PACIS 2023 Proceedings

Pacific Asia Conference on Information Systems (PACIS)

7-8-2023

Identifying Crypto Addresses with Gambling Behaviors: A Graph Neural Network Approach

Jiaxin Wang
Nanjing University, mf21140113@smail.nju.edu.cn

Qian'ang Mao
Nanjing University, maoxs2@gmail.com

Jiaqi Yan
Nanjing University, jiaqiyan@nju.edu.cn

Hongliang Sun
Nanjing University of Finance &Economics, hlsun84@mail.ustc.edu.cn

Peixuan Qi
Nanjing University, 3455636126@qq.com

Recommended Citation

Wang, Jiaxin; Mao, Qian'ang; Yan, Jiaqi; Sun, Hongliang; and Qi, Peixuan, "Identifying Crypto Addresses with Gambling Behaviors: A Graph Neural Network Approach" (2023). *PACIS 2023 Proceedings*. 210.
https://aisel.aisnet.org/pacis2023/210

Identifying Crypto Addresses with Gambling Behaviors: A Graph Neural Network Approach

Completed Research Paper

Jiaxin Wang
Nanjing University
Nanjing, Jiangsu Province, China
mf21140113@smail.nju.edu.cn

Qian'ang Mao
Nanjing University
Nanjing, Jiangsu Province, China
maoxs2@gmail.com

Jiaqi Yan
Nanjing University
Nanjing, Jiangsu Province, China
jiaqiyan@nju.edu.cn

Hongliang Sun
Nanjing University of Finance &Economics
Nanjing, Jiangsu Province, China
hlsun84@mail.ustc.edu.cn

Peixuan Qi
Nanjing University
Nanjing, Jiangsu Province, China
peixuanqi@smail.nju.edu.cn

Abstract

The development of blockchain technology has brought prosperity to the cryptocurrency market and has made the blockchain platform a hotbed of crimes. As one of the most rampant crimes, crypto gambling has more high risk of illegal activities due to the lack of regulation. As a result, identifying crypto addresses with gambling behaviors has emerged as a significant research topic. In this work, we propose a novel detection approach based on Graph Neural Networks named CGDetector, consisting of Graph Construction, Subgraph Extractor, Statistical Feature Extraction, and Gambling Address Classification. Extensive experiments of large-scale and heterogeneous Ethereum transaction data are implemented to demonstrate that our proposed approach outperforms state-of-the-art address classifiers of traditional machine learning methods. This work makes the first attempt to detect suspicious crypto gambling addresses via Graph Neural Networks by all EVM-compatible blockchain systems, providing new insights into the field of cryptocurrency crime detection and blockchain security regulation.

Keywords: blockchain, cryptocurrency, gambling, Graph Neural Networks, Ethereum

Introduction

With the rapid development of blockchain technology(Yli-Huumo et al., 2016) in recent years, crypto gambling(Delfabbro et al., 2021), involving playing virtual crypto casino games using cryptocurrencies(Sockin & Xiong, 2023), has become a new and mainstream form of online gambling. The number of applications and users engaged in crypto gambling activities has shown a sharp growth trend(Min & Cai, 2022). Compared to traditional offline and online gambling, gamblers, agents and organizations utilizes crypto wallet addresses(Suratkar et al., 2020) to process payments and calculate the results of gambling games, helping us track and analyze gambling behaviors more comprehensively due to highly transparent and accessible characteristics of blockchain(Brown, 2021). A crypto address is a unique and public sequence of strings identifying wallets and is used by users to receive and send cryptocurrency funds(Suratkar et al., 2020). It transfers gambling funds or tokens in strict accordance with predefined rules of smart contracts(Wang et al., 2018) deployed on decentralized gambling platforms, which will solve key problems in traditional centralized gambling platforms, such as fictitious bonus pools, opaque game processes, high operating costs, and refusal to pay bonuses(Steinmetz & Fiedler, 2019). Blockchain technology allows for a high level of transparency and accessibility to data, helping us track and analyze gambling behaviors more comprehensively.

Cryptocurrency used in crypto gambling is a kind of digital currency, which is not issued by legal tender institutions and not controlled by central banks. With the help of blockchain(Zheng et al., 2017) and other emerging technologies, cryptocurrency mainly presents the characteristics and advantages such as decentralization, low transaction costs, international circulation, consensus mechanism, high anonymity, and distributed storage, which is more likely to become a tool for criminals to carry out illegal activities(Foley et al., 2019). Existing cryptocurrencies such as Bitcoin(Böhme et al., 2015) and Ethereum(Wood, 2014) are widely used in criminal activities such as cybercrime, gambling, money laundering, ransomware, theft, scam, terrorism, darknet market, etc.

Smart contracts adopted by crypto gambling are the core carrier of blockchain applications. It is a kind of automated script that allows trusted transactions without a third party(Zou et al., 2019). All business asset operations are defined by smart contracts, which are often used to build decentralized applications(DApps). DApps are not controlled by centralized institutions and are widely used in finance, gambling, gaming, etc(Cai et al., 2018). In the last few years, the number of decentralized gambling applications has grown dramatically, and these gambling DApps have implemented the use of cryptocurrencies for deposits and payments.

Naturally, crypto gambling has a high risk of fraud and illegal transaction activities(Brito et al., 2014), which includes cross-border money laundering, false advertising, tampering with game rules, exploiting vulnerabilities or attacking codes to control game outcomes or steal user information, and placing bets using stolen or fake digital assets due to a lack of regulation(K. Wu et al., 2021).

As a result, detecting and identifying crypto gambling addresses in large-scale blockchain networks is of great research significance, which not only provides the ecological picture of crypto gambling but also provides theoretical references for monitoring and cracking down on crypto gambling behaviors.

However, there is still no available approach to the identification of crypto gambling addresses. In comparison to traditional gambling, we concluded three key challenges to detecting crypto gambling addresses: (1)The gambling transactions, accounts, and smart contracts in blockchain are heterogeneous and multi-modal, thus lacking extraction, mining, and organization of these gambling information. (2)Large-scale crypto gambling transaction data and complex gambling transaction behaviors make it difficult to efficiently carry out graph learning for all gambling addresses. (3)The scarce annotated labels of crypto addresses and only a single dimension of manual-designed features result in the poor performance of detection.

To tackle these challenges, we seek to formulate crypto gambling address identification problems. We first collect addresses directly involved in gambling transactions from the blockchain browser and decentralized gambling sites, and then simultaneously synchronize and parse transactions associated with gambling addresses. In particular, we design a novel effective detection approach based on Graph Neural Network named **CGDetector**(Crypto Gambling Detector). The approach consists of four key components:

(1)**Graph Construction**: We construct a homogeneous weighted directed *lightweight*-Account Transaction Graph(ATG) model to match addresses and transactions in cyberspace with physical space. (2)**Subgraph Extractor**: Since large-scale account transaction graphs are time-consuming and computation resource-consuming during full-batch training, we extract neighborhood subgraphs of target gambling addresses from the full account transaction graph to achieve efficient mini-batch training. (3) **Statistical Feature Extraction:** we manually extract statistical features of addresses including domain-based features related to basic transaction information and graph-based features related to graph structural information. (4)**Gambling Address Classification**: On one hand, we utilize Graph Neural Networks(GNN) to learn graph embeddings of each address, which can effectively describe node-level gambling characteristics and behavior patterns. On the other hand, the final graph representation is aggregated into the address-level classification to detect whether an address is a gambling address.

The two key contributions of this research are summarized as follows:

(1) To the best of our knowledge, this work is the first investigation of crypto gambling identification via graph neural network models. Compared to traditional gambling activities**,** we focus on a new research scenario of identifying crypto addresses with gambling behaviors from large-scale, heterogeneous, and multi-modal on-chain data, which presents a new insight into the field of cryptocurrency crime detection and blockchain ecological security regulation.

(2) We propose a novel crypto gambling addresses detection approach (**CGDetector**) for all EVM-compatible cryptocurrencies. Within the approach, we innovatively design Graph Construction, Subgraph Extractor, Statistical Feature Extraction, and Gambling Address Classification based on Graph Neural Networks. Extensive experiments for Ethereum are implemented to demonstrate the efficiency of the proposed approach and yield new discoveries.

The remainder of the paper is organized as follows: Sect. 2 provides a retrospect of the existing research on crypto address identification; Sect. 3 elaborates the design of our proposed approach; Sect. 4. presents experimental datasets and results, and answers two research questions respectively; Sect. 5 illustrates relevant discussion and research implication; Sect. 6 concludes our research and recommends future perspectives.

Related Work

There has been a lot of work concentrating on detecting abnormal behaviors on blockchain such as Ponzi, frauds, and phishing scams. In this section, we summarized the existing studies on the detection of these abnormal behaviors on the blockchain in recent years. A considerable part of existing methods regards crypto address identification as a classification task, mainly focusing on machine learning algorithms. Supervised machine learning, unsupervised machine learning, and deep learning algorithms have been widely applied to capture transaction characteristics of different accounts. Table 1 shows the comparison of the state-of-the-art detection methods. Among them, the types of crypto addresses are mainly concentrated in the two largest and relatively mature cryptocurrencies, namely Bitcoin and Ethereum.

Methods based on supervised machine learning and unsupervised machine learning algorithms have been widely used to identify different types of crypto addresses. Supervised machine learning methods are mainly divided into two paradigms: binary classification and multi-classification. A large number of studies show that different identity types have different transaction behavior patterns, so feature engineering plays a crucial role in distinguishing and identifying the identity entities of addresses. However, these types of methods ignore the existence of financial flows and fail to take advantage of important neighborhood relationships and rich transaction structure features. Only a single dimension of manual-designed features further lead to the weak capabilities of these detection methods.

Therefore, a considerable number of existing research analyze cryptocurrency transactions from the perspective of networks(J. Wu et al., 2021). By abstracting the objects in the cryptocurrency system (such as address, transaction, smart contract, user, etc.) as nodes and taking the interaction between objects as connections, we can construct graph modeling of interactive activities from different perspectives, automatically analyze graph characteristics, reveal abnormal transaction graph structures, so as to better identify nodes with abnormal transaction behavior. Graph deep learning provides a perspective for large-scale heterogeneous data modeling in cryptocurrency transactions. Especially, graph neural networks(Z. Wu et al., 2020) have been proven to be very powerful in graph representation learning ability in various fields. Currently, it has been applied widely in the research of crypto address identification, which is mainly divided into two paradigms: binary classification and multi-classification.

However, existing detection methods based on graph neural networks cannot be directly applied to our research. Crypto gambling ecosystem involves abundant token transfers, and crypto gambling addresses have typical transaction characteristics and structural characteristics. Therefore, we propose a novel crypto gambling address detection approach based on graph learning technology for all EVM-compatible cryptocurrencies, in which Graph Construction, Subgraph Extractor and Statistical Feature Extraction Module are well designed.

Type	Reference	Crypto	Identification Problem	Models	Classification
Supervised machine learning	(Bartoletti et al., 2018)	Bitcoin	Ponzi scheme	BN, RF	Binary classfication
	(Ostapowicz & Żbikowski, 2019)	Bitcoin	Fraud	RF, SVM, XGBoost	
	(Li et al., 2020)	Bitcoin	Illicit addresses	RF, SVM, XGBoost, ANN	
	(Toyoda et al., 2019)	Bitcoin	Illicit addresses	RF, XGBoost, ANN, SVM, KNN	
	(Farrugia et al., 2020)	Ethereum	Illicit accounts	XGBoost	
	(Ibrahim et al., 2021)	Ethereum	Fraud	DT, RF, KNN	
	(Poursafaei et al., 2020)	Ethereum	Malicious addresses	LR, SVM, RF, AdaBoost	
	(Chen et al., 2020)	Ethereum	Phishing scam	LightGBM	
	(Michalski et al., 2020)	Bitcoin	Mining pools, mixing services, gambling, exchanges, etc	RF	Multi-classfication
	(Liang et al., 2019)	Bitcoin	Exchange, gambling, service, etc	Decision Tree	
Unsupervised learning algorithms	(P. Monamo et al., 2016)	Bitcoin	Fraud	Evolve K-means	—
	(P. M. Monamo et al., 2016)	Bitcoin	Fraud	Kd-tree	
	(Poursafaei et al., 2020)	Ethereum	Malicious addresses	K-means, Unsupervised SVM	
Deep learning	(Tian et al., 2021)	Bitcoin	Illicit addresses	Evolve GNN	Binary classfication
	(Shao et al., 2018)	Bitcoin	Address classification	Graph Embedding	Multi-classfication
	(Wen et al., 2023)	Ethereum	Phishing scam	Deep Neural Network	Binary classfication
	(Zhou et al., 2022)	Ethereum	Account classification	GCN	Multi-classfication
	(X. Liu et al., 2022)	Ethereum	Account classification	GCN	
	(J. Liu et al., 2022)	Ethereum	Account classification	Evolve GNN	
	Our work	**Ethereum, Tron, EOS, etc**	**Gambling**	**GNNs**	**Binary classfication**

Table 1. Machine Learning Algorithms of Crypto Address Identification

(Note: BN--Bayes Network, RF--Random Forest, SVM--Support Vector Machine, DT—Decision Tree, KNN--K-nearest Neighbor, LR--Logistic Regression, GCN--Graph Convolutional Network)

Our Method

Problem Formulation

In view of research gaps proposed above, we mainly focus on detecting crypto gambling addresses via graph neural networks in this paper. We consider crypto gambling detection as a graph classification task. Cryptocurrency transaction data can be modeled and simplified as a uniformed network $\boldsymbol{G} = (V, E, W)$, where $V = \{v_1, v_2, v_3, \dots, v_n\}$ represents the set of addresses, $E = \{(v_i, v_j) | v_i, v_j \in V\}$ represents the set of transactions with the set of amount values for $W = \{w_1, w_2, w_3, \dots, w_n\}$. We set $\boldsymbol{Y} = \{(v_i, y_i) | \; v_i \in V\} \in \mathbb{R}^{n \times 2}$ as the address label matrix representing whether node v_i is related to gambling or not. $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ represents the adjacency matrix, where $\boldsymbol{A_{ij}} = 1$ denotes there is an edge between v_i and v_j else $\boldsymbol{A_{ij}} = 0$. Each node $v_i \in V$ has a corresponding feature vector $\boldsymbol{x_i} \in \mathbb{R}^{F_1 \times 1}$. $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_n]^{\boldsymbol{T}} \in \mathbb{R}^{n \times F_1}$ denotes the node feature matrix. For a given subgraph $G_{v_i} \subset G$ for address v_i, we need to learn a function $f(G_{v_i}) \longrightarrow \hat{y_i}$ mapping the subgraph G_{v_i} to label $\hat{y_i}$ to detect whether the target address v_i is a gambling account or not.

Method Overview

As is depicted in Fig. 1, we present the details of our proposed research approach **CGDetector**, consisting of four key components: (1) Graph Construction, (2)Subgraph Extractor (3) Statistical Feature Extraction, and (4)Gambling Address Classification. For a target address v_i, the input of **CGDetector** is the subgraph ATS_i sampled from ATG, and the output is the predictive identity label y_i. Next, we will introduce the four components, respectively.

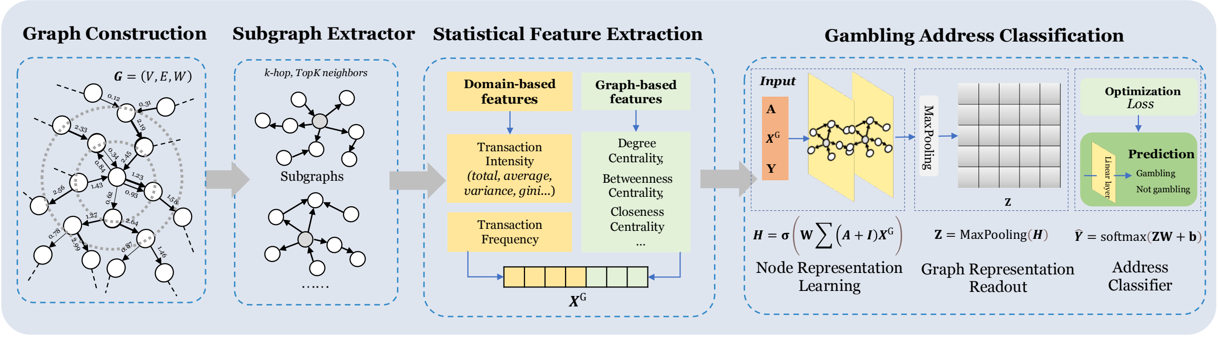


Figure 1. The Research Approach of Crypto Gambling Detector (CGDetector)

Graph Construction

In most crypto gambling cases, multiple account addresses have a tendency to point to the same receiving address, that is, multiple money may be transferred to the same gambling pool. These frequent behaviors are usually not easy to be captured by traditional features but are clearly detected by a graph structure. At the same time, crypto gambling transactions, accounts, and smart contracts are heterogeneous and multi-modal. As a result, the first step of the **CGDetector** approach is to convert address transactions into a unified graph structure from heterogeneous gambling information.

To be specific, we track the source and flow of transactions from collected gambling addresses in order to deeply sort out the flow structure of transactions and visually present the complex Account Transaction Graph**(ATG)**. This paper mainly depends on Breadth First Search(BFS) algorithm in breadth-first search(Kozen, 1992).

At present, crypto gambling mainly relies on three public chains namely Ethereum, EOS, and Tron, which are all the Account Model(Brünjes & Gabbay, 2020) in their transaction execution. Based on the Account Model, every transaction has only one input address and one output address, so the input or output relationship is clear. To visualize and simplify transaction data, we model ***lightweight*-ATG** as a homogeneous, weighted and directed graph $G = (V, E, W)$ for every address ad_i , where $V = \{v_1, v_2, v_3, \dots, v_n\}$ represents the set of addresses, $E = \{(v_i, v_j)|v_i, v_j \in V\}$ represents the set of transactions sent and received between addresses, and $W = \{w_1, w_2, w_3, \dots, w_n\}$ represents the set of transaction amounts transferred by the address. Notably, G has two types of edges, that is, $E = (Input, Output)$, where $Input$ represents input transactions of the address, and $Output$ represents output transactions of the address. Figure 2 below shows a subgraph example of a lightweight Account Transaction Graph(ATG) for Ethereum. The nodes in the figure represent addresses, the directed edges between nodes respectively represent input transactions and output transactions, and the weight of the edges represents transaction amounts.

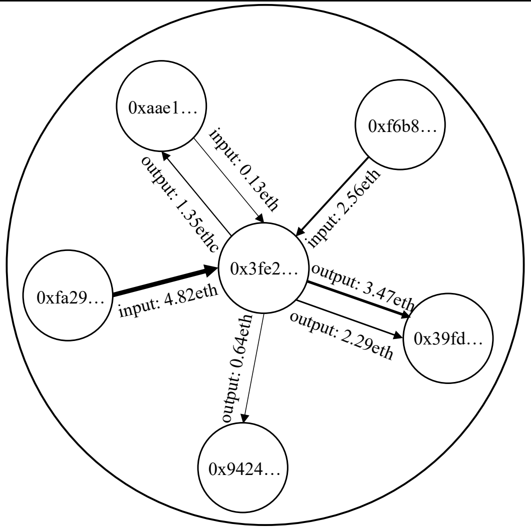


Figure 2. A Subgraph Example of Lightweight Account Transaction Graph(ATG) for Ethereum

Subgraph Extractor

Considering huge transaction data and algorithm complexity, a large-scale Account Transaction Graph**(ATG)** is not feasible for full-batch training of GNNs. On the contrary, subgraphs are the adjacent fields of center target nodes, which is much smaller than the whole graph and allows for mini-batch training. Furthermore, subgraphs consisting of the target account and its local neighborhood information imply partial transaction behavior patterns, which play a vital role in address identification.

Thus, we convert crypto address identification as a subgraph-level classification task by subgraph sampling strategies that allow for mini-batch training of GNNs from a large-scale Account Transaction Graph. On one hand, we restrict the number of tracing layers to obtain k-hop transaction subgraphs. On the other hand, for a target address node v_i, we perform node-level sampling to obtain the top-*K* most important neighbors based on edge attribute(transaction amount) and perform recursive operations on each account address sampled in the previous hop with the same strategy. At last, we generate Account Transaction Subgraph for each targeted address v_i. The subgraph sampling can be formulated as follows:

$$N_k = \bigcup_{v_i \in V_{k-1}} \boldsymbol{SE}(k, K, v_i, E_w(v_i)) \tag{1}$$

where N_k denotes the number of sampled address nodes at hop k. $v_i \in V_{k-1}$ represents the set of nodes at hop k-1. ***SE*** represents the subgraph extraction function that returns the top-*K* most important nodes of each address at limited tracing hops. k denotes the number of hops. K denotes the number of sampled neighbors per hop. $E_w(v_i)$ denotes the set of transaction amounts of node v_i at hop k-1.

Statistical Feature Extraction

In view of existing research on anomaly detection on blockchain from related work, it can be concluded that the attributes of addresses can reflect account categories to a certain extent. It is intuitive to understand that a single dimension of features is unable to characterize gambling behaviors well and reduce the performance of the classification algorithms based on manual statistical features.

Unlike other criminal activities, gambling activities have the following different characteristics:

(1) Transfer values each time between gambling platforms and their neighbor nodes(gamblers) are usually similar.
(2) Gamblers participate in gambling activities at a high frequency, which is reflected in the graph structure as multiple edges between gambling platforms and gamblers over a period of time.
(3) The input transfers of gambling platform addresses always outnumber the output transfers. Gambling platforms will transfer the balance out, indicating that they are essentially profitable entities.
(4) Generally speaking, the number of gamblers who win money in gambling activities is less than the total number of participants, so the number of transactions transferred by gamblers to the gambling platform is far greater than the number of transactions transferred by the gambling platform to gamblers.
(5) There is a higher probability that gamblers will have few transfers with each other, which reflects the neighbors of the contract address in the graph tend to disperse.

Our key insight is that the differences of these characteristics are mainly reflected in the transfer amount, transfer volume, and network structure. Therefore, we now introduce a set of manually extracted features, ranging from domain-based features related to basic transaction information to graph-based features related to graph structural information. Next, we will specifically introduce the extraction of these two types of features.

• Domain-based features

Domain-based features refer to basic characteristics associated with addresses and transactions, including two types: Transaction intensity and transaction frequency. Transaction intensity characteristics refer to those related to transaction amounts, balance changes, gas fees, etc, while transaction frequency characteristics refer to those related to input and output transaction numbers within a certain period of time, the ratio between input and output transactions, etc. At the same time, we further obtain new aggregated features to characterize the attributes of nodes by summing, averaging, variance, Gini coefficient, etc. The detailed features are described in Table 2 below.

Type	Symbol	Description
Transaction Intensity	Total_amount	The total amount of all transactions of the address
	Total_in_amount	The total amount of input transactions of the address
	Total_out_amount	The total amount of output transactions of the address
	Avg_amount	The average amount of all transactions of the address
	Avg_in_amount	The average amount of input transactions of the address
	Avg_out_amount	The average amount of output transactions of the address
	Var_amount	The standard deviation amount of all transactions of the address
	Var_in_amount	The standard deviation amount of input transactions of the address
	Var_in_amount	The standard deviation amount of output transactions of the address
	Gini_amount	The Gini coefficient amount of all transactions of the address
	Gini_in_amount	The Gini coefficient amount of input transactions of the address
	Gini_out_amount	The Gini coefficient amount of output transactions of the address
	Balance	The number of Ether tokens owned by the address

	Nounce	The total transaction amount of the address
	Gas	The transaction gas supplied by the sender of the address
	Gas_in	The input transaction gas supplied by the sender of the address
	Gas_out	The output transaction gas supplied by the sender of the address
	Gas_price	The gas price set by the transaction sender of the address
	Gas_in_price	The gas price set by the input transaction sender of the address
	Gas_out_price	The gas price set by the output transaction sender of the address
Transaction Frequency	Num_all_tran	The number of all transactions of the address
	Num_in_tran	The number of input transactions of the address
	Num_out_tran	The number of output transactions of the address
	Fre_all_tran	The frequency of all transactions of the address
	Fre_in_tran	The frequency of input transactions of the address
	Fre_out_tran	The frequency of output transactions of the address
	R_in_out	The ratio of the number of input/output transactions of the address

Table 2. Domain-based Features

• **Graph-based features**

We further need to extract graph structural features to capture structural information of nodes in the graph. Here we select Degree centrality, Closeness centrality, Betweenness centrality, and Eigenvector centrality metrics. We incorporate such features of network centrality into graph node features.

Degree Centrality(Zhang & Luo, 2017) is the most direct measure to describe node centrality in network analysis. The more degree centrality a node has, the more important it is in the network. Degree centrality of node v_i is defined as:

$$DC(v_i) = \frac{k_{v_i}}{N-1} \tag{2}$$

where N is the total number of nodes in the network, k_{v_i} is the degree of node v_i.

Closeness Centrality is the sum of the lengths of the shortest paths between one node and all the other nodes in the diagram. Therefore, the higher the closeness centrality value of a node, the closer it is to all other nodes. Closeness centrality of node v_i is defined as:

$$CC(v_i) = \frac{N-1}{\sum_{j \neq i}^{n} g(v_i, v_j)} \tag{3}$$

where N is the total number of nodes in the network, $g(v_i, v_j)$ is the shortest distance between node v_i and v_j.

Betweenness Centrality is the number of shortest paths through nodes in a network, that is, all the shortest paths between any two nodes in the network are counted. If any of the shortest paths pass through a node, the node has a high betweenness centrality. Betweenness centrality of a node v_i is defined as:

$$BC(v_i) = \sum_{v_m \neq v_i \neq v_n, m<n} \frac{\sigma_{mn}(v_i)}{\sigma_{mn}} \tag{4}$$

where σ_{mn} is the total number of shortest paths from node m to node n, $\sigma_{mn}(v_i)$ is the total number of shortest paths that pass through the node v_i.

Eigenvector Centrality indicates that the importance of a node depends both on the number and importance of its neighbors. That is, the contribution of high-score nodes to this node is greater than that of low-score nodes. Eigenvector centrality of a node v_i for a graph $G(V, E)$ is defined as:

$$EC(v_i) = \frac{1}{\lambda}\sum A_{v_i v_j} EC(v_j) \tag{5}$$

where v_j is the neighbor of v_i, λ is a constant, which is the eigenvalue of the matrix A, and A is the adjacency matrix of the network.

Finally, we can obtain the aggregated feature matrix X^G of nodes as below:

$$\boldsymbol{X}^G = [\boldsymbol{X}^d, \boldsymbol{X}^c] \tag{6}$$

where X^G is the aggregated graph feature matrix of nodes, X^d is the graph-based features matrix of nodes, X^c is the domain-based feature matrix of nodes.

Gambling Address Classification

In this section, we present the details of crypto gambling addresses classification, which is divided into three steps: (1) automatically aggregate nodes' information through Node Representation Learning, (2) generate the complete embedding of target nodes through Graph Representation Readout, (3) classify gambling-related addresses through the full connection layer.

Node Representation Learning This step aims to represent graph nodes with low-dimensional vectors and preserve as much topological information as possible. We employ graph neural networks to capture more effective graph features to provide a more abstract representation of nodes in a graph. This process can be expressed as follows:

$$\boldsymbol{H} = \boldsymbol{\sigma}\left(\mathbf{W}\sum(\boldsymbol{A} + \boldsymbol{I})\boldsymbol{X}^{\mathrm{G}}\right) \tag{7}$$

where $\boldsymbol{\sigma}(\cdot)$ is a nonlinear activation function such as ReLU and W is the trainable parameters of the neural network. $\boldsymbol{A}$ is the graph adjacency matrix and $\boldsymbol{I}$ is the identity matrix.

Graph Representation Readout The output function produces the graph representation by aggregating representation of all nodes in the graph. Here we adopt the Max pooling function to aggregate all the node embeddings in the graph to obtain the final graph embedding.

$$\mathbf{Z} = \mathrm{MaxPooling}(\boldsymbol{H}), v \in V^{\mathrm{G}} \tag{8}$$

where the graph embedding $\mathbf{Z}$ is the final representation of graph G, $\mathrm{MaxPooling}(\cdot)$ is a pooling function.

Address Classifier The goal of the address classification phase is to use the output embedding $\mathbf{Z}$ in Eq. (8) for semi-supervised classification. Suppose $\widehat{\boldsymbol{Y}} \in \mathbb{R}^{\mathrm{n}\times 2}$ denotes the probability of nodes related to crypto gambling behaviors or normal nodes. Then $\widehat{\boldsymbol{Y}}$ can be calculated with a linear transformation and a softmax function:

$$\widehat{\boldsymbol{Y}} = \mathrm{softmax}(\mathbf{ZW} + \mathbf{b}) \tag{9}$$

where $\mathbf{W}$ and $\mathbf{b}$ are parameters of full connected layer. The model uses binary_cross_entropy_with_logits loss function by the classification task:

$$\mathcal{L} = -w_i \sum_{i=1}^{N} [y_i \cdot log\widehat{y_i} + (1 - y_i) \cdot log(1 - \widehat{y_i})] \tag{10}$$

where N denotes the number of all nodes. w_i denotes the weight of categories. $\widehat{y_i} \in \widehat{\boldsymbol{Y}}$ denotes the predictive identity label. $y_i \in \boldsymbol{Y}$ denotes the actual identity label.

Experiment

In this section, we empirically evaluate our proposed research approach on large-scale real-world crypto gambling addresses. We seek to answer the following research questions:

• RQ1: How to evaluate the performance of CGDetector using Graph Neural Network models compared to traditional machine learning methods?

• RQ2: How do different hyperparameters affect our proposed approach?

Next, we first present the experimental settings and data preparation, followed by answering the above research questions one by one.

Experimental Settings

Implementation details

We conducted extensive experiments on a server that has dual AMD EPYC 7763 64-Core Processors running at 1.50GHz, equipped with 2TiB of memory and 47GiB of swap space. Additionally, the server has 8 NVIDIA GeForce RTX 3090 graphics cards and stores all data in solid-state drives (SSD). We used Rust programming language to implement the data gathering modules, while the training modules were implemented in Python, utilizing the PyTorch approach. Furthermore, we leveraged the pytorch-geometric package to support graph neural network models.

Evaluation metrics

For binary classification problems, classification results often appear in four situations: that is, positive class is determined as positive class (TP), positive class is determined as negative class (FN), negative class is determined as positive class (FP), and negative class is determined as negative class (TN). The confusion matrix is shown in Table 3 below.

		Prediction	
		Positive	Negative
Reference	Positive	True Positive, TP	False Positive, FP
	Negative	False Negative, FN	True Negative, TN

Table 3. Confusion Matrix Table

Accuracy, precision, recall and F1-score value are basic evaluation indicators, which can be obtained according to the confusion matrix(Goutte & Gaussier, 2005). Their concrete definitions will be briefly introduced below:

• **Accuracy**: Accuracy is used to measure the proportion of samples with correct classification in the total number of samples. However, this evaluation method cannot fully evaluate the performance of the model. Accuracy is defined as:

$$\text{Accuracy}=\frac{TP+TN}{TP+TN+FP+FN} \tag{11}$$

• **Precision**: Precision is used to measure the proportion of the sample that is predicted to be positive. Precision is defined as:

$$\text{Precision}=\frac{TP}{TP+FP} \tag{12}$$

• **Recall:** Recall rate is used to measure the proportion of samples that can be identified in a true positive sample. Recall is defined as:

$$\text{Recall}=\frac{TP}{TP+FN} \tag{13}$$

• **F1-score:** Since accuracy rate and recall rate are contradictory, they need to be integrated. F1-score is the index of harmonizing accuracy rate and recall rate. It is more objective and fair to use F1-score value to evaluate the model. F1-score is defined as:

$$F_1=2\times\frac{1}{\frac{1}{Precision}+\frac{1}{Recall}}=\frac{2\times Precision\times Recall}{Precision+Recall} \tag{14}$$

Based on above basic indicators, we further adopt two widely-used and complementary metrics: AUC-ROC and AUC-PR(Khan & Ali Rana, 2019). To be specific, the X-axis of ROC curve is false positive rate(FP), and the Y-axis is true positive rate(TP). AUC-ROC is the area formed by ROC curve and X-axis. Compared to Accuracy and Precision, AUC-ROC and AUC-PR are not sensitive to the ratio of positive and negative samples, which are suitable for unbalanced sample datasets. After comprehensive consideration, we finally select Recall, F1-score, AUC-ROC, and AUC-PR for model performance comparison.

Dataset Preparation

To the best of our knowledge, Ethereum is an important part of the cryptocurrency ecosystem and by far the most popular digital asset among gambling criminals. The following two types of experimental datasets were selected for this paper: Ethereum transaction dataset and Ethereum address label dataset. Ethereum transaction dataset records all transaction data for addresses. Ethereum address label dataset helps map anonymous Ethereum users to real-world identities, which includes Ethereum gambling contract addresses. The sources and processing of these datasets are detailed below.

Ethereum transaction dataset

All Ethereum transactions are recorded on a distributed public ledger. Each Ethereum block contains hundreds of public transaction records. In this paper, the official open-source Ethereum client go-ethereum[1] is used to download Ethereum transaction data in the corresponding time window and automatically synchronize the data. Go-ethereum is one of the earliest and most popular Ethereum clients that allows us to interact with Ethereum network environment based on an interactive command console.

The transaction data downloaded by go-ethereum needs to be further parsed before it can be stored in the graph database, so the raw data is parsed into individual transactions using the open source tool ethereum-etl[2], which is a tool for bulk parsing of Ethereum transaction data. Finally, the parsed data will be imported into Neo4j graph database[3].

Ethereum address label dataset

In this study, we obtained publicly available Ethereum address label datasets mainly from etherscan.io[4] and xblock.pro[5]. Address labels cover different types of entities, including but not limited to exchanges, Defi, cross-chain Bridges, games, gambling services, mining pools, scams and other services.

In particular, gambling services on blockchain often need to be developed and maintained by people with technical backgrounds and experience, and gambling service providers usually get certain profits, so these people are both gambling service providers and gambling players. Thus, we explored a number of decentralized gambling applications and obtained 1-hop addresses directly related to these gambling DApps. To sum up, we select 5283 gambling addresses as our target datasets.

[1] https://geth.ethereum.org/
[2] https://github.com/blockchain-etl/ethereum-etl
[3] https://neo4j.com/
[4] https://etherscan.io/labelcloud
[5] https://xblock.pro/#/datasets

Notably, gambling is only one kind of DApps on the blockchain. As a result, gambling addresses make up only a small fraction of Ethereum addresses. To make the classification model aware of this problem, we constructed an unbalanced dataset in which gambling addresses accounted for only about 15%. These non-gambling addresses mainly include exchanges, games, mining, etc.

As the number of graph layers increases, the number of addresses and transactions increases exponentially. Considering large computational consumption and high time cost, we sample 2-hop addresses and top-100 transactions for each address v_i separately to get subgraph ATS_{v_i}. The total number of nodes in gambling-related subgraphs is 66,637, while the total number of nodes in non-gambling-related subgraphs is 321,427.

RQ1: Model Performance Comparison

Model Selection

In this section, we evaluate the proposed **CGDetector** approach by conducting comparative experiments. We selected five mainstream machine learning algorithms(Logistic Regression, Decision Tree, Random Forest, LightGBM, XGBoost)(Olden et al., 2008) with manually extracted statistical features and four commonly-used GNN models (GraphSAGE(Hamilton et al., 2017), GCN(Kipf & Welling, 2017), GAT(Veličković et al., 2018), and GIN(Xu et al., 2019)) for comparison.

For the model training, we normalize the feature matrix and use the ImbalancedSampler function to oversample for unbalanced datasets. We divided the training set, verification set, and test set in a ratio of 7:2:1. We adopt the method of variance filtering for feature selection to retain features that are found to be greater than zero, covering more feature dimensions and information. For all GNN-based methods, we set the embedding dimension, learning rate, and dropout as 128, 0.001, and 0.2, respectively. All other traditional machine learning methods are initialized with the same parameters suggested by their official codes and have been carefully fine-tuned. For all methods, we report the average results of 10 independent runs.

Model Performance Evaluation

We compare our CGDetector with traditional machine learning baseline methods to evaluate its effectiveness in identifying gambling accounts. A detailed description of the experimental results are reported in Table 4 below. For each column in Table 4, Recall, F1-score, AUC-PR, and AUC-ROC are respectively demonstrated from left to right. We observe that experiments demonstrate that GNN models achieve state-of-the-art results in classification tasks compared to machine learning models.

Specifically, our CGDetector approach outperforms traditional machine learning methods, yielding relative improvement over best baselines in terms of Recall, AUC-PR, AUC-ROC. For GNN models, GIN shows significant advantages in terms of Recall and AUC-PR, which reach up to 89.48% and 47.62% respectively. GraphSAGE demonstrates performance advantage in terms of F1-score and AUC-ROC, which reach up to 59.45% and 91.55% respectively. To sum up, GIN shows the best comprehensive performance.

Type	Models	Recall	F1-score	AUC-PR	AUC-ROC
Traditional machine learning models	Logistic Regression	50.00	49.63	7.49	77.86
	Decision Tree	**62.18**	**60.05**	23.89	61.59
	Random Forest	54.34	57.52	**37.36**	90.53
	LightGBM	54.82	57.74	32.14	91.06
	XGBoost	55.41	56.32	36.31	**91.33**
CGDetector (ours)	GraphSAGE	70.30±0.02	**59.45±0.07**	11.98±0.46	**91.55±0.34**
	GCN	60.25±0.85	58.26±1.22	13.36±0.13	87.70±0.08
	GAT	62.61±0.04	58.37±0.06	7.95±0.03	72.58±0.02
	GIN	**89.48±0.89**	56.89±1.42	**47.62±1.61**	91.46±0.92

Table 4. Experimental Results of Machine Learning VS GNNs (%)

These findings are consistent with our points above. Traditional machine learning methods with manual feature engineering cannot learn related features in an end-to-end manner and relies heavily on classifier selection. GNN models can learn graph topologies and hidden features from both address and edge level, reasonably achieving better performance on crypto gambling detection. This suggests that our **CGDetector** approach is able to effectively identify crypto addresses involved in gambling.

To allow for more intuitive and clear understandings, we also visualized the comparison results in Fig.3 below. The figure on the left shows the performance of traditional machine learning models based on manually extracted statistical features, while the figure on the right shows the performance of Graph Neural Network models of our proposed research approach.

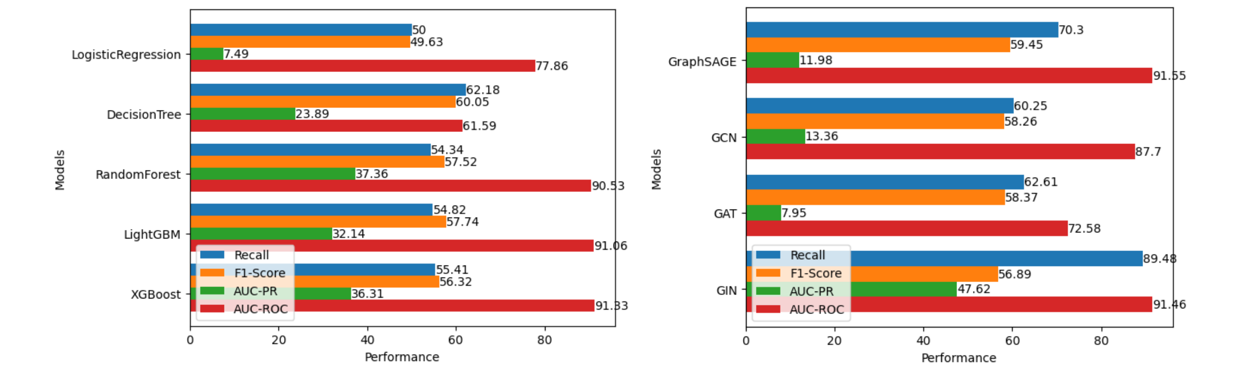


Figure 3. Performance Visualization of Machine Learning VS GNNs

RQ2: Sensitivity Analysis

In this study, we analyze the impacts of the graph neural network (GNN)-based embedding layer on the model's performance. Specifically, we evaluate the effects of several key factors, including the number of convolution layers, the embedding size, and the learning rate. After comprehensive analysis, we select the GIN model as our research object due to its superior performance in our proposed approach.

(a) Effect of Layer Numbers.

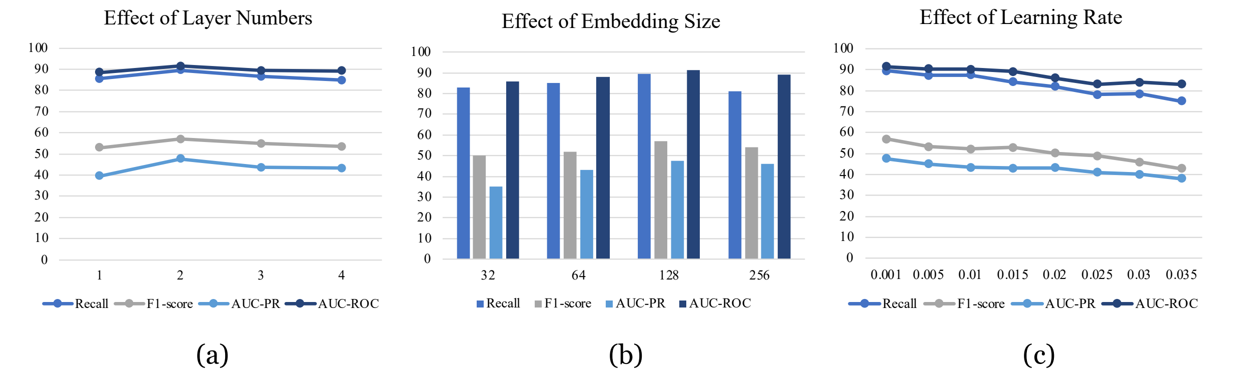
We investigate the impact of GNN layer depth on model performance by varying the number of layers between 1 and 4. Results are presented in Figure 4(a), showcasing the trends for four evaluation metrics across different GIN layer depths. We observe that increasing the depth of GIN leads to improved model performance, with peak performance at two layers. However, excessively increasing the number of convolutional layers results in slightly reduced overall performance. This is because high-order neighbors lose their individuality due to the small world property, leading to a weakened performance(Watts & Strogatz, 1998).

(b) Effect of Embedding Size.

The dimension of the embedding vector is another parameter that affects graph representational capacity. An excessively high dimension tends to lead to overfitting and increases computational complexity, while a low dimension reduces graph representation performance. Therefore, we investigate the impact of embedding vector dimension on classification performance by varying the size from approximately 32 to 256 and presenting the results in Figure 4(b). Our analysis shows that the performance first increases and then slowly decreases as the embedding vector dimension increases.

(c) Effect of Learning Rate.

We explore the effect of the learning rate on our model's performance. Figure 4(c) presents the test performance for different learning rates, indicating a general downward trend across metrics.



(a)

(b)

(c)

Figure 4. Impact of hyper parameters of sensitivity analysis

Discussion and Implication

Theoretical implications

In our work, we propose a novel model for detecting phishing scam accounts on Ethereum, which provides new ideas and methods on transaction analysis for future research. Specifically, we propose a detection model named CGDetector, which combines manual feature engineering and transaction records analysis with graph neural networks. This unique structure enables our model to efficiently obtain implicit relationship features and outperform traditional machine learning methods. We believe that this contribution can provide valuable insights for researchers investigating gambling account detection on blockchain-based systems and inspire similar efforts in related research tasks, such as malicious account detection.

Practical implications

Our approach provides valuable guidance for regulators and law enforcement in identifying crypto addresses associated with gambling behaviors. With our novel detection method, they can more efficiently evaluate suspicious gambling accounts across all EVM-compatible blockchain systems, potentially mitigating economic losses. Additionally, this approach can help regulators to take action against illegal transactions and scams related to gambling activities on the blockchain, which is typically considered challenging due to the complex and anonymous nature of blockchain technology.

Conclusion and Future Work

With the help of blockchain and smart contracts, crypto gambling has a high risk of fraud and illegal transaction activities due to the lack of regulation. As a result, identifying crypto addresses with gambling behaviors has emerged as a significant research topic, providing valuable enlightenment in the field of cryptocurrency crime detection and blockchain security regulation. In this work, we propose a novel detection approach based on Graph Neural Networks named CGDetector, consisting of Graph Construction, Subgraph Extractor, Statistical Feature Extraction, and Gambling Address Classification. Extensive experiments for Ethereum are implemented to demonstrate that our proposed approach outperforms state-of-the-art address classifiers of traditional machine learning methods.

In the future, we will focus more on refining and deepening the identification of crypto gambling behaviors to provide additional guidance for law enforcement organizations. To achieve this, we have outlined several key objectives that we plan to pursue. Firstly, we aim to accumulate more crypto gambling transaction datasets and entity label datasets to further enhance our understanding of these behaviors. Specifically, we intend to collect transaction data from Tron and EOS chains, which will allow us to verify the generalizability and universality of our approach. Secondly, we plan to investigate deeper and temporal transaction characteristics to improve our ability to identify crypto gambling behaviors. Lastly, we will work on improving our graph neural network models to enable efficient detection of crypto gambling addresses in the context of imbalanced dataset sampling scenarios.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (NSFC) under grant nos. 72171115 and 71701091.

References

Bartoletti, M., Pes, B., & Serusi, S. (2018). Data mining for detecting bitcoin ponzi schemes. *2018 Crypto Valley Conference on Blockchain Technology (CVCBT)*, 75–84.

Böhme, R., Christin, N., Edelman, B., & Moore, T. (2015). Bitcoin: Economics, technology, and governance. *Journal of Economic Perspectives*, *29*(2), 213–238.

Brito, J., Shadab, H., & Castillo, A. (2014). Bitcoin financial regulation: Securities, derivatives, prediction markets, and gambling. *Colum. Sci. & Tech. L. Rev.*, *16*, 144.

Brown, S. H. V. (2021). Gambling on the Blockchain: How the Unlawful Internet Gambling Enforcement Act Has Opened the Door for Offshore Crypto Casinos. *Vand. J. Ent. & Tech. L.*, *24*, 535.

Brünjes, L., & Gabbay, M. J. (2020). UTxO- vs Account-Based Smart Contract Blockchain Programming Paradigms. In T. Margaria & B. Steffen (Eds.), *Leveraging Applications of Formal Methods, Verification and Validation: Applications* (pp. 73–88). Springer International Publishing. https://doi.org/10.1007/978-3-030-61467-6_6

Cai, W., Wang, Z., Ernst, J. B., Hong, Z., Feng, C., & Leung, V. C. (2018). Decentralized applications: The blockchain-empowered software system. *IEEE Access*, *6*, 53019–53033.

Chen, W., Guo, X., Chen, Z., Zheng, Z., & Lu, Y. (2020). Phishing Scam Detection on Ethereum: Towards Financial Security for Blockchain Ecosystem. *IJCAI*, *7*, 4456–4462.

Delfabbro, P., King, D., Williams, J., & Georgiou, N. (2021). Cryptocurrency trading, gambling and problem gambling. *Addictive Behaviors*, *122*, 107021. https://doi.org/10.1016/j.addbeh.2021.107021

Farrugia, S., Ellul, J., & Azzopardi, G. (2020). Detection of illicit accounts over the Ethereum blockchain. *Expert Systems with Applications*, *150*, 113318.

Foley, S., Karlsen, J. R., & Putniņš, T. J. (2019). Sex, drugs, and bitcoin: How much illegal activity is financed through cryptocurrencies? *The Review of Financial Studies*, *32*(5), 1798–1853.

Goutte, C., & Gaussier, E. (2005). A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation. In D. E. Losada & J. M. Fernández-Luna (Eds.), *Advances in Information Retrieval* (pp. 345–359). Springer. https://doi.org/10.1007/978-3-540-31865-1_25

Hamilton, W., Ying, Z., & Leskovec, J. (2017). Inductive Representation Learning on Large Graphs. *Advances in Neural Information Processing Systems*, *30*. https://proceedings.neurips.cc/paper/2017/hash/5dd9db5e033da9c6fb5ba83c7a7ebea9-Abstract.html

Ibrahim, R. F., Elian, A. M., & Ababneh, M. (2021). Illicit account detection in the ethereum blockchain using machine learning. *2021 International Conference on Information Technology (ICIT)*, 488–493.

Khan, S. A., & Ali Rana, Z. (2019). Evaluating Performance of Software Defect Prediction Models Using Area Under Precision-Recall Curve (AUC-PR). *2019 2nd International Conference on Advancements in Computational Sciences (ICACS)*, 1–6. https://doi.org/10.23919/ICACS.2019.8689135

Kipf, T. N., & Welling, M. (2017). *Semi-Supervised Classification with Graph Convolutional Networks* (arXiv:1609.02907). arXiv. https://doi.org/10.48550/arXiv.1609.02907

Kozen, D. C. (1992). Depth-First and Breadth-First Search. In D. C. Kozen (Ed.), *The Design and Analysis of Algorithms* (pp. 19–24). Springer. https://doi.org/10.1007/978-1-4612-4400-4_4

Li, Y., Cai, Y., Tian, H., Xue, G., & Zheng, Z. (2020). Identifying illicit addresses in bitcoin network. *Blockchain and Trustworthy Systems: Second International Conference, BlockSys 2020, Dali, China, August 6–7, 2020, Revised Selected Papers 2*, 99–111.

Liang, J., Li, L., Chen, W., & Zeng, D. (2019). Targeted addresses identification for bitcoin with network representation learning. *2019 IEEE International Conference on Intelligence and Security Informatics (ISI)*, 158–160.

Liu, J., Zheng, J., Wu, J., & Zheng, Z. (2022). FA-GNN: Filter and Augment Graph Neural Networks for Account Classification in Ethereum. *IEEE Transactions on Network Science and Engineering*, *9*(4), Article 4. https://doi.org/10.1109/TNSE.2022.3166655

Liu, X., Tang, Z., Li, P., Guo, S., Fan, X., & Zhang, J. (2022). A Graph Learning Based Approach or Identity Inference in DApp Platform Blockchain. *Ieee Transactions on Emerging Topics in Computing*, *10*(1), Article 1. https://doi.org/10.1109/TETC.2020.3027309

Michalski, R., Dziuba\ltowska, D., & Macek, P. (2020). Revealing the character of nodes in a blockchain with supervised learning. *Ieee Access*, *8*, 109639–109647.

Min, T., & Cai, W. (2022). Portrait of decentralized application users: An overview based on large-scale Ethereum data. *CCF Transactions on Pervasive Computing and Interaction*, *4*(2), 124–141. https://doi.org/10.1007/s42486-022-00094-6

Monamo, P. M., Marivate, V., & Twala, B. (2016). A Multifaceted Approach to Bitcoin Fraud Detection: Global and Local Outliers. *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 188–194. https://doi.org/10.1109/ICMLA.2016.0039

Monamo, P., Marivate, V., & Twala, B. (2016). Unsupervised learning for robust Bitcoin fraud detection. *2016 Information Security for South Africa (ISSA)*, 129–134. https://doi.org/10.1109/ISSA.2016.7802939

Olden, J. D., Lawler, J. J., & Poff, N. L. (2008). Machine Learning Methods Without Tears: A Primer for Ecologists. *The Quarterly Review of Biology*, *83*(2), 171–193. https://doi.org/10.1086/587826

Ostapowicz, M., & Żbikowski, K. (2019). Detecting fraudulent accounts on blockchain: A supervised approach. *Web Information Systems Engineering–WISE 2019: 20th International Conference, Hong Kong, China, January 19–22, 2020, Proceedings 20*, 18–31.

Poursafaei, F., Hamad, G. B., & Zilic, Z. (2020). Detecting malicious Ethereum entities via application of machine learning classification. *2020 2nd Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS)*, 120–127.

Shao, W., Li, H., Chen, M., Jia, C., Liu, C., & Wang, Z. (2018). Identifying Bitcoin Users Using Deep Neural Network. In J. Vaidya & J. Li (Eds.), *Algorithms and Architectures for Parallel Processing* (pp. 178–192). Springer International Publishing. https://doi.org/10.1007/978-3-030-05063-4_15

Sockin, M., & Xiong, W. (2023). A Model of Cryptocurrencies. *Management Science*. https://doi.org/10.1287/mnsc.2023.4756

Steinmetz, F., & Fiedler, I. (2019). A State-Operated Blockchain-Based System for the Transparent Processing of Online Gambling Payments in Germany. *Gaming Law Review*, *23*(10), 715–725.

Suratkar, S., Shirole, M., & Bhirud, S. (2020). Cryptocurrency wallet: A review. *2020 4th International Conference on Computer, Communication and Signal Processing (ICCCSP)*, 1–7.

Tian, H., Li, Y., Cai, Y., Shi, X., & Zheng, Z. (2021). Attention-Based Graph Neural Network for Identifying Illicit Bitcoin Addresses. In H.-N. Dai, X. Liu, D. X. Luo, J. Xiao, & X. Chen (Eds.), *Blockchain and Trustworthy Systems* (pp. 147–162). Springer. https://doi.org/10.1007/978-981-16-7993-3_11

Toyoda, K., Mathiopoulos, P. T., & Ohtsuki, T. (2019). A novel methodology for hyip operators' bitcoin addresses identification. *IEEE Access*, *7*, 74835–74848.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., & Bengio, Y. (2018). *Graph Attention Networks* (arXiv:1710.10903). arXiv. https://doi.org/10.48550/arXiv.1710.10903

Wang, S., Yuan, Y., Wang, X., Li, J., Qin, R., & Wang, F.-Y. (2018). An Overview of Smart Contract: Architecture, Applications, and Future Trends. *2018 IEEE Intelligent Vehicles Symposium (IV)*, 108–113. https://doi.org/10.1109/IVS.2018.8500488

Watts, D. J., & Strogatz, S. H. (1998). Collective dynamics of 'small-world'networks. *Nature*, *393*(6684), 440–442.

Wen, T., Xiao, Y., Wang, A., & Wang, H. (2023). A novel hybrid feature fusion model for detecting phishing scam on Ethereum using deep neural network. *Expert Systems with Applications*, *211*, 118463.

Wood, G. (2014). Ethereum: A secure decentralised generalised transaction ledger. *Ethereum Project Yellow Paper*, *151*(2014), 1–32.

Wu, J., Liu, J., Zhao, Y., & Zheng, Z. (2021). Analysis of cryptocurrency transactions from a network perspective: An overview. *Journal of Network and Computer Applications*, *190*, 103139. https://doi.org/10.1016/j.jnca.2021.103139

Wu, K., Ma, Y., Huang, G., & Liu, X. (2021). A first look at blockchain-based decentralized applications. *Software: Practice and Experience*, *51*(10), 2033–2050.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Philip, S. Y. (2020). A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, *32*(1), 4–24.

Xu, K., Hu, W., Leskovec, J., & Jegelka, S. (2019). *How Powerful are Graph Neural Networks?* (arXiv:1810.00826). arXiv. https://doi.org/10.48550/arXiv.1810.00826

Yli-Huumo, J., Ko, D., Choi, S., Park, S., & Smolander, K. (2016). Where Is Current Research on Blockchain Technology?—A Systematic Review. *PLOS ONE*, *11*(10), e0163477. https://doi.org/10.1371/journal.pone.0163477

Zhang, J., & Luo, Y. (2017). *Degree Centrality, Betweenness Centrality, and Closeness Centrality in Social Network*. 300–303. https://doi.org/10.2991/msam-17.2017.68

Zheng, Z., Xie, S., Dai, H.-N., Chen, X., & Wang, H. (2017). Blockchain challenges and opportunities: A survey. *International Journal of Web and Grid Services*, *14*(4).

Zhou, J., Hu, C., Chi, J., Wu, J., Shen, M., & Xuan, Q. (2022). Behavior-aware Account De-anonymization on Ethereum Interaction Graph. *IEEE Transactions on Information Forensics and Security*, *17*, 3433–3448. https://doi.org/10.1109/TIFS.2022.3208471

Zou, W., Lo, D., Kochhar, P. S., Le, X.-B. D., Xia, X., Feng, Y., Chen, Z., & Xu, B. (2019). Smart contract development: Challenges and opportunities. *IEEE Transactions on Software Engineering*, *47*(10), 2084–2106.