SIGTRAN: *Signature Vectors for Detecting Illicit Activities in Blockchain Transaction Networks*

Farimah Poursafaei[1,3], Reihaneh Rabbany[2,3], Zeljko Zilic[1]
farimah.poursafaei@mila.quebec, rrabba@cs.mcgill.ca, and zeljko.zilic@mcgill.ca

[1] ECE, McGill University, Montreal, Canada
[2] CSC, McGill University, Montreal, Canada
[3] Mila, Montreal, Canada

Abstract. Cryptocurrency networks have evolved into multi-billion-dollar havens for a variety of disputable financial activities, including phishing, ponzi schemes, money-laundering, and ransomware. In this paper, we propose an *efficient* graph-based method, SIGTRAN, for detecting illicit nodes on blockchain networks. SIGTRAN first generates a graph based on the transaction records from blockchain. It then represents the nodes based on their structural and transactional characteristics. These node representations *accurately* differentiate nodes involved in illicit activities. SIGTRAN is *generic* and can be applied to records extracted from different networks. SIGTRAN achieves an F_1 score of 0.92 on Bitcoin and 0.94 on Ethereum, which outperforms the state-of-the-art performance on these benchmarks obtained by much more complex, platform-dependent models.

1 Introduction

Blockchain-based cryptocurrencies, such as Bitcoin and Ethereum, have taken a considerable share of financial market [11]. Malicious users increasingly exploit these platforms to undermine legal control or conduct illicit activities [5,11,4,17]. In particular, billions of dollars attained through illegal activities such as drug smuggling and human trafficking are laundered smoothly through blockchain-based cryptocurrencies exploiting their pseudonymity [21]. Given the openness and transparency of blockchain [19], it is of paramount importance to mine this data for detecting such illicit activities.

Although the recent machine learning advances have enhanced the exploration of large-scale complex networks, the performance of these methods relies on the quality of the data representations and extracted features [14]. Likewise, in blockchain networks with high anonymity and large number of participants with diverse transactional patterns, any illicit node detection method is effective only when the extracted characteristics of the data efficiently distinguish the illicit and licit components of the network. Hence, developing effective illicit node detection method depends heavily on the efficiency of the data representations and extracted features. Considering that network topologies can reflect the roles of the different nodes, graph representation learning methods have been potentially conceived as great means for capturing neighborhood similarities and community detection [22]. Additionally, machine learning analysis on large networks is becoming viable due to the efficiency, scalability and ease of use of graph representation learning methods [22,15].

Driven by the need to enhance the security of the blockchain through transaction network analysis, and by recent advances in graph representation learning, we propose an efficient graph-based method SIGTRAN for detecting illicit nodes in the transaction network of blockchain-based cryptocurrencies. SIGTRAN can be applied for warning honest parties against transferring assets to illicit nodes. In addition to providing a trading ledger for cryptocurrencies, blockchain can be perceived as a network that analyzing its dynamic properties enhances our understanding of the interactions within the network [26,9]. SIGTRAN first constructs a graph based on the extracted transactions from the blockchain ledger considering integral characteristics of the blockchain network. Then it extracts structural, transactional and higher-order features for graph nodes; these features strengthen the ability to classify the illicit nodes. SIGTRAN shows superior performance compared to the previous platform-dependant state-of-the-arts (SOTAs), while it is generic and applicable to different blockchain network models. Particularly, SIGTRAN achieves an F_1 score of 0.92 and 0.94 for detecting illicit nodes in Bitcoin and Ethereum network, respectively. Moreover, SIGTRAN is scalable and simpler compared to much more complex SOTA contenders. In short, SIGTRAN is:

- **Generic:** SIGTRAN is platform independent and applicable to different blockchain networks, unlike current contenders.
- **Accurate:** SIGTRAN outperforms much more complex SOTA methods on the platforms they are designed for.
- **Reproducible**: we use publicly available datasets, and the code for our method and scripts to reproduce the results is available at: https://github.com/fpour/SigTran.

2 Background and Related Work

The increasingly huge amount of data being appended to the blockchain ledger makes the manual analysis of the transactions impossible. Multiple works proposed different machine learning techniques for detection of the illicit entities on cryptocurrency networks [28,12,16,21,13,31]. Specifically, [12,16,28] investigate supervised detection methods for de-anonymizing and classifying illegitimate activities on the Bitcoin network. In parallel, Farrugia et al. [13] focus on examining the transaction histories on the Ethereum aiming to detect illicit accounts through a supervised classification approach.

Lorenz et al. [21] address the problem of tracking money laundering activities on the Bitcoin network. Concentrating on the scarcity of the labeled data on a crypocurrency network such as Bitcoin, Lorenz et al. [21] argue against the unsupervised anomaly detection methods for the detection of illicit patterns on the Bitcoin transaction network. Instead, they propose an active learning solution as effective as its supervised counterpart using a relatively small labeled dataset.

Previous work often focuses on analyzing the transactions on the cryptocurrency networks with *platform-dependent* features, e.g. works in [25,27,21,13]. There are also several *task-dependent* studies investigating a specific type of illicit activity, for example [6,10] focus on ponzi schemes, which are illicit investments. Here, we propose a method that is *independent of the platform and/or task* and looks at basic structural features of the underlying transaction networks. This is inline with more recent graph based methods which we summarize below.

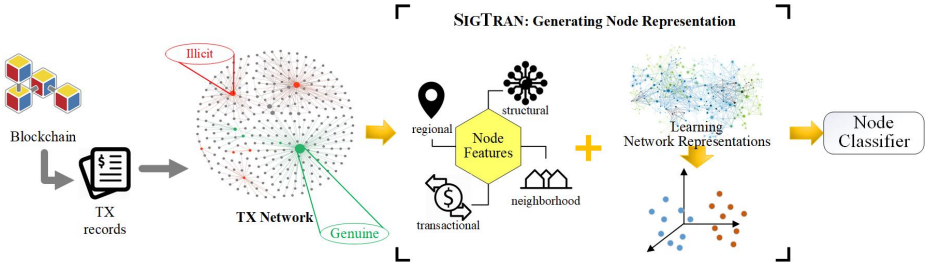


Fig. 1: Overview of SIGTRAN to detect illicit nodes on a blockchain network.

Graph-based anomaly detection approaches have emerged recently as a promising solution for analyzing the growing data of the blockchain ledger, for both Bitcoin [18], and Ethereum [29,30,20,31]. In particular, Wu et al. [30] model the transaction network of Ethereum as a multi-edge directed graph where edges represent transactions among Ethereum public addresses, and are assigned weights based on the amount of transactions and also a timestamp. They propose *trans2vec* that is based on adopting a random walk-based graph representation embedding method and is specifically designed for the Ethereum network. Weber et al. [28] model the Bitcoin transaction network as a directed acyclic graph where the nodes represent the transactions and the edges illustrate the flow of cryptocurrency among them. They construct a set of features for the transactions of the network based on the publicly available information. Then, they apply Graph Convolutional Networks to detect illicit nodes. For node features, they consider local information (e.g. transactions fee, average Bitcoin sent/received), and other aggregated features representing characteristics of the neighborhood of the nodes. In the experiment section we show that SIGTRAN is more accurate than these two platform-dependent SOTAs.

More generally, graph representation learning has became commonplace in network analysis with superior performance in a wide-range of real-world tasks. As a pioneering work, *node2vec* [15] explores the neighborhood of each node through truncated random walks which are used for achieving representations that preserve the local neighborhood of the nodes. *RiWalk* [22] mainly focuses on learning the structural node representations through coupling a role identification procedure and a network embedding step. For a more detailed survey of network embedding methods, see [14].

3 Proposed Method

Given the publicity of the transaction ledger, our aim is to detect illicit activities on a blockchain-based cryptocurrency network. We formulate the problem as a node classification task in the transaction graph. Specifically, given the transaction records of a set of blockchain nodes, we devise the transaction graph and investigate the authenticity of different nodes by predicting the probability of each being involved in an illegitimate activities such as phishing, scam, malware, etc. We propose an efficient feature vector generation approach for nodes in these networks which demonstrates node activity signatures which can be used to distinguish illicit nodes. An overview of SIGTRAN framework is illustrated in Fig. 1. SIGTRAN extracts the transaction (*TX*) history from the blockchain ledger and constructs a transaction network from those records. To gen-

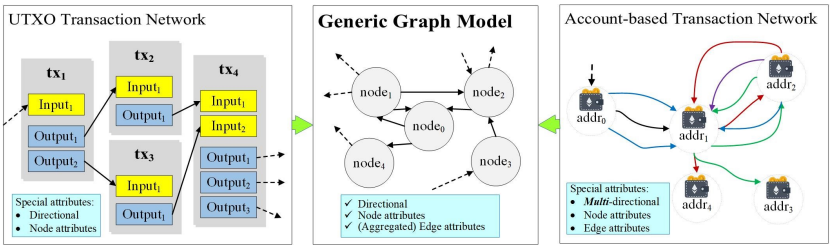


Fig. 2: SIGTRAN creates a generic graph model based on the transaction networks.

erate node representations, it then extracts a set of useful features which are fused with the corresponding node representations produced by a node embedding method. The final representations are then classified to detect illicit nodes. These steps are explained in detail in the following.

3.1 Transactions History Retrieval

The required transaction records can be obtained directly from the blockchain public ledger of the target cryptocurrency. For instance, for Bitcoin or Ethereum, we can use the client software of these peer-to-peer networks to pull down the blockchain data in binary format which is converted to human-readable formats like CSV via an appropriate parser. As an example, for converting the binary records of the Bitcoin and Ethereum ledger, *SoChain* [1] and [3] can be employed respectively. The transaction records contain details such as timestamp, amount sent or received, incoming and outgoing addresses, and other related information. Different authoritative websites (such as *EtherScamDB* [2] for Ethereum network) helps in gathering a list of illicit nodes on the blockchain network. Together transaction records and the information about the authenticity of the network nodes constitute the dataset required.

3.2 Network Construction

A cryptocurrency transactions network is modeled as a graph demonstrating the interactions among participants of the network. We model a blockhchain network as a graph $G = (V, E)$, where V represents the set of nodes and E expresses the set of edges. Nodes and edges could have extra attributes, such as labels for nodes, and amount and timestamp of transaction for edges. Essentially, blockchain networks can be classified into two categories: (a) unspent transaction output *(UTXO)* model where the nodes specify the transactions, and the edges denote the flow of the cryptocurrency among nodes. Bitcoin, Dash, Z-Cash, and Litecoin are cyrptocurrencies based on the UTXO model [26], and (b) *account-based* model, where the account addresses are considered as the nodes and the transactions among addresses as the edges of the graph. Ethereum network is based on the account-based model. Considering the different categories of blockchain networks, we construct a generic graph model, as illustrated in Fig. 2, to which the instances of both the UTXO as well as the account-based network models are easily convertible. In the generic graph, the nodes specify the network entities in which we are interested to investigate their authenticity, while the edges denote the interactions among the nodes. The generated graph model entails any features associated with the nodes, whereas multiple edges between any two nodes with the same

direction are aggregated into a single edge. It is noteworthy that based on the underlying blockchain network (i.e. UTXO or account-based), nodes and edges of the generic graph can have different intuitions. Particularly, if the graph is constructed based on an UTXO blockchain, the nodes represent cryprocurrency transactions which may belong to licit or illicit categories of real entities. However, if the graph is constructed based on an account-based blockchain, each node represents either an illicit or licit address. In both cases, node representations and classification are applied incognizant of the underlying blockchain model.

3.3 SIGTRAN

After modeling the blockchain transactions network as a graph, we need to develop proper representations for the nodes. This consists of a set of carefully crafted features which are fused with learned node representations, explained below respectively.

SIGTRAN-Feature Extraction. For each node u, we gain a diverse set of features consisting of four main categories as follows. It is important to note that the features of the nodes (e.g., labels) and edges (e.g., amount and timestamp) of the original network are preserved in the constructed generic model, since we employ these attributes for extracting the features of the nodes.

• *Structural features* consist of in-degree ($D_{in}(u) = \sum_{v \in N_u} |e_{vu}|$), out-degree ($D_{out}(u) = \sum_{v \in N_u} |e_{uv}|$), and total degree ($D_{tot}(u) = D_{in}(u) + D_{out}(u)$) of node u. As there may exist multiple edges between two nodes, $|e_{vu}|$ determines the number of edges from v to u, and N_u consists of all first-order neighbors of node u.

• *Transactional features* investigate the characteristics related to the amount and time interval of the transactions. Indeed, blockchain specific information of the transaction network is mainly enriched in this set of features. Each edge e_{uv} from u to v is associated with a set of attributes including the amount and time interval of the transactions from node u to node v. For obtaining transactional features, we consider a set of aggregation functions, G, which includes *summation*, *maximum*, *minimum*, *average*, *standard deviation*, and *entropy* operations over an arbitrary given distribution x as follows:

$$G = \{\sum(x), \max(x), \min(x), \overline{x}, \sigma(x), H(x)\} \tag{1}$$

With the set of aggregation functions G, transactional features of node u are defined as:

$$tx_u^{amnt} = \{g(e_u^a) \mid g \in G, e_u^a \subseteq \{e_{uv}^a, e_{vu}^a\}\}, tx_u^{freq} = \{g(e_u^\tau) \mid g \in G, e_u^\tau \subseteq \{e_{uv}^\tau, e_{vu}^\tau\}\}$$

where e_u^a denotes the amount related to (in/out) edges of node u. Similarly, e_u^τ denotes the time interval related to (in/out) edges of node u.

• *Regional features* are defined with regard to the ego network of a node. We consider the *egonet* of node u ($S_u = (V_u, E_u)$) as a subgraph of the original graph consisting of u and its first-order neighbors (i.e. N_u), with all the edges amongst these nodes. As an example, considering the generic graph model in Fig. 2, the egonet of $node_0$ consists of $\{node_1, node_2, node_4\}$. Having the definition of the egonet in mind, we consider the number of edges of S_u as one of the regional features of node u. Besides, the in-degree, out-degree, and total degree of S_u are considered as the other regional features according to $D_{in}(S_u) = |\{e_{wv} \in E \mid w \notin V_u, v \in V_u\}|$, $D_{out}(S_u) = |\{e_{wv} \in E \mid w \in V_u, v \notin V_u\}|$, and $D_{tot}(S_u) = D_{in}(S_u) + D_{out}(S_u)$, where $V_u = u \cup N_u$.

• *Neighborhood features* analyze the aggregated characteristics of neighbors of node u. Considering the aggregation functions in (1), the neighborhood features of node u are defined as: $D_{in}(N_u) = \{g(D_{in}(v)) \mid g \in G, v \in N_u\}$, $D_{out}(N_u) = \{g(D_{out}(v)) \mid g \in G, v \in N_u\}$, and $D_{tot}(N_u) = \{g(D_{tot}(v)) \mid g \in G, v \in N_u\}$.

Network Representation Learning. In order to learn node representations which fuse topological perspectives of the nodes in a cryptocurrency transaction network, SIGTRAN combines the extracted features explained in above (which are obtained focusing on the specific characteristics of the cryptocurrency networks such as amount and time interval of transactions) with the node representations that are learned automatically through a network embedding procedure. For retrieving more efficient node representations, we exploit a common network embedding method for learning the features of the nodes in the generic graph model. Then, we fuse the extracted features with the node embeddings in an effective manner so that the ultimate node representations effectively demonstrate the fundamental characteristics of the nodes. For fusing the extracted features and the node embeddigns, we investigate two approaches explained in the following subsections.

RiWalk-enhanced. In this approach, we focus on the fact that nodes with different functionalities have different roles in a network, and the structure of the network can be investigated for gleaning these roles [22]. Hence, we consider the SIGTRAN-features as powerful indicators of similarity among nodes, and decouple the node embedding procedure into two steps. First, we identify the top ten SIGTRAN-features with the highest importance in detecting the illicit nodes and retrieve the values of those features for each node u as $\mathbf{f}_u^*$. We then relabel each neighbor of node u such as v according to the function $\phi(v) = h(\mathbf{f}_u^*) \oplus h(\mathbf{f}_v^*) \oplus d_{uv}$. Here, d_{uv} denotes the shortest path length from u to v, $\oplus$ is the concatenation operation, and $h(x)$ is defined as $h(x) = \lfloor log_2(x+1) \rfloor$. The new labels which are generated based on the node features roughly indicates the role of the nodes (thus, *Ri: Role identification*). Thereafter, the second step consists of a random-walk-based network embedding method for learning the node representations. Specifically, we generate several random walks starting from each node, then merge the random walks to construct a corpus and adopt the Skip-Gram model with negative sampling of *word2vec* [24] to learn the node representations.

SIGTRAN. In this approach, we consider the fusion of the SIGTRAN-features and automatically generated node embeddings through a concatenation procedure. Particularly, we apply a random-walk-based node embedding method such as node2vec [15] and for each node u obtain its embedding as $\mathbf{e_u^*}$. Then, we generate the final representations by concatenating the SIGTRAN-features $\mathbf{f_u^*}$ with the node embeddings for each node (i.e., $\mathbf{e_u^*} \oplus \mathbf{f_u^*}$) intending to achieve accurate node representations.

3.4 Node Classification

The generated node representations can then be used in the downstream task for classification of the illicit and genuine nodes. The illicit node detection task is akin to the common task of fraud detection and anti-money laundering applications. We simply employ *Logistic Regression* for the classification task because of its widespread adoption in similar tasks as well as its high interpretability [28,8,7,25]. This simple choice enables us to better compare the effect of different embedding techniques.

Table 1: Statistics of the investigated Blockchain-based Cryptocurrency Networks.

Dataset	Nodes	Edges	Illicit Nodes
Bitcoin	203,769	234,355	4,545
Ethereum	2,973,489	13,551,303	1,165

4 Experiments

This section evaluates SIGTRAN experimentally. We present the datasets in Section 4.1, baseline methods in Section 4.2, and discuss the results in Section 4.3.

4.1 Dataset Description

We investigated two real-world transaction datasets consisting of the most widely adopted cryptocurrencies: (a) Bitcoin blockchain network which is the largest cryptocurrency system based on UTXO model, and (b) Ethereum that support smart contracts, holds the second largest cryptocurrency, and provides an account-based model.

We employed Bitcoin transactions dataset shared by Weber et al. [28] in which 21% of the transactions are labeled as *licit* (corresponding to different legitimate categories such as exchanges, miners, wallet provider, etc.), 2% as *illicit* (corresponding to different categories of illegitimate activities such as scams, malware, ponzi scheme, ransomeware, etc.), and there are no labels for the rest of the transactions. In addition to the transaction records, the Bitcoin dataset consists of a set of handcrafted features representing the characteristics of the considered transactions. Since the dataset is fully anonymized, we could only generate *structural*, *regional*, and *neighborhood* features for the nodes of the Bitcoin graph. We combined SIGTRAN-features with the initial attributes available in the dataset to form the node features of the Bitcoin network. In addition, we investigated the Ethereum transactions data shared by Wu et al. [30]. This dataset consists of Ethereum transaction records for a set of addresses consisting of licit addresses as well as illicit addresses reported to be involved in phishing and scam activities. The statistical details of the Bitcoin and Ethereum dataset are shown in Table 1.

4.2 Baseline Methods

Several SOTA methods were evaluated and compared.

- *node2vec* [15] is a random-walk-based node representation method which employs biased random walks to explore the neighborhood of the nodes with the consideration of local and global network similarities. Default parameters of the node2vec are set in line with the typical values mentioned in the paper [15]: context size $k = 10$, embedding size $d = 64$, walk length $l = 5$, and number of walk per node $r = 20$. We have also considered setting $p = 0.25$ and $q = 4$ to better exploit the structural equivalency of the nodes according to the discussion in the paper [15].
- *RiWalk* [22] is another random-walk-based node embedding methods which focuses on learning structural node representations through decoupling the role identification and the network embedding procedures [22]. We considered the *RiWalk-WL* which aims to imitate the neighborhood aggregation notion of the Weisfeiler-Lehman graph kernels, and captures fine-grained connection similarity patterns.

Table 2: **Bitcoin performance:** SIGTRAN significantly outperforms baselines on the Bitcoin dataset. The last three rows are introduced in this paper.

	Training Set Performance					*Test Set Performance*				
Algorithm	Precision	Recall	F_1	Accuracy	AUC	Precision	Recall	F_1	Accuracy	AUC
Weber et al. [28]	0.910	0.933	0.921	0.921	0.980	0.901	0.929	0.915	0.913	0.976
node2vec	0.626	0.300	0.405	0.560	0.580	0.627	0.312	0.415	0.563	0.580
RiWalk	0.556	0.348	0.426	0.535	0.556	0.549	0.343	0.421	0.530	0.547
RiWalk-enhanced	0.584	0.478	0.518	0.573	0.620	0.582	0.486	0.522	0.573	0.619
SIGTRAN-Features	0.911	0.936	0.924	0.923	0.980	**0.905**	0.926	0.915	0.914	0.976
SIGTRAN	0.898	0.940	0.919	0.917	0.978	0.890	**0.947**	**0.918**	**0.915**	**0.976**

Table 3: **Ethereum performance**: SIGTRAN significantly outperforms baselines on the Ethereum dataset. The last three rows are introduced in this paper.

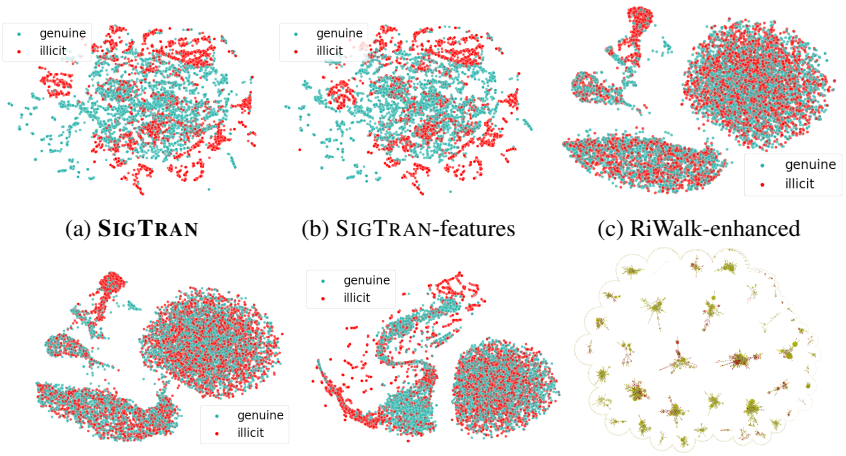
	Training Set Performance					*Test Set Performance*				
Algorithm	Precision	Recall	F_1	Accuracy	AUC	Precision	Recall	F_1	Accuracy	AUC
trans2vec [30]	0.912	0.909	0.910	0.911	0.966	0.919	0.894	0.906	0.908	0.967
node2vec	0.908	0.935	0.921	0.920	0.970	0.917	0.907	0.912	0.912	0.964
RiWalk	0.922	0.772	0.840	0.852	0.904	0.931	0.764	0.838	0.853	0.894
RiWalk-enhanced	0.921	0.846	0.882	0.887	0.908	0.928	0.832	0.877	0.884	0.899
SIGTRAN-Features	0.926	0.945	0.935	0.934	0.962	0.923	0.926	0.925	0.925	0.958
SIGTRAN	0.946	0.953	0.949	0.949	0.983	**0.944**	**0.940**	**0.942**	**0.942**	**0.976**

We also compared the performance of SIGTRAN with methods specifically designed for Bitcoin or Ethereum network.

- Bitcoin: we considered the method proposed by *Weber et al. [28]* as the baseline.
- Ethereum: we considered phishing scams detection method by Wu et al. [30] denoted as trans2vec as the baseline method. To make a fair comparison, we set the default parameters of trans2vec inline with the parameters of the node2vec.

4.3 Performance Evaluation

To evaluate the performance of SIGTRAN, we considered the illicit nodes as the target of the detection approach and randomly selected an equal number of genuine nodes to form our set of anchor nodes. We extracted the first-order neighbors of all the anchor nodes and all edges among these nodes to construct a subgraph for investigation. Random selection of genuine nodes was repeated for 50 times, thus 50 different subgraphs were examined and the average performance was reported. Logistic regression with *L1* regularization was implemented in Scikit-learn Python package as the node classifier. The performance evaluation results for the Bitcoin and Ethereum network are illustrated in Table 2 and Table 3, respectively. To investigate the importance of SIGTRAN-*features*, both tables also report the performance of the classification tasks when only SIGTRAN-features were used as the node representations.

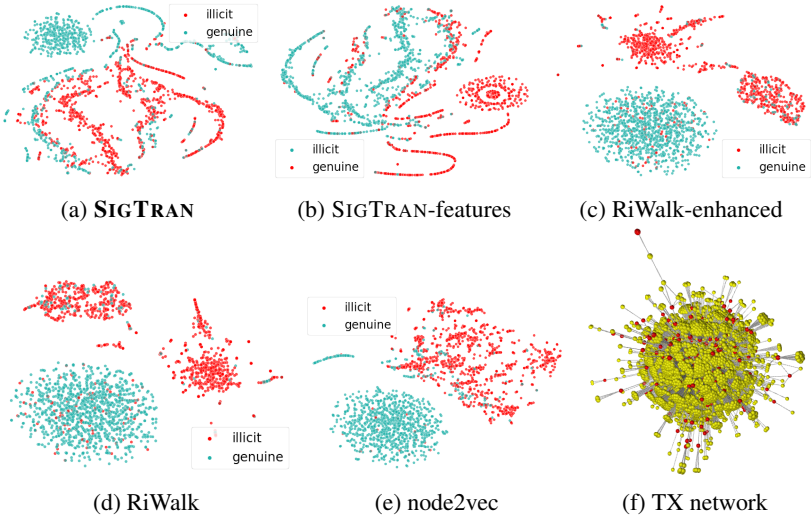


(a) **SIGTRAN** (b) SIGTRAN-features (c) RiWalk-enhanced

(d) RiWalk (e) node2vec (f) TX network

Fig. 3: **Bitcoin Embeddings:** SIGTRAN better separate illicit (red) and genuine (blue) transactions in Bitcoin network (plotted in (f)) compared to other baselines.

Bitcoin. Considering the results of illicit node detection on Bitcoin network in Table 2, it can be observed that node embedding methods namely node2vec and RiWalk did not generate efficient node representations. Therefore, the classification task had very low performance in detecting illicit nodes. The poor performance of node2vec and RiWalk is due to the fact that these methods are not specifically dealing with the intrinsic characteristics of financial networks, such as having multiple edges among nodes, or being dependent on the amount and time interval of the transactions. These methods mainly focus on exploiting the structural similarities in order to maximize the likelihood of preserving neighborhoods of nodes. However, the results demonstrate that ignoring the specific characteristics of cryptocurrency networks, such as amount and timestamp of the transactions, results in embeddings that are not efficient for achieving decent illicit node classification performance. On the other hand, methods likes Weber et al. [28] and SIGTRAN that are designed specifically for cryptocurreny networks show much better performance. Superior performance of SIGTRAN compared to Weber et al. [28] is due to its extended set of features as well as the exploitation of the structural information via node embedding methods. It is noteworthy to mention that SIGTRAN is more efficient than the proposed RiWalk-enhanced method. This can be attributed to two main reasons. First, in RiWalk-enhanced, we employed the extracted features only for relabeling the nodes. Although the labels of the nodes impact the node embeddings, the exact values of the extracted features do not directly influence the embeddings values which are later used for the node classification task. Moreover, it should be noted that the new labels combine the extracted features of the anchor and neighbor nodes as well as their shortest path distance. Thus, modified values of the extracted features are used for labeling. However, it is noteworthy that RiWalk-enhanced outperforms its counterpart RiWalk, which underlines the importance of fusing the extracted features with the node embeddings in terms of improving the performance of the node classifi-



(a) **SIGTRAN** (b) SIGTRAN-features (c) RiWalk-enhanced

(d) RiWalk (e) node2vec (f) TX network

Fig. 4: **Ethereum Embeddings:** SIGTRAN better separate illicit (red) and genuine (blue) accounts in Ethereum network (plotted in (f)) compared to other baselines. Notice the red nodes mixed in the blue cluster in (c-e).

cation task. For a qualitative comparison of the different embedding methods, we have depicted the t-SNE [23] transformations of different node representations methods for one of the subgraphs of the Bitcoin network in Fig. 3. According to Fig. 3, it can be observed that the embeddings produced by SIGTRAN shape more separable distributions.

Ethereum. For the Ethereum dataset as shown in Table 3, it can be observed that SIGTRAN demonstrates considerably better performance than the other methods. Although trans2vec and node2vec demonstrate high performance, the superior performance of SIGTRAN underlines its efficiency in employing the native characteristics of the cryptocurrency networks as well as structural information obtained by the node embedding methods. Besides, we can observe that the extracted features improved the performance of the RiWalk-enhanced compared to RiWalk. Due to the fact that SIGTRAN better incorporates the extracted features with the network structural embeddings, it achieves the most decent performance on the Ethereum network as well. We have also depicted t-SNE [23] transformations of different node embedding methods for a subgraph of the Ethereum network in Fig. 4. Considering Fig. 4, it is observable that embeddings obtained by SIGTRAN show considerable distinction between illicit and licit nodes, while for example in Fig. 4e, there are several illicit nodes (marked with red) in the licit cluster (marked with blue).

5 Conclusions

We propose SIGTRAN that extracts signature vectors for detecting illicit activities in blockchain network. Our proposed SIGTRAN transforms the blockchain network into

a simple graph and then extracts carefully designed features which explain structural, transactional, regional, and neighborhood features of the nodes. These features are then combined with generic node representations which encode the roles of the nodes in a given graph. SIGTRAN should be considered as a simple and strong baseline when developing more complex models. Our proposed SIGTRAN baseline is:

- **Accurate**: SIGTRAN outperforms state-of-the-art alternatives in detecting illicit activities in blockchain transaction records.
- **Generic**: SIGTRAN is platform independent and we apply it to blockchain data extracted from both Bitcoin and Ethereum.

Reproducibility: the code and data are available at https://github.com/fpour/SigTran.

References

1. Bitcoin block explorer and api. `https://sochain.com/`, (Accessed on 09/15/2020) 4
2. Ethereum scam database. `https://etherscamdb.info/scams`, (Accessed on 05/14/2020) 4
3. Github - blockchain-etl/ethereum-etl: Python scripts for etl (extract, transform and load) jobs for ethereum blocks, transactions, erc20 / erc721 tokens, transfers, receipts, logs, contracts, internal transactions. data is available in google bigquery https://goo.gl/oy5bcq. `https://github.com/blockchain-etl/ethereum-etl`, (Accessed on 09/15/2020) 4
4. Atzei, N., Bartoletti, M., Cimoli, T.: A survey of attacks on ethereum smart contracts (sok). In: Principles of Security and Trust, pp. 164–186. Springer (2017) 1
5. Badawi, E., Jourdan, G.V.: Cryptocurrencies emerging threats and defensive mechanisms: A systematic literature review. IEEE Access (2020) 1
6. Bartoletti, M., Pes, B., Serusi, S.: Data mining for detecting bitcoin ponzi schemes. In: 2018 Crypto Valley Conference on Blockchain Technology (CVCBT). pp. 75–84. IEEE (2018) 2
7. Bhattacharyya, S., Jha, S., Tharakunnel, K., Westland, J.C.: Data mining for credit card fraud: A comparative study. Decision Support Systems **50**(3), 602–613 (2011) 6
8. Carneiro, N., Figueira, G., Costa, M.: A data mining based system for credit-card fraud detection in e-tail. Decision Support Systems **95**, 91–101 (2017) 6
9. Chen, T., Zhu, Y., Li, Z., Chen, J., Li, X., Luo, X., Lin, X., Zhange, X.: Understanding ethereum via graph analysis. In: IEEE INFOCOM 2018-IEEE Conference on Computer Communications. pp. 1484–1492. IEEE (2018) 2
10. Chen, W., Zheng, Z., Ngai, E.C.H., Zheng, P., Zhou, Y.: Exploiting blockchain data to detect smart ponzi schemes on ethereum. IEEE Access **7**, 37575–37586 (2019) 2
11. Conti, M., Kumar, E.S., Lal, C., Ruj, S.: A survey on security and privacy issues of bitcoin. IEEE Communications Surveys & Tutorials **20**(4), 3416–3452 (2018) 1
12. Dey, S.: Securing majority-attack in blockchain using machine learning and algorithmic game theory: A proof of work. arXiv preprint arXiv:1806.05477 (2018) 2
13. Farrugia, S., Ellul, J., Azzopardi, G.: Detection of illicit accounts over the ethereum blockchain. Expert Systems with Applications **150**, 113318 (2020) 2
14. Goyal, P., Ferrara, E.: Graph embedding techniques, applications, and performance: A survey. Knowledge-Based Systems **151**, 78–94 (2018) 1, 3
15. Grover, A., Leskovec, J.: node2vec: Scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 855–864 (2016) 1, 3, 6, 7

16. Harlev, M.A., Sun Yin, H., Langenheldt, K.C., Mukkamala, R., Vatrapu, R.: Breaking bad: De-anonymising entity types on the bitcoin blockchain using supervised machine learning. In: Proceedings of the 51st Hawaii International Conference on System Sciences (2018) 2
17. Howell, B.E., Potgieter, P.H.: Industry self-regulation of cryptocurrency exchanges (2019) 1
18. Hu, Y., Seneviratne, S., Thilakarathna, K., Fukuda, K., Seneviratne, A.: Characterizing and detecting money laundering activities on the bitcoin network. arXiv preprint arXiv:1912.12060 (2019) 3
19. Huang, H., Kong, W., Zhou, S., Zheng, Z., Guo, S.: A survey of state-of-the-art on blockchains: Theories, modelings, and tools. arXiv preprint arXiv:2007.03520 (2020) 1
20. Lin, D., Wu, J., Yuan, Q., Zheng, Z.: Modeling and understanding ethereum transaction records via a complex network approach. IEEE Transactions on Circuits and Systems II: Express Briefs (2020) 3
21. Lorenz, J., Silva, M.I., Aparício, D., Ascensão, J.T., Bizarro, P.: Machine learning methods to detect money laundering in the bitcoin blockchain in the presence of label scarcity. arXiv preprint arXiv:2005.14635 (2020) 1, 2
22. Ma, X., Qin, G., Qiu, Z., Zheng, M., Wang, Z.: Riwalk: Fast structural node embedding via role identification. In: 2019 IEEE International Conference on Data Mining (ICDM). pp. 478–487. IEEE (2019) 1, 3, 6, 7
23. Maaten, L.v.d., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(Nov), 2579–2605 (2008) 10
24. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. Advances in neural information processing systems **26**, 3111–3119 (2013) 6
25. Monamo, P.M., Marivate, V., Twala, B.: A multifaceted approach to bitcoin fraud detection: Global and local outliers. In: Machine Learning and Applications (ICMLA), 2016 15th IEEE International Conference on. pp. 188–194. IEEE (2016) 2, 6
26. Motamed, A.P., Bahrak, B.: Quantitative analysis of cryptocurrencies transaction graph. Applied Network Science **4**(1), 1–21 (2019) 2, 4
27. Pham, T., Lee, S.: Anomaly detection in the bitcoin system-a network perspective. arXiv preprint arXiv:1611.03942 (2016) 2
28. Weber, M., Domeniconi, G., Chen, J., Weidele, D.K.I., Bellei, C., Robinson, T., Leiserson, C.E.: Anti-money laundering in bitcoin: Experimenting with graph convolutional networks for financial forensics. arXiv preprint arXiv:1908.02591 (2019) 2, 3, 6, 7, 8, 9
29. Wu, J., Lin, D., Zheng, Z., Yuan, Q.: T-edge: temporal weighted multidigraph embedding for ethereum transaction network analysis. arXiv preprint arXiv:1905.08038 (2019) 3
30. Wu, J., Yuan, Q., Lin, D., You, W., Chen, W., Chen, C., Zheng, Z.: Who are the phishers? phishing scam detection on ethereum via network embedding. IEEE Transactions on Systems, Man, and Cybernetics: Systems (2020) 3, 7, 8
31. Yuan, Z., Yuan, Q., Wu, J.: Phishing detection on ethereum via learning representation of transaction subgraphs. In: International Conference on Blockchain and Trustworthy Systems. pp. 178–191. Springer (2020) 2, 3