# Structural Identity Representation Learning for Blockchain-Enabled Metaverse Based on Complex Network Analysis

Bishenghui Tao, *Graduate Student Member, IEEE*, Hong-Ning Dai, *Senior Member, IEEE*, Haoran Xie, *Senior Member, IEEE*, and Fu Lee Wang, *Senior Member, IEEE*

***Abstract*—The metaverse and its underlying blockchain technology have attracted extensive attention in the past few years. How to mine, process, and analyze the tremendous data generated by the metaverse systems has posed a number of challenges. Aiming to address them, we mainly focus on modeling and understanding the blockchain transaction network from a structural identity perspective, which represents the entire network structure and reveals the relations among multiple entities. In this article, we analyze three metaverse-related systems: non-fungible token (NFT), Ethereum (ETH), and Bitcoin (BTC) from the structural-identity perspective. First, we conduct the complex network analysis of the metaverse network and obtain several new insights (i.e., power-law degree distribution, disconnection, disassortativity, preferential attachment, and non-rich-club effect). Secondly, based on such findings, we propose a novel representation learning method named structure-to-vector with random pace (SVRP) for learning both the latent representation and structural identity of the network. Thirdly, we conduct node classification and link prediction tasks with the integration of graph neural networks (GNNs). Empirical results on three real-world datasets demonstrate that our proposed SVRP outperforms other existing methods in multiple tasks. In particular, our SVRP achieves the highest node classification accuracy (Acc) (99.3%) and $F1$-score (96.7%) while only requiring original non-attributed graphs.**

***Index Terms*—Blockchain, complex networks, graph neural networks (GNNs), graph representation, metaverse.**

## I. Introduction

RECENTLY, the metaverse has enticed massive attention from the world with the development of novel technologies. The metaverse, as a combination of “meta” (meaning beyond) and the stem “verse” from “universe,” denotes the next-generation Internet, in which users not only produce contents but also edit the world [1]. As a new type of Internet application and social forms with the integration of a variety of new technologies, the metaverse has been a paradigm that is constantly evolving. Different participants are enriching its meaning in their own ways.

The metaverse is not only the synthesis of virtual reality (VR), augmented reality (AR), and extended reality (XR) but also the underlying blockchain technology, which ensures trust across multiple entities in the metaverse ecosystem. Judging from many existing successful projects, the metaverse provides an immersive experience based on blockchain systems, while tightly integrating the virtual world into the economic system, the information system, and the trusted identity system. In particular, the initial metaverse startups usually create blockchain-based platforms that deeply involve leveraging cryptocurrencies, decentralized Finance (DeFi), and non-fungible tokens (NFTs) [2], [3]. For example, several blockchain-based platforms, such as Decentraland’s MANA and Sandbox’s SAND, require cryptocurrency tokens to purchase and sell virtual assets. Owing to its immutability and traceability, blockchain has been a critical infrastructure for diverse metaverse applications.

As the fundamental deployment technology of the metaverse, the advent of blockchain systems has gained popularity in a myriad of applications [4]. Despite unprecedented opportunities brought by decentralization, anonymity, and non-repudiation, blockchains are also notorious due to malicious activities, such as money laundering [5], market manipulation [6], scams [7] and ransomware [8]. As the primary transaction and payment channel of the metaverse, blockchain technology has an impact on the network security of the metaverse ecosystem. In particular, a large number of transactions have been executed on the underlying blockchain to underpin the upper metaverse applications. Therefore, it is a necessity to ensure the security of blockchain transactions, which can be regarded as the *metaverse transaction security*. However, the evolution of metaverse transaction security is still in its infancy, with great potential for improvement. Facing such challenges, it is crucial to mine, process, and analyze massive volumes of blockchain transaction data especially when dealing with enormous unstructured blockchain transaction data.

Manuscript received 16 July 2022; revised 23 September 2022 and 13 December 2022; accepted 22 December 2022. Date of publication 24 January 2023; date of current version 2 October 2023. This work was supported in part by the Macao Science and Technology Development Fund under Macao Funding Scheme for Key Research and Development Projects under Grant 0025/2019/AKP; in part by the Department of Computer Science Startup Fund of Hong Kong Baptist University; in part by the Lam Woo Research Fund under Grant LWP20019; and in part by the Faculty Research Grants of Lingnan University, Hong Kong, under Grant DB22B4. *(Corresponding author: Hong-Ning Dai.)*

Bishenghui Tao is with the School of Computer Science and Engineering, Macau University of Science and Technology, Taipa, Macau, and also with the School of Science and Technology, Hong Kong Metropolitan University, Hong Kong (e-mail: bishenghui.tao@connect.polyu.hk).

Hong-Ning Dai is with the Department of Computer Science, Hong Kong Baptist University, Hong Kong (e-mail: hndai@ieee.org).

Haoran Xie is with the Department of Computing and Decision Sciences, Lingnan University, Hong Kong (e-mail: hrxie@ln.edu.hk).

Fu Lee Wang is with the School of Science and Technology, Hong Kong Metropolitan University, Hong Kong (e-mail: pwang@hkmu.edu.hk).

Digital Object Identifier 10.1109/TCSS.2022.3233059

Structural identity reveals the relationships among multiple entities (e.g., users' accounts) and expresses the graph structure of the network. Thus, it is of great importance to study the structural identity of blockchain transaction networks for enhancing metaverse transaction security. For example, a suitable latent representation can represent high-dimensional network topology as embeddings, which can be applied to machine learning tasks to discern the type of users in the metaverse (e.g., malicious or common, illegal or legitimate, etc.).

Inspired by recent advances in structure-to-vector (s2v) methods in learning the structural identity of other networks [9], we present a novel structural identity representation method, namely s2v with random pace (SVRP) for the *Blockchain networks* (BCNs). Our SVRP can learn the structural identity of BCNs and characterize similar nodes with the random pace strategy. Integrating with graph neural networks (GNNs), our SVRP also outperforms existing methods in both node classification and link prediction tasks. To summarize, the main contributions of this article are as follows.

1) In this article, we conduct a complex-network analysis of three different BCNs and obtain several important observations including the power-law degree distribution, disconnection, disassortativity, preferential attachment, and non-rich-club effect.
2) To our best knowledge, we are the first to propose a novel representation learning method–SVRP for analyzing BCNs from the structural identity perspective. Our SVRP can well learn both the latent representation and structural identity of BCN.
3) Extensive experiments on three datasets demonstrate that our SVRP outperforms existing state-of-the-art baselines when integrating our SVRP with GNNs in node classification and link prediction tasks. Meanwhile, the results reveal that our SVRP requires less attributed data (which contain naturally or artificially-constructed node features) to achieve the highest accuracy (Acc) (99.3%) and $F1$-score (96.7%).

The rest of the article is organized as follows. Section II first presents a literature review on related studies. Then, a complex network analysis on BCNs is given in Section III. Section IV presents our proposed structural identity representation learning method. Experimental results are then given in Section V. Finally, Section VI concludes the article and discusses the future direction.

## II. Related Work

As a basic unit to record interactions between different blockchain accounts, a transaction can characterize activities occurring on the blockchain. Therefore, it is crucial to analyze the *blockchain transaction network* consisting of multiple correlated transactions from a network perspective. Unlike some existing methods relying on artificial or natural features [5], [10], [11], [12], network representation learning methods have been adopted to analyze blockchain transaction networks. Despite advances in [13], [14], [15], [16], and [17], most of them are mainly based on Deepwalk [18] or node-to-vector (node2vec) [19], which can well learn *latent representations* of nodes while cannot accurately capture the *structural identity* of the entire network. Take Fig. 1 as an example, in which nodes $u$ and $v$ have similar local structures, i.e., the same degree and similar neighbor structures (e.g., either $u$ or $v$ is connected with a triangular cycle). However, they cannot be well learned by existing approaches since they do not share similar latent representations as they are far away from each other or even disconnected.

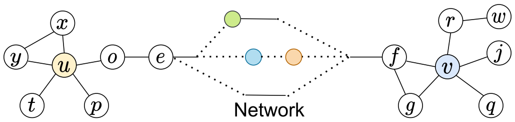


Fig. 1. Nodes $u$ and $v$ have similar neighborhood structures, but they are far away from each other in the network.

This section presents a brief literature survey on related studies to our work. We roughly categorize them into network analysis on understanding BCNs, modeling BCNs, and representation learning on graphs.

### A. Understanding BCNs

There are myriad studies on modeling blockchain transaction networks by considering unstructured cryptocurrency data as a network structure. Most of them mainly focus on data analysis, such as [20], [21], [22], [23], [24], and [25] explored BCN with a focus on descriptive statistics. Thereafter, studies like [26], [27], and [28] analyze the BCN from perspectives of the long-term evolutionary dynamics. Meanwhile, studies [29], [30], and [31] provide large-scale insights into the BCN transaction network. However, full graph analyses require a tremendous computational cost of processing and analyzing data. Therefore, the study [30] reduces the size of BCN by utilizing a scalable clustering algorithm.

### B. Modeling BCNs

In the line of studies on modeling BCNs, a substantial part of them focuses on price prediction. Literatures [11], [32], [33], [34], [35], and [36] provide various models for the analysis of the cryptocurrency prices or transaction fees, such as Gradient Boosting Decision Tree, neural networks, and long short-term memory (LSTM) networks. On the other hand, other studies focus on characterizing graphs and detecting cryptocurrency BCNs. In this context, studies [37], [38], [39], and [29] detect unusual behaviors. Moreover, studies [14], [16], and [40] propose graph-representation methods and graph-feature explanation methods to identify and capture informative graphs. It is worth noting that the study [14] and [16] employs DeepWalk and node2vec methods while [40] uses the struc2vec model to label graphs.

### C. Representation Learning on Graphs

Despite the above advances in analyzing BCNs, there are few studies on representation learning for the BCN. One of the major challenges is learning to capture the structural identity representation of nodes. There are some studies on

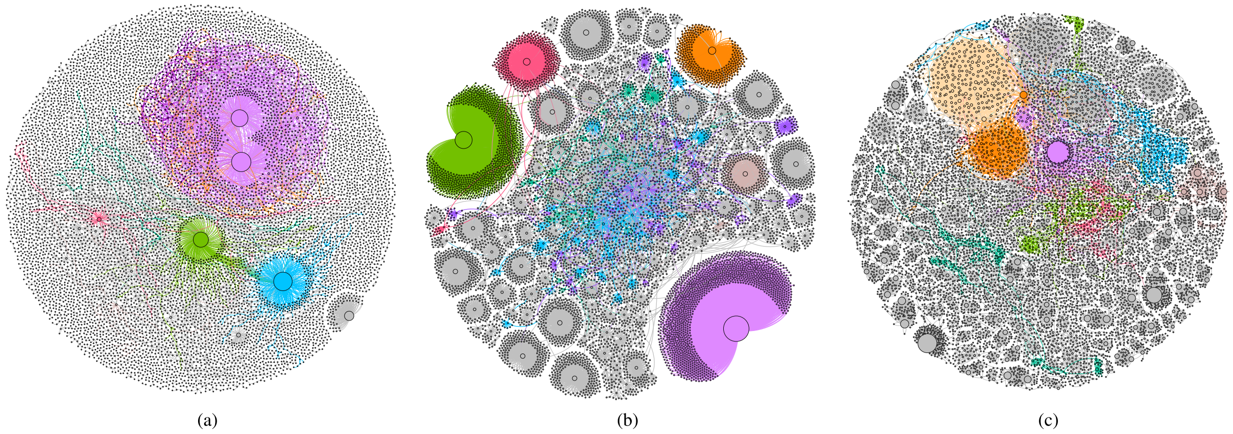


Fig. 2. Visualization of BCNs containing 10 000 selected nodes. (a) NFT. (b) ETH. (c) BTC.

network representation learning for other types of networks. For example, Deepwalk [18] firstly proposes the random walk with the natural language processing (NLP) model to process the graph. Moreover, node2vec [19] extended Deepwalk as a second-order biased random walk model, which generates the context of nodes more flexibly. However, these random walk-based methods are limited by the Skip-Grim distant window size. In other words, some similar nodes, which are far away in the network, cannot have approximating representations. Therefore, struc2vec [9] proposes a network representation method based on the structural similarity, to construct a multilayer network, thereby avoiding random walks in the original network topology. It is worth mentioning that we design a representation method, which is more suitable for the BCN inspired by the idea of struc2vec (see Section IV).

Although we present part of the main findings in our previous short conference paper [41], the current version is significantly different from the short paper in the following perspectives: 1) we conduct a comprehensive complex network analysis in Section III on three types of typical BCNs, which are highly related to the Metaverse and 2) we present new results on representation learning in Section V.

## III. Complex Network Analysis

In blockchain cryptocurrency systems, a transaction records transaction hash, timestamp, input addresses, and output addresses. The correlated transactions between addresses then construct BCN, which can be represented by a directed graph denoted by $G = (V, E)$, where $V$ is a set of nodes, and $E$ is a set of edges. In this graph, a node is a blockchain cryptocurrency address and an edge denotes a transaction between two addresses (i.e., nodes).

In particular, each edge is represented as $e_{ij} = (i, j)$, where $i$ is the input node, and $j$ is the output node. The set $E$ of a graph with $N$ nodes can then be represented as an $N \times N$ matrix, which is essentially an adjacency matrix denoted by $\mathbf{A}$. For any element $a_{ij}$ in $\mathbf{A}$, we have $a_{ij} = 1$ if there exists a link between $i$ and $j$; $a_{ij} = 0$ otherwise [42]. In particular, we have

$$a_{ij} = \begin{cases} 1, & \text{if } \; e_{ij} \; \text{ is defined} \\ 0, & \text{if } \; e_{ij} \; \text{ is not defined.} \end{cases} \tag{1}$$

In this article, we focus on three different BCNs which are closely related to the metaverse: the NFT network, Ethereum (ETH) network, and Bitcoin (BTC) network. The data of the NFT network are constructed purely from on-chain data from [43], representing the activities on the ETH NFT market between 1 April 2021 and 25 September 2021. The data of the ETH network are from [44], after using etherscan.io to crawl the transaction data with 100 804 nodes from 07 August 2015 to 31 December 2019. For the BTC network, we crawl the BTC transaction data via the cryptocurrency explorer from block height 520 890–520 910. Section V will present more details of datasets.

Fig. 2 presents the graph view of the three networks with 10 000 selected nodes respectively. It is worth noting that the size of a node is proportional to the number of its links, while the colors distinguish different communities based on their modularity. From Fig. 2, we have some preliminary observations: 1) all three networks have rare large nodes; 2) most of the nodes have small sizes and a few numbers of edges; and 3) they all have some disconnected nodes being scattered far away so that the networks are not completely connected.

In order to study and prove these conjectures in-depth, to figure out what kind of networks the BCNs are, how they work, and how they are formed, we introduce the measurement method of complex networks in the following.

### A. Degree Distribution

It is critical to examine the node degree distribution of BCNs. In complex networks, the total number of adjacent edges of a node is defined as a *degree* denoted by $k$. In BCNs, the degree $k$ is calculated for every cryptocurrency address after the summation of both incoming and outgoing transactions. Moreover, we also introduce the degree distribution denoted by $P(k)$ on degree $k$. The degree distribution $P(k)$ is the probability that a randomly-selected node has the degree equivalent to $k$ [42]. Also, if the degree $k$ obeys a power law, we then have

$$P(k) \propto k^{-\alpha} \tag{2}$$

where $\alpha$ is the scaling parameter of the power-law distribution.

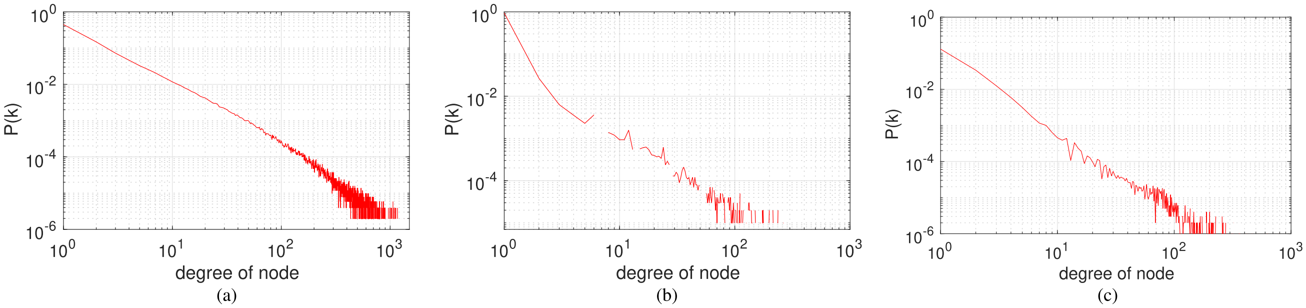


Fig. 3. Degree distribution of (a) NFT, (b) ETH, and (c) BTC networks.

TABLE I
CONNECTED COMPONENT METRICS OF NETWORKS

| Networks | #Nodes | Largest SCC (Ratio) | #SCC | Largest WCC (Ratio) | #WCC |
|---|---|---|---|---|---|
| NFT | 509577 | 146667 (28.8%) | 360446 | 490368 (96.2%) | 8034 |
| ETH | 100804 | 1149 (1.1%) | 96065 | 94079 (93.3%) | 640 |
| BTC | 168677 | 3311 (2%) | 165295 | 127752 (75.7%) | 8984 |

Fig. 3 exhibits the degree distributions of the three BCNs in logarithmic scale in both horizontal and vertical axes. We can observe from the results that all degree distributions follow the power-law distribution with the heavy tail. In particular, the scaling parameters of the network degree distribution are $\alpha_{\mathrm{NFT}} = 1.61$, $\alpha_{\mathrm{ETH}} = 3.85$, and $\alpha_{\mathrm{BTC}} = 2.07$. The results imply that they are all scale-free networks, in which only rare nodes have plenty of connections [45] while the plurality of nodes is of low degrees (i.e., nodes have fewer connections). The formation of this degree distribution can be owed to the growth mechanism of the network, such as the *preferential attachment* of self-organizing networks. This feature will be further elaborated in Section III-C.

### B. Connected Component

In complex networks, if any pair of nodes in a sub-graph has at least one connected path, we call this sub-graph a *connected component*. Meanwhile, in the case of a directed network, we measure its *strongly-connected components* (SCCs), in which any random node pair $(i, j)$ has a directed path from $i$ to $j$, and also a directed path from $j$ to $i$ simultaneously. Likewise, *weakly-connected components* (WCCs) refer to undirected connected components. Table I presents the connected component calculation results of three BCNs, including the number of SCCs, the size of the largest SCC, the number of WCCs, and the size of the largest WCC.

We observe that in all of the BCNs, both the largest SCC and the largest WCC are relatively sizable compared to the entire graph's size, implying disconnected networks. To be specific, the largest SCC covers about 30% of nodes of the NFT graph and 2% of the ETH and BTC graph. In addition, the largest WCC covers about 95% of nodes of the NFT graph, 93% of the ETH, and 75% BTC graph, respectively. We note that the NFT network is more closely connected than the others. This can be explained by the fact that NFT is still in the initial transaction phase within some small communities, and the user group is relatively fixed. Therefore, there are more frequent transactions between nodes in the NFT network than BTC and ETH networks.

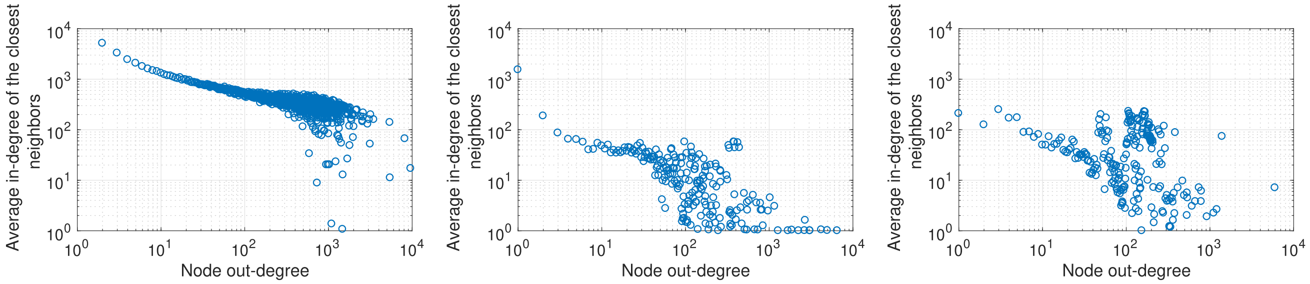
We also assume that existing hub nodes bridge many solitary nodes, as shown in Fig. 2 (see big nodes). In reality, such hub nodes may be exchanges, financial institutions, or trading platforms. Meanwhile, we also noticed that the number of WCCs is far more undersized than that of SCCs, implying that many transactions are only one-way in these graphs. In other words, the majority of nodes do not make bidirectional transactions frequently, i.e., they only pay or only accept tokens. At the same time, only a few nodes transact bidirectionally, i.e., frequently paying and receiving.

### C. Disassortativity

According to the analysis of degree distributions of BCNs, there is a gap between the number of high-degree nodes and that of low-degree nodes, implying the high *heterogeneity* of BCNs. To further explore this connection tendency, we introduce the assortativity analysis. Firstly, we adopt the Pearson correlation coefficient denoted by $\rho$ to characterize the network assortativity [42]. To be specific, a negative value of the Pearson correlation coefficient denotes disassortativity. The total number of edges in the graph is denoted by $|E|$. We then have Pearson correlation coefficient $\rho$ as in (3), shown at the bottom of the page, where $k_i$ is the out-degree of node $i$ at the beginning of link $e_{ij} \in E(G)$ and $k_j$ is the in-degree of the node at the end of link $e_{ij} \in E(G)$.

In this case, the values of the Pearson correlation coefficient are $\rho_{\mathrm{NFT}} = -0.03$, $\rho_{\mathrm{ETH}} = -0.21$, and $\rho_{\mathrm{BTC}} = -0.023$, indicating that all the networks are disassortative. Particularly, it means that high-degree nodes tend to link with low-degree

$$\rho = \frac{|E|^{-1}\sum_{e_{ij}\in E(G)} k_i k_j - \left[|E|^{-1}\sum_{e_{ij}\in E(G)} 1/2\left(k_i + k_j\right)\right]^2}{|E|^{-1}\sum_{e_{ij}\in E(G)} 1/2\left(k_i^2 + k_j^2\right) - \left[|E|^{-1}\sum_{e_{ij}\in E(G)} 1/2\left(k_i + k_j\right)\right]^2} \tag{3}$$



(a) (b) (c)

Fig. 4. Average in-degree of the closest-neighbor nodes of node $i$ with out-degree $k_i^{\text{out}}$. (a) NFT. (b) ETH. (c) BTC.

nodes while low-degree nodes also prefer linking with high-degree nodes.

However, the negative value of $\rho$ cannot fully indicate the *disassortativity*. We also adopt the following measure $k^{\text{cn-in}}(k^{\text{out}})$ to describe the average in-degree of the closest-neighbor nodes of node $i$ that has out-degree $k_i^{\text{out}}$ [42]

$$k^{\text{cn-in}}\left(k^{\text{out}}\right) = \sum_{i=1,k_i^{\text{out}}=k^{\text{out}}}^{N} \left(\frac{k_i^{\text{cn-out}}}{N}\right) P\left(k^{\text{out}}\right) \tag{4}$$

where $k_i^{\text{cn-out}} = \sum_{j=1}^{N} a_{ij} k_j^{\text{in}} / k_i^{\text{out}}$, $a_{ij}$ is the $(i, j)$th entry of the adjacency matrix in (1), and $P(k^{\text{out}})$ is the out-degree distribution function. If the value of $k^{\text{cn-in}}(k^{\text{out}})$ shows a downward trend with respect to the variable $k^{\text{out}}$, then the graph is disassortative, as shown in the results in Fig. 4. It means that high-degree nodes prefer connecting to low-degree nodes while low-degree nodes also prefer connecting to high-degree nodes. This effect can also be explained by the *preferential attachment*, in which newly-joined nodes prefer connecting to high-degree nodes (i.e., the rich get richer) [22].

### *D. Rich-Club Coefficient*

In addition to disassortativity, we also focus on the connectivity propensity between nodes with high degrees. In complex networks, the rich club refers to the phenomenon of a tight connection between high-degree nodes. In other words, the nodes with a large number of edges are regarded as the rich nodes, which nevertheless are more likely to gather into clubs in contrast to those low-degree nodes. We denote the *rich-club coefficient* by $\phi(k)$, which is defined as follows [46]:

$$\phi(k) = \frac{2E_{>k}}{N_{>k}(N_{>k} - 1)} \tag{5}$$

where $N_{>k}(N_{>k} - 1)/2$ is the maximum possible edges of all $N_{>k}$ nodes whose degree is higher than $k$; similarly, $E_{>k}$ denotes the number of edges among $N_{>k}$ nodes. It is worth noting that the rich-club coefficient can be regarded as a more specific measurement than the assortativity coefficient since the rich-club coefficient focuses on the possibility of connection to a node over degree $k$. For example, a network with several rich nodes and some low-degree nodes exhibits disassortativity because the rich nodes are not directly connected; however, it still shows the rich-club phenomenon if the rich nodes are closely connected in the same sub-graph. Only when the graph is disassortative and there is no rich-club ordering, we can say the center nodes are far from each other.

Fig. 5 plots the rich club ordering of NFT, ETH, and BTC networks. We can observe from Fig. 5(a)–(c) that the rich-club coefficient $\phi(k)$ does not monotonically increase with the increment of $k$, implying no obvious rich-club phenomenon in all the three BCNs. However, as discussed in [46], the rich-club coefficient alone cannot well reflect the rich-club effect of large networks. For a more accurate evaluation, we also compare the BCNs with corresponding random networks. Thus, we adopt the normalized rich-club coefficient denoted by $\phi_{\text{norm}}(k)$ as follows:

$$\phi_{\text{norm}}(k) = \frac{\phi(k)}{\phi_{\text{rand}}(k)} \tag{6}$$

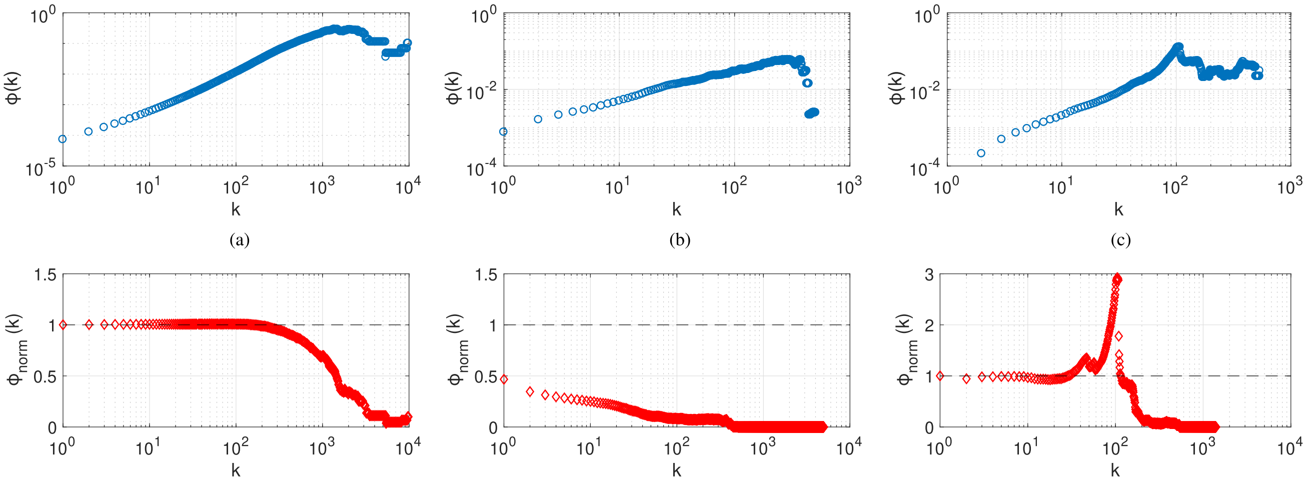
where $\phi(k)$ is the rich-club coefficient of the BCN and $\phi_{\text{rand}}(k)$ is the rich-club coefficient of a random network with the same degree distribution. Fig. 5(d)–(f) plots the normalized rich-club coefficient, whereas the rich-club ordering, depends on whether $\phi_{\text{norm}}(k) > 1$. The results also reveal an absence of rich-club ordering on most $k$ values.

In general, the BCNs exhibit the "non-rich-club" phenomenon, implying that the high-degree central nodes in these networks tend to disconnect from each other. They are evenly distributed in different connective sub-graphs. This effect can be explained by the fact that the central nodes are more likely to be some exchanges or large institutions with their own relatively-fixed customer groups. Therefore, the rich nodes are not closely connected to each other in the BCNs.

### *E. Summary*

Through the complex network analysis, we found that BCNs are scale-free networks that follow the power-law distribution with a heavy tail. In other words, when a node is randomly selected from these networks, there is a high probability that this node has many similar counterparts (according to the node-degree distribution). On the other hand, we also reveal the disconnection, disassortativity, and non-rich-club effect of BCNs. The disassortativity implies that neighbors of a node are not similar to each other most of the time in these networks. Meanwhile, the non-rich-club effect indicates that multiple central nodes serve as hubs, which do not tend to connect.

These observations imply that existing representation learning methods, such as DeepWalk and node2vec may not be suitable for BCNs since they have inconsistent assumptions



(a) (b) (c)

(d) (e) (f)

Fig. 5. Rich club ordering of (a) and (d) NFT, (b) and (e) ETH, and (c) and (f) BTC networks.

with the above observations (e.g., they are based on the assumption that the closest neighbors are similar to each other). In addition, the traditional schemes are also limited by node distance and network connectivity. For instance, cryptocurrency exchanges in DeFi services tend to be embedded as completely different representations [47] because they are usually not directly connected, even though they play similar roles and have similar transaction patterns.

## IV. Structural Identity Representation

Structural identity can represent the entire structure of the BCN and express the relationships among multiple entities. Therefore, given the characteristics of BCNs, it is of great significance to design a suitable representation learning method that can capture the structural identity.

To this end, s2v methods are able to learn the structural identity of complex networks. After measuring node similarity from a structural-role perspective, s2v methods construct a multilayer graph with different scales. Similar to other representation learning methods, s2v methods also require *random walk* operations. However, unlike other representation learning methods, such as DeepWalk and node2vec, s2v methods do not require a fully-connected original network when conducting random walks on the multilayer graph.

Inspired by the above observations of BCNs and the merits of s2v methods, we propose a novel structural identity representation method, namely SVRP for BCNs. Unlike other s2v methods, we design a novel random walk strategy–*random pace* to conduct a *flexible* random walk while pacing around. Consequently, our random pace strategy can sample more similar neighbors for the target node, and better retain structural characteristics of BCNs, thereby achieving more accurate representation learning than existing methods. Fig. 6 depicts the working flow of our SVRP.

### A. s2v With Random Pace

In our SVRP, the structural distance between nodes $u$ and $v$ in their $k$-hop neighborhoods is denoted by $f_k(u,v)$, which is expressed as $f_k(u,v) = f_{k-1}(u,v) + g(s[R_k(u)], s[R_k(v)])$, where $R_k(u)$ denotes the set of nodes at distance $k$ from $u$ in $G$ ($k \geq 0$, $f_{-1} = 0$, $|R_k(u)| > 0$ and $|R_k(v)| > 0$), $s(S)$ denotes the ordered degree sequence of a node set $S$, and $g[s_1, s_2]$ measures the distance between the ordered degree sequences $s_1$ and $s_2$ [9].

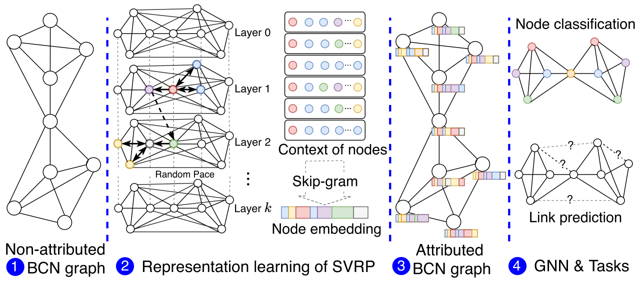
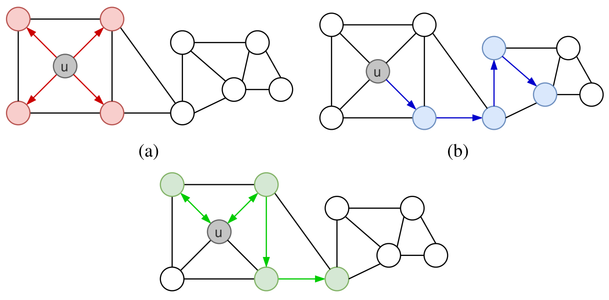


Fig. 6. Overview of the proposed framework.

Our SVRP constructs a multilayer weighted graph $M$ while layer $k$ is defined by the $k$-hop neighborhoods of the node. We use the multilayer graph $M$ for each node in $V$ to generate node-sequence contexts (as shown in Step 2 of Fig. 6).

Different from conventional s2v methods, we design a novel *random pace* strategy for generating more accurate structure contexts. Similar to struc2vec [9], SVRP first determines which layer to walk. At each step, SVRP then determines two random states with respect to the decaying factor $d$: 1) *transferring* step or 2) *pacing* around. If the state is "transfer," the walk continues after choosing a neighbor to walk; the state is "pace," otherwise. Accordingly, several neighbors are added to the walking path while this strategy does not change the starting step next time, acting like "pacing" around.

As shown in Fig. 7(a) and (b), traditional search strategies, such as Breadth-first Sampling (BFS) and Depth-first Sampling (DFS) focus on the local and global characteristics, respectively. By contrast, the random-pace strategy in Fig. 7(c) provides a more adaptive manner for neighbor sampling. We denote the probabilities of passing by a neighbor node via "transfer" and "pace" by $P_{\text{trans}}$ and $P_{\text{pace}}$, respectively. They are



(a)

(b)

(c)

Fig. 7. Different random-walk strategies (walk length $l = 4$). (a) BFS. (b) DFS. (c) Random pace ($s = 2$).

expressed by $P_{\text{trans}}(u, v) = P_k(u, v)(1-d)$ and $P_{\text{pace}}(u, v) = dP_k \sum_{i=1}^{s}(1 - P_k(u, v))^{i-1}$, where $s$ is the number of pacing steps and $P_k$ is the probability of choosing neighbors in layer $k$. Meanwhile, $P_k(u, v)$ denotes the probability that node $u$ chooses its neighbor node $v$ in layer $k$. In particular, $P_k(u, v)$ is expressed as follows [9]:

$$P_k(u, v) = \frac{e^{-f_k(u,v)}}{\sum_{\substack{v \in V \\ v \neq u}} e^{-f_k(u,v)}}. \tag{7}$$

We then have Markov Chain transition matrix $\mathbf{P}_k$ for the entire network by extending $P_k(u, v)$

$$\mathbf{P}_k = \begin{pmatrix} P_k(1,1) & \cdots & P_k(1,v) & \cdots & P_k(1,n) \\ \vdots & & \vdots & & \vdots \\ P_k(u,1) & \cdots & P_k(u,v) & \cdots & P_k(u,n) \\ \vdots & & \vdots & & \vdots \\ P_k(n,1) & \cdots & P_k(n,v) & \cdots & P_k(n,n) \end{pmatrix} \tag{8}$$

where $n$ is the number of vertices. We then have the probability of selecting $v$ from the neighbors of $u$ in one action

$$P(u, v) = (1-d)P_k(u, v) + dP_k \sum_{i=1}^{s} (1 - P_k(u, v))^{i-1}. \tag{9}$$

We next express (9) via the above matrix as follows:

$$\mathbf{P} = \mathbf{P}_k + d \sum_{i=1}^{s-1} \left(\mathbf{P}_k^{-}\right)^i \tag{10}$$

where $\mathbf{P}_k^{-}$ is denoted by

$$\mathbf{P}_k^{-} = \begin{pmatrix} 1 - P_k(1,1) & \cdots & 1 - P_k(1,v) & \cdots & 1 - P_k(1,n) \\ \vdots & & \vdots & & \vdots \\ 1 - P_k(u,1) & \cdots & 1 - P_k(u,v) & \cdots & 1 - P_k(u,n) \\ \vdots & & \vdots & & \vdots \\ 1 - P_k(n,1) & \cdots & 1 - P_k(n,v) & \cdots & 1 - P_k(n,n) \end{pmatrix}. \tag{11}$$

Thereafter, for every node of $V$, we conduct $r$ independent walks with length $l$ representing the total number of steps for every walk.

Unlike Deepwalk and node2vec, the walk of SVRP is conducted on the multilayer network $M$, in which each layer is a complete graph with edges weighted by the structural distance. Since any pair of nodes in the complete network is connected, there is a chance to select all nodes except the current node when sampling the neighbors. In this case, the traditional biased random walk is not as advisable as the random pace to generate the context for the nodes. As mentioned in (7), the node-transfer probability in the random pace strategy depends on the structural similarity distance $f_k(u, v)$. In other words, the more similar node pair $(u, v)$ is, the higher probability $P_k(u, v)$ is. Thus, for each target node, SVRP can sample more similar neighbor nodes as context sequences, thereby being beneficial to the computing of embeddings when using NLP modules, such as Skip-Gram [48].

Moreover, we can implement BFS and DFS strategies by choosing different values of $s$ and $d$. When $s$ or $d$ is large, it can be regarded as BFS; when $s = 1$ or $d$ is small, it can be regarded as DFS. Finally, the finished walks of nodes can be regarded as the node sequence, which is the graph structure context for the NLP module to compute the embeddings. In SVRP, we choose the NLP context prediction algorithm, e.g., Skip-Gram [48], which has been proven to be superior to other existing methods in similar tasks. Skip-Gram has also been applied to sequence embeddings for many representation-learning methods [9], [18], [19]. Despite the effectiveness of Skip-Gram in learning node representation from graph data, we can achieve more accurate representation learning by providing the sequence of nodes with higher similarity to the target node. Algorithms 1 and 2 depict SVRP and RandomPace, respectively.

**Algorithm 1** SVRP

```
Input: $G(V, E)$.
Output: Embeddings of nodes $\{\mathbf{X}_v, \forall v \in V\}$.
  Initialization : (walk_iter = 0, Number of walks $r$, Walk
  length $l$, Decaying factor $d$, Pace step $s$ per node, Dimen-
  sions dim);
for $u \in V$ do
    $K \leftarrow$ Append node degree $k(u)$;
    return Degree list $K$
end for
for $k \in K$ do
    for node pair $(u, v) \in V$ do
        $S(u, v) \leftarrow$ structural distances $S(R_k(u), R_k(v))$;
        $M \leftarrow$ distance network $M_k = (V, E, W_k)$;
    end for
    return Multilayer graph $M$
end for
while walk_iter $<$ $r$ do
    for $u \in V$ do
        walk $\leftarrow$ RandomPace $(M, u, l, d, s)$;
        walks $\leftarrow$ Append walk;
    end for
    walk_iter $\leftarrow$ walk_iter + 1
end while
$\mathbf{X}_v \leftarrow$ SkipGram($dim$, walks);
return $\{\mathbf{X}_v, \forall v \in V\}$
```

## *B. Learning Edge Features*

It is important to learn edge features, especially for learning tasks done by GNNs. Thus, we present a semi-supervised

**Algorithm 2** RandomPace

```
Input: Multilayer graph $M$, Current node $u$, Walk length $l$,
    Decaying factor $d$, Pace step $s$.
Output: walk.
    Initialization : (step_iter = 0, pace_iter = $s$, walk $\leftarrow$ append
    $u$, current_layer $\leftarrow$ randomLayer($u$));
while step_iter $< l$ do
    $V_{\text{curr}} \leftarrow$ walk[$-1$];
    layer $\leftarrow$ chooseLayer(current_layer);
    if pace then
        while pace_iter $> 0$ do
            $V_{\text{step}} \leftarrow$ chooseNeighbor($V_{\text{curr}}$, $M$);
            walk $\leftarrow$ Append $V_{\text{step}}$;
            pace_iter $\leftarrow$ pace_iter $- 1$;
        end while
        step_iter $\leftarrow$ step_iter $+ s$;
    else
        $V_{\text{curr}} \leftarrow$ chooseNeighbor($V_{\text{curr}}$, $M$);
        walk $\leftarrow$ Append $V_{\text{curr}}$;
        step_iter $\leftarrow$ step_iter + 1;
    end if
end while
return walk
```

TABLE II
BINARY OPERATORS

| Operator | Definition |
|---|---|
| Weighted-L1 | $\|h(u) \cdot h(v)\|_{\bar{1}i} = \|h_i(u) - h_i(v)\|$ |
| Weighted-L2 | $\|h(u) \cdot h(v)\|_{\bar{2}i} = \|h_i(u) - h_i(v)\|^2$ |
| ip | $\|h(u) \cdot h(v)\| = \sum_{i=1}^{\dim} [h_i(u) \cdot h_i(v)]$ |

method to learn edge representation for the BCN. Since our random pace strategy requires walks crossing layers among the multilayer graph, it is efficient to generate edge features along with the representation of nodes. SVRP provides Skip-Gram with appropriate node contexts to learn node embeddings while also considering the edge representation. We construct the edge features by aggregating the embeddings of nodes with various binary operators. Let $F(u)$ represent the features learned from node $u$ to the $i$th component. Then, the features of edge $e_{uv}$ can be generated by both $F(u)$ and $F(v)$. We consider three binary operators [19]: Weighted-$L1$, Weighted-$L2$ and ip defined as in Table II, where dim denotes the dimension of node embeddings. For each node pair $(u, v)$, we aggregate their embeddings to generate the edge representation between them even if there is no existing connection. Thus, there are positive edge features and negative edge features (nonexistent edges) for downstream tasks such as link prediction.

## V. Experimental Evaluation

### A. Label Data Collection

To evaluate the performance of the proposed SVRP, we construct five main datasets. In particular, we first obtain the addresses of the NFT-transferring transactions from "`Moonstream`" of the dataset [43]. Consequently, we then construct the NFT dataset, in which the addresses are labeled as three different trader types. Secondly, we use the ETH phishing-node labels from [44], in which the addresses are classified as either phishing nodes or non-phishing nodes.

TABLE III
INFORMATION OF DATASETS

| Datasets | #Nodes | #Edges | #Labels | #Classes |
|---|---|---|---|---|
| NFT [43] | 509577 | 2922108 | 509577 | 3 |
| ETH [44] | 100804 | 108647 | 3360 | 3 |
| BTC [17] | 168,677 | 370,823 | 8,800 | 7 |
| Ronin bridge | 178 | 185 | - | - |
| 3Kv [49] | 347 | 350 | - | - |

Thirdly, we obtain the BTC address-label dataset from Harvard dataverse [17] including six different instance labels: "`mining pool`," "`miner`," "`coinjoin`," "`gambling`," "`exchanges`," and "`service`." Moreover, we construct the Roninbridge graph of the ETH attack event activities, and $3Kv$ graph on the BTC mixing-coin services provided by [49]. Table III summarizes the details of all the mentioned datasets.

### B. Case Study: Ronin Bridge

We conduct the case study by tracking the attack event of the most popular NFT game–Axie Infinity, whose side-chain Ronin bridge was exploited for 173 600 ETH on 23 March 2022. Since the hacker's wallet account $0 \times$`098b716b8aaf21512996dc57eb0615e2383e2f96` (in short $0 \times 098$) is publicly visible, we then represent the fund-transferring activities as a local sub-graph, Fig. 8(a) depicts the visualization. For easy illustration, we denote hacker addresses in red and suspicious transfer addresses in varied-hue orange colors according to the amount of ETH transferred. We initially observe that the $0 \times 098$ graph shares similar properties to the typical BCNs, i.e., power-law distribution (the majority of nodes have only a few connections), and the multicenter effect.

We then perform representation learning on this network using different embedding algorithms and visualize the results. Fig. 8(b)–(e) show the latent representations in two dimensions for the $0 \times 098$ graph learned by DeepWalk, node2vec, struct2vec, and our SVRP, respectively. It is worth mentioning that the malicious or suspicious nodes (in red) should be captured and distinguished from the others in the ideal representations of learning results.

As shown in Fig. 8(b) and (c), the malicious nodes (red nodes) are mixed with others (i.e., blue hollow nodes), implying that DeepWalk and node2vec cannot capture the structural identities of nodes. This is because DeepWalk and node2vec mainly work for assortative networks while the BCNs do not fulfill this feature. By contrast, s2v schemes (e.g., struct2vec and our SVRP) show better performance than DeepWalk and node2vec. For example, Fig. 8(d) illustrates that struct2vec differentiates many suspicious nodes from the rest of the graph though there is still no clear boundary between suspect nodes and common nodes. Thus, it is still difficult for struct2vec to distinguish them completely. Moreover, Fig. 8(e) demonstrates that our SVRP clearly clusters similar type of nodes so as to easily distinguish the common nodes from the malicious nodes, implying that our SVRP well preserves graph structural identities.

### C. Case Study: 3Kv Graph

Mixing services (also known as tumblers) are used to mix one's own cryptocurrency tokens with other users, designed

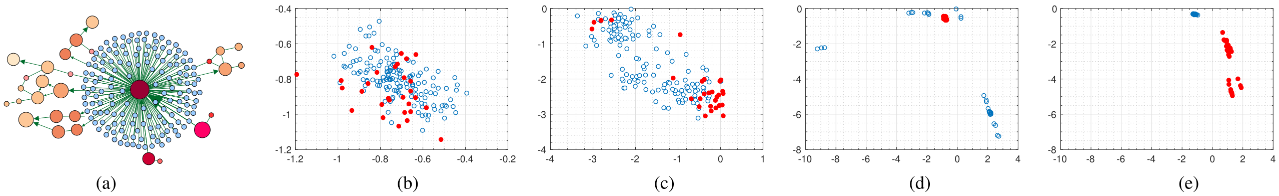


Fig. 8. Case study: Ronin bridge attack event. (a) 0 × 098 graph. (b) DeepWalk. (c) Node2vec. (d) Struc2vec. (e) SVRP.

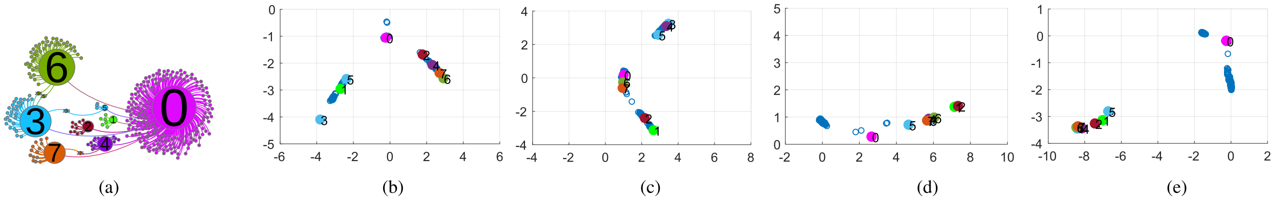


Fig. 9. Case study: BTC mixing service. (a) $3K\upsilon$ graph. (b) DeepWalk. (c) Node2vec. (d) S2V. (e) SVRP.

to obfuscate the original source of funds. In the case study of $3K\upsilon$ graph, we track a BTC Mixing Service [50] account $3K\upsilon$ (3Kvvf2yAQScujZpQj32ii2NjXZnBBpc6Dp) and obtain 350 transactions from timestamp 31 March 2019. We then represent $3K\upsilon$ as a local sub-graph based on the transfer direction of funds. For simplicity, we use numeric IDs instead of addresses, where node 0 is the initial Mixing Service address of $3K\upsilon$. Fig. 9(a) depicts the visualization of $3K\upsilon$. We initially observe that the $3K\upsilon$ graph shares similar properties to the BTC graph, i.e., disassortativity and non-rich-club effect. Moreover, we also observe that nodes from zero to seven are central nodes with more connections while they are not directly connected with each other [i.e., indirectly connected through some bridging nodes such as nodes 239, 256, 257, and 275 in Fig. 9(a)].

Fig. 9(a), (c)–(e) show the latent representations in two dimensions for the $3K\upsilon$ graph learned by DeepWalk, node2vec, struc2vec, and SVRP, respectively. It is worth mentioning that some prominent nodes (e.g., nodes from 0 to 7) should be captured and distinguished from the others in the ideal representations of learning results.

In Fig. 9(b) and (c), nodes 0–7 are mingling with normal nodes, implying that DeepWalk and node2vec fail to capture the structural identities of nodes, similar to the $0 \times 098$ graph. This is because DeepWalk and node2vec mainly work for assortative networks while BCNs are not (see Section III). Moreover, the random walk-based representation learning is limited by the window size of Skip-Gram. By contrast, s2v schemes also show better performance than DeepWalk and node2vec. For example, Fig. 9(d) illustrates that s2v differentiates central nodes from the rest of the graph though the distance between similar nodes is still relatively far and there is no tendency for clustering. Moreover, Fig. 9(e) demonstrates that our SVRP clearly clusters similar nodes, which have long distances from each other, implying that SVRP well preserves the structural identities of nodes.

### D. Node Classification

In order to better perform node classification tasks, we adopt GNNs to further explore BCNs. GNNs have been widely used in various graph-related tasks due to their superior performance by appropriately learning graph natural features. In particular, we take the embeddings from representation learning as the node features for the non-attributed BCN graphs (i.e., original graph topology without feature data), and construct the graph with attributes (attributed graph), i.e., Step 3 as shown in Fig. 6. We then adopt graph sample and aggregate (GraphSAGE) [51] with sum aggregator to conduct a fair evaluation on the effectiveness of learning node features for different node-feature initialization (NFI) methods across eight types of graph-mining tasks.

We consider the following methods: 1) six intuitive methods focusing on the structural aspects: *random normal*, *one hot*, *degree*, *PageRank*, *shared*, and *bucket range*; and 2) four embedding methods: DeepWalk, node2vec, struc2vec, and SVRP. To confirm the applicability of the model, for each dataset, we measure its Acc, and weighted-averaged $F1$ score ($F1$) as in (12) and (15) on different methods. We consider parameters for all the methods as follows: the number of walks per node $r = 10$, walk length $l = 40$, Skip-Gram window size

TABLE IV
NODE CLASSIFICATION RESULTS OF GNN (GRAPHSAGE)

| Algorithms | NFT | | ETH | | BTC | |
|---|---|---|---|---|---|---|
| | Acc | F1 | Acc | F1 | Acc | F1 |
| Random normal [51] | 0.478 | 0.464 | 0.986 | 0.800 | 0.923 | 0.907 |
| One hot [52] | 0.475 | 0.467 | 0.987 | 0.803 | 0.941 | 0.910 |
| Degree [51] | 0.601 | 0.466 | 0.988 | 0.782 | 0.945 | 0.924 |
| Pagerank [53] | 0.607 | 0.458 | 0.986 | 0.769 | 0.941 | 0.923 |
| Bucket range [54] | 0.606 | 0.459 | 0.988 | 0.821 | 0.941 | 0.929 |
| Shared [55] | 0.607 | 0.459 | 0.986 | 0.773 | 0.941 | 0.923 |
| DeepWalk [18] | 0.502 | 0.479 | 0.991 | 0.853 | 0.954 | 0.946 |
| Node2vec [19] | 0.538 | 0.480 | 0.989 | 0.805 | 0.954 | 0.944 |
| Struc2vec [9] | 0.579 | 0.485 | 0.991 | 0.852 | 0.957 | 0.950 |
| Our SVRP | **0.610** | **0.511** | **0.993** | **0.885** | **0.971** | **0.967** |

TABLE V
NODE CLASSIFICATION RESULTS OF LR

| Algorithms | NFT | | ETH | | BTC | |
|---|---|---|---|---|---|---|
| | Acc | F1 | Acc | F1 | Acc | F1 |
| DeepWalk [18] | 0.450 | 0.357 | 0.457 | 0.383 | 0.603 | 0.453 |
| Node2vec [19] | 0.460 | 0.371 | 0.558 | 0.451 | 0.682 | 0.619 |
| Struc2vec [9] | 0.571 | 0.471 | 0.774 | 0.765 | 0.877 | 0.865 |
| SVRP | **0.583** | **0.476** | **0.794** | **0.787** | **0.895** | **0.883** |

TABLE VI
LINK PREDICTION OF LR AND GNN

| Algorithms | | Datasets | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | NFT | | | ETH | | | BTC | | |
| | | **L2** | **L1** | **ip** | **L2** | **L1** | **ip** | **L2** | **L1** | **ip** |
| **LR** | DeepWalk [18] | 0.565 | 0.557 | 0.563 | 0.593 | 0.607 | 0.536 | 0.587 | 0.630 | 0.670 |
| | Node2vec [19] | 0.576 | 0.579 | 0.574 | 0.625 | 0.641 | 0.538 | 0.678 | 0.649 | 0.680 |
| | Struc2vec [9] | 0.772 | 0.804 | 0.815 | 0.670 | 0.682 | 0.742 | 0.669 | 0.689 | 0.924 |
| | SVRP | **0.775** | **0.828** | **0.838** | **0.746** | **0.782** | **0.755** | **0.919** | **0.931** | **0.947** |
| **GNN** | DeepWalk$_{\text{GCN}}$ | 0.547 | 0.554 | 0.624 | 0.923 | 0.931 | 0.920 | 0.873 | 0.882 | 0.841 |
| | Node2vec$_{\text{GCN}}$ | 0.566 | 0.552 | 0.637 | 0.931 | 0.927 | 0.924 | 0.898 | 0.822 | 0.885 |
| | Struc2vec$_{\text{GCN}}$ | 0.699 | 0.709 | 0.753 | 0.931 | 0.934 | 0.938 | 0.909 | 0.930 | 0.867 |
| | SVRP$_{\text{GCN}}$ | **0.853** | **0.862** | **0.824** | **0.937** | **0.944** | **0.945** | **0.934** | **0.931** | **0.912** |
| | DeepWalk$_{\text{GraphSAGE}}$ | 0.926 | 0.907 | 0.892 | 0.925 | 0.896 | 0.864 | 0.952 | 0.955 | 0.876 |
| | Node2vec$_{\text{GraphSAGE}}$ | 0.885 | 0.957 | 0.887 | 0.936 | 0.957 | 0.914 | 0.960 | 0.953 | 0.874 |
| | Struc2vec$_{\text{GraphSAGE}}$ | 0.961 | 0.979 | 0.889 | 0.961 | 0.958 | 0.927 | 0.960 | 0.937 | 0.871 |
| | SVRP$_{\text{GraphSAGE}}$ | **0.984** | **0.987** | **0.918** | **0.979** | **0.967** | **0.940** | **0.965** | **0.968** | **0.885** |

$w = 5$; $p = 1$ and $q = 2$ for node2vec; pace step per node $s = 2$ and decaying factor $d = 0.25$ for SVRP.

In particular, the metrics are defined as in (12)–(15), shown at the bottom of the page.

Table IV lists the results of GNN classification. We observe that embedding methods perform better than intuitive NFI methods. While in intuitive methods, the *bucket range* method outperforms the others. Meanwhile, over all datasets, our SVRP obtains the best results among all the methods, e.g., reaching 0.993 of Acc on the ETH graph and 0.967 of $F1$ score on the BTC graph. On the same dataset, compared with other existing work [17], additional artificially calculated features are used to reach the best $F1$ score of 0.960, but the SVRP result of 0.967 is still higher than that without using the node attribution data. In other words, in the SVRP method, we used fewer data to achieve approximate performance.

To further confirm that GNN is the most suitable classifier model, specifically, we apply the latent representation embeddings to train a one-versus-rest logistic regression (LR) classifier model. We observe from Table V that the overall performance of all the methods is not as good as the results of GNNs (as shown in Table IV). Similarly, the proposed SVRP still outperforms other representation learning methods in LR.

### *E. Link Prediction*

Link prediction tasks aim to predict the occurrence of links in a given graph based on observed information. As mentioned in Section IV-B, we choose three binary operators: Weighted-$L1$, Weighted-$L2$, and ip for learning edge features. After conducting link prediction tasks on the three datasets, we obtain the results of LR and the GNN method in terms of area under curve (AUC), as shown in Table VI. Regarding the LR classifier, we use the one-versus-rest scheme and cross-entropy loss. Regarding the GNN approaches, we also consider graph convolutional networks (GCN) [56] as another comparison method in addition to GraphSage.

We observe from the results that our SVRP outperforms other methods on all three datasets, especially for NFT (i.e., reaching the highest AUC of 0.987). Meanwhile, GNN methods perform better than basic LR methods. Moreover, GraphSAGE shows superior results than GCN. In addition, the results manifest that the representation learning of structural identities is quite meaningful for edge analysis and is beneficial for understanding BCNs. And it also demonstrates that our SVRP indeed helps to obtain a more comprehensive link representation for predictive tasks than other methods.

## VI. CONCLUSION

In this article, we accomplish a series of analyses and in-depth mining on the metaverse BCNs. First, we conduct the complex network analysis of three BCNs that play significant roles in the metaverse. Secondly, based on the novel findings as characteristics of BCNs (i.e., power-law distribution, disassortativity, disconnection, one-way deals, non-rich-club phenomenon), we proposed the SVRP method to model the network via a structural identity representation. Thirdly, two

$$\text{Acc} = \frac{\text{true positive} + \text{true negative}}{\text{true positive} + \text{false positive} + \text{false negative} + \text{true negative}} \tag{12}$$

$$\text{Precision} = \frac{\text{true positive}}{\text{true positive} + \text{false positive}} \tag{13}$$

$$\text{Recall} = \frac{\text{true positive}}{\text{true positive} + \text{false negative}} \tag{14}$$

$$F1\ \text{score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{15}$$

cases of real-time malicious activities study confirmed that SVRP is quite fitting for the representation learning of the current blockchain transaction networks. It can distinguish nodes with more significant structural identities from the others when creating embeddings.

Last but not least, we applied our method to the blockchain graphs in the GNN node classification task, and then conducted a comparative experiment with other representation learning and intuitive NFI methods. Our results demonstrate that our SVRP reaches 96.7% in the $F1$ score, which is the best performance among all the compared methods in non-attributed data. We also paralleled and compared the combination of the same representation learning method to the LR classifier on the classification task. Moreover, we found that the GNN model achieves better performance than LR when dealing with the classification of network data. Hereafter, we show the superiority of the SVRP in link prediction tasks. It means that our method can be well adapted to a variety of network analysis tasks. Such findings might lead us to a better understanding of the structural behavior of blockchain cryptocurrency transaction networks. Meanwhile, it may provide insightful implications for detecting mixing services, malicious addresses, and fund-tracing tasks in metaverse security problems in the future.

## REFERENCES

[1] H. Duan, J. Li, S. Fan, Z. Lin, X. Wu, and W. Cai, "Metaverse for social good: A university campus prototype," in *Proc. 29th ACM Int. Conf. Multimedia*, Oct. 2021, pp. 153–161.

[2] F. Victor and A. M. Weintraud, "Detecting and quantifying wash trading on decentralized cryptocurrency exchanges," in *Proc. Web Conf.*, Apr. 2021, pp. 23–32.

[3] L. Kugler, "Non-fungible tokens and the future of art," *Commun. ACM*, vol. 64, no. 9, pp. 19–20, Sep. 2021.

[4] V. Hassija, V. Saxena, and V. Chamola, "A mobile data offloading framework based on a combination of blockchain and virtual voting," *Softw., Pract. Exper.*, vol. 51, no. 12, pp. 2428–2445, Dec. 2021.

[5] J. Lorenz, M. I. Silva, D. Aparício, J. T. Ascensão, and P. Bizarro, "Machine learning methods to detect money laundering in the Bitcoin blockchain in the presence of label scarcity," in *Proc. 1st ACM Int. Conf. AI Finance*, Oct. 2020, pp. 1–8.

[6] N. Gandal, J. Hamrick, T. Moore, and T. Oberman, "Price manipulation in the Bitcoin ecosystem," *J. Monetary Econ.*, vol. 95, pp. 86–96, May 2018.

[7] Y. Boshmaf, C. Elvitigala, H. A. Jawaheri, P. Wijesekera, and M. A. Sabah, "Investigating MMM Ponzi scheme on Bitcoin," in *Proc. 15th ACM Asia Conf. Comput. Commun. Secur.*, Oct. 2020, pp. 519–530.

[8] K. Wang et al., "A large-scale empirical analysis of ransomware activities in Bitcoin," *ACM Trans. Web*, vol. 16, no. 2, pp. 1–29, 2021.

[9] L. F. R. Ribeiro, P. H. P. Saverese, and D. R. Figueiredo, "struc2vec: Learning node representations from structural identity," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2017, pp. 385–394.

[10] I. Alarab, S. Prakoonwit, and M. I. Nacer, "Comparative analysis using supervised learning methods for anti-money laundering in Bitcoin," in *Proc. 5th Int. Conf. Mach. Learn. Technol.*, New York, NY, USA, Jun. 2020, pp. 11–17.

[11] G. C. Cerda and J. L. Reutter, "Bitcoin price prediction through opinion mining," in *Proc. Companion World Wide Web Conf.*, New York, NY, USA, May 2019, pp. 755–762.

[12] R. S. Portnoff, D. Y. Huang, P. Doerfler, S. Afroz, and D. McCoy, "Backpage and Bitcoin: Uncovering human traffickers," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, New York, NY, USA, Aug. 2017, pp. 1595–1604.

[13] A. Turner, S. Mccombie, and A. Uhlmann, "Follow the money: Revealing risky nodes in a ransomware-Bitcoin network," in *Proc. Annu. Hawaii Int. Conf. Syst. Sci.*, 2021, pp. 1560–1572.

[14] J. Liang, L. Li, W. Chen, and D. Zeng, "Targeted addresses identification for Bitcoin with network representation learning," in *Proc. IEEE Int. Conf. Intell. Secur. Informat. (ISI)*, Shenzhen, China, Jul. 2019, pp. 158–160.

[15] J. Liang, L. Li, S. Luan, L. Gan, and D. Zeng, "Bitcoin exchange addresses identification and its application in online drug trading regulation," in *Proc. 23rd Pacific Asia Conf. Inf. Syst., Secure ICT Platform 4th Ind. Revolution*, Xi'an, China, 2019, pp. 1–9.

[16] Y. Hu, S. Seneviratne, K. Thilakarathna, K. Fukuda, and A. Seneviratne, "Characterizing and detecting money laundering activities on the Bitcoin network," 2019, *arXiv:1912.12060*.

[17] R. Michalski, D. Dziubaltowska, and P. Macek, "Revealing the character of nodes in a blockchain with supervised learning," *IEEE Access*, vol. 8, pp. 109639–109647, 2020.

[18] B. Perozzi, R. Al-Rfou, and S. Skiena, "DeepWalk: Online learning of social representations," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2014, pp. 701–710.

[19] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 855–864.

[20] D. Ron and A. Shamir, "Quantitative analysis of the full Bitcoin transaction graph," in *Proc. Int. Conf. Financial Cryptogr. Data Secur.* Berlin, Germany: Springer, 2013, pp. 6–24.

[21] A. Baumann, B. Fabian, and M. Lischke, "Exploring the Bitcoin network," in *Proc. WEBIST*, Barcelona, Spain, 2014, pp. 369–374.

[22] D. Kondor, M. Pósfai, I. Csabai, and G. Vattay, "Do the rich get richer? An empirical analysis of the Bitcoin transaction network," *PLoS ONE*, vol. 9, no. 2, Feb. 2014, Art. no. e86197.

[23] M. Lischke and B. Fabian, "Analyzing the Bitcoin network: The first four years," *Future Internet*, vol. 8, no. 4, p. 7, Mar. 2016.

[24] I. Alqassem, I. Rahwan, and D. Svetinovic, "The anti-social system properties: Bitcoin network data analysis," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 50, no. 1, pp. 21–31, Jan. 2020.

[25] M. A. Javarone and C. S. Wright, "From Bitcoin to Bitcoin cash: A network analysis," in *Proc. 1st Workshop Cryptocurrencies Blockchains Distrib. Syst.*, New York, NY, USA, Jun. 2018, pp. 77–81.

[26] P. Nerurkar, D. Patel, Y. Busnel, R. Ludinard, S. Kumari, and M. K. Khan, "Dissecting Bitcoin blockchain: Empirical analysis of Bitcoin network (2009–2020)," *J. Netw. Comput. Appl.*, vol. 177, Mar. 2021, Art. no. 102940.

[27] J. Liang, L. Li, and D. Zeng, "Evolutionary dynamics of cryptocurrency transaction networks: An empirical study," *PLoS ONE*, vol. 13, no. 8, Aug. 2018, Art. no. e0202202.

[28] A. P. Motamed and B. Bahrak, "Quantitative analysis of cryptocurrencies transaction graph," *Appl. Netw. Sci.*, vol. 4, no. 1, pp. 1–21, Dec. 2019.

[29] P. Monamo, V. Marivate, and B. Twala, "Unsupervised learning for robust Bitcoin fraud detection," in *Proc. Inf. Secur. South Afr. (ISSA)*, Johannesburg, South Africa, Aug. 2016, pp. 129–134.

[30] D. Di Francesco Maesa, A. Marino, and L. Ricci, "Uncovering the Bitcoin blockchain: An analysis of the full users graph," in *Proc. IEEE Int. Conf. Data Sci. Adv. Anal. (DSAA)*, Montreal, QC, Canada, Oct. 2016, pp. 537–546.

[31] T. Min and W. Cai, "Portrait of decentralized application users: An overview based on large-scale Ethereum data," *CCF Trans. Pervasive Comput. Interact.*, vol. 4, no. 2, pp. 1–18, 2022.

[32] X. Sun, M. Liu, and Z. Sima, "A novel cryptocurrency price trend forecasting model based on LightGBM," *Finance Res. Lett.*, vol. 32, Jan. 2020, Art. no. 101084.

[33] S. Lahmiri and S. Bekiros, "Cryptocurrency forecasting with deep learning chaotic neural networks," *Chaos, Solitons Fractals*, vol. 118, pp. 35–40, Jan. 2019.

[34] L. Alessandretti, A. ElBahrawy, L. M. Aiello, and A. Baronchelli, "Anticipating cryptocurrency prices using machine learning," *Complexity*, vol. 2018, Nov. 2018, Art. no. 8983590.

[35] C.-H. Wu, C.-C. Lu, Y.-F. Ma, and R.-S. Lu, "A new forecasting framework for Bitcoin price with LSTM," in *Proc. IEEE Int. Conf. Data Mining Workshops (ICDMW)*, Singapore, Nov. 2018, pp. 168–175.

[36] H. Zhang, S. Fan, Z. Fang, and W. Cai, "Economic analysis of decentralized exchange market with transaction fee mining," in *Proc. 4th ACM Int. Symp. Blockchain Secure Crit. Infrastruct.*, May 2022, pp. 59–70.

[37] D. Di Francesco Maesa, A. Marino, and L. Ricci, "Detecting artificial behaviours in the Bitcoin users graph," *Online Social Netw. Media*, vols. 3–4, pp. 63–74, Oct. 2017.

[38] Q.-B. Nguyen, A.-Q. Nguyen, V.-H. Nguyen, T. Nguyen-Le, and K. Nguyen-An, “Detect abnormal behaviours in Ethereum smart contracts using attack vectors,” in *Proc. Int. Conf. Future Data Secur. Eng.* Cham, Switzerland: Springer, 2019, pp. 485–505.

[39] A. Turner and A. S. M. Irwin, “Bitcoin transactions: A digital discovery of illicit activity on the blockchain,” *J. Financial Crime*, vol. 25, no. 1, pp. 109–130, Jan. 2018.

[40] L. Yu, N. Zhang, and W. Wen, “Abnormal transaction detection based on graph networks,” in *Proc. IEEE 45th Annu. Comput., Softw., Appl. Conf. (COMPSAC)*, Madrid, Spain, Jul. 2021, pp. 312–317.

[41] B. Tao, H.-N. Dai, H. Xie, and F. L. Wang, “Structural identity representation learning of blockchain transaction network for metaverse,” in *Proc. IEEE 24th Int. Workshop Multimedia Signal Process. (MMSP)*, Sep. 2022, pp. 1–6.

[42] G. Chen, X. Wang, and X. Li, *Fundamentals of Complex Networks: Models, Structures and Dynamics*. Singapore: Wiley, 2014.

[43] “Ethereum NFTs,” Ethereum, Bern, Switzerland, Tech. Rep., 2021. [Online]. Available: https://ethereum.org/en/nft/

[44] Z. Yuan, Q. Yuan, and J. Wu, “Phishing detection on Ethereum via learning representation of transaction subgraphs,” in *Proc. Int. Conf. Blockchain Trustworthy Syst.* Singapore: Springer, 2020, pp. 178–191.

[45] A. Clauset, C. R. Shalizi, and M. E. J. Newman, “Power-law distributions in empirical data,” *SIAM Rev.*, vol. 51, no. 4, pp. 661–703, 2009.

[46] V. Colizza, A. Flammini, M. A. Serrano, and A. Vespignani, “Detecting rich-club ordering in complex networks,” *Nature Phys.*, vol. 2, no. 2, pp. 110–115, Feb. 2006.

[47] V. Hassija, G. Bansal, V. Chamola, V. Saxena, and B. Sikdar, “BlockCom: A blockchain based commerce model for smart communities using auction mechanism,” in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, May 2019, pp. 1–6.

[48] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” in *Proc. Workshop ICLR*, 2013, pp. 1–12.

[49] J. Wu, J. Liu, W. Chen, H. Huang, Z. Zheng, and Y. Zhang, “Detecting mixing services via mining Bitcoin transaction network with hybrid motifs,” *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 52, no. 4, pp. 2237–2249, Apr. 2022.

[50] L. Wu et al., “Towards understanding and demystifying Bitcoin mixing services,” in *Proc. Web Conf.*, New York, NY, USA, 2021, pp. 33–44.

[51] W. L. Hamilton, R. Ying, and J. Leskovec, “Inductive representation learning on large graphs,” in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.* Red Hook, NY, USA: Curran Associates, 2017, pp. 1025–1035.

[52] J. You, R. Ying, and J. Leskovec, “Position-aware graph neural networks,” in *Proc. Int. Conf. Mach. Learn.*, Long Beach, CA, USA, 2019, pp. 7134–7143.

[53] S. Brin and L. Page, “The anatomy of a large-scale hypertextual web search engine,” *Comput. Netw. ISDN Syst.*, vol. 30, nos. 1–7, pp. 107–117, Apr. 1998.

[54] H. Cui, Z. Lu, P. Li, and C. Yang, “On positional and structural node features for graph neural networks on non-attributed graphs,” in *Proc. 31st ACM Int. Conf. Inf. Knowl. Manage.*, New York, NY, USA, Oct. 2022, pp. 3898–3902.

[55] F. Errica, M. Podda, D. Bacciu, and A. Micheli, “A fair comparison of graph neural networks for graph classification,” in *Proc. Int. Conf. Learn. Represent.*, Addis Ababa, Ethiopia, 2020, pp. 11–27.

[56] M. Welling and T. N. Kipf, “Semi-supervised classification with graph convolutional networks,” in *Proc. Int. Conf. Learn. Represent.*, 2017, pp. 1–15.

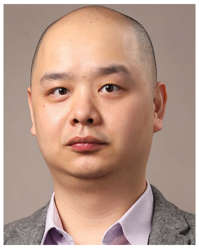
**Bishenghui Tao** (Graduate Student Member, IEEE) received the M.Sc. degree from The Hong Kong Polytechnic University, Hong Kong, in 2020. She is currently pursuing the Ph.D. degree with the Macau University of Science and Technology, Cotai, Macau.

She is also a Teaching Assistant with Hong Kong Metropolitan University, Hong Kong. Her research interests include complex networks, Metaverse, blockchain, and graph neural networks. She has extensive research experience and has publications in refereed journals and conferences.

**Hong-Ning Dai** (Senior Member, IEEE) received the Ph.D. degree in computer science and engineering from The Chinese University of Hong Kong, Hong Kong, in 2008.

He is currently an Associate Professor with the Department of Computer Science, Hong Kong Baptist University, Hong Kong. Before joining Hong Kong Baptist University, he was with the School of Computer Science and Engineering, Macau University of Science and Technology, Cotai, Macau, as an Assistant Professor/Associate Professor from 2010 to 2021; and the Department of Computing and Decision Sciences, Lingnan University, Hong Kong, as an Associate Professor from 2021 to 2022. He has coauthored/co-edited four monograph books and published more than 200 peer-reviewed papers in top-tier journals and conferences, including the IEEE COMMUNICATIONS SURVEYS & TUTORIALS, PROCEEDINGS OF THE IEEE, *Association for Computing Machinery (ACM) Computing Surveys*, IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, IEEE TRANSACTIONS ON MOBILE COMPUTING, IEEE TRANSACTIONS ON COMPUTERS, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, ACM/IEEE The International Conference on Automated Software Engineering (ASE), IEEE International Conference on Computer Communications (INFOCOM), and Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence (AAAI). His research interests include the Internet of Things, blockchains, and big data analytics.

Dr. Dai has served as an Associate Editor for IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, and *Ad Hoc Networks*.



**Haoran Xie** (Senior Member, IEEE) received the Ph.D. degree from the City University of Hong Kong, Hong Kong, in 2013, and the Ed.D. degree from the University of Bristol, Bristol, U.K., in 2022.

He is currently an Associate Professor with Lingnan University, Hong Kong. His research interests include artificial intelligence in education, big data, and educational technology. He has over 320 publications, including 170 journals like IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE (TPAMI), IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING (TKDE), IEEE TRANSACTIONS ON AFFECTIVE COMPUTING (TAFFC), and IEEE TRANSACTIONS ON CYBERNETICS (TCYB); and top-tier conferences like Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence (AAAI), International Joint Conference on Artificial Intelligence (IJCAI), Association for Computational Linguistics (ACL), International Conference on Empirical Methods in Natural Language Processing (EMNLP), International Conference on Computer Vision (ICCV), and The IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR).

Dr. Xie is a Senior Member of Association for Computing Machinery (ACM), has been listed as one of the World’s Top 2% Scientists by Stanford University, and was honored with 20 research awards. He is the Editor-in-Chief of *Natural Language Processing Journal*, *Computers and Education: Artificial Intelligence*, and *Computers and Education: X Reality*.

**Fu Lee Wang** (Senior Member, IEEE) received the Ph.D. degree from The Chinese University of Hong Kong, Hong Kong, in 2003.

He is currently the Dean and a Professor with the School of Science and Technology, Hong Kong Metropolitan University, Hong Kong. He has over 300 publications and 40 grants with more than 80 million Hong Kong dollars. His research interests include e-learning, information retrieval, educational data mining, recommender systems, and educational technology.

Dr. Wang is a fellow of British Computer Society (BCS), The Hong Kong Institution of Engineers (HKIE), and Institution of Engineering and Technology (IET). He was the Chair of the Association for Computing Machinery (ACM) Hong Kong Chapter and the IEEE Hong Kong Section Computer Society.