

# Characterizing and Detecting Money Laundering Activities on the Bitcoin Network

Yining Hu<sup>1,2</sup>[0000-0002-6233-743X], Suranga Seneviratne<sup>3</sup>[0000-0002-5485-5595],  
Kanchana Thilakarathna<sup>3</sup>[0000-0003-4332-0082], Kensuke  
Fukuda<sup>4</sup>[0000-0001-8372-2807], and Aruna Seneviratne<sup>2</sup>[0000-0001-6894-7987]

<sup>1</sup> Data61-CSIRO, Sydney, Australia

Yining.Hu@data61.csiro.au

<sup>2</sup> University of New South Wales, Sydney, Australia

a.seneviratne@unsw.edu.au

<sup>3</sup> University of Sydney, Sydney, Australia

{firstname.lastname}@sydney.edu.au

<sup>4</sup> National Institute of Informatics, Tokyo, Japan

kensuke@nii.ac.jp

**Abstract.** Bitcoin is by far the most popular crypto-currency solution enabling peer-to-peer payments. Despite some studies highlighting the network does not provide full anonymity, it is still being heavily used for a wide variety of dubious financial activities such as money laundering, ponzi schemes, and ransom-ware payments. In this paper, we explore the landscape of potential money laundering activities occurring across the Bitcoin network. Using data collected over three years, we create transaction graphs and provide an in-depth analysis on various graph characteristics to differentiate money laundering transactions from regular transactions. We found that the main difference between laundering and regular transactions lies in their output values and neighbourhood information. Then, we propose and evaluate a set of classifiers based on four types of graph features: immediate neighbours, curated features, deepwalk embeddings, and node2vec embeddings to classify money laundering and regular transactions. Results show that the node2vec-based classifier outperforms other classifiers in binary classification reaching an average accuracy of 92.29% and an F1-measure of 0.93 and high robustness over a 2.5-year time span. Finally, we demonstrate how effective our classifiers are in discovering unknown laundering services. The classifier performance dropped compared to binary classification, however, the prediction can be improved with simple ensemble techniques for some services.

**Keywords:** Bitcoin, Transaction Graph, Money Laundering

## 1 Introduction

The first successful peer-to-peer (P2P) financial system, Bitcoin [30], has evolved over the years into a complex financial ecosystem with a large number of users,

different transaction types, and various support services. As of now, Bitcoin has the highest market share of 50% among crypto-currencies [6] and performs over 300k transactions per day [7].

Driven by the pseudonymity provided by the network, many cyber-criminals and hackers have started using Bitcoin for illegal activities. For example, *Silk Road* [12], an online market place for illegal goods and services, accepted bitcoins as their payment method to hide the identities of the sellers and buyers. More recently, ransom-ware attacks *WannaCry* and *Petaya* [14] also accepted Bitcoin payments from victimized computer owners to unlock their machines. Other activities that involve the misuse of Bitcoin include money laundering via mixing to hide the origin of illegally obtained money [29,16], and ponzi scheme, a pyramid scheme that pays old users with investments from new users [17].

In this paper, we investigate money laundering activities, one of the main misuses of Bitcoin [29]. While there are public websites such as Blockchain Info [7], BitcoinWhosWho [5], WalletExplorer.com [13] that collect de-anonymized Bitcoin services and tag those involved in money laundering, new money laundering services emerge frequently due to the unregulated and P2P nature of Bitcoin. Although several efforts have been made to understand and detect laundering activities [17,16,39], existing studies have not looked at the graph properties of laundering and regular transactions in detail, they also have not fully explored the potential of automatically created node embeddings using techniques such as deepwalk [32] and node2vec [24]. In this paper, we first explore money laundering transactions on the Bitcoin network from a graph theoretic perspective and compare their characteristics to regular transactions. We find that although some manually extracted statistical and network features follow different distributions for laundering and regular transactions, they are not sufficiently effective in detecting laundering transactions. We show that random-walk based graph representation learning algorithms—deepwalk [32] and node2vec [24] significantly outperform manually created features for this task. This paper makes the following contributions:

- With Bitcoin data collected over three years, we characterize graph properties of money laundering transactions and highlight their differences in comparison to regular transactions.
- We show that laundering transactions are distinguishable from regular transactions in several statistical and network features including in-degree/out-degree ratio, sum/mean/standard deviation of output values, and number of weakly connected components—the size of the subgraph a transaction belongs to. Nonetheless, these metrics are not effective for the binary classification of money laundering and regular transactions.
- We show a node2vec-based classifier achieves the highest performance in classifying laundering and regular transactions. We also show the robustness of the classifier by applying it to randomly selected weeks across a large timescale of two and a half years and show that results remain consistent.
- Finally, we demonstrate the performance of our classifiers in detecting unknown money laundering transactions. The classifier performance decreases

compared to the binary classification, but can be improved with simple ensemble techniques.

The remainder of the paper is organized as follows. Section 2 discusses related work that motivated our research. Section 3 presents details of our data collection, ground truth labelling and how we created the Bitcoin transaction graphs. Section 4 characterizes the properties of money laundering transactions in comparison to regular transactions. Section 5 presents the classifier design and classification results, followed by Section 6 which presents the new money laundering service discovery results. Section 7 concludes the paper.

## 2 Related Work

**Bitcoin Network Characterization** Ron and Shamir [37] found a significant variance in the distribution of various Bitcoin addresses, accumulated balance and number of transactions per user providing empirical evidence that a limited number of Bitcoin entities control the majority of addresses, transactions and bitcoins. Lischke et al. [27] conducted a study on the Bitcoin transaction graph during its first four years. Authors observed the distribution of several graph metrics such as in-degree, out-degree and clustering coefficient of the entire transaction graph. They also analyzed the economic and network aspects of multiple major Bitcoin businesses and markets including *SatoshiDice* and *Mt.Gox*. Similar studies include [31,38]. In this paper, we analyse the graph characteristics of potential laundering and regular transactions separately with the objective of coming up with a transaction classifier.

**Address De-anonymization** Multiple studies explored the possibility of de-anonymizing Bitcoin addresses [36,28,20]. Reid et al. [36] found external information, such as user registration details and voluntary disclosure of public-keys, can be used to link Bitcoin addresses to real-life users. Meiklejohn et al. [28] proposed two address clustering heuristics to aggregate Bitcoin transactions: i) addresses associated with the input UTXOs of a transaction belonging to the same user and ii) the change address that is created when the sum of input UTXOs exceeds the amount to pay also belongs to the sender. Using these heuristics, a number of services and their Bitcoin addresses have been identified in the literature [16,17] and online forums [4,13,5,7]. We establish our ground truth labelling based on these identified services and their addresses.

**Service Analysis and Anomaly Detection** Driven by its pseudonymity, there also exists a number of dubious and potentially illegal services in the Bitcoin network. Moser et al. [29] found services such as *Bitcoin Fog*, that hide transaction origins by withholding multiple small inputs and bundling them into a smaller number of larger outputs. Ferrin et al. [21] later discovered a common pattern of transaction mixing which is to form a “mixing cloud” that contains multiple interconnected “joint transactions”, which result from layering multiple transactions into a single larger transaction.

More recently, machine learning methods are being applied to detect potentially illegal Bitcoin activities. Early studies often relied on the metadata and the temporal features of addresses or transactions. Pham et al. [33] performed k-means clustering on both the transaction graph and user graph and applied outlier detection to find suspicious transactions and users, however they were only able to detect one out of 30 known cases of theft. Bartoletti et al. [17] applied several supervised learning techniques to detect Bitcoin addresses associated with ponzi schemes using temporal features such as total address lifetime and active days. Weber et al. [39] recently presented a study on detecting Bitcoin laundering transactions using network features and node embeddings. The authors created 49 independent graphs over a total period of 2 weeks, using the first 34 graphs for training and the rest for testing and achieved an F1-measure of over 0.7 with Graph Convolutional Networks (GCN) and EvolveGCN. They also published a dataset containing the extracted network features, an anatomized edge list and labels [8]. However, no information was provided on the ground truth labelling process as well as the exact features the authors used, making it hard to apply the solution in other datasets. While our work is complementary to this study, we analyse data in a much larger time span of over three years, evaluate quantitatively different types of metadata and graph features, and address more realistic and challenging scenarios of operating on larger graphs and discovering new money laundering instances.

### 3 Data

We next describe how the transaction graphs are built, our data collection process, and the methodology of establishing the ground truth.

#### 3.1 Bitcoin Transaction Graph

A *Bitcoin transaction*, identified by a unique id, comprises a list of input and output *unspent transaction outputs*, or *UTXOs*. UTXOs are indivisible chunks of bitcoins attached to specific owners. A Bitcoin transaction consumes UTXOs by unlocking them with the sender’s signature, and creates new UTXOs designated to recipients. This is how bitcoins are transferred among users, i.e., by creating chains of transactions as shown in Figure 1. Each node represents a transaction and a directed edge between two nodes exists if an output UTXO of a transaction becomes an input UTXO of a succeeding transaction. For example, an edge is created from transaction  $Tx3$  to transaction  $Tx4$  as UTXO8 serves as an output for  $Tx3$  and an input for  $Tx4$ . We utilize these transaction chains to create transaction graphs. We did not explore user (address) graphs in this paper, as new addresses can be easily created and manipulated.

#### 3.2 Data Collection and Labelling

**Data Collection** We ran a Bitcoin Core client under the version bitcoin-0.15.0 [3] to collect block data, and parsed the block data with a simple parser [2]

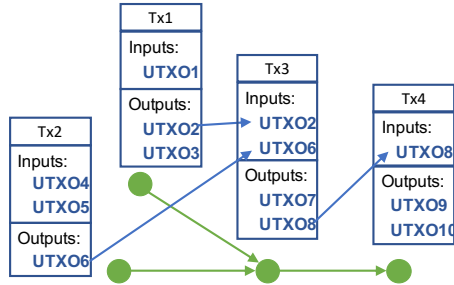


Fig. 1: A simple Bitcoin transaction chain.

to obtain transaction information. For each transaction, we extracted its timestamp, previous transactions (cf. Part 3.1), number and value of input and output UTXOs. Our dataset was collected between 07/2014 and 05/2017.

**Ground Truth Labelling** We identify several laundering and regular services and their addresses. We consider all transactions associated with these addresses as our labelled ground truth.

*Money laundering* Laundering services disguise the origin of bitcoins by mixing different users' transactions, and many of them are potentially malicious [29]. Our selection of *money laundering services* is based on existing literature [29,16], news articles and address tags from trusted online resources, e.g., WalletExplorer.com [13], a website that tracks Bitcoin wallets and aggregates relevant addresses. In total we found 4 major laundering services with more than 22,000 transactions each. Our selection include *AlphaBay* [10], *BTC-e* [1], *Bitmixer* [9] and *HelixMixer* [16]. Figure 2 shows the accumulated number of daily transactions during the active periods of the four laundering services, as well as a combined sum across all these services. For example, *BTC-e* was active between 08/14-05/17, and generated nearly six million transactions in total.

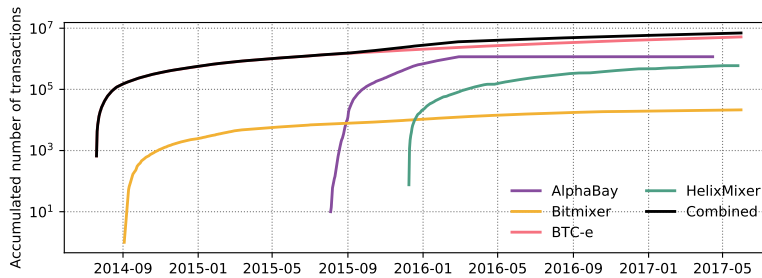


Fig. 2: Service active period and accumulated number of transactions.

*Regular* The exchange sector has the highest number of operating entities, as most new users begin trading on Bitcoin via exchanges [23]. This is followed by wallet services with 5.8 – 11.5 million active users [35]. Payment services such as *CoinPayment* and *Bitpay* are also often linked to banking institutions and existing payment networks. As the lines between different sectors have become increasingly blurred [35], when labelling *regular services*, we randomly selected 9 active, large-scale exchange, payment and wallet services from the top wallet list on WalletExplorer.com [13], with reference to existing studies [34].

We label all transactions related to regular services as *regular* and those associated with laundering services as *laundering*, similar to the ground truth labelling in [17]. To ensure the correctness of labelling, we have eliminated all transactions with conflicting labels [20]. Table 1 summarizes our data collection with the total number of laundering and regular transactions, active periods of laundering services, as well as the percentage of labelled laundering and regular transactions among all existing transactions.

Table 1: Data collection.

Service	Number of Tx.s	Active period	Percentage
AlphaBay	1,168,382	08/15-04/17	0.97%
Bitmixer	22,122	09/14-06/17	0.01%
BTC-e	5,665,400	08/14-06/17	2.93%
HelixMixer	605,991	12/15-08/16	0.56%
Laundering	7,461,895	-	4.29%
Regular	37,907,769	-	22.98%

## 4 Graph Characterization

We next provide a characterization of Bitcoin transaction graphs, with a focus on the difference between laundering and regular transactions.

### 4.1 Graph Evolution

We first show the growth of Bitcoin network using directed daily transaction graphs created between 07/2014-05/2017. We calculated the number of nodes, number of edges, and graph density for each graph. Figure 3 shows the evolution of these 3 metrics over the approximate 3-year time span. Both number of nodes and number of edges continued to increase, and started to saturate in early 2017. The decrease in graph density suggests transactions have fewer inputs and outputs on average in more recent years. This could be due to the growing number of wallet or payment services that run specific algorithms to construct transactions. The periodic fluctuations can be ascribed to the difficulty adjustment embedded in the Bitcoin Proof-of-Work (PoW) protocol [15].

As the total hashrate changes, to maintain the constant 10 minute block time, Bitcoin adjusts PoW difficulty every 2016 blocks, which approximately was 2 weeks in 2016 [11].

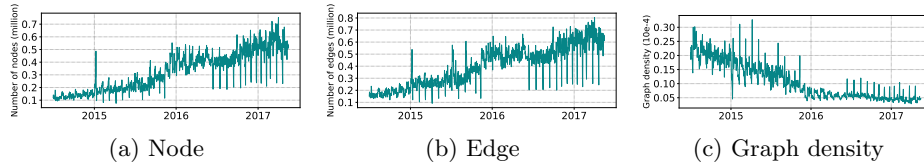


Fig. 3: Evolution of daily transaction graphs.

## 4.2 Feature Characterization

We then look at transaction values, calculated as the sum of all output UTXOs. Figure 4 shows the daily average value of laundering and regular transactions. On average, laundering transactions carry 38.8 bitcoins per day, while regular transactions only have 29.1 bitcoins per day. Laundering transactions also carry higher values than regular ones 77.3% of the time.

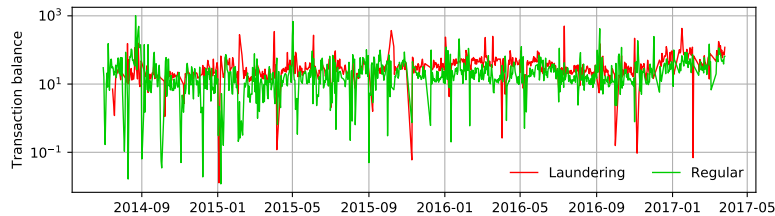


Fig. 4: Average daily transaction value by service type.

For each directed daily transaction graph, we calculate basic features such as number of input UTXOs (or in-degree), number of output UTXOs (or out-degree), sum/mean/standard deviation (std) of outputs, and network features such as centrality measures, PageRank, and clustering co-efficient [19]. We extracted a total of 14 features for each transaction. We performed a feature importance analysis using an *Adaptive Boost (Adaboost)* [22] classifier with a Decision Tree based estimator of maximum depth 5. We selected 5 most discriminating features based on the importance score, among which the in-degree/out-degree ratio provides information on edges; the sum/mean/std of outputs relate to UTXO values; and the weakly connected components relate to a transaction’s neighbourhood.

Figure 5 shows the distribution of these 5 features of laundering and regular transactions separately. Compared to regular transactions, a larger portion of laundering transactions have high in-degree/out-degree ratio. The resulting distribution confirms laundering services operate by bundling small transactions from various users and forming new ones, which is also discussed in Section 2. The CDF curves for sum, mean and std of output UTXOs of laundering transactions are slightly steeper than those of regular transactions. This indicates laundering transactions create similar number and value of output UTXOs, which also confirms the findings in [29]. Laundering transactions also have a slightly smaller number of weakly connected components than regular transactions. This is resulted from the scales of different services. As services tend to send and receive transactions using addresses associated to them, directly connected transactions are more likely to belong to a same service. We leverage this observation later to design the baseline classifier in Section 5.

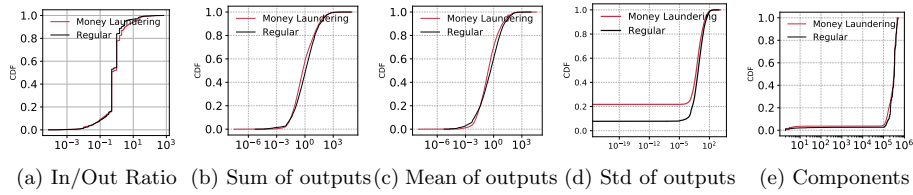


Fig. 5: Feature distribution for money laundering and regular transactions.

To further understand the service-wise differences, we provide a more detailed comparison in Figure 6. Although there are some differences among laundering transactions related to different services, these transactions are individually distinguishable from regular ones in the selected features. Among the 4 laundering services studied, *BTC-e* shows the most similar feature distributions as regular transactions. *Bitmixer*, on the other hand, behaves similarly to *HelixMixer*.

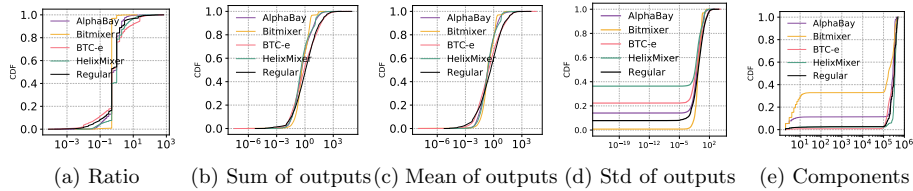


Fig. 6: Service-wise feature distribution.



## 5 Detecting Money Laundering Activities

We next build multiple classifiers to detect laundering transactions using the partially labelled dataset described in Section 3. We train our classifiers with four different features—nodes’ immediate neighbours, curated features, deepwalk embeddings and node2vec embeddings. We used accuracy—the total percentage of regular and laundering transactions that are correctly identified, and F1-measure—the harmonic mean of precision and recall—as evaluation metrics.

### 5.1 Transaction Classifiers

The design details of our four classifiers are discussed below.

**Immediate Neighbour-Based Classifier** Since transactions from a same service tend to be more connected (cf. Section 4.2), as a first step, we classify unlabelled transactions based on their nearest neighbours. This classifier works under the following criteria: 1) a node with more immediate regular neighbours than laundering neighbours is classified regular, 2) a node with more or equal number of immediate laundering than regular neighbours is classified laundering, and 3) a node with no regular or laundering neighbours is classified regular—this criterion is required due to the unlabelled nodes in the graph.

**Curated Features** We trained an Adaboost classifier with a Decision Tree base-estimator using the 14 statistical and network features discussed in Section 4. Hyper-parameter tuning results when varying the maximum depth and number of estimators are presented in Appendix A.

**Deepwalk** Graph representation learning via random walks has become a common technique to analyse graph structures in recent years [26]. These algorithms can automatically create feature vectors for graph nodes in an unsupervised manner, and achieve better scalability and performance than manually extracted features in various scenarios. A widely used algorithm—*deepwalk* [32]—leverages random walks of specified *number of walks per node* and *walk length* to uniformly sample a node’s neighbourhood.

**Node2vec** A more recent technique, *node2vec* [24], uses 2 more parameters—the return parameter  $p$  and the in-out parameter  $q$ —to more precisely guide the walks. When  $p$  is high ( $> \max(q, 1)$ ), the same nodes will not be revisited in the next 2 steps [24], and when  $p$  is low ( $< \max(q, 1)$ ), the walks will remain close to the starting node. When  $q$  and  $p$  both equal to 1, node2vec creates random walks in a uniform manner, similar to deepwalk.

Deepwalk and node2vec are strictly transductive [25], and can only predict unlabelled nodes on the same graph, hence we created transaction graphs that cover the entire duration of training and testing, and only considered nodes on the giant component [18]. We also used undirected graphs to better explore the

graph structure. For both binary classification and new instance prediction in Section 6, we started random-walks from all labelled nodes in the training set and all labelled and unlabelled nodes in the testing set to ensure no pre-assumed knowledge on the testing nodes. We used 100 (*number of walks per node*), 100 (*walk length*), 25 (*embedding dimension*) for both deepwalk and node2vec, and 2 (*p*), and 0.5 (*q*) for node2vec. We fed the embedding vectors into an Adaboost classifier with a Decision Tree base classifier (*maximum depth* = 5, *number of estimators* = 40) for binary classification. Details on the selection of random-walk parameters and walk starting nodes are provided in Appendix A.

## 5.2 Classifier Performance Comparison

We set the experiment duration to 1 week with the first 5 days for training and last 2 days for testing. To compare the performance of the above 4 classifiers, we selected 5 consecutive weeks in late 2014, and present the results averaged over these weeks below. In addition, we also applied two simple ensemble techniques to the prediction results using *curated features* and *node2vec embeddings*. In “OR” ensemble, we label a transaction as laundering if either classifier labels it as laundering. While in “AND” ensemble, we only label a transaction as laundering if both classifiers do so.

Table 2 shows the node2vec-based classifier with proper parameter settings outperforms the other three classifiers achieving an average accuracy of 92.05% and an average F1-measure of 0.94. Information from a transaction’s immediate neighbours is not sufficient in detecting laundering transactions. Although some differences can be observed in the feature distribution (cf. Section 4), manually extracted statistical and network features cannot effectively differentiate laundering and regular transactions. Calculating network features on these large transaction graphs can also be very time-consuming. Both deepwalk and node2vec performed well in binary classification, while node2vec produced slightly better results than deepwalk. “OR” ensemble improved the node2vec classifier performance as it helps to capture more laundering nodes. “AND” ensemble on the other hand drastically diminished the performance of both classifiers, as more laundering transactions are wrongly detected as regular.

Table 2: Classifier Performance.

<b>Classifier</b>	<b>Accuracy (%)</b>	<b>F1-measure</b>
Neighbourhood	28.46	0.09
Manually extracted features	65.34	0.45
Deepwalk	91.72	0.94
Node2vec	92.05	0.94
“OR” ensemble	92.74	0.95
“AND” ensemble	21.47	0.02

### 5.3 Over-time Classification Results With Node2vec

We then applied the node2vec-based classifier with proper parameters on one random week per month between 08/2014-01/2017. Figure 7 illustrates the classifier performance over the entire experimentation period. To ensure training reliability, we have removed scenarios with less than 150 laundering samples in training. The classifier performance was robust and achieved an average accuracy of 92.29% and F1-measure of 0.93 across the selected weeks.

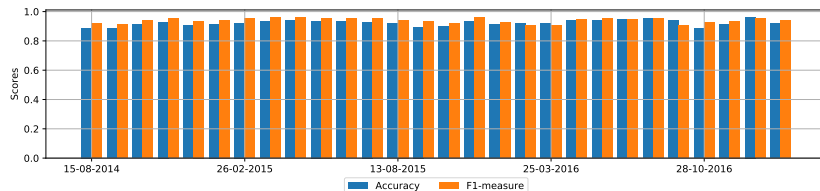


Fig. 7: Node2vec classifier performance over time.

## 6 Predicting New Money Laundering Instances

We next apply a curated feature-based classifier and a node2vec-based classifier with corresponding parameters to detect previously unseen laundering transactions. We selected *AlphaBay*, *Bitmixer* and *HelixMixer* for comparison. For every 2 or 3 months during a service’s active period, we randomly selected one week to leave the target service out and train the model using the remaining services. Then, we evaluated the model’s ability to discover the left-out service. Figure 8 compares the classifier performance and prediction ensemble for the 3 selected services, averaged over their respective experiment periods.

Compared to the classification results in Section 5, the classifier performance decreases when predicting instances from new services. Nonetheless, the node2vec classifier was able to achieve 95.2% accuracy in detecting *HelixMixer* with a F1-measure of 0.3. As observed in Figure 6, there are differences in several features among these services. When applying node2vec, the random walks cannot effectively explore the neighbourhoods of transactions belonging to a different service. The class imbalance also becomes more significant when splitting training and testing sets by service, resulting in more malicious nodes being classified as regular. “OR” ensemble improves the F1-measure, as it tends to capture more malicious transactions. “AND” ensemble on the other hand improves the accuracy as more regular transactions are correctly identified.

In the case of *Bitmixer*, there are on average 66 laundering transactions and 24,428 regular transactions in the test set of each week. This significant class imbalance causes an extremely low F1-measure, which is not improved even with the proposed ensemble techniques. As for *AlphaBay* and *HelixMixer*, which have

discriminating features compared to regular transactions, the F1-measure using the curated feature-based classifier increases as more laundering transactions can be easily distinguished from regular ones. Both services also have sufficient samples, with an average money laundering to regular ratio being 10,485/18,727 and 2,287/24,842 respectively. When applying node2vec, their transactions are more often covered by random walks and are hence more correctly identified.

Since the prediction results are also affected by the number of testing samples, it can be inferred that laundering services need to grow to a certain extent for our classifiers to be effective. Services with a small number of transactions are hard to predict, and hence they can operate unnoticed and last longer than services who create a higher transaction volume.

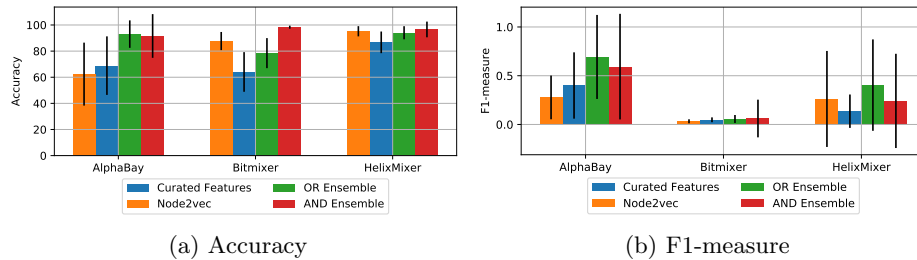


Fig. 8: Comparison of classifier performance for different services.

## 7 Concluding Remarks

We characterized Bitcoin transaction graphs and studied patterns of money laundering and regular transactions on data collected over three years. Our study found that money laundering transactions tend to have a higher in-degree/out-degree ratio, more uniform sum, mean and std of outputs, and a slightly smaller number of weakly connected components compared to regular transactions. We also show that *BTC-e* behaves similarly to regular services, while *Bitmixer* and *HelixMixer* have the most similar behaviors among the 4 selected laundering services. The preliminary classification and prediction results using an Adaboost classifier proved we can differentiate money laundering from regular transactions, and predict new instances using node2vec embeddings. Results also showed the classifier performance can be improved with ensemble techniques. Future work will mainly focus on the two aspects below.

*Ground Truth Labelling* We operated on a fraction ( $\approx 27\%$ ) of Bitcoin transactions belonging to a limited number of identified services. A larger fraction of transactions remain unlabelled. Finding sufficient volumes of reliably labelled

data for potentially illegal activities such as money laundering is always challenging. As we used data labelled by volunteers, there exist conflicting labels for certain transactions. In this paper, we limit ourselves to the most reliable tags based on information from existing literature and trusted websites. In the future, semi-supervised approaches such as *label propagation* [41] and *label spreading* [40] can be leveraged to further improve the performance.

*Transductive Graph Embeddings* As mentioned in Section 5, node2vec is strictly transductive and can only be used to predict unknown nodes on the same graph. One requirement for analysing temporal graphs is the possibility of making predictions to previously unseen nodes. Recent developments based on graph convolutions such as GraphSage [25] which allow embeddings learned from one graph to be used in prediction on a completely different graph can be explored. Nonetheless, their success on very large and dynamic graphs remains to be seen.

## A Parameter Tuning

### A.1 Curated Features

We varied the maximum depth of the base estimator from 5 to 50 and number of estimators from 10 to 80 as illustrated in Figure 9. The classifier performance remains low and does not vary significantly with maximum depth and number of estimators. This confirms manually-extracted statistical and network features are not effective in distinguishing laundering and regular transactions.

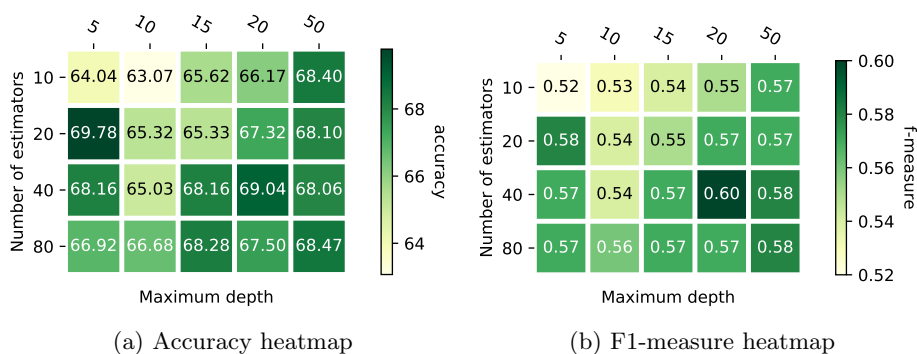


Fig. 9: Adaboost classifier parameter tuning.

### A.2 Node2vec/Deepwalk

*Hyper-parameter Tuning* Results in Table 3 show that a higher number of walks per node and longer walk path produce better performance. Also, for node2vec it is better to set  $q$  to a relatively low value and  $p$  higher than  $\max(q, 1)$  to ensure outwards exploration from transaction nodes, as it is important to understand how UTXOs are propagated via transactions for different services.

*Effect of Random-Walk Starting Nodes* Further, to understand the effect of random-walk starting nodes, we compared the classifier performance when varying the volume of labelled and unlabelled test nodes to start random-walks from and present the results in Figure 10. The random-walks are started from all labelled training nodes,  $x\%$  ( $x \in \{100, 80\}$ ) of all labelled testing nodes,  $y\%$  ( $y \in \{100, 80, 50, 20, 0\}$ ) of all unlabelled testing nodes. Results show that to achieve high classifier performance, random-walks should be started from as many labelled testing nodes as possible. The volume of unlabelled random-walk starting nodes does not significantly affect the classifier performance.

Table 3: Node2vec/Deepwalk hyper-parameter tuning.

# of walks per node	Walk length	$p$	$q$	Accuracy (%)	F1-measure
50	50	1	1	90.20	0.93
50	50	2	0.5	90.48	0.93
50	50	4	0.5	89.72	0.92
50	50	0.5	2	88.35	0.91
50	50	0.5	4	86.85	0.90
100	100	1	1	89.84	0.92
100	100	2	0.5	91.01	0.93
100	100	4	0.5	89.62	0.92

\* The graph has 634,739 nodes and 1,147,979 edges.

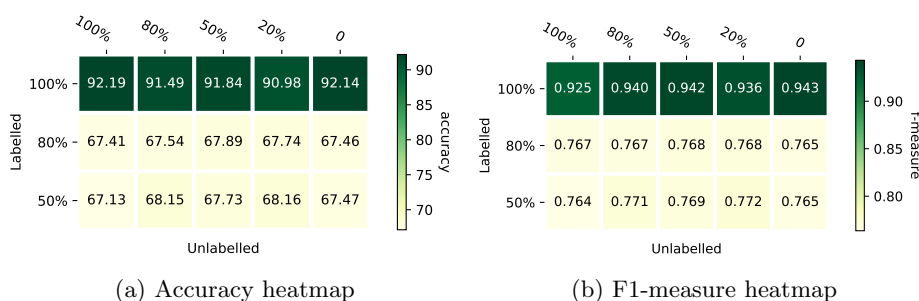


Fig. 10: Effect of random-walk starting nodes on classifier performance.

## References

1. Alleged btc-e operator alexander vinnik seeks extradition to russia. <https://www.coindesk.com/alleged-btc-e-operator-alexander-vinnik-seeks-extradition-to-russia>, accessed: 2019-10-14
2. bitcoin-blk-file-reader. <https://github.com/mrqc/bitcoin-blk-file-reader>, accessed: 2019-04-26
3. Bitcoin core version 0.15.0 released. <https://bitcoin.org/en/release/v0.15.0>, accessed: 2019-04-26
4. Bitcoin forum. <https://bitcointalk.org>, accessed: 2019-05-04
5. Bitcoin who is who. <https://bitcoinwhoswho.com>, accessed: 2019-04-27
6. Bitcoin's share of total crypto market slips back toward 50%. <https://www.coindesk.com/bitcoins-share-of-total-crypto-market-slips-back-toward-50>, accessed: 2019-05-04
7. Blockchain. <https://www.blockchain.com>, accessed: 2019-04-27
8. Elliptic data set. <https://www.kaggle.com/ellipticco/elliptic-data-set>, accessed: 2019-10-06
9. European authorities swoop on bitcoin mixing service. <https://bravenewcoin.com/insights/european-authorities-swoop-on-bitcoin-mixing-service>, accessed: 2019-10-20

10. The golden age of dark web drug markets is over. <https://www.theverge.com/2019/2/17/18226718/alphabay-takedown-drug-marketplace-federal-arrest>, accessed: 2019-10-14
11. Proof of work. [https://en.bitcoin.it/wiki/Proof\\_of\\_work](https://en.bitcoin.it/wiki/Proof_of_work), accessed: 2019-05-10
12. Silk road (marketplace). [https://en.wikipedia.org/wiki/Silk\\_Road\\_\(marketplace\)](https://en.wikipedia.org/wiki/Silk_Road_(marketplace)), accessed: 2019-04-26
13. Wallet explorer. <https://www.walletexplorer.com>, accessed: 2019-04-27
14. Wannacry, petya, notpetya: how ransomware hit the big time in 2017. <https://www.theguardian.com/technology/2017/dec/30/wannacry-petya-notpetya-ransomware>, accessed: 2019-05-06
15. Antonopoulos, A.M.: Mastering Bitcoin: unlocking digital cryptocurrencies. ” O’Reilly Media, Inc.” (2014)
16. de Balthasar, T., Hernandez-Castro, J.: An analysis of bitcoin laundry services. In: Nordic Conference on Secure IT Systems. pp. 297–312. Springer (2017)
17. Bartoletti, M., Pes, B., Serusi, S.: Data mining for detecting bitcoin ponzi schemes. arXiv preprint arXiv:1803.00646 (2018)
18. Bollobás, B.: The evolution of random graphsthe giant component. In: Random graphs. vol. 184, pp. 130–59. Cambridge university press Cambridge (2001)
19. Bondy, J.A., Murty, U.S.R., et al.: Graph theory with applications, vol. 290. Cite-seer (1976)
20. Ermilov, D., Panov, M., Yanovich, Y.: Automatic bitcoin address clustering. In: Machine Learning and Applications (ICMLA), 2017 16th IEEE International Conference on. pp. 461–466. IEEE (2017)
21. Ferrin, D.: A preliminary field guide for bitcoin transaction patterns. In: Texas Bitcoin Conference (2015)
22. Freund, Y., Schapire, R.E., et al.: Experiments with a new boosting algorithm. In: International Conference on Machine Learning (ICML). pp. 148–156 (1996)
23. Glaser, F., Zimmermann, K., Haferkorn, M., Weber, M., Siering, M.: Bitcoin-asset or currency? revealing users’ hidden intentions. ECIS (2014)
24. 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. ACM (2016)
25. Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. In: Advances in Neural Information Processing Systems. pp. 1024–1034 (2017)
26. Hamilton, W.L., Ying, R., Leskovec, J.: Representation learning on graphs: Methods and applications. arXiv preprint arXiv:1709.05584 (2017)
27. Lischke, M., Fabian, B.: Analyzing the bitcoin network: The first four years. Future Internet **8**(1), 7 (2016)
28. Meiklejohn, S., Pomarole, M., Jordan, G., Levchenko, K., McCoy, D., Voelker, G.M., Savage, S.: A fistful of bitcoins: characterizing payments among men with no names. In: Proceedings of the 2013 conference on Internet measurement conference. pp. 127–140. ACM (2013)
29. Moser, M., Bohme, R., Breuker, D.: An inquiry into money laundering tools in the bitcoin ecosystem. In: eCrime Researchers Summit (eCRS), 2013. pp. 1–14. IEEE (2013)
30. Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system (2008)
31. Parino, F., Beiró, M.G., Gauvin, L.: Analysis of the bitcoin blockchain: socio-economic factors behind the adoption. EPJ Data Science **7**(1), 38 (2018)



32. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: Online learning of social representations. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 701–710. ACM (2014)
33. Pham, T., Lee, S.: Anomaly detection in the bitcoin system-a network perspective. arXiv preprint arXiv:1611.03942 (2016)
34. Ranshous, S., Joslyn, C.A., Kreyling, S., Nowak, K., Samatova, N.F., West, C.L., Winters, S.: Exchange pattern mining in the bitcoin transaction directed hypergraph. In: International Conference on Financial Cryptography and Data Security. pp. 248–263. Springer (2017)
35. Rauchs, M., Hileman, G., et al.: Global cryptocurrency benchmarking study. Tech. rep., Cambridge Centre for Alternative Finance, Cambridge Judge Business School (2017)
36. Reid, F., Harrigan, M.: An analysis of anonymity in the bitcoin system. In: Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third International Conference on Social Computing (SocialCom), 2011 IEEE Third International Conference on. pp. 1318–1326. IEEE (2011)
37. Ron, D., Shamir, A.: Quantitative analysis of the full bitcoin transaction graph. In: International Conference on Financial Cryptography and Data Security. pp. 6–24. Springer (2013)
38. Somin, S., Gordon, G., Altshuler, Y.: Social signals in the ethereum trading network. arXiv preprint arXiv:1805.12097 (2018)
39. 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)
40. Zhou, D., Bousquet, O., Lal, T.N., Weston, J., Schölkopf, B.: Learning with local and global consistency. In: Advances in neural information processing systems. pp. 321–328 (2004)
41. Zhu, X., Ghahramani, Z.: Learning from labeled and unlabeled data with label propagation. Tech. rep., CMU-CALD-02-107, Carnegie Mellon University (2002)