Cryptocurrency Transaction Network Embedding From Static and Dynamic Perspectives: An Overview

Yue Zhou, Xin Luo, *Senior Member, IEEE*, and MengChu Zhou, *Fellow, IEEE*

***Abstract*— Cryptocurrency, as a typical application scene of blockchain, has attracted broad interests from both industrial and academic communities. With its rapid development, the cryptocurrency transaction network embedding (CTNE) has become a hot topic. It embeds transaction nodes into low-dimensional feature space while effectively maintaining a network structure, thereby discovering desired patterns demonstrating involved users' normal and abnormal behaviors. Based on a wide investigation into the state-of-the-art CTNE, this survey has made the following efforts: 1) categorizing recent progress of CTNE methods, 2) summarizing the publicly available cryptocurrency transaction network datasets, 3) evaluating several widely-adopted methods to show their performance in several typical evaluation protocols, and 4) discussing the future trends of CTNE. By doing so, it strives to provide a systematic and comprehensive overview of existing CTNE methods from static to dynamic perspectives, thereby promoting further research into this emerging and important field.**

***Index Terms*—Big data analysis., cryptocurrency transaction network embedding (CTNE), dynamic network, network embedding, network representation, static network.**

I. Introduction

CRYPTOCURRENCY is a typical application of blockchain for facilitating verified transactions through Internet [1]. Different from the traditional currency requiring central authority to supervise transactions, it establishes distributed consensus-based protocols for efficient and secured transactions [2], [3]. With the rapid progress and widely applications of the blockchain technology [4]−[6], it developed rapidly in the past decade [7], [8]. To date, represented by the well-known Bitcoins, there are more than 5000 active cryptocurrencies in the market [9]−[11].

Manuscript received October 4, 2022; revised January 4, 2023; accepted January 19, 2023. This work was supported in part by the National Natural Science Foundation of China (62272078), the CAAI-Huawei MindSpore Open Fund (CAAIXSJLJJ-2021-035A), and the Doctoral Student Talent Training Program of Chongqing University of Posts and Telecommunications (BYJS202009). Recommended by Associate Editor Jiacun Wang. *(Corresponding authors: Xin Luo and MengChu Zhou.)*

Citation: Y. Zhou, X. Luo, and M. C. Zhou, "Cryptocurrency transaction network embedding from static and dynamic perspectives: An overview," *IEEE/CAA J. Autom. Sinica*, vol. 10, no. 5, pp. 1105–1121, May 2023.

Y. Zhou is with the School of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, the Chongqing Key Laboratory of Big Data and Intelligent Computing, Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences, Chongqing 400714, and the Chongqing School, University of Chinese Academy of Sciences, Chongqing 400714, China (e-mail: yueyzhou@outlook.com).

X. Luo is with the College of Computer and Information Science, Southwest University, Chongqing 400715, China (e-mail: luoxin@swu.edu.cn).

M. C. Zhou is with the School of Information and Electronic Engineering, Zhejiang Gongshang University, Hangzhou 310018, China, and also with the Helen and John C. Hartmann Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark NJ 07102 USA (e-mail: zhou@njit.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JAS.2023.123450

Therefore, cryptocurrency data analysis, as an emerging topic in academic communities, has attracted lots of attentions. Vujičić *et al.* [11] conduct a brief overview of two most frequently adopted cryptocurrencies, i.e., Bitcoin and Ethereum, and further analyze their differences. Conti *et al.* [12] conduct a comprehensive survey regarding Bitcoin's security and privacy, thereby further discussing the feasibility of their security and privacy protection schemes. Wang *et al.* [13] review the blockchain-enabled smart contracts that play a key role in cryptocurrency. Tschorsch and Scheuermann [14] survey the fundamental Bitcoin protocol and its relationship to security and privacy. Khalilov and Levi [15] present a comprehensive investigation into the anonymity and privacy of cryptocurrency systems.

Owing to the blockchain technology, the cryptocurrency transaction records are verifiable and immutable [16], [17]. The growing list of transaction records stored in the chain is publicly accessible, which contains a wealth of user behavior patterns [4], [18], [19]. Cryptocurrency transaction data analysis is of great significance owing to the following reasons:

1) Studies on financial data mining are limited due to the confidentiality of traditional financial data. Fortunately, cryptocurrency transactions are mostly accessible on the chain, which opens the opportunity to conduct studies on financial data analysis and pattern mining; and

2) Cybercrimes like money laundering and smuggling trade are frequently encountered in the application of cryptocurrency due to its anonymity and decentralization [20], [21]. Therefore, it is extremely interesting to perform data analysis and pattern mining on cryptocurrency transaction data for identifying abnormal transactions, tracking illegal cash flows, and establishing cryptocurrency security [22]−[25].

Cryptocurrency transaction records among numerous users can be well modeled into a static or dynamic network. Hence, cryptocurrency transaction network embedding (CTNE) becomes an important topic in the area of cryptocurrency transaction data analysis. In recent years, network embedding methods have proven to be highly efficient in mining the rich

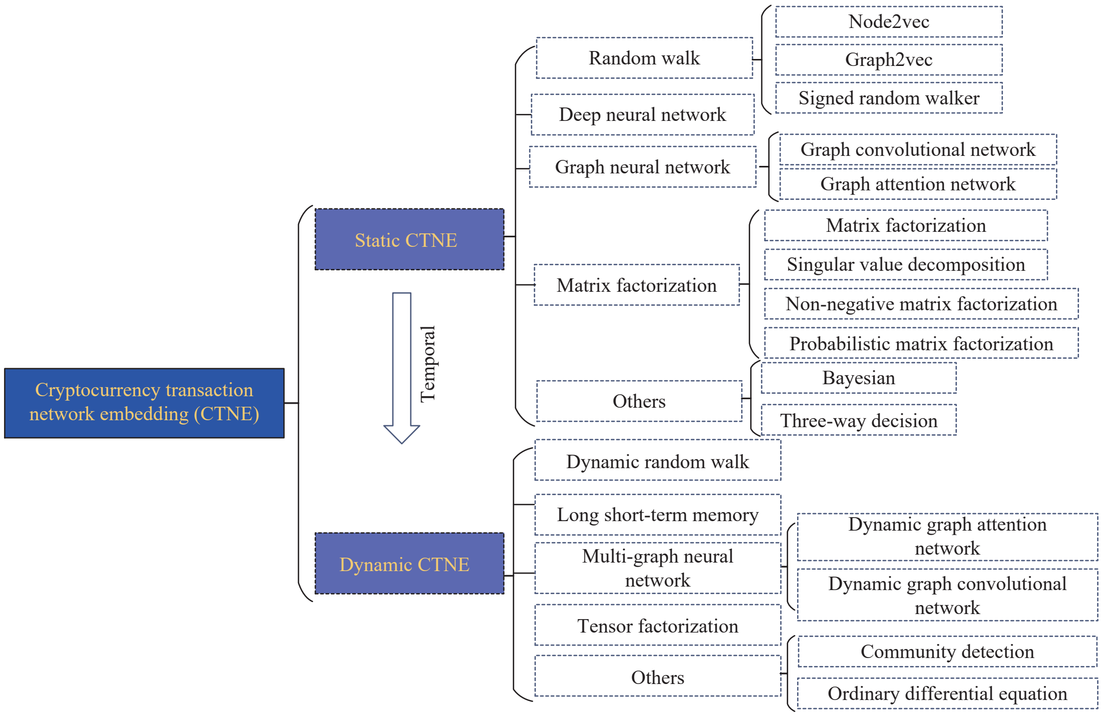


Fig. 1. Classification of cryptocurrency transaction network embedding. Kindly note that blockchain is the underlying technology of cryptocurrency, it does not affect the transactions data evidently. The most significance of the blockchain technique is to provide the unchangeable transactions records on the cryptocurrency transaction network. Hence, the classifications illustrated does not depend on different blockchain techniques, but on different embedding techniques.

information hidden in an intricate network by building the low-dimensional latent representation to its nodes, thereby

facilitating pattern analyses [26]−[30], e.g., node classification, link prediction, community detection, network visualization. Collectively, network embedding is widely adopted to analyze various complex networks, i.e., biological [31]−[33], citation [34]−[36], and social networks [37]−[42].

Traditionally, network embedding is primarily dependent on matrix eigenvector decomposition, i.e., matrix factorization [33], [43]−[45] and non-negative matrix factorization [46]−[48]. Recently, Grover and Leskovec [32] propose a node-to-vector algorithm that maximizes the likelihood of neighborhood preservation via embedding involved nodes into a low-dimensional feature space. Cao *et al.* [36] propose deep neural network-based graph representation that works by embedding nodes into a low-dimensional feature space. Perozzi *et al.* [37] propose a DeepWalk algorithm that introduces the deep learning principle into the random walk sequence. Wang *et al.* [38] propose a structural deep network embedding model that embeds a network into a nonlinear latent space to reserve its topology.

Motivated by the above mentioned successes of network embedding, CTNE has attracted widespread attentions, yielding a rapidly increasing number of related studies. However, a survey regarding its state-of-the-arts is missing.

This paper presents a comprehensive survey of existing CTNE methods. The existed CTNE methods are categorized into static and dynamic methods and summarized in Fig. 1. Either of them can be further divided into five branches: 1) random walk, 2) neural network, 3) graph neural network, 4) matrix/tensor factorization, and 5) others. We discuss the application of these networks embedded in cryptocurrency transaction networks, as presented in Sections III-A and III-B. This work intends to make the following contributions:

i) Summarizing the progress of CTNE from static to dynamic perspectives, where the state-of-the-art is carefully reviewed and categorized;

ii) Summarizing typical evaluation metrics and commonly adopted datasets for CTNE, as well as several empirically validated CTNE methods on two large-scale datasets to illustrate their performance; and

iii) Discussing the CTNE development trends.

Section II details the background. Section III reviews state-of-the-art CTNE methods. Section IV summarizes typical evaluation metrics and datasets for CTNE, and conducts the empirical studies. Section V discusses CTNE's future research directions and potential applications. Eventually, Section VI draws the conclusions.

II. Background

This section covers: 1) the introduction to cryptocurrency transaction networks, and 2) the existing network embedding methods from static to dynamic perspectives.

A. Cryptocurrency Transaction Networks

As a digital cash system of virtual assets protected by blockchain technology, cryptocurrency enables users to trade directly without any trustiness authorization [2]−[4]. More specifically, in cryptocurrency, the blockchain stores data with encrypted chained blocks and validates data with distributed consensus algorithms. In addition, it adopts cryptography to guarantee the security and privacy of data access and transmission as well as utilizes self-executed program scripts to handle data. [3], [4], [11], [49].

Notably, Bitcoin is recognized as the first decentralized cryptocurrency. Owing to its anonymity and low transaction costs, it has become the most widely-adopted cryptocurrency, and taken a dominant share of the cryptocurrency market [20], [49], [50]. After it, Ethereum is the second largest public blockchain platform. Unlike Bitcoin, Ethereum provides a Turing-complete script language that allows the users to design an arbitrary smart contract or transaction.

Blockchain stores the whole cryptocurrency transaction data, e.g., amount value, sender, receiver, and transaction time in blocks. Nearly all data are explicitly accessible. Hence, these transaction data can be abstracted into a huge and complex network, where each node represents the transaction address of a user, and each edge represents the transaction process between two nodes. Note that such a cryptocurrency transaction network (CTN) has the following features:

1) Directed since each transaction is related to a unique sender and receiver only;

2) Weighted since the amount value varies with transactions;

3) Attributed since each transaction can be measured from several different dimensions, thus resulting in multiple attributes;

4) Temporal since cryptocurrency transactions accumulate as time elapses.

B. Network Embedding

1) Static Network Embedding: Considering a static network, it can be simply described by an adjacency matrix whose embedding mostly means decomposing this adjacency matrix to learn latent feature vectors of nodes and edges [51], [52]. Qiu *et al.* [33] propose network embedding as a matrix factorization (NetMF) algorithm that adopts the approximation closed-form of the Deepwalk's implicit matrix. Qiu *et al.* [43] further propose network embedding as a sparse matrix factorization (NetSMF) algorithm, which achieves a sparsification of the NetMF matrix by leveraging the spectral graph sparsification technique. Wang *et al.* [53] propose a modularized nonnegative matrix factorization (M-NMF) model by incorporating community structures into the embedding objective for preserving both the microscopic and mesoscopic structures of a target network. The approaches based on alternating direction method of multipliers (ADMM) [54] and non-negative latent factor analysis (NLFA) [55] are also implemented to facilitate static network embedding. On the other hand, various random walk-based embedding methods emerge. Their examples are node2vec [32], DeepWalk [37], and Line [39]. Neural network-based embedding methods, e.g., deep neural networks for graph representations (DNGR) [36] and structural deep network embedding (SDNE) [38] are also investigated.

2) Dynamic Network Embedding: Recently, research on dynamic network embedding emerges, since a static network is only an abstraction of the real application scene concerning a dynamic network. Li *et al.* [56] propose a dynamic attributed network embedding (DANE) framework, which adopts matrix perturbation theory to keep the freshness of the embedding results in an online manner. Zhu *et al.* [57] propose a dynamic high-order proximity preserved embedding (DHPE) method that adopts generalized singular value decomposition (GSVD) to maintain the high-order proximity of the embedding vectors in a dynamic network. Chen *et al.* [58] propose a SuRep method that utilizes matrix factorization techniques to succinctly represent a dynamic network. Zhiyuli *et al.* [59] propose a damping-based positive-negative sampling (DNPS) algorithm that precisely learns the dynamic and hierarchical structures of a dynamic network. Xiang *et al.* [60] propose a time interval graph convolutional network (TI-GCN) model that embeds a dynamic network's each snapshot based on the embeddings of the previous ones. Zhou *et al.* [61] propose a semantic evolution method for dynamic network embedding (DynSEM). To conclude, existing dynamic network embedding methods are mostly straightforward combinations of static network embedding methods, which somehow limits their scalability when the dynamic patterns are becoming increasingly complex, e.g., a network varies continuously.

III. CTNE METHODS

Note that a CTNE method takes a static or dynamic graph as its fundamental input:

1) Static Graph: As depicted in Fig. 2, a static graph ignores the temporal dynamics. Let $G = (V, E)$ denote a static CTN with V and E being the node and edge sets respectively. Thus, $\forall e \in E$ can be defined as $e = (u, v, w)$ with u being a sender, v a receiver, and w a transaction amount value.

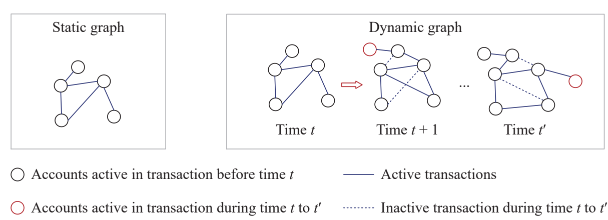


Fig. 2. Illustration of static and dynamic graphs.

2) Dynamic Graph: As illustrated in Fig. 2, a dynamic graph includes the temporal dynamics of a cryptocurrency network. Let $G = (V, E, T)$ denote a dynamic CTN as $V = \{V^{(t)}\}_{t \in T}$ denotes a node set, $E = \{E^{(t)}\}_{t \in T}$ denotes a edge set, and T denotes a time-span set. Thus, $\forall t \in T$, the snapshot $G^{(t)} = (V^{(t)}, E^{(T)})$ denotes the static state of G during the t-th time span.

We next review state-of-the-art CTNE methods from static to dynamic perspectives. In addition, in Tables I and II, we summarize the main characteristics and classification of existing static and dynamic CTNE methods, and illustrate their pros and cons.

A. Static CTNE Methods

1) Random Walk-Based Methods

A random walk-based method extracts the network topology by calculating the distance among nodes [62]. As shown in Tables I(a)−I(c), the random walk methods are mainly adopted in CTNE, i.e., node2vec [32], graph2vec [63], and

TABLE I
SUMMARY OF STATIC CTNE ARCHITECTURES

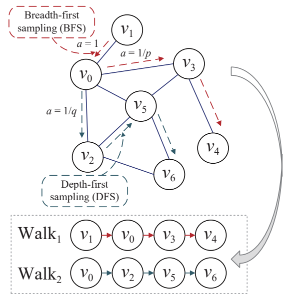
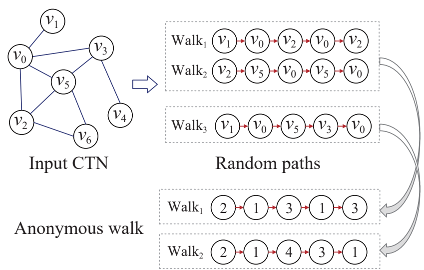
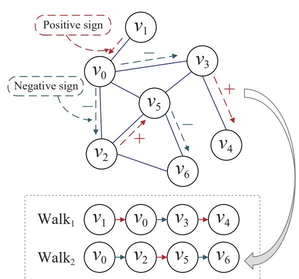
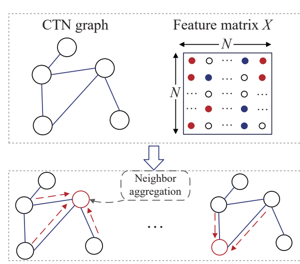
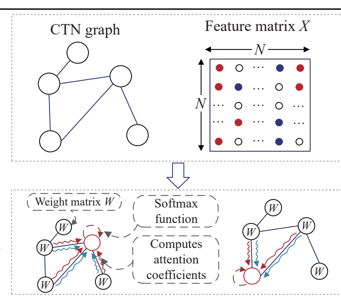
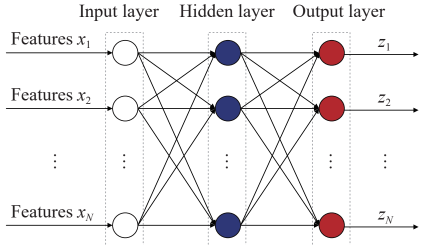
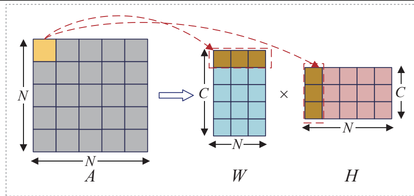
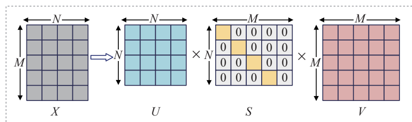
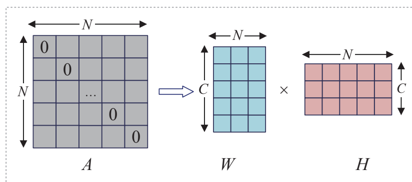
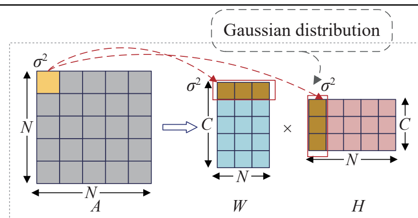
		Architecture	Description	Characteristics
Random walk-based methods	(a) Node2vec	Input CTN Random paths 	It can effectively explore different neighborhoods by defining the flexible neighborhood of each node and designing a biased random walk process.	Pros. a) A random walk-based method is easy to implement since it only considers the node pairs that co-occur during the random walk process. b) It is interpretable owing to the flexibly stochastic definition of node similarity. Cons. The computation cost is high and representation learning ability is limited.
	(b) Graph2vec	Input CTN Random paths Anonymous walk 	It extracts rooted subgraphs from a target graph, and then conduct representation learning to all sub-graphs.	
	(c) Signed random walk (RW)	Input CTN Random paths 	It adopts a signed network that considers the sign attribute (i.e., positive or negative) for each edge.	
Graph neural network-based methods	(d) Graph convolutional network (GCN)	Input Graph convolutional layer 	It extends the concept of convolution to the graph embedding domain, thereby achieving high practicability in CTN embedding.	Pros. a) It encodes graph structures and node features effectively. b) The graph attention layer is computationally efficient since it does not require expensive matrix operations and is parallel on all nodes in the graph. Cons. It ignores the dynamics of a CTN and is lack of interpretability.
	(e) Graph attention network (GAT)	Input Graph attention layer 	It further incorporates the attention mechanisms for precisely aggregating similar nodes.	
Deep neural network-based methods	(f) Deep neural network (DNN)		It is a multi-layer structure. Each layer constitutes a non-linear information processing unit, which is used to learn multi-level feature representation.	Pros. It adopts nonlinear activation functions to precisely learn the network structure. Cons. It is computationally expensive and lack of interpretability.

TABLE I
SUMMARY OF STATIC CTNE ARCHITECTURES (CONTINUED)

		Architecture	Description	Characteristics
Matrix factorization-based methods	(g) Matrix factorization (MF)		A target matrix is mapped into low-rank latent space and decomposed into two latent feature matrices with rank C, where the achieved feature matrices are interpreted as the CTNE results.	Pros. a) It captures the hidden information in CTN, such as abnormal transactions; b) It has excellent scalability. Cons. a) It ignores the dynamics of a CTN; b) The generally linear structure restricts its representation learning ability.
	(h) Singular value decomposition (SVD)		It decomposes the matrix into three simple matrices: two orthogonal matrices and a diagonal matrix.	
	(i) Nonnegative matrix factorization (NMF)		It applies the non-negativity constraints to each involved node for better representing the 'non-negative' conceptions such as the possibility that a node belongs to a specific community.	
	(j) Probabilistic matrix factorization (PMF)		It adds the probability analysis process to the decomposition process of the target CTN's adjacency matrix.	

signed random walk. Note that they all are based on the core idea of random walk, i.e., the network structure has a random path created by a certain probability distribution of a point movement on a regular lattice.

a) Node2vec: As shown in Table I(a), Node2vec facilitates a second-order random walk strategy to sample the neighborhood nodes, thus smoothing the interpolation between width-first sampling (BFS) and depth-first sampling (DFS). As shown in Table I(a), p and q denote the return and in-out parameters that adjust the transition probability during a walking process. For CTNE, Yuan *et al.* [64] utilize node2vec to extract the latent features of the Ethereum CTN accounts. Tao *et al*. [65] propose a random walk with a flying-back properties (RWFB) method, which extracts the features of a Bitcoin CTN via multi-dimensional analysis regarding degree distribution, clustering coefficient, shortest path length, assortactivity analysis, and rich club coefficient.

b) Graph2vec: As shown in Table I(b), this method represents the entire graph with a series of root subgraphs around each node. It utilizes anonymous walk embedding to generate subgraphs for capturing the graph state corresponding to the index of the initially-visited node during walking. Considering CTNE, Yuan *et al.* [66] embed the transaction topology of the sub-network of each target account into the latent feature space via Graph2vec.

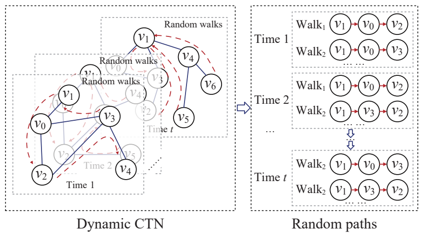
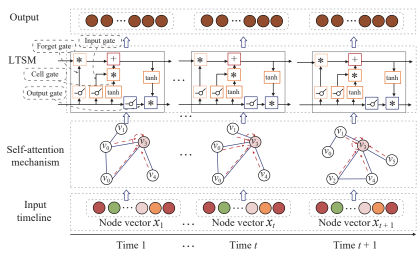
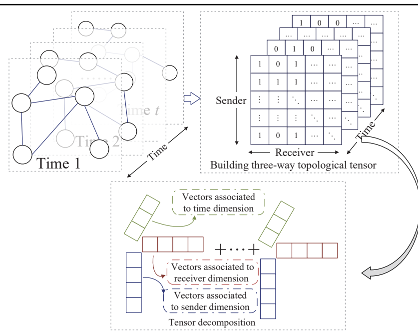
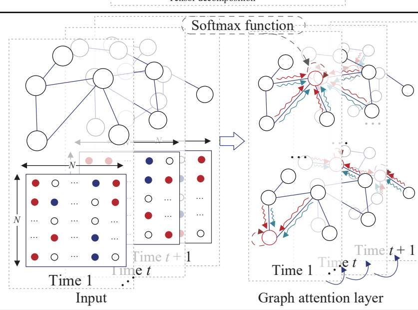
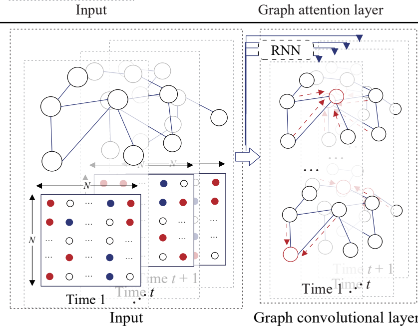
c) Signed random walk (RW): As shown in Table I(c), it incorporates the positive and negative signs of the edges into the random walking process, thereby modeling the social advantage of each node and its neighbors. Note that a random walk process is designated more likely to visit a potential "friend", i.e., a positively-linked node rather than a potential enemy, i.e., a negatively-linked node. Given a signed CTN, Ma *et al.* [67] propose a signed network embedding approach based on the framework of generative adversarial networks to learn its low-dimensional node representation while maintaining its link structures and edge signs. It incorporates the graph softmax function and signed random walk into a generator for approximating the underlying true connectivity distribution of a signed CTN. Li *et al.* [68] propose a signed supervised random walk method that is able to capture CTN users' different preferences on their neighbors, so as to better promote the task of personalized user ranking.

2) Graph Neural Network-Based Methods

A graph neural network (GNN)-based model aggregates the features of adjacent nodes and computes new feature vectors in a layer-by-layer way. As shown in Tables I(d) and I(e), the commonly adopted graph neural network-based methods in CTN can be divided into two categories: Table I(d) graph convolution network (GCN)-based ones, and Table I(e) graph attention network (GAT)-based ones.

d) Graph convolution network (GCN): As shown in Table I(d), GCN aggregates the CTN node information from neighborhood by convolution. Specifically, the graph convolution layer collects information according to the graph structure, and then updates the state of hidden nodes accordingly. It can precisely represent the non-Euclidean structure of the target CTN [69]. Patel *et al.* [70] propose a one class graph neural network (OCGNN)-based anomaly detection framework that incorporates the support vector data description (SVDD) into a GCN for learning the structure of an Ethereum CTN.

TABLE II
SUMMARY OF DYNAMIC CTNE ARCHITECTURES

Architecture			Description	Characteristics
Dynamic random walk-based methods	(a) Dynamic random walk (DRW)		After selecting the initial edge from the initial snapshot of the network, repeat selecting the next time-efficient neighbor from the given node until traversing the network slice at the initial time. The same random walk is then continued in snapshot order.	Pros. a) It integrates the weight and temporal into the feature vectors. b) It can capture the temporally valid links from a dynamic network. Cons. a) It is hard to find the optimal sampling strategy. b) It adopts the same probability to perform random walk, which can vary in real cases.
Long short-term memory-based methods	(b) Long short-term memory (LSTM)		It retains long-term dependencies and connect information from past to present. It contains three types of gates: 1) forget gate, deciding what information to forget from the previous; 2) input gate, deciding what new information to remember; and 3) output gate, deciding which part of the output cell state.	Pros. a) It adopts self-attention to enhance the embedding and maintain the node diversity. b) The LSTM layer can simulate the dynamic evolution of latent space. Cons. It is computationally expensive and without interpretability.
Tensor factorization-based methods	(c) Tensor factorization (TF)		It considers the target dynamic CTN as a three-way topological tensor, and then decompose it into several rank-one tensors following the Canonical Polyadic Decomposition (CPD) or Tucker frameworks.	Pros. a) It can effectively capture temporal patterns in a dynamic CTN. b) It has excellent scalability. Cons. It models the temporal dynamics in the target CTN from the numerical perspective only, yet lacking of considerations from the modeling perspective.
Multi-graph neural network-based methods	(d) Dynamic graph attention network (DGAT)		It efficiently captures the evolutionary patterns from the graph sequences by learning the impact of previous multiple graph snapshots on the current one as well as utilizes a self-attention mechanism for neighborhood aggregation.	Pros. It can effectively capture temporal patterns in a graph sequence. Cons. It is computationally expensive and lack of interpretability.
	(e) Dynamic graph convolutional network (DGCN)		It combines GCN with recurrent neural network (RNN), where the former is adopted for structure information extraction, and the latter is adopted for sequence modeling.	

Huang *et al*. [71] propose a mix-grain GCN model that adopts the fine-grained and coarse-grained aggregators to address the issue of insufficient information collection as well as learn the embedding of a large-scale graph efficiently. Tam *et al.* [72] propose an EdgeProp method based on an end-to-end GCN, which is applied to node and edge embedding of a large-scale time evolution graph. Derr *et al.* [73] propose a signed GCN that applies the balance theory to interlayer information aggregation and propagation, thereby achieving a complex CTN's embedding in an efficient way.

By combining the virtues of statistical relation learning and GCN, Qu *et al.* [74] propose a graph Markov neural network model that identifies node representation and label dependencies. Agrawal and Alfaro [75] present a deep structured embedding model that learns edge representations based on aggregation of paths. It is capable of embedding an arbitrary edge attribute without feature extraction. Verma *et al.* [76] propose a GraphMix model that adopts a full-connection network to improve the training efficiency of a GCN-based embedding model. Huang *et al.* [77] propose a signed directed GNN model that adopts multiple-layers to capture high-order structure information in a Bitcoin CTN. Kudo *et al.* [78] propose a GCN with expended balance theory, which aggregates the edge signs and directions for identifying fraudulent users in CTN. Liu *et al.* [79] propose an identity inference approach by graph deep learning, where blockchain accounts and related transactions are represented by graphs and the accounts are represented as nodes with low-dimensional features via GNN-based graph learning.

e) Graph attention network (GAT): As shown in Table I(e), GAT further applies the shared linear transformation to each node, and then computes the attention coefficients for better aggregating nodes with similar behavior patterns in the target CTN. Huang *et al.* [80] propose a signed GAT that simultaneously combines balance and state theories to achieve accurate CTN embeddings. Li *et al.* [81] propose a signed network embedding via a graph attention model that leverages the graph attentional layer to aggregate multi-source information based on the balance theory. Wu *et al.* [82] propose a hierarchical attention signed network model that precisely maintains the balance and state theories with hierarchical attentions.

3) Deep Neural Network-Based Methods

f) Deep neural network (DNN): As shown in Table I(f), The target CTN is embedded though a DNN. The input features are propagated from the input layer, via the hide layer, and to the output layer. During the propagation, the state of each layer only affects the state of the next layer. When the output layer fails in achieving the expected output, it can be switched to error signal back-propagation [83]. Based on the principle of transfer learning [84], Liu *et al.* [85] design an asymmetric tri-training back propagation neural network model for accurately predict the unlabeled relationships in a Bitcoin CTN.

4) Matrix Factorization-Based Methods

As shown in Tables I(g)−I(j), matrix factorization-based CTNE models depend on latent feature matrices describing the topology of a target CTN (such as its adjacency or Laplacian matrices). They can be divided into the following branches:

g) Matrix factorization (MF): As shown in Table I(g), it maps each involved node in the target CTN to the same low-dimensional latent feature space by decomposing the adjacency matrix of the target CTN into two low-rank latent feature matrices. By doing so, each node is represented by a dense feature vector that can facilitate subsequent tasks like link prediction. Meo [86] propose a pairwise trust prediction method through a matrix factorization algorithm, which incorporates the trustor and trustee behavior biases into the learning objective for predicting the intensity of trust and distrust relations in CTN.

h) Singular value decomposition (SVD): As depicted in Table I(h), SVD factorizes adjacency matrix A into three matrices, i.e., two orthogonal matrices and one diagonal matrix to achieve the low-dimensional embeddings of each involved node. However, its computational cost is huge. The widely adopted SVD algorithm is also utilized to implement CTNE [87]. Chen *et al.* [9] apply the standard SVD to a Bitcoin CTN to discover the relationship between node behavior and Bitcoin price.

i) Nonnegative matrix factorization (NMF): As shown in Table I(i), NMF is a classical low rank method, which incorporates non-negative constraints, resulting in part-based representations and correspondingly enhanced problem interpretability [88]−[90]. It has been applied to CTNE scenes. Yu *et al.* [91] propose a double NMF model that integrates the node in-degree as a regularization term into the learning objective to build a node transaction probability matrix. Wang and Mu [92] propose a regularized convex NMF model that considers graph regularization, thus constraining positively-connected nodes to enter the same community and ensuing negatively-connected nodes to represent the hidden structure in a target CTN. Reference [93] proposes an analogous preserving overlapping community detection method. It extracts node similarity and geometric structures from link topology, which are further fused to implement accurate community detection via a graph-regularized binary semi-NMF model.

j) Probabilistic matrix factorization (PMF): According to Table I(j), different from the other matrix factorization methods, PMF adopts a probabilistic linear model with Gaussian noise to correlate a target CTN with potential variables linearly [94]. Muzammal *et al.* [95] decompose a target CTN into several sub-graphs, and then adopt Bayesian probabilistic matrix factorization to extract latent features from them to achieve its low-rank embeddings.

5) Other Methods

Liu *et al.* [96] propose a signed local naive Bayesian model, which achieves highly accurate link prediction on small-scale CTNs. Qiu *et al.* [97] propose a directed edge weight prediction model based on a decision tree ensemble, which utilizes network topology without dependence on involved nodes' private attribute information. Liu *et al.* [98] propose a single motif naive Bayesian model that not only explains the prediction mechanism of the single edge-dependent motif based method, but also considers the roles of different nodes and edges when adopting multiple motif information for sign prediction in a CTN. Pang *et al.* [99] propose a sign prediction method based on tri-domain relationship pattern method, which adopts the three-domain relationship pattern to predict the signs of links on the unlabeled domains from a Bitcoin CTN. Liu *et al.* [100] propose a three-way decisions functional network model that incorporates the three-way decision into functional network modeling, thereby implementing three-way decision making to for precisely identifying boundary samples with high performance.

B. Dynamic CTNE Methods

The previous section discusses existing static CTNE methods. However, a real CTN commonly changes over time as shown in Fig. 2. Therefore, it is highly significant to study dynamic CTNE methods.

1) Dynamic Random Walk-Based Methods

a) Dynamic random walk (DRW): Most existing dynamic random walk-based CTNE methods are based on a static algorithm like DeepWalk and node2vec. As shown in Table II(a), DRW considers the temporal dependence, adopts the temporal walk for CTN, and then obtains the random walk sequence on each time slice. For instance, Lin *et al.* [101] propose a temporally-weighted multidigraph embedding algorithm based on DeepWalk, which is designed to learn significant node representation from a dynamic CTN.

2) Long Short-Term Memory-Based Methods

b) Long short-term memory (LSTM): As shown in Table II(b), LSTM firstly generates a temporal embedding vector, which reflects the changes of network topology, and then enhances the embedding and maintains the diversity of nodes through a self-attention mechanism. The LSTM memory architecture is utilized to preserve the important features of the target CTN, and the forget gate and output gate are adopted to preserve the basic relations and drop disturbance information. This method leads to an effective and scalable model for capturing long-term temporal dependencies [102].

Wang *et al.* [103] integrate self-tokenization into a sequence modeling framework based on LSTM, thereby predicting the future links in a temporal network. Jiao *et al.* [104] propose a temporal network embedding method based on a variational autoencoder. It combines a self-attention mechanism and LSTM, thus not only generating low-dimensional embedding vectors for nodes, but also maintaining the dynamic nonlinear features of a temporal network.

3) Tensor Factorization-Based Methods

c) Tensor factorization (TF): As shown in Table II(c), a three-way tensor can be defined according to a dynamic CTN with its first dimension being the sender set, the second dimension the receiver set, and the third dimension the time slots. When there is a transaction between a specific sender

account and a specific receiver at a specific time slot, the corresponding entry in the built tensor is filled, and otherwise it is unknown as depicted in Table II(c). Note that a three-way tensor is a natural yet highly-precise way to describe a dynamic CTN, where the temporal dynamics can be taken into consideration in a natural way.

Considering dynamic CTNE methods, existing tensor factorization-based methods are based on the Canonical Polyadic decomposition or Tucker factorization frameworks. For instance, Charlier *et al.* [105], [106] adopt a CPD-based tensor factorization model to represent smart contract data, and then utilize a log-normal-mean-reverting stochastic model to predict future smart contract sequences. They prove their method’s efficiency. However, few studies fall to this category and the community’s further efforts are required.

4) Multi-Graph Neural Network-Based Methods

These methods are able to capture the dynamic evolution of target CTN. They can be further divided into dynamic graph attention network (DGAT) and dynamic graph convolutional network (DGCN)-based ones.

d) Dynamic graph attention network (DGAT): According to Table II(d), a DGAT method accepts multiple graph snapshots as the input according to the time line, and then adopts the self-attention mechanism to aggregate the graph neighborhood for capturing the temporal tendency. The graph attention layer mainly captures weights implicitly through an end-to-end neural network architecture. Li *et al.* [69] propose a dynamic GCN that facilitates spatial and temporal convolution in an interleaving manner.

It adopts an S-stack temporal self-attention architecture, which integrates the effects of several previous graph snapshots into the current one with self-adapting importance, therefore effectively capturing the evolutionary patterns hidden in a dynamic CTN. Li *et al.* [107] propose a graph temporal edge aggregation framework that integrates an attention mechanism into LSTM to represent the temporal interactions among involved nodes. Wang *et al.* [108] propose a co-evolutionary GNN model.

e) Dynamic graph convolutional network (DGCN): According to Table II(e), A DGCN-based method accepts multiple graph snapshots as the input according to the time line. The graph topology information is extracted by GCN, and the temporal information is captured by a recurrent neural network (RNN). Bonner *et al.* [109] propose a temporal neighborhood aggregation method by combining graph convolution with recurrent vertex representation, thereby capturing both topological and temporal variations from a dynamic graph. Dave and Hasan [110] propose a graphlet and node-based time-conserving embedding framework based on neural networks, where an edge representation vector learning model is designed to embed the edges with similar triangle completion time into the latent space. Pareja *et al.* [111] propose an evolving GCN model that captures the temporal patterns by evolving GCN parameters following the RNN principle. Wu *et al.* [112] propose a dynamic graph evolution network prediction model that adopts a recurrently structured GNN to represent a dynamic graph. Cai *et al.* [113] propose an end-to-end structural temporal GNN model that detects anomalous edge by mining the unusual and temporal sub-graphs. Malik *et al.* [114], [115] incorporate a tensor M-product into GCN, thus capturing correlations over time to learn the embedding.

5) Other Methods

Ao *et al.* [116] propose a temporal high-order proximity aware community detection model, which can efficiently analyze the temporal user behavior in Ethereum block transactions. It consists of an temporal-motif mining algorithm, a high-order proximity computing algorithm and a temporal motif-aware community detection algorithm. Cao *et al.* [117] propose a novel graph representation learning framework based on ordinary differential equations used to model the continuous dynamics of CTN, thus capturing the temporal patterns in a natural way.

C. Summary

We summarize the progress of CTNE from the static to the dynamic perspectives, where the state-of-the-art is comprehensively reviewed and categorized. The main characteristics of existing studies are summarized in Tables I and II, including the proposed models, tasks, datasets, and evaluation metrics. Both static and dynamic CNTE models are mostly related to GNN, GCN and GAT [69]–[74], [76]–[81], [103], [107]–[114]. Relatively fewer studies have been conducted on random walks and tensor factorization, where the latter possesses the potential to achieve highly accurate representations to establish dynamic CTNE.

Considering the limitations of existing CTNE methods:

a) They mostly focus on static CNTE, while ignoring the temporal nature of CTN in real applications. Dynamic CTNE methods are relatively scarce and deserve more studies.

b) Existing studies on dynamic CNTE mostly focuses on (graph) neural network-based methods. However, their scalability is rather limited due to the high computational and storage costs in spite of their excellent representation learning ability. From this point of view, it is urgent to facilitate research on relatively light-weight CTNE models like a tensor factorization-based one for performing highly scalable CTNE on large-scale networks.

IV. Metric and Dataset

A. Metric

In this section, we summarize and briefly introduce the commonly-adopted evaluation metrics for a CTNE model. As shown in Tables III and IV, most existing studies focus on node classification and link prediction, whose evaluation metrics are listed as follows.

1) Precision: It reflects whether the missing edge in the target CTN is accurately predicted. Higher precision denotes more accurate predictions, which is defined as

$$Precision = \frac{TP}{TP+FP} \tag{1}$$

where *TP* denotes the correctly predicted link count, and *FP* denotes the number of falsely predicted ones.

2) Recall: It denotes the fraction of missing links that have been retrieved over the total number of potentially existing ones, which is defined as

$$Recall = \frac{TP}{TP+FN} \tag{2}$$

where *FN* denotes the number of potential links missed by the evaluated CTNE model.

3) Average Precision (AP): It is a metric that balances the precision and recall. As the recall threshold increases from 0 to 1, AP increases as precision increases.

4) F1: It is an important metric in the statistical analysis of node classification because it is a harmonic average of accuracy and recall. By combining (1) and (2), F1 is calculated as

$$F1 = \frac{2\times Precision\times Recall}{Precision+Recall}. \tag{3}$$

5) Macro-F1: It refers to the calculating the average of precision and recall of each category for F1.

6) Accuracy: It focuses on overall performance, and is defined as the proportion of correctly predicted links in all potential links. It is calculated as

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{4}$$

where *TN* denotes the number of links that is not in the target CTN and also correctly ignored by the CTNE model.

7) RMSE: The root mean squared error describes the difference between the real link weights and the predicted ones by a CTNE model, i.e.,

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i-\hat{y}_i)^2}{n}} \tag{5}$$

where n denotes the number of the samples, y_i denotes the actual value in the test set, and $\hat{y}_i$ denotes the estimated one generated by the learning model.

8) MAE: The mean absolute error measures the absolute difference between the real link weights and the predicted ones by a CTNE model, i.e.,

$$MAE = \frac{\sum_{i=1}^{n}|y_i-\hat{y}_i|}{n}. \tag{6}$$

9) PCC: The Pearson correlation coefficient measures the linear correlation between two variables, i.e., the real link weights and the predicted ones by a CTNE model in our context, and large PCC denotes strong linear relationship between them. It is defined as

$$PCC = \frac{\left(\sum_{i=1}^{n}\left(\frac{x_i-\bar{x}}{\sigma_x}\right)\left(\frac{y_i-\bar{y}}{\sigma_y}\right)\right)}{n-1} \tag{7}$$

where $\bar{x}$ and $\bar{y}$ denote the average values of the sample values of variables x and y, and σ_x and σ_y denote the standard deviation of the sample values of x and y, respectively.

10) AUC: It measures the probability that the predictions for the randomly selected positive samples are higher than those for the randomly selected negative samples. Note that high AUC stands for high model performance.

B. Datasets

The entire cryptocurrency data are currently open for access, providing unprecedented opportunities for CTNE research.

Mostly adopted datasets are summarized as follows:

1) Mt. Gox [9]: It is the bitcoin history transaction dataset leaked by the world's leading bitcoin intermediary in early 2014, which recorded the largest bitcoin transactions from April 2011 to November 2013. It includes 119 343 transaction nodes and 2 682 719 transaction edges.

2) Blockchain transaction [70]: It is a collection of Ethereum transaction data from August 2016 to January 2017. It does not contain time information, and only contains 50 422 edges among 25 257 nodes.

3) API-Ethereum [71]: It contains the Ethereum transaction data of 2 815 028 edges among 1 402 220 nodes. Each node contains 13 features extracted from the transaction records.

4) Bitcoin-Alpha: It is collected from a bitcoin trading plat-

TABLE III
SUMMARY OF STATIC CTNE

Reference	Model	Task	Baseline	Metric	Dataset	Application
[9]	SVD	–	–	–	Mt. Gox transaction network*	Market analysis
[64]	Node2vec	Node classification	DeepWalk, Non-embedding method	Precision, Recall, F1	Ethereum transaction network*	Phishing detection
[65]	RWFB	–	–	–	Bitcoin blockchain network*	Phishing detection
[66]	Line-Graph2Vec	Classification prediction	Node2Vec, WL-kernel, original Graph2Vec	Precision, Recall, F1	Ethereum transaction network*	Phishing detection
[67]	SNEGAN	Link prediction, reconstruction	DeepWalk, Node2vec, GraphGAN, DNE-SBP	AUC, Average precision	Bitcoin-Alpha[1]	–
[85]	BPtri-train	Trust relationship predictions	SVM, Random Forest, TranFG	Accuracy, Precision, Recall, F1	Bitcoin-Alpha[1], Bitcoin-OTC[2]	–
[70]	OCGNN	Node classification	Isolation Forest	Accuracy, F1	Ethereum blockchain network[3]	Anomaly detection
[71]	Mix-grained GCN	Node classification	GCN, GraphSAGE, Fast-GCN, SGC	Accuracy	Ethereum*	–
[72]	EdgeProp	Node classification	LR, Random Forest, Gradient Boosting Decision Tree, DeepWalk, Line, GraphSAGE	Accuracy, Precision, recall F1	Ethereum transactions*	Identifying Illicit Accounts
[73]	SGCN	Link prediction	Signed Spectral Embedding, SiNE, SIDE	AUC, F1	Bitcoin-Alpha[1], Bitcoin-OTC[2]	–
[74]	CMNN	Object classification, link classification, unsupervised node representation learning	DeepWalk, GCN, GAN, PRM, LP, RMN, MLN	F1	Bitcoin-Alpha[1], Bitcoin-OTC[2]	–
[75]	LEAP	Link prediction, user rating prediction	WLNM, SEAL, Adamic-Adar, Katz, PageRank, node2vec	AUC	Bitcoin-Alpha[1], Bitcoin-OTC[2]	–
[76]	GraphMix	Link classification	DeepWalk, GMNN, GCN	F1	Bitcoin-Alpha[1], Bitcoin-OTC[2]	–
[77]	SDGNN	Link prediction	Random, Deepwalk, Node2vec, LINE, SiNE, SIGNet, BESIDE, FeExtra, SGCN, SiGAT	Binary-F1, Micro-F1, Macro-F1, AUC	Bitcoin-Alpha[1], Bitcoin-OTC[2]	–
[78]	GCNEXT	Node classification	Rev2, R-GCN, SIDE, SGCN	AUC	Bitcoin-Alpha[1], Bitcoin-OTC[2]	Fraud detection
[80]	SiGAT	Link prediction	Deepwalk, Node2vec, LINE, SiNE, SIDE SIGNet, SGCN	Accuracy, F1, Macro-F1, AUC	Bitcoin-Alpha[1]	–
[81]	SENA	Link prediction	TSVD, SiNE, SIDE, SGCN, SiGAT	AUC, F1	Bitcoin-Alpha[1], Bitcoin-OTC[2]	–
[82]	HASN	Link prediction	DeepWalk, LINE, SINE, SIDE, FExtra, SGCN, SiGAT	Accuracy, Micro-F1, Macro-F1, AUC	Bitcoin-Alpha[1]	–
[86]	PTP-MF	Link prediction	FG	RMSE	Bitcoin-Alpha[1], Bitcoin-OTC[2]	–
[91]	DouNMF	Link prediction	AA, ACT, CN, CRA, Jaccard, Salton	Generalized AUC	Bitcoin-OTC[2]	–
[92]	RC-NMF	Link prediction	CN, Jaccard, Salton, Res-NMTF	Generalized AUC	Bitcoin-OTC[2]	–
[93]	SPOCD	Overlapping community detection	LPOCSIN, SPM, ReSNMF, SDMID, MEAs-SN	F1	Bitcoin*	–
[95]	PLF	Link prediction	Bayesian PMF, NeLP	Precision, Recall, F1	Bitcoin-Alpha[1], Bitcoin-OTC[2]	–
[96]	SLNB	Link prediction	FriendTNS, Status theory	AUC, Precision	Bitcoin-Alpha[1], Bitcoin-OTC[2]	–
[97]	DEWP	Link prediction	FG, reciprocal, TidalTrust, SEC	RMSE, PCC	Bitcoin-Alpha[1], Bitcoin-OTC[2]	–
[98]	SMNB	Link prediction	SEDM	AUC, PCC	Bitcoin-Alpha[1], Bitcoin-OTC[2]	–
[99]	SP-TDRP	Link prediction	SLATTL, SLATL, TTL, TL, EasyTTL, EasyTL	Accuracy, F1	Bitcoin-Alpha[1], Bitcoin-OTC[2]	–
[100]	3WD-FN	Link prediction	LR, LP, LS, e-Trust, FN	Precision, Recall, F1, Accuracy	Bitcoin-Alpha[1], Bitcoin-OTC[2]	–

[1] http://www.btcalpha.com; [2] http://www.bitcoin-otc.com; [3] https://github.com/vatsalpatels/Graph-DL-Based-Anomaly-Detection-in-Ethereum.git; * Not available.

form Bitcoin-Alpha. Due to the fact that bitcoin accounts are anonymous, users need to set up an online trust network to reserve safety. Members of Bitcoin-Alpha rate other members on a scale from −10 (full distrust) to 10 (full trust) in steps of 1, which helps preventing transactions with fraudulent and risky users. The dataset covers transactions from October 2010 to January 2016, with 3784 users and 12 729 entries.

5) Bitcoin-OTC: is collected from a bitcoin trading platform Bitcoin-OTC. Like the Bitcoin-Alpha dataset, Bitcoin-OTC is a who-trusts-whom network dataset, with scores rang-

TABLE IV
SUMMARY OF DYNAMIC CTNE

Reference	Model	Task	Baseline	Metric	Dataset	Application
[68]	SSRW	Link prediction	SPNR, TNS, SFM, RWR	GeneralizedAUC	Bitcoins*	–
[69]	DynGCN	Temporal link prediction, edge classification	GCN, GCN-GRU, EvolveGCN	Accuracy, F1	Bitcoin-Alpha[1], Bit-coin-OTC[2]	–
[79]	I^2GL	Node classification	DeepWalk, PARW, rGCN	Precision, Recall, F1	Ethereum transaction network*	Phis92hing detection
[101]	T-EDGE	Node classification	DeepWalk, Node2vec	Micro-F1, macro-F1	Ethereum[3]	–
[103]	GLSM	Temporal link prediction	LPGNN, GAN, AA, MF, GG, TMF, JC	AUC	Bitcoin*	–
[109]	TNA	Temporal link prediction	GAE, GVAE, TO-GAE, TO-GVAE, DynAE, DynRNN	AUC, AP	Bitcoin-Alpha[1]	–
[107]	GTEA	Node classification	XGBoost, GCN, Graph-SAGE, GAT, APPNP, ECConv, EGNN, TGAT	Accuracy, macro-F1	Ethereum-Role Dataset, Ethereum Phishing Large Dataset[4]	–
[108]	CoEvoGNN	Node attribute prediction, link prediction	GCN, GAT, GraphSAGE, DynamicTrian, DySAT, DCRNN, STGCN	MAE, RMSE, AUC, F1, Precision	Bitcoin-Alpha[1], Bit-coin-OTC[2]	–
[110]	GraNiTE	Temporal link prediction	LINE, Node2vec, Graph-SAGE, AROPE, VERSE	MAE	Bitcoin-OTC[2]	–
[111]	EvolveGCN	Temporal link prediction, node classification	GCN, GCN-GRU, DynGEM, dyngraph2vec	Mean AP, F1	Bitcoin-Alpha[1], Bit-coin-OTC[2]	–
[112]	EvoNet	Temporal link prediction	ER, SBM, BA, Power, Kron-Rand, Kron-Fix	AUC	Bitcoin-Alpha[1], Bit-coin-OTC[2]	–
[113]	StrGNN	Node classification	DeepWalk, Node2vec, Spec-tral Clusteing, Netwalk	AUC	Bitcoin-Alpha[1], Bit-coin-OTC[2]	Anomaly detection
[114]	TensorGCN	Edge classification	WD-GCN, EvolveGCN, GCN	Accuracy, F1	Bitcoin-Alpha[1], Bit-coin-OTC[2]	–
[115]	TM-GNN	Edge classification, tempo-ral link prediction	WD-GCN, EvolveGCN, GCN	Accuracy, F1	Bitcoin-Alpha[1], Bit-coin-OTC[2]	–
[105]	non-negative CP	Link prediction	–	AUC	Smart contracts*	–
[106]	Tensor	Link predictions	–	AUC	Smart contracts*	–
[116]	THCD	Community detected	Louvain, Motif, EdMot	Modularity	Ethereum*	Market analy-sis
[117]	Graph-ODE	Link prediction, node classi-fication	GCN, GraphSage, DNE, CTDNE, EvolveGCN	Mean AP, AUC	Bitcoin-Alpha[1], Bit-coin-OTC[2]	
[104]	TVAE	Link prediction	DeepWalk, Line, GAT-AE, DynamicTriad, DynAERNN, TNE, DySAT	Mean AP, AUC	Bitcoin-Alpha[1], Bit-coin-OTC[2]	–

[1] http://www.btcalpha.com; [2] http://www.bitcoin-otc.com; [3] https://github.com/lindan113/xblock-network_analysis/tree/master; [4] https://www.kaggle.com/xblock/ethereumphishing-transaction-network; * Not available.

ing from −10 to 10. In addition, the dataset covers bitcoin ratings from November 2010 to January 2016, with 881 users and 35 592 entries.

6) Client-Ethereum [79]: It is collected by the Ethereum client, which synchronizes all historical transaction records from the Ethereum blockchain. Also, the dataset has 116 293 867 transactions and 16 599 825 active accounts from January 2018 to December 2018.

7) Ethereum-Role [72], [107]: It is an Ethereum transactions dataset consisting of 2.18 million nodes, 3.75 million edges and 445 ground truth labels.

Each node and each edge respectively have 23 and 5 features acquired from the transaction records.

8) Phishing Small and Phishing Large [107]: They are Ethereum phishing datasets. Their data are binary-classified to detect phishing accounts in the Ethereum payment network. The former has 1 329 729 nodes, 2 161 573 edges and 3360 ground truth labels, while the latter has 2 973 489, 5 355 155 and 6165 in correspondence.

9) Smart Contract [105], [106]: It was collected from the Ethereum platform from August 2015 to March 2016. During this period, about two million transactions have happened between 241 385 senders and 359 798 receivers.

10) Bitcoin Transaction [102]: It contains 297 816 881 accounts and 298 325 122 transactions from January 2009 to February 2018. The edges are weighted according to the bitcoin amount transferred among accounts. As shown in Table V, we summarize the commonly adopted CTN datasets. Most datasets are temporally dynamic and highly incomplete.

C. Experiments and Results

In this section, we test eight models in terms of link prediction. As shown in Table VI, M1−M4 are the static while M5−M8 are dynamic CTNE models. M1−M8 all adopt the default hyper parameter settings as reported in prior studies [77], [80], [95], [111], [115], [118]. Since static CTNE models have no temporal settings, datasets adopted by M1−M4 are cleaned to remove the time information.

The experiments are conducted on two real-world bitcoin datasets, i.e., Bitcoin-Alpha (D1) and Bitcoin-OTC (D2). In particular, for static link prediction, each dataset is randomly split into ten disjoint subsets for implementing 80%−20%

TABLE V
SUMMARY OF CRYPTOCURRENCY TRANSACTION DATASETS

Dataset	Nodes	Edges	Time	Reference
Mt. Gox	119 343	2 682 719	2011.04−2013.11	[9]
Blockchain transaction	25 257	50 422	2016.08−2017.01	[70]
API-Ethereum	1 402 220	2 815 028	–	[71]
Bitcoin-Alpha	3784	12 729	2010.09−2016.01	–
Bitcoin-OTC	5881	35 592	2010.11−2016.01	–
Client-Ethereum	16 599 825	116 293 867	2018.01−2018.12	[79]
Ethereum-Role	2 180 689	3 745 858	2018.01−2018.12	[72], [107]
Phishing Small	1 329 729	2 161 573	–	[107]
Phishing Large	2 973 489	5 355 155	–	[107]
Smart contracts	359 798	2 000 000	2015.08−2016.03	[105], [106]

TABLE VI
COMPARED MODELS

No.	Name	Description
M1	PMF	A probabilistic linear matrix factorization model with Gaussian observation noise [95].
M2	Bayesian PMF	An extended PMF model that introduces a complete Bayesian prior into PMF [95].
M3	SiGAT	A SiGAT model that incorporates graph motifs into GAT [80].
M4	SDGNN	A SDGNN model that redesign aggregators and loss function [77].
M5	EvolveGCN-O	A EvolveGCN-O model that recurrent hidden state realized by GRU [111].
M6	EvolveGCN-H	A EvolveGCN-H model that recurrent input-output relationship realized by LSTM [111].
M7	WD-GCN	A Waterfall Dynamic-GCN model that combines LSTMs and GCNs to exploit structural and temporal information of data [118].
M8	TM-GNN	A TM-GNN model that incorporates tensor M-product technique into GCN [115].

training-testing. For dynamic link prediction, each dataset is divided into time series, with the first 80% form the training set and the last 20% for the testing set.

Precision, Recall and F1 are chosen as the evaluation metrics. Tables VII and VIII compare the performance of static and dynamic CTNE models on D1 and D2, respectively. Table IX summarizes the total time cost of compared models. Thus, we find that:

1) For Static Link Prediction, the Prediction Precision of a Neural Network Model is Better: As shown in Table VII, on D1, M4's precision, recall and F1 are 0.4319, 0.1315, and 0.2060, respectively. It is 99.07%, 49.43% and 96.36% higher than M1's 0.0040, 0.0665 and 0.0075, respectively.

2) The Accuracy of Dynamic Link Prediction Model Is Generally Higher Than That of Static One: As shown in Tables VII and VIII, the precision of the dynamic link prediction model

TABLE VII
PERFORMANCE COMPARISON AMONG M1–M4 (PRECISION/RECALL/F1)

No.	D1	D2
M1	0.0040/0.0665/0.0075	0.0061/0.0576/0.0111
M2	0.0010/0.0172/0.0019	0.0010/0.0091/0.0017
M3	0.2781/0.1100/0.1559	0.2184/0.1315/0.1642
M4	**0.4319**/**0.1315**/**0.2060**	**0.2445**/**0.1412**/**0.1790**

TABLE VIII
PERFORMANCE COMPARISON AMONG M5–M8 (PRECISION/RECALL/F1)

No.	D1	D2
M5	0.7506/0.0332/0.0635	0.6957/0.0221/0.0429
M6	**0.8760**/0.0331/0.0638	0.6479/0.0230/0.0444
M7	0.7133/**0.0873**/**0.1556**	0.6035/**0.1314**/**0.2158**
M8	0.6911/0.0385/0.0730	**0.9753**/0.0550/0.1042

TABLE IX
TOTAL TIME COST AMONG M1–M8 (SECONDS)

No.	D1	D2
M1	**286**	**214**
M2	529	234
M3	1495	1528
M4	2273	1964
M5	2809	4582
M6	5865	6989
M7	4019	3161
M8	4355	3540

is superior to the static one. For example, on D1, the precision of M5−M8 is 0.7506, 0.8760, 0.7133, and 0.6911, which is 42.46%, 50.70%, 39.45% and 37.50% higher than M4's corresponding values, respectively. However, the difference among M5−M8's recall and F1's M1−M4 is insignificant.

3) Dynamic Link Prediction Model Needs More Time Consumption Than Static One: As shown in Table IX, on D1, total time cost of M5−M8 is 2809, 5865, 4019 and 4355, which is 19.08%, 61.24%, 43.44% and 47.81% higher than M4's corresponding time, respectively.

V. FUTURE DEVELOPMENT TRENDS

Based on the above literature review, we present the future development trends in this section.

1) Efficient Dynamic CTNE Models: As shown in Table V, most of the cryptocurrency transaction datasets are temporal. However, existing CTNE models are mostly static. On the other hand, according to Table IV and Section III-C, existing dynamic CTNE models commonly focus on GNN, and GAT, which lead to high time complexity. Is it possible to achieve highly efficient dynamic CTNE models with the guarantee of high representation learning ability [119]–[132]? This question remains unveiled and calls for future efforts.

2) Multi-Attributed CTNE Models: In a CTN, most edges and nodes are attributive. For instance, the edges of the Mt. Gox transaction network [9] possesses four attributes: currently traded Bitcoin count, transaction time, the dollar count for Bitcoins, and the price per Bitcoin. However, existing CTNE models only focus on the time and Bitcoin count, while ignoring the other attributes. From this point of view, it appears necessary to embed a multi-attributive CTN into a multi-attributed CTNE model to precisely represent a target CTN's structure and semantic characteristics.

3) Applications in Market Analysis: Since the birth of Bitcoin, block-chain-based cryptocurrencies have attracted more and more investors and played an indispensable role in today's financial markets. Meanwhile, huge price fluctuation in cryptocurrency can be frequently encountered. Therefore, whether artificial manipulation exists in cryptocurrency has attracted extensive attention, which is a critical applications scene of CTNE research.

4) Application in Anomaly Detection: Blockchain technology provides anonymity protection, which brings convenience to users, but also become a hotbed of crime [133]–[137]. Fortunately, the transparency of blockchain and the irreversibility of cryptocurrency transactions provide researchers with the opportunity to detect abnormal transactions. As shown in Tables III and IV, anomaly detection includes phishing account detection [64]–[66], [79], fraud detection [78], and anomaly account detection [72], [113]. However, existing models are limited by their unsatisfactory accuracy, which should be carefully addressed as an important issue.

VI. Conclusions

With the successful application of blockchain technology, cryptocurrency is becoming more and more popular in people's daily life. Benefitting from the open and transparent nature of the blockchain technology, great convenience is provided for researchers to study the complete traces of financial activities in cryptocurrencies. Therefore, CTNE has attracted wide attentions. This paper thoroughly reviews the latest research of CTNE from static to dynamic versions. Firstly, CTN and the research status of network embedding models are introduced. Afterwards, the progress of CTNE from the static to dynamic perspectives is introduced, where the state-of-the-art is comprehensively reviewed, categorized and discussed. Subsequently, the typical CTNE evaluation metrics and datasets are summarized. Several popular CTNE models are also empirically validated to show their performance. Eventually, potential opportunity and direction of future research are summarized. We hope that this review can stimulate researchers and engineers to perform more and more research of CTNE and its applications.

References

[1] M. Saad, J. Choi, D. Nyang, J. Kim, and A. Mohaisen, "Toward characterizing blockchain-based cryptocurrencies for highly accurate predictions," *IEEE Syst. J.*, vol. 14, no. 1, pp. 321–332, Mar. 2020.

[2] X. Han, Y. Yuan, and F.-Y. Wang, "A blockchain-based framework for central bank digital currency," in *Proc. IEEE Int. Conf. Service Operations and Logistics, and Informatics*, Zhengzhou, China, 2019, pp. 263–268.

[3] A. Ghosh, S. Gupta, A. Dua, and N. Kumar, "Security of cryptocurrencies in blockchain technology: State-of-art, challenges and future prospects," *J. Netw. Comput. Appl.*, vol. 163, p. 102635, Aug. 2020.

[4] M. H. U. Rehman, K. Salah, E. Damiani, and D. Svetinovic, "Trust in blockchain cryptocurrency ecosystem," *IEEE Trans. Eng. Manage.*, vol. 67, no. 4, pp. 1196–1212, Nov. 2020.

[5] J. Bonneau, A. Miller, J. Clark, A. Narayanan, J. A. Kroll, and E. W. Felten, "SoK: Research perspectives and challenges for bitcoin and cryptocurrencies," in *Proc. IEEE Symp. Security and Privacy*, San Jose, USA, 2015, pp. 104–121.

[6] S. Dustdar, Fernández, J. M. García, and A. Ruiz-Cortés, "Elastic smart contracts in blockchains," *IEEE/CAA J. Autom. Sinica*, vol. 8, no. 12, pp. 1901–1912, Dec. 2021.

[7] H. M. Lu, Y. Tang, and Y. Sun, "DRRS-BC: Decentralized routing registration system based on blockchain," *IEEE/CAA J. Autom. Sinica*, vol. 8, no. 12, pp. 1868–1876, Dec. 2021.

[8] Y. Zhang and M. C. Zhou, "Security and trust in blockchains: Architecture, key technologies, and open issues," *IEEE Trans. Comput. Soc. Syst.*, vol. 7, no. 3, pp. 790–801, Jun. 2020.

[9] W. L. Chen, J. Wu, Z. B. Zheng, C. Chen, and Y. R. Zhou, "Market manipulation of bitcoin: Evidence from mining the Mt. Gox transaction network," in *Proc. IEEE Conf. Computer Communications*, Paris, France, 2019, pp. 964–972.

[10] B. Y. Gao, H. Y. Wang, C. Xia, S. W. Wu, Y. J Zhou, X. P Luo, and G. Tyson, "Tracking counterfeit cryptocurrency end-to-end," *Proc. ACM Meas. Analy. Comput. Syst.*, vol. 4, no. 3, p. 50, Jun. 2021.

[11] D. Vujičić, D. Jagodić, and S. Ranđić, "Blockchain technology, bitcoin, and Ethereum: A brief overview," in *Proc. 17th Int. Symp. Infoteh-Jahorina*, East Sarajevo, Bosnia and Herzegovina, 2018, pp. 1–6.

[12] M. Conti, E. S. Kumar, C. Lal, and S. Ruj, "A survey on security and privacy issues of bitcoin," *IEEE Commun. Surv. Tutorials*, vol. 20, no. 4, pp. 3416–3452, Apr. 2018.

[13] S. Wang, L. W. Ouyang, Y. Yuan, X. C. Ni, X. Han, and F.-Y. Wang, "Blockchain-enabled smart contracts: Architecture, applications, and future trends," *IEEE Trans. Syst.*, *Man*, *Cybern.: Syst.*, vol. 49, no. 11, pp. 2266–2277, Nov. 2019.

[14] F. Tschorsch and B. Scheuermann, "Bitcoin and beyond: A technical survey on decentralized digital currencies," *IEEE Commun. Surv. Tutorials*, vol. 18, no. 3, pp. 2084–2123, Mar. 2016.

[15] M. C. K. Khalilov and A. Levi, "A survey on anonymity and privacy in bitcoin-like digital cash systems," *IEEE Commun. Surv. Tutorials*, vol. 20, no. 3, pp. 2543–2585, Mar. 2018.

[16] Helo and Y. Hao, "Blockchains in operations and supply chains: A model and reference implementation," *Comput. Ind. Eng.*, vol. 136, pp. 242–251, Oct. 2019.

[17] Y. Zhang, M. C. Zhou, Q. X. Zhao, A. Abusorrah, and O. O. Bamasag, "A performance-optimized consensus mechanism for consortium blockchains consisting of trust-varying nodes," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 3, pp. 2147–2159, Jul.–Sept. 2021.

[18] Y. M. Liu, F. R. Yu, X. Li, H. Ji, and V. C. M. Leung, "Blockchain and machine learning for communications and networking systems," *IEEE Commun. Surv. Tutorials*, vol. 22, no. 2, pp. 1392–1431, Feb. 2020.

[19] L. Yang, "The blockchain: State-of-the-art and research challenges," *J. Ind. Inf. Integr.*, vol. 15, pp. 80–90, Sept. 2019.

[20] J. J. Wu, J. L. Liu, W. L. Chen, H. W. Huang, Z. B. 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.

[21] M. Mirtaheri, S. Abu-El-Haija, F. Morstatter, G. V. Steeg, and A. Galstyan, "Identifying and analyzing cryptocurrency manipulations in social media," *IEEE Trans. Comput. Soc. Syst.*, vol. 8, no. 3, pp. 607–617, Jun. 2021.

[22] S. Farrugia, J. Ellul, and G. Azzopardi, "Detection of illicit accounts over the Ethereum blockchain," *Expert Syst. Appl.*, vol. 150, p. 113318, Jul. 2020.

[23] J. J. Wu, J. L. Liu, Y. J. Zhao, and Z. B. Zheng, "Analysis of cryptocurrency transactions from a network perspective: An overview," *J. Netw. Comput. Appl.*, vol. 190, p. 103139, Sept. 2021.

[24] F. Zola, M. Eguimendia, J. L. Bruse, and R. O. Urrutia, "Cascading machine learning to attack bitcoin anonymity," in *Proc. IEEE Int. Conf. Blockchain*, Atlanta, USA, 2019, pp. 10–17.

[25] M. Caprolu, S. Raponi, G. Oligeri, and R. Di Pietro, "Cryptomining makes noise: Detecting cryptojacking via machine learning," *Comput. Commun.*, vol. 171, pp. 126–139, Apr. 2021.

[26] Z. C. He, J. Liu, N. Li, and Y. L. Huang, "Learning network-to-network model for content-rich network embedding," in *Proc. 25th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, Anchorage, USA, 2019, pp. 1037–1045.

[27] L. Z. Liao, X. N. He, H. W. Zhang, and T. S. Chua, "Attributed social network embedding," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 12, pp. 2257–2270, Dec. 2018.

[28] B. B. Hu, Y. Fang, and C. Shi, "Adversarial learning on heterogeneous information networks," in *Proc. 25th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, Anchorage, USA, 2019, pp. 120–129.

[29] P. Cui, X. Wang, J. Pei, and W. W. Zhu, "A survey on network embedding," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 5, pp. 833–852, May 2019.

[30] X. Wang, Y. F. Lu, C. Shi, R. J. Wang, P. Cui, and S. Mou, "Dynamic heterogeneous information network embedding with meta-path based proximity," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 3, pp. 1117–1132, Mar. 2022.

[31] L. C. Xu, X. K. Wei, J. N. Cao, and P. S. Yu, "Embedding of embedding (EOE): Joint embedding for coupled heterogeneous networks," in *Proc. 10th ACM Int. Conf. Web Search and Data Mining*, Cambridge, UK, 2017, pp. 741–749.

[32] A. Grover and J. Leskovec, "Node2vec: Scalable feature learning for networks," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, California, USA, 2016, pp. 855–864.

[33] J. Z. Qiu, Y. X. Dong, H. Ma, J. Li, K. S. Wang, and J. Tang, "Network embedding as matrix factorization: Unifying DeepWalk, LINE, PTE, and Node2vec," in *Proc. 11th ACM Int. Conf. Web Search and Data Mining*, Marina Del Rey, USA, 2018, pp. 459–467.

[34] C. C. Tu, W. C. Zhang, Z. Y. Liu, and M. S. Sun, "Max-margin DeepWalk: Discriminative learning of network representation," in *Proc. 25th Int. Joint Conf. Artificial Intelligence*, New York, USA, 2016, pp. 3889–3895.

[35] C. Yang, Z. Y. Liu, D. L. Zhao, M. S. Sun, and E. Y. Chang, "Network representation learning with rich text information," in *Proc. 24th Int. Conf. Artificial Intelligence*, Buenos, Argentina, 2015, pp. 2111–2117.

[36] S. S. Cao, W. Lu, and Q. K. Xu, "Deep neural networks for learning graph representations," in *Proc. AAAI Conf. Artif. Intell.*, vol. 30, no. 1, pp. 1145–1152, Feb. 2016.

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

[38] D. X. Wang, P. Cui, and W. W. Zhu, "Structural deep network embedding," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, California, USA, 2016, pp. 1225–1234.

[39] J. Tang, M. Qu, M. Z. Wang, M. Zhang, J. Yan, and Q. Z. Mei, "LINE: Large-scale information network embedding," in *Proc. 24th Int. Conf. World Wide Web*, Florence, Italy, 2015, pp. 1067–1077.

[40] X. Huang, J. D. Li, and X. Hu, "Label informed attributed network embedding," in *Proc. 10th ACM Int. Conf. Web Search and Data Mining*, Cambridge, UK, 2017, pp. 731–739.

[41] Z. H. Huang, Z. Y. Wang, R. Zhang, Y. Y. Zhao, X. H. Xie, and S. Mehrotra, "Network2Vec: Learning node representation based on space mapping in networks," in *Proc. Int. Conf. Data Mining Workshops*, Beijing, China, 2019, pp. 727–734.

[42] Z. Y. Zhao, H. Zhou, L. Qi, L. Chang, and M. C. Zhou, "Inductive representation learning via CNN for partially-unseen attributed networks," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 1, pp. 695–706, Jan.–Mar. 2021.

[43] J. Z. Qiu, Y. X. Dong, H. Ma, J. Li, C. Wang, K. S. Wang, and J. Tang, "NetSMF: Large-scale network embedding as sparse matrix factorization," in *Proc. World Wide Web Conf.*, San Francisco, USA, 2019, pp. 1509–1520.

[44] X. Liu, T. Murata, K. S. Kim, C. Kotarasu, and C. Y. Zhuang, "A general view for network embedding as matrix factorization," in *Proc. 12th ACM Int. Conf. Web Search and Data Mining*, Melbourne, Australia, 2019, pp. 375–383.

[45] B. Y. Chen, Z. Yang, and Z. W. Yang, "An algorithm for low-rank matrix factorization and its applications," *Neurocomputing*, vol. 275, pp. 1012–1020, Jan. 2018.

[46] Y. Song, M. Li, X. Luo, G. S. Yang, and C. J. Wang, "Improved symmetric and nonnegative matrix factorization models for undirected, sparse and large-scaled networks: A triple factorization-based approach," *IEEE Trans. Ind. Inf.*, vol. 16, no. 5, pp. 3006–3017, May 2020.

[47] G. F. Chen, C. Xu, J. Y. Wang, J. W. Feng, and J. Q. Feng, "Graph regularization weighted nonnegative matrix factorization for link prediction in weighted complex network," *Neurocomputing*, vol. 369, pp. 50–60, Dec. 2019.

[48] H. A. Song, B. K. Kim, T. L. Xuan, and S. Y. Lee, "Hierarchical feature extraction by multi-layer non-negative matrix factorization network for classification task," *Neurocomputing*, vol. 165, pp. 63–74, Oct. 2015.

[49] Y. Yuan and F.-Y. Wang, "Blockchain and cryptocurrencies: Model, techniques, and applications," *IEEE Trans. Syst., Man, Cybern.: Syst.*, vol. 48, no. 9, pp. 1421–1428, Sept. 2018.

[50] 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.

[51] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *Proc. 14th Int. Conf. Neural Information Processing System*, Vancouver, Canada, 2001, pp. 585–591.

[52] M. D. Ou, P. Cui, J. Pei, Z. W. Zhang, and W. W. Zhu, "Asymmetric transitivity preserving graph embedding," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, California, USA, 2016, pp. 1105–1114.

[53] X. Wang, P. Cui, J. Wang, J. Pei, W. W. Zhu, and S. Q. Yang, "Community preserving network embedding," in *Proc. 31st AAAI Conf. Artificial Intelligence*, San Francisco, USA, 2017, pp. 203–209.

[54] L. Hu, X. H. Yuan, X. Liu, S. W. Xiong, and X. Luo, "Efficiently detecting protein complexes from protein interaction networks via alternating direction method of multipliers," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 16, no. 6, pp. 1922–1935, Nov.–Dec. 2019.

[55] X. Luo, M. S. Shang, and S. Li, "Efficient extraction of non-negative latent factors from high-dimensional and sparse matrices in industrial applications," in *Proc. IEEE 16th Int. Conf. Data Mining*, Barcelona, Spain, 2016, pp. 311–319.

[56] J. D. Li, H. Dani, X. Hu, J. L. Tang, Y. Chang, and H. Liu, "Attributed network embedding for learning in a dynamic environment," in *Proc. ACM Conf. Information and Knowledge Management*, Singapore, Singapore, 2017, pp. 387–396.

[57] D. Y. Zhu, P. Cui, Z. W. Zhang, J. Pei, and W. W. Zhu, "High-order proximity preserved embedding for dynamic networks," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 11, pp. 2134–2144, Nov. 2018.

[58] K. Q. Chen, L. L. Yu, T. T. Zhu, L i, and J. Kurths, "Succinct representation of dynamic networks," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 7, pp. 2983–2994, Jul. 2021.

[59] A. Zhiyuli, X. Liang, Y. F. Chen, and X. Y. Du, "Modeling large-scale dynamic social networks via node embeddings," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 10, pp. 1994–2007, Oct. 2019.

[60] Y. L. Xiang, Y. Xiong, and Y. Y. Zhu, "TI-GCN: A dynamic network embedding method with time interval information," in *Proc. IEEE Int. Conf. Big Data*, Atlanta, USA, 2020, pp. 838–847.

[61] Y. J, Zhou, W. L. Liu, Y. Pei, L. Wang, D. R. Zha, and T. S. Fu, "Dynamic network embedding by semantic evolution," in *Proc. Int. Joint Conf. Neural Networks*, Budapest, Hungary, 2019, pp. 1–8.

[62] F. Xia, J. Y. Liu, H. S. Nie, Y. H. Fu, L. T. Wan, and X. J. Kong, "Random walks: A review of algorithms and applications," *IEEE Trans. Emerg. Top. Comput. Intell.*, vol. 4, no. 2, pp. 95–107, Apr. 2020.

[63] A. Narayanan, M. Chandramohan, R. Venkatesan, L. H. Chen, Y. Liu, and S. Jaiswal, "Graph2vec: Learning distributed representations of graphs," 2017. arXiv preprint arXiv: 1707.05005.

[64] Q. Yuan, B. Y. Huang, J. Zhang, J. J. Wu, H. N. Zhang, and X. Zhang, "Detecting phishing scams on ethereum based on transaction records," in *Proc. IEEE Int. Symp. Circuits and Systems*, Seville, Spain, 2020, pp. 1–5.

[65] B. Tao, I. W. H. Ho, and H. N. Dai, "Complex network analysis of the bitcoin blockchain network," in *Proc. IEEE Int. Symp. Circuits and Systems*, Daegu, Korea (South), 2021, pp. 1–5.

[66] Z. H. Yuan, Q. Yuan, and J. J. Wu, "Phishing detection on ethereum via learning representation of transaction subgraphs," in *Proc. 2nd Int. Conf. Blockchain and Trustworthy Systems*, Dali, China, 2020, pp. 178–191.

[67] L. J. Ma, Y. C. Ma, Q. Z. Lin, J. K. Ji, C. A. C. Coello, and M. G. Gong, "SNEGAN: Signed network embedding by using generative adversarial nets," *IEEE Trans. Emerg. Top. Comput. Intell.*, vol. 6, no. 1, pp. 136–149, Feb. 2022.

[68] X. M. Li, H. Fang, and J. Zhang, "Supervised user ranking in signed social networks," in *Proc. 33rd AAAI Conf. Artif. Intell.*, Palo Alto, USA, 2019, pp. 184–191.

[69] J. Li, Y. Liu, and L. Zou, "DynGCN: A dynamic graph convolutional network based on spatial-temporal modeling," in *Proc. 21st Int. Conf. Web Information Systems Engineering*, Amsterdam, The Netherlands, 2020, pp. 83–95.

[70] V. Patel, L. Pan, and S. Rajasegarar, "Graph deep learning based anomaly detection in ethereum blockchain network," in *Proc. 14th Int. Conf. Network and System Security*, Melbourne, Australia, 2020, pp. 132–148.

[71] T. Huang, Y. H. Zhang, J. J. Wu, J. Y. Fang, and Z. B. Zheng, "MG-GCN: Fast and effective learning with mix-grained aggregators for training large graph convolutional networks," 2020. arXiv preprint arXiv: 2011.09900.

[72] D. S. H. Tam, W. C. Lau, B. Hu, Q. F. Ying, D. M. Chiu, and H. Liu, "Identifying illicit accounts in large scale e-payment networks — A graph representation learning approach," 2019. arXiv preprint arXiv: 1906.05546.

[73] T. Derr, Y. Ma, and J. L. Tang, "Signed graph convolutional networks," in *Proc. IEEE Int. Conf. Data Mining*, Singapore, Singapore, 2018, pp. 929–934.

[74] M. Qu, Y. Bengio, and J. Tang, "GMNN: Graph Markov neural networks," in *Proc. 36th Int. Conf. Machine Learning*, Long Beach, USA, 2019, pp. 5241–5250.

[75] R. Agrawal and L. De Alfaro, "Learning edge properties in graphs from path aggregations," in *Proc. World Wide Web Conf.*, San Francisco, USA, 2019, pp. 15–25.

[76] V. Verma, M. Qu, K. Kawaguchi, A. Lamb, Y. Bengio, J. Kannala, and J. Tang, "GraphMix: Improved training of GNNs for semi-supervised learning," in *Proc. 35th AAAI Conf. Artif. Intell.*, Palo Alto, USA, 2021, pp. 10024–10032.

[77] J. J. Huang, H. W. Shen, L. Hou, and X. Q. Cheng, "SDGNN: Learning node representation for signed directed networks," in *Proc. 35th AAAI Conf. Artif. Intell.*, Palo Alto, USA, 2021, pp. 196–203.

[78] W. Kudo, M. Nishiguchi, and F. Toriumi, "GCNEXT: Graph convolutional network with expanded balance theory for fraudulent user detection," *Soc. Netw. Anal. Min.*, vol. 10, no. 1, p. 85, Oct. 2020.

[79] X. Liu, Z. Y. Tang, P. Li, S. Guo, X. Fan, and J. B. Zhang, "A graph learning based approach for identity inference in DApp platform blockchain," *IEEE Trans. Emerg. Top. Comput.*, vol. 10, no. 1, pp. 438–449, Jan.–Mar. 2022.

[80] J. J. Huang, H. W. Shen, L. Hou, and X. Q. Cheng, "Signed graph attention networks," in *Proc. 28th Int. Conf. Artificial Neural Networks*, Munich, Germany, 2019, pp. 566–577.

[81] Y. Li, Y. Tian, J. W. Zhang, and Y. Chang, "Learning signed network embedding via graph attention," in *Proc. 34th AAAI Conf. Artif. Intell.*, New York, USA, 2020, pp. 4772–4779.

[82] Y. Wu, B. J. Wang, and W. Li, "Hierarchical attention signed network," in *Proc. 4th Int. Conf. Imaging, Signal Processing and Communications*, Kumamoto, Japan, 2020, pp. 54–61.

[83] W. B. Liu, Z. D. Wang, X. H. Liu, N. Y. Zeng, Y. R. Liu, and F. E. Alsaadi, "A survey of deep neural network architectures and their applications," *Neurocomputing*, vol. 234, pp. 11–26, Apr. 2017.

[84] S. Y. Yao, Q. Kang, M. C. Zhou, M. J. Rawa, and A. Abusorrah, "A survey of transfer learning for machinery diagnostics and prognostics," *Artif. Intell. Rev.*, vol. 56, no. 4, pp. 2871–2922, Apr. 2023.

[85] Q. Liu, S. X. Liu, G. Y. Wang, and S. Y. Xia, "Social relationship prediction across networks using tri-training BP neural networks," *Neurocomputing*, vol. 401, pp. 377–391, Aug. 2020.

[86] De Meo, "Trust prediction via matrix factorisation," *ACM Trans. Internet Technol.*, vol. 19, no. 4, p. 44, Sept. 2019.

[87] H. Abdi, "Singular value decomposition (SVD) and generalized singular value decomposition," in *Encyclopedia Measurement and Statistics*, N. Salkind, Ed. Thousand Oaks, USA: Sage, 2007, pp. 907–912.

[88] Z. G. Liu, Y. G. Yi, and X. Luo, "A high-order proximity-incorporated nonnegative matrix factorization-based community detector," *IEEE Trans. Emerg. Top. Comput. Intell.*, 2022, DOI: 10.1109/TETCI.2022.3230930.

[89] Z. G. Liu, G. X. Yuan, and X. Luo, "Symmetry and nonnegativity-constrained matrix factorization for community detection," *IEEE/CAA J. Autom. Sinica*, vol. 9, no. 9, pp. 1691–1693, Sept. 2022.

[90] X. Luo, W. Qin, A. Dong, K. Sedraoui, and M. C. Zhou, "Efficient and high-quality recommendations via momentum-incorporated parallel stochastic gradient descent-based learning," *IEEE/CAA J. Autom. Sinica*, vol. 8, no. 2, pp. 402–411, Feb. 2021.

[91] W. Yu, R. J. Mu, Y. Sun, X. Chen, W. J. Wang, and H. M. Wu, "A double non-negative matrix factorization model for signed network analysis," in *Proc. IEEE Int. Conf. Parallel and Distributed Processing With Applications, Big Data and Cloud Computing, Sustainable Computing and Communications, Social Computing and Networking*, Xiamen, China, 2019, pp. 936–943.

[92] J. Wang and R. J. Mu, "A regularized convex nonnegative matrix factorization model for signed network analysis," *Soc. Netw. Anal. Min.*, vol. 11, no. 1, p. 7, Jan. 2021.

[93] C. B. He, H. Liu, Y. Tang, S. Y. Liu, X. Fei, Q. W. Cheng, and H. C. Li, "Similarity preserving overlapping community detection in signed networks," *Future Gener. Comput. Syst.*, vol. 116, pp. 275–290, Mar. 2021.

[94] R. Salakhutdinov and A. Mnih, "Probabilistic matrix factorization," in *Proc. 20th Int. Conf. Neural Information Processing Systems*, Vancouver, Canada, 2008, pp. 1257–1264.

[95] M. Muzammal, F. Abbasi, Q. Qu, R. Talat, and J. Fan, "A decentralised approach for link inference in large signed graphs," *Future Gener. Comput. Syst.*, vol. 102, pp. 827–837, Jan. 2020.

[96] S. Y. Liu, J. Xiao, and X. K. Xu, "Link prediction in signed social networks: From status theory to Motif families," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 3, pp. 1724–1735, Jul.–Sept. 2020.

[97] T. Qiu, M. Zhang, X. Z. Liu, J. Liu, C. Chen, and W. B. Zhao, "A directed edge weight prediction model using decision tree ensembles in industrial internet of things," *IEEE Trans. Ind. Inf.*, vol. 17, no. 3, pp. 2160–2168, Mar. 2021.

[98] S. Y. Liu, J. Xiao, and X. K. Xu, "Sign prediction by Motif naive Bayes model in social networks," *Inf. Sci.*, vol. 541, pp. 316–331, Dec. 2020.

[99] J. L. Pang, W. W. Yuan, and D. H. Guan, "Tri-domain pattern preserving sign prediction for signed networks," *Neurocomputing*, vol. 421, pp. 234–243, Jan. 2021.

[100] Q. Liu, Y. Chen, G. Q. Zhang, and G. Y. Wang, "A novel functional network based on three-way decision for link prediction in signed social networks," *Cogn. Comput.*, vol. 14, no. 6, pp. 1942–1954, Jun. 2021.

[101] D. Lin, J. J. Wu, Q. Yuan, and Z. B. Zheng, "T-edge: Temporal weighted multidigraph embedding for Ethereum transaction network analysis," *Front. Phys.*, vol. 8, p. 204, Jun. 2020.

[102] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space Odyssey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2222–2232, Oct. 2017.

[103] Y. Wang, C. W. Zhang, S. Wang, S. Yu, L. Bai, L. X. Cui, and G. D. Xu, "Generative temporal link prediction via self-tokenized sequence modeling," *World Wide Web*, vol. 23, no. 4, pp. 2471–2488, May 2020.

[104] F. Jiao, X. Guo, X. Jing, D. X. He, H. M. Wu, S. R. Pan, M. G. Gong, and W. J. Wang, "Temporal network embedding for link prediction via VAE joint attention mechanism," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 12, pp. 7400–7413, Dec. 2022.

[105] J. Charlier, R. Statem, and J. Hilger, "Modeling smart contracts activities: A tensor based approach," 2019. arXiv preprint arXiv: 1905.09868.

[106] J. Charlier, S. Lagraa, R. State, and J. François, "Profiling smart contracts interactions tensor decomposition and graph mining," in *Proc. 2nd Workshop Mining Data for Financial Applications*, Skopje, Macedonia, 2017, pp. 31–42.

[107] Y. M. Li, D. S. H. Tam, S. Y. Xie, X. X. Liu, Q. F. Ying, W. C. Lau, D. M. Chiu, and S. Z. Chen, "GTEA: Representation learning for temporal interaction graphs via edge aggregation," 2020. arXiv preprint arXiv: 2009.05266.

[108] D. H. Wang, Z. H. Zhang, Y. H. Ma, T. Zhao, T. W. Jiang, N. V. Chawla, and M. Jiang, "Learning attribute-structure co-evolutions in dynamic graphs," 2020. arXiv preprint arXiv: 2007.13004.

[109] S. Bonner, A. Atapour-Abarghouei, P. T. Jackson, J. Brennan, I. Kureshi, G. Theodoropoulos, A. S. McGough, and B. Obara, "Temporal neighbourhood aggregation: Predicting future links in temporal graphs via recurrent variational graph convolutions," in *Proc. IEEE Int. Conf. Big Data*, Los Angeles, USA, 2019, pp. 5336–5345.

[110] V. S. Dave and M. Al Hasan, "Triangle completion time prediction using time-conserving embedding," in *Proc. Joint European Conf. Machine Learning and Knowledge Discovery Databases*, Würzburg, Germany, 2020, pp. 541–557.

[111] A. Pareja, G. Domeniconi, J. Chen, T. F. Ma, T. Suzumura, H. Kanezashi, T. Kaler, T. Schardl, and C. Leiserson, "EvolveGCN: Evolving graph convolutional networks for dynamic graphs," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 4, pp. 5363–5370, Apr. 2020.

[112] C. M. Wu, G. Nikolentzos, and M. Vazirgiannis, "EvoNet: A neural network for predicting the evolution of dynamic graphs," in *Proc. 29th Int. Conf. Artificial Neural Networks*, Bratislava, Slovakia, 2020, pp. 594–606.

[113] L. Cai, Z. Z. Chen, C. Luo, J. P. Gui, J. C. Ni, D. Li, and H. F. Chen, "Structural temporal graph neural networks for anomaly detection in dynamic graphs," in *Proc. 30th ACM Int. Conf. Information and Knowledge Management*, Queensland, Australia, 2021, pp. 3747–3756.

[114] O. A. Malik, S. Ubaru, L. Horesh, M. E. Kilmer, and H. Avron, "Dynamic graph convolutional networks using the tensor M-product," 2019. arXiv preprint arXiv: 1910.07643.

[115] L. Akoglu, E. Terzi, O. A. Malik, S. Ubaru, L. Horesh, M. E. Kilmer, and H. Avron, "Dynamic graph convolutional networks using the tensor M-product," in *Proc. SIAM Int. Conf. Data Mining*, 2021, pp. 729–737.

[116] X. Ao, Y. Liu, Z. D. Qin, Y. Sun, and Q. He, "Temporal high-order proximity aware behavior analysis on Ethereum," *World Wide Web*, vol. 24, no. 5, pp. 1565–1585, Mar. 2021.

[117] H. F. Cao, Z. B. Zhang, L. Sun, and Z. Wang, "Inductive and irregular dynamic network representation based on ordinary differential equations," *Knowl.-Based Syst.*, vol. 227, p. 107271, Sept. 2021.

[118] F. Manessi, A. Rozza, and M. Manzo, "Dynamic graph convolutional networks," *Pattern Recogn.*, vol. 97, p. 107000, Jan. 2020.

[119] X. Luo, H. Wu, and Z. C. Li, "NeuLFT: A novel approach to nonlinear canonical polyadic decomposition on high-dimensional incomplete tensors," *IEEE Trans. Knowl. Data Eng.*, 2022, DOI: 10.1109/TKDE. 2022.3176466.

[120] L. Chen and X. Luo, "Tensor distribution regression based on the 3D conventional neural networks," *IEEE/CAA J. Autom. Sinica*, 2022, DOI: 10.1109/JAS.2022.123591.

[121] W. J. Qin, H. L. Wang, F. Zhang, J. J. Wang, X. Luo, and T. W. Huang, "Low-rank high-order tensor completion with applications in visual data," *IEEE Trans. Image Process.*, vol. 31, pp. 2433–2448, Mar. 2022.

[122] X. Luo, M. Z. Chen, H. Wu, Z. G. Liu, H. Q. Yuan, and M. C. Zhou, "Adjusting learning depth in nonnegative latent factorization of tensors for accurately modeling temporal patterns in dynamic QoS data," *IEEE Trans. Autom. Sci. Eng.*, vol. 18, no. 4, pp. 2142–2155, Oct. 2021.

[123] H. Wu, X. Luo, and M. C. Zhou, "Advancing non-negative latent factorization of tensors with diversified regularization schemes," *IEEE Trans. Serv. Comput.*, vol. 15, no. 3, pp. 1334–1344, May–Jun. 2022.

[124] H. Wu, X. Luo, M. C. Zhou, M. J. Rawa, K. Sedraoui, and A. Albeshri, "A PID-incorporated latent factorization of tensors approach to dynamically weighted directed network analysis," *IEEE/CAA J. Autom. Sinica*, vol. 9, no. 3, pp. 533–546, Mar. 2022.

[125] X. Luo, H. Wu, H. Q. Yuan, and M. C. Zhou, "Temporal pattern-aware QoS prediction via biased non-negative latent factorization of tensors," *IEEE Trans. Cybern.*, vol. 50, no. 5, pp. 1798–1809, May 2020.

[126] M. Z. Chen, C. L. He, and X. Luo, "MNL: A highly-efficient model for large-scale dynamic weighted directed network representation," *IEEE Trans. Big Data*, 2022, DOI: 10.1109/TBDATA.2022.3218064.

[127] D. Wu and X. Luo, "Robust latent factor analysis for precise representation of high-dimensional and sparse data," *IEEE/CAA J. Autom. Sinica*, vol. 8, no. 4, pp. 796–805, Apr. 2021.

[128] M. S. Shang, X. Luo, Z. G. Liu, J. Chen, Y. Yuan, and M. C. Zhou, "Randomized latent factor model for high-dimensional and sparse matrices from industrial applications," *IEEE/CAA J. Autom. Sinica*, vol. 6, no. 1, pp. 131–141, Jan. 2019.

[129] A. H. Khan, X. W. Cao, S. Li, V. N. Katsikis, and L. F. Liao, "BAS-ADAM: An ADAM based approach to improve the performance of beetle antennae search optimizer," *IEEE/CAA J. Autom. Sinica*, vol. 7, no. 2, pp. 461–471, Mar. 2020.

[130] C. C. Leng, H. Zhang, G. R. Cai, I. Cheng, and A. Basu, "Graph regularized L_p smooth non-negative matrix factorization for data representation," *IEEE/CAA J. Autom. Sinica*, vol. 6, no. 2, pp. 584–595, Mar. 2019.

[131] N. X. Yu, R. Yang, and M. J. Huang, "Deep common spatial pattern based motor imagery classification with improved objective function," *Int. J. Netw. Dyn. Intell.*, vol. 1, no. 1, pp. 73–84, Dec. 2022.

[132] E. R. Pinto, E. G. Nepomuceno, and A. S. L. O. Campanharo, "Campanharo.Individual-based modelling of animal brucellosis spread with the use of complex networks," *Int. J. Netw. Dyn. Intell.*, vol. 1, no. 1, pp. 120–129, Dec. 2022.

[133] M. M. Hassan, G. Fortino, L. T. Yang, H. Jiang, K. K. R. Choo, J. Zhang, and F.-Y. Wang, "Guest editorial for special issue on blockchain for Internet-of-Things and cyber-physical systems," *IEEE/CAA J. Autom. Sinica*, vol. 8, no. 12, p. 1867, Dec. 2021.

[134] S. K. Dwivedi, R. Amin, and S. Vollala, "Blockchain-based secured IPFS-enable event storage technique with authentication protocol in VANET," *IEEE/CAA J. Autom. Sinica*, vol. 8, no. 12, pp. 1913–1922, Dec. 2021.

[135] D. L. Xu, W. Shi, W. S. Zhai, and Z. H. Tian, "Multi-candidate voting model based on blockchain," *IEEE/CAA J. Autom. Sinica*, vol. 8, no. 12, pp. 1891–1900, Dec. 2021.

[136] Mamta, B. B. Gupta, K. C. Li, V. C. M. Leung, K. E. Psannis, and S. Yamaguchi, "Blockchain-assisted secure fine-grained searchable encryption for a cloud-based healthcare cyber-physical system," *IEEE/CAA J. Autom. Sinica*, vol. 8, no. 12, pp. 1877–1890, Dec. 2021.

[137] Y. Zhang, M. C. Zhou, C. X. Li, and A. Abusorrah, "Dynamic evolutionary game-based modeling, analysis and performance enhancement of blockchain channels," *IEEE/CAA J. Autom. Sinica*, vol. 10, no. 1, pp. 188–202, Jan. 2023.

Yue Zhou received the B.S. degree and the M.S. degree in computer science from China West Normal University, in 2017 and 2020, respectively. She is currently pursuing the Ph.D. degree in computer science with Chongqing University of Posts and Telecommunications (CQUPT), united training by Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences. Her research interests include big data analysis and algorithm design.

Xin Luo (Senior Member, IEEE) received the B.S. degree in computer science from the University of Electronic Science and Technology of China in 2005, and the Ph.D. degree in computer science from the Beihang University in 2011. He is currently a Professor of data science and computational intelligence with the College of Computer and Information Science, Southwest University. His current research interests include big data analysis and intelligent control. He has authored or coauthored over 200 papers (including over 100 IEEE Transactions papers) in the areas of big data analysis, graph learning, and manipulator control. He is currently serving as a Deputy Editor-in-Chief for *IEEE/CAA Journal of Automatica Sinica*, and an Associate Editor for *IEEE Transactions on Neural Networks and Learning Systems*. His Google page: https://scholar.google.com/citations?user=hyGlDs4AAAAJ&hl=zh-TW.

MengChu Zhou (Fellow, IEEE) received his B.S. degree in Control Engineering from Nanjing University of Science and Technology, Nanjing, China in 1983, M.S. degree in Automatic Control from Beijing Institute of Technology, Beijing, China in 1986, and Ph. D. degree in Computer and Systems Engineering from Rensselaer Polytechnic Institute, Troy, NY in 1990. He joined New Jersey Institute of Technology (NJIT), Newark, NJ in 1990, and has become a Distinguished Professor of Electrical and Computer Engineering. His research interests are in intelligent automation, machine leaning, Petri nets, robotics, Internet of Things, big data, cloud/edge computing, transportation and energy systems. He has over 1100 publications including 14 books, over 750 journal papers (over 600 in IEEE transactions), and 32 book-chapters. He holds 31 patents and several pending ones. His recently co-authored books include Sustainable Manufacturing Systems: An Energy Perspective, IEEE Press/Wiley, Hoboken, Hoboken, NJ, 2022 (with L. Li) and Supervisory Control and Scheduling of Resource Allocation Systems: Reachability Graph Perspective, IEEE Press/Wiley, Hoboken, NJ, 2020 (with B. Huang).

He is the founding Editor of IEEE Press Book Series on Systems Science and Engineering. He served as Editor-in-Chief of IEEE/CAA Journal of Automatica Sinica, Associate Editor of IEEE Transactions on Robotics and Automation, IEEE Transactions on Automation Science and Engineering, and IEEE Transactions on Industrial Informatics, and Editor of IEEE Transactions on Automation Science and Engineering. He served as a Guest-Editor for many journals including IEEE Internet of Things Journal, IEEE Transactions on Industrial Electronics, and IEEE Transactions on Semiconductor Manufacturing. He is presently Associate Editor of Research, IEEE Transactions on Intelligent Transportation Systems, IEEE Internet of Things Journal, IEEE Transactions on Systems, Man, and Cybernetics: Systems, and Frontiers of Information Technology & Electronic Engineering. He is founding Chair/Co-chair of Technical Committee on AI-based Smart Manufacturing Systems and Technical Committee on Humanized Crowd Computing of IEEE Systems, Man, and Cybernetics Society, Technical Committee on Semiconductor Manufacturing Automation and Technical Committee on Digital Manufacturing and Human-Centered Automation of IEEE Robotics and Automation Society. He is Chair of Fellow Evaluation Committee of Chinese Association of Automation. He is also a member of IEEE TAB Periodicals Committee and Periodicals Review and Advisory Committee. He was General Chair of IEEE Conf. on Automation Science and Engineering, Washington D.C., August 23-26, 2008, General Co-Chair of 2003 IEEE International Conference on System, Man and Cybernetics (SMC), Washington DC, October 5-8, 2003 and 2019 IEEE International Conference on SMC, Bari, Italy, Oct. 6-9, 2019, Founding General Co-Chair of 2004 IEEE Int. Conf. on Networking, Sensing and Control, Taipei, China, March 21-23, 2004, and General Chair of 2006 IEEE Int. Conf. on Networking, Sensing and Control, Ft. Lauderdale, Florida, U.S.A. April 23-25, 2006. He was Program Chair of 2010 IEEE International Conference on Mechatronics and Automation, August 4-7, 2010, Xi'an, China, 1998 and 2001 IEEE International Conference on SMC and 1997 IEEE International Conference on Emerging Technologies and Factory Automation. Dr. Zhou has led or participated in over 60 research and education projects with total budget over $12M, funded by National Science Foundation, Department of Defense, NIST, New Jersey Science and Technology Commission, and industry. He is a recipient of Excellence in Research Prize and Medal from NJIT, Humboldt Research Award for US Senior Scientists from Alexander von Humboldt Foundation, and Franklin V. Taylor Memorial Award and the Norbert Wiener Award from IEEE SMC Society, Computer-Integrated Manufacturing UNIVERSITY-LEAD Award from Society of Manufacturing Engineers, Distinguished Service Award from IEEE Robotics and Automation Society, and Edison Patent Award from the Research & Development Council of New Jersey. He has been among most highly cited scholars since 2012 and ranked top one in the field of engineering worldwide in 2012 by Web of Science. He has over 56,800 GoogleScholar citations with h-index 120. He was ranked #89 in the world and #58 in the United States among the 2022 Top 1000 Scientists in Computer Science in the World, Research.com. He is a life member of Chinese Association for Science and Technology-USA and served as its President in 1999. He is Fellow of IEEE, International Federation of Automatic Control (IFAC), American Association for the Advancement of Science (AAAS), Chinese Association of Automation (CAA) and National Academy of Inventors (NAI).I.