur, John G. Breslin, An edge colouring-based collaborative routing protocol for blockchain offline channels, in: 2020 IEEE International Conference on Blockchain (Blockchain), IEEE, Rhodes Island, Greece, November 2020, pp. 343–350.
[68] Jeyakumar Samantha Tharani, Eugene Yougarajah Andrew Charles, Zhe Hou, Marimuthu Palaniswami, Vallipuram Muthukkumarasamy, Graph based visualisation techniques for analysis of blockchain transactions, in: 2021 IEEE 46th Conference on Local Computer Networks (LCN), IEEE, Edmonton, AB, Canada, October 2021, pp. 427–430.
[69] Dan Wang, Jindong Zhao, Chunxiao Mu, Research on blockchain-based E-bidding system, Appl. Sci. 11 (9) (April 2021) 4011.
[70] Hongyu Wang, Bing Yao, On multiple topological authentications in the encryption of tree-like networks, in: 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), IEEE, Chongqing, China, June 2020, pp. 2017–2022.
[71] Hanqing Wu, Jiannong Cao, Shan Jiang, Ruosong Yang, Yanni Yang, Jianfei Hey, TSAR: a fully-distributed trustless data ShARing platform, in: 2018 IEEE International Conference on Smart Computing (SMARTCOMP), IEEE, Taormina, June 2018, pp. 350–355.
[72] Lian Yu, Lijun Liu, Yanbing Jiang, Qi Jing, Bei Zhao, Chen Zhang, Attack graph auto-generation for blockchains based on bigraphical reaction systems, in: 2020 IEEE 20th International Conference on Software Quality, Reliability and Security (QRS), IEEE, Macau, China, December 2020, pp. 285–296.

[73] Yuhang Zhang, Jun Wang, Fei Zhao, Transaction community identification in bitcoin, in: 2020 13th International Symposium on Computational Intelligence and Design (ISCID), IEEE, Hangzhou, China, December 2020, pp. 140–144.

[74] Lin Zhao, Sourav Sen Gupta, Arijit Khan, Robby Luo, Temporal analysis of the entire Ethereum blockchain network, in: Proceedings of the Web Conference 2021, ACM, Ljubljana Slovenia, April 2021, pp. 2258–2269.

[75] Xuanchen Zhou, Wenzhong Yang, Xiaodan Tian, Detecting phishing accounts on Ethereum based on transaction records and EGAT, Electronics 12 (4) (2023) 993.

Jie Song received his Ph.D. degree in computer science from Northeastern University in 2008. He is a Professor at Software College, Northeastern University. His research interest includes big data management, distributed computing, and machine learning.



Pengyi Zhang took a successive postgraduate and doctoral program of Northeastern University from 2019. He is now a master of Software College, Northeastern University. His research interest includes blockchain platforms and graph computing.

Qiang Qu received the MSc degree in computer science from Peking University, and the PhD degree from Aarhus University. He is now a professor with the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences. His current research interests include large-scale data management and mining.

Yongjie Bai received his Mater Degree in computer science from Peking University in 2021. He is now a software engineer with the Blockchain Lab, Huawei Cloud Tech Co., Ltd. His current research interests include distributed systems and computer network.

Yu Gu received his Ph.D. degree from Northeastern University in 2010. He is a professor of Computer Science, Northeastern University China. His current research interests include graph data management, spatial data management, and big data analysis.

Ge Yu received his Ph.D. degree in computer science from Kyushu University of Japan in 1996. He is a Professor at Computer Science and Engineering, Northeastern University. His research interest includes database theory and technology, distributed and parallel systems.