Resurrecting Address Clustering in Bitcoin

Malte Möser and Arvind Narayanan

Princeton University
{mmoeser,arvindn}@cs.princeton.edu

Abstract. Blockchain analysis is essential for understanding how cryptocurrencies like Bitcoin are used in practice, and address clustering is a cornerstone of blockchain analysis. However, current techniques rely on heuristics that have not been rigorously evaluated or optimized. In this paper, we tackle several challenges of change address identification and clustering. First, we build a ground truth set of transactions with known change from the Bitcoin blockchain that can be used to validate the efficacy of individual change address detection heuristics. Equipped with this data set, we develop new techniques to predict change outputs with low false positive rates. After applying our prediction model to the Bitcoin blockchain, we analyze the resulting clustering and develop ways to detect and prevent cluster collapse. Finally, we assess the impact our enhanced clustering has on two exemplary applications.

1 Introduction

Blockchain analysis techniques are essential for understanding how cryptocurrencies like Bitcoin are used in practice. A major challenge in analyzing blockchains is grouping transactions belonging to the same user. Users can create an unlimited amount of addresses to receive and send coins. As a result, their activity is often split among a multitude of such addresses. *Address clustering heuristics* aim to identify addresses under an individual user's control based on assumptions about how wallets create transactions. As the term *heuristic* suggests, address clustering today is more intuitive than rigorous; our overarching goal in this paper is to elevate it to a science.

There are at least four applications for which accurate address clustering is important. First, a law enforcement agency may be interested in evaluating the transactions of a specific entity. They may supplement their own investigation with a set of reliable heuristics to identify relevant transactions. Second, and conversely, the ability to accurately determine a user's transactions directly impacts their privacy. This tension between law enforcement needs and everyday users' privacy is inherent to cryptocurrencies due to their transparency and pseudonymity. Advocates from one side push for greater privacy and from the other side for stronger regulation. To better understand this tug-of-war, it is important to quantify how reliable change address heuristics are in practice. Third, accurate grouping of transaction activity is important for aggregate analyses such as studying economic activity over time. This usually requires a full clustering

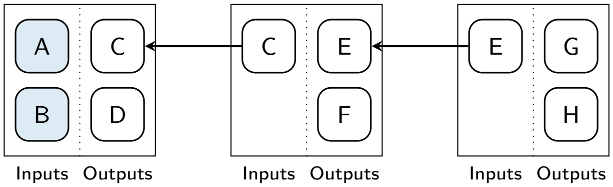


Fig. 1. The multi-input heuristic clusters addresses of inputs jointly spent in the same transaction. It does not cluster addresses that are never co-spent with other addresses (such as C and E).

of all addresses on the blockchain. Finally, the unique challenges of address clustering may be interesting for researchers outside of cryptocurrencies. For example, it may pose as an application domain for machine learning models and could be used as a benchmarking application.

The current state of address clustering techniques available to researchers is sub-optimal in multiple ways. The most common heuristic, *multi-input*, groups addresses that are jointly used in inputs of a transaction [31, 32]. This heuristic is easy to apply, moderately effective in practice [15], and widely used. However, it misses addresses that are never co-spent with other addresses (cf. Figure 1).

Many of these addresses can be clustered using *change address* heuristics: as coins in Bitcoin cannot be spent partially, transactions return the surplus value back to the sender. Identifying the change output thus allows grouping the associated address with the inputs' addresses. However, as the Bitcoin protocol does not explicitly distinguish between change and spend outputs, heuristics need to be used to identify them.

While the importance of change address identification and clustering has been demonstrated empirically and through simulation [1, 25], it remains difficult to assess how well it works in practice. A major issue is that researchers currently lack ground truth data on change outputs to assess the accuracy of individual heuristics. We are only aware of one prior study from 2015 that exploited weaknesses in a lightweight client [28], which allowed to extract the addresses of 37 585 wallets to assess four different clustering heuristics. Blockchain intelligence companies might possess manually curated and refined data sets and clusterings, but their techniques and data aren't openly available to researchers (or only shared in limited form, e.g., [14, 37]). As a result, analyses of new heuristics often fall short of quantifying their accuracy and resort to analyzing the resulting clusterings only (e.g., [7, 38]). Furthermore, clustering is applied inconsistently across studies: many forgo change address clustering entirely (e.g., [18, 19, 22, 33]), whereas some simply apply a single change heuristic (e.g., [8, 29]).

Considering this state of affairs, our goals in this paper are to address the lack of ground truth data and assessment methods, develop new techniques to apply heuristics to predict change and use them to create improved clusterings.

Contributions, methods and findings.

1. **A new ground truth method and dataset:** We put forward a procedure to select and filter transactions for which the change output has been revealed on the blockchain. Our approach exploits that future transactions of users can reveal change outputs in past transactions. We extract a set of 35.26 million transactions, carefully filtered down from 53 million candidate transactions, that can be used as ground truth for validation and prediction. (Section 2)
2. **Evaluating existing heuristics:** We've compiled and evaluate a set of 26 change address heuristics based on previous literature and community resources. Most heuristics individually produce few false positives at low to medium true positive rates. We find that due to changes in the protocol and usage patterns, heuristics wax and wane in their effectiveness over time, showing the need to use multiple heuristics and combine them in an adaptive way rather than rely on a fixed algorithm. (Section 3.1)
3. **Improved prediction:** We use a random forest classifier to identify change outputs and compare it against a baseline: the majority vote of individual heuristics. While machine learning has been used to classify the type of entity behind a transaction (e.g., [3, 14, 16, 18, 20, 36, 37]), to the best of our knowledge our work is the first to apply it to change identification. Our random forest model outperforms the vote, correctly detecting twice as many change outputs for low false positive rates. (Sections 3.2 to 3.4)
4. **Preventing cluster collapse:** We find that a naive clustering of predicted change outputs leads to cluster collapse, despite using a high threshold to prevent false positives. We then apply constraints to the union-find algorithm underlying our clustering to prevent cluster collapse stemming from frequent, repeated interaction between entities. This prevents large-scale cluster collapse while still enhancing a majority of the involved clusters. (Section 4)
5. **Assessing impact:** We assess the impact our enhanced clustering has on two exemplary applications: cash-out flows from darknet markets to exchanges and the velocity of bitcoins. We find that the results of such typical longitudinal analyses are off by at least 11 % to 14 % if they don't fully account for clustering. (Section 5)

Limitations. Our results in this paper are limited by the availability of "real" (i.e. manually collected and validated) ground truth. As such, our analysis should be treated as a first step towards better understanding the feasibility of change address detection and clustering. However, we do not expect our high-level insights to change significantly in the light of minor corrections to our ground truth data set. We make our data set publicly available to allow other researchers to evaluate it using their own private ground truth or analysis techniques.

Our extraction mechanism relies on change outputs revealed by the multi-input heuristic. This heuristic is effective in practice [15] and widely used, but vulnerable to false positives from techniques like CoinJoin and PayJoin that are intentionally designed to break the heuristic (e.g., [9, 23, 24, 26]). While we take measures to detect CoinJoin transactions and pre-existing cluster collapse, some

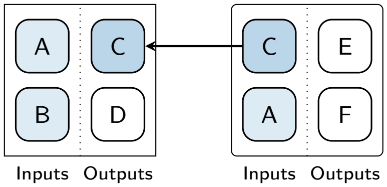


Fig. 2. The multi-input heuristic adds address C to the same cluster as addresses A and B, thereby revealing it as the change address of the first transaction.

errors can remain. Furthermore, entities that more effectively prevent address reuse are less likely to be included in our data set.

2 Building a Ground Truth Data Set

Core assumption. We focus on the feasibility of detecting the change output in Bitcoin transactions with exactly two spendable outputs, by far the most common type of transaction as of June 2021 (75.8 % of all transactions, see Figure 3). Our core assumption is that one of these outputs is a payment, and the other output receives the change. We call this type of transaction a *standard* transaction, as they are created by typical end-user wallet software.[1]

For transactions with only one output there is no good indicator to directly and reliably determine whether the output belongs to the same user. The transaction may correspond to a user sweeping the balance of their wallet, but the destination address may not be under the same user's control (e.g., it could be managed by a cryptocurrency exchange).

Transactions with more than two outputs are less likely to originate from an ordinary wallet. They may belong to an exchange that batches payouts to multiple users, or correspond to a restructuring of their hot and cold wallets. Our assumption that exactly one of the outputs receives change may not hold here.

Method. Our approach leverages that change outputs are sometimes revealed by the multi-input heuristic at a later point in time due to address reuse. Figure 2 shows how such disclosure may unintentionally happen: a user spends coins at addresses A and B, their wallet directs the change to a new address C. Later, they spend the change at address C along with other coins at address A. At this point, the multi-input heuristic reveals that A, B and C belong to the same user, thus C is the change address in the first transaction. By identifying transactions that have their change revealed in this way, we can build a ground truth set of transactions with known change.

Comparison to interactive collection. In contrast to prior deanonymization studies (e.g., [25]) our primary interest is not in identifying address clusters of specific entities but to identify change outputs in their transactions. To achieve

[1] Our definition is unrelated to the `isStandard` test in the Bitcoin reference implementation that checks whether a transaction uses common script types.

this interactively, we would need to induce them to make a transaction to an address under our control. This would likely yield inferior ground truth:

- Heterogeneous ground truth requires transactions from a *variety of different use cases, entities and wallets*. We would only be able to directly interact with some types of intermediaries (such as exchanges). Our non-interactive method, instead, is not limited to a small set of intermediaries of our choosing.
- Interactive collection would be hard to *scale* beyond a few hundred transactions, as we would have to individually engage with the intermediaries. Our non-interactive approach instead yields a data set of millions of transactions.
- Interactive collection cannot be done retroactively and is therefore limited to a short, current *time frame*. The resulting data set wouldn't capture shifting patterns over different epochs of Bitcoin's history. Our non-interactive approach however can be applied to Bitcoin's entire history.

Our method has a few important limitations. First, because we extract ground truth data non-interactively from the blockchain, we are not able to fully verify its correctness. Second, our core assumption that exactly one of the outputs belongs to the user may not hold in every scenario. For example, a user sending funds to an address under their control could lead to ambiguous or incorrect labeling of change outputs. We take specific care to remove transactions likely to violate the core assumption in this way. Similarly, there could be instances where none of the outputs is a change output. As this would require a user to make a payment to two different entities using a perfectly matching set of inputs, we expect it to be rare. Third, our ground truth set could be biased towards entities or wallet implementations that are more prone to reuse and merge addresses.

2.1 Data collection and overview

We use and build upon BlockSci v0.7 [19], an open-source blockchain analysis framework that provides fast access to blockchain data upon which we implement custom heuristics and extraction procedures. We parse the Bitcoin blockchain until the end of June 2021 (block height 689 256) and create a *base clustering* using the multi-input heuristic (where we heuristically exclude CoinJoin transactions).

As of June 2021, the blockchain contains 91 million transactions with one output, 495 million with two outputs, and 67 million with three or more outputs (see Figure 3). We divide the transactions into mutually exclusive categories. Transactions with unspendable OP_RETURN outputs often signal the use of an overlay application that stores metadata in the blockchain [4]. Such transactions may follow unique rules for their construction, potentially making change detection unreliable. Transactions directly reusing an input address have their change output trivially revealed and applying change heuristics is not necessary. We thus focus on transactions where the change has been revealed by the multi-input heuristic and use them to construct our ground truth data set. For the remaining transactions, i.e. those with yet unknown change, we will later predict their change output.

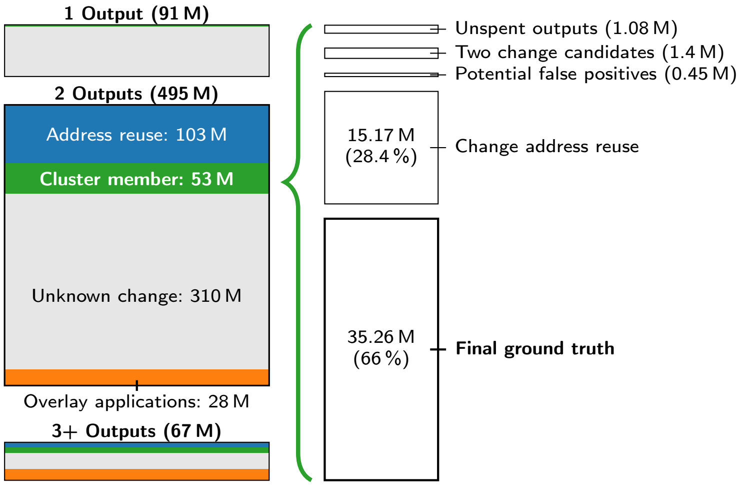
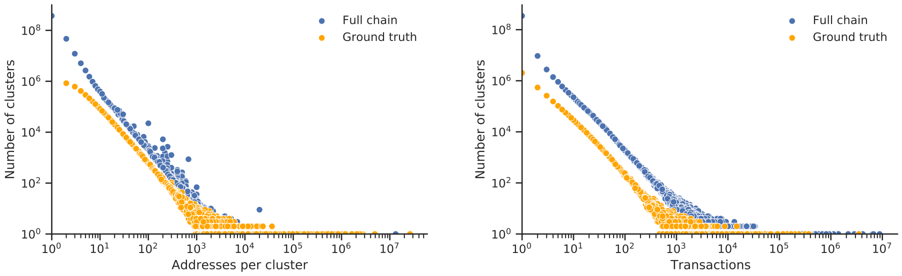


Fig. 3. Breakdown of different types of transactions in the Bitcoin blockchain until end of June 2021. Transactions with two outputs and change revealed through base cluster membership form the basis of our ground truth data, which we further refine to a final selection of 35.26 million transactions.

2.2 Refining the candidate set of ground truth transactions

Our candidate set of ground truth transactions consists of transactions with two outputs (ignoring overlay transactions) where no input address is reused for change and where at least one output is in the same base cluster as the inputs. This yields a total of 53.41 million transactions. We further filter them as follows (see Figure 3 for a visual breakdown, and Appendix A for additional details):

1. We remove 1.08 million transactions with unspent outputs, as our subsequent analyses rely upon the spending transactions being known.
2. For 0.97 million transactions both outputs are in the same base cluster, violating our core assumption. We remove these transactions. As some base clusters appear to be more likely to produce such transactions, we exclude transactions from base clusters where more than 10 % of transactions exhibit this behavior. This removes 0.48 million transactions in 9967 base clusters.
3. We check our base clustering for preexisting cluster collapse, which could create false positives. We remove 0.37 million transactions belonging to the Mt.Gox supercluster (cf. [15]) as well as 0.09 million transactions from one possible instance of cluster collapse detected using address tags from the website WalletExplorer.com.
4. We find many instances where the change address did not appear in the inputs, but had been used before and was known to be the change at the time the transaction was created through multi-input clustering. For example, there are 5.77 million transactions originating from the gambling service "SatoshiDice" that use only a total of 50 change addresses, and 1.27 million



(a) Number of base clusters represented in our ground truth by total address count.

(b) Number of transactions in ground truth and full blockchain per base cluster.

Fig. 4. Address and transaction counts for base clusters in our ground truth.

transactions from "LuckyB.it" that use a single change address. For such transactions, applying change address heuristics is never necessary. We remove 15.17 million transactions where the change output was already known at the time the transaction was created.

2.3 Assessing the final set of ground truth transactions

Scale and time frame. Our final ground truth set of 35.26 million transactions makes up about 7.6 % of standard transactions and about 5.4 % of all transactions. These percentages are relatively stable over time.

Variety of included clusters. Our ground truth includes transactions from 3.6 million base clusters. Figure 4a shows the distribution of address counts of base clusters that are represented with at least one transaction in our ground truth. Our ground truth contains transactions from base clusters of all sizes, giving us confidence that it can be representative of the blockchain overall.

Figure 4b shows the number of transactions per base cluster included in the ground truth compared to the total number of transactions per cluster, showing an overall similar distribution. The largest number of transactions from a single base cluster is 3.49 million, which has 8.85 million transactions in total. We did not find a label for it on WalletExplorer.com. The second highest number of transactions is 383 519, again from an unlabeled cluster.

Transaction composition and use of protocol features. Compared to standard transactions with yet unknown change, our ground truth transactions have more inputs (38.9 % of transactions have three or more inputs, compared to 7.6 % for the remaining transactions). This is an expected artifact of our selection method, which relies on transactions with more than one input to reveal change outputs. The share of transactions using SegWit serialization or allowing for fee bumping (RBF) is also higher in the set of remaining transactions. In Appendix B we provide additional details on the characteristics of our ground truth data.

2.4 Data release

We make our ground truth data set publicly available to allow other researchers to evaluate it using their own tools and techniques.[2] We believe that making this data public does not create significant new privacy risks: all information necessary to recreate the data set is already publicly available on the Bitcoin blockchain and our method—extracting change outputs revealed by the multi-input heuristic—is easy to reproduce with open-source tools like BlockSci.

3 Predicting Change Outputs

The Bitcoin protocol does not explicitly distinguish between change and spend outputs. However, wallets create change outputs automatically to return surplus value when users make payments. This allows to guess the change using a variety of heuristics targeted at identifying specific wallet or user behavior.

In this paper we evaluate two general types of heuristics. *Universal* heuristics use characteristics of the transaction and change output to determine the change. For example, the address type of a change output is likely to match the address types of the inputs, and rounded output values may indicate spend amounts. *Fingerprint* heuristics determine change based on matching characteristics of the transactions spending the outputs. For example, if a transaction sets a positive locktime to prevent fee sniping [35] and only one of the outputs is spent in a transaction with the same behavior, it is likely the change. We are not aware of any prior work that has evaluated fingerprinting across the range of available protocol characteristics. In total, we use 9 variants of universal heuristics and 17 variants of fingerprinting heuristics (cf. Table 1). To prevent cluster collapse, we explicitly encode our constraint that only one output can be the potential change: if both outputs are change candidates, none is returned by our heuristics.

在本文中，我们评估了两种一般类型的启发式。通用启发式方法使用事务和更改输出的特征来确定更改。例如，变更输出的地址类型很可能与输入的地址类型匹配，四舍五入的输出值可能表示开销。指纹启发式算法根据花费输出的交易的匹配特征确定变更。例如，如果一个事务设置了一个正的锁时间来防止费用狙击[35]，并且只有一个输出被花费在具有相同行为的事务中，那么很可能会发生更改。我们不知道之前有任何工作评估了指纹的范围内可用的协议特征。总的来说，我们使用了9种通用启发式和17种指纹启发式(参见表1)。为了防止集群崩溃，我们显式地对我们的约束进行编码，即只有一个输出可以是潜在的更改:如果两个输出都是更改候选，启发式不会返回任何结果。

3.1 Assessing individual change heuristics

In a first step we assess each of the heuristics using our ground truth data set. We find that most heuristics produce few false positives but often only apply to a small share of transactions (most heuristics have true positive rates between 10 % to 30 %; detailed individual results are provided in Appendix D). Figure 5 shows the average number of correct and incorrect predictions per transaction over time, grouped by the type of heuristic.

We see three important trends: first, the universal heuristics drop over time, likely due to rounded values becoming less common. Second, the consistent fingerprint heuristics see a steady uptick in the number of correct votes, enabled

[2] `https://github.com/maltemoeser/address-clustering-data`

Table 1. Change heuristics proposed in the literature and used in this paper.

Heuristic	Notes and limitations	Used
Optimal change: There should be no unnecessary inputs: if one output is smaller than any of the (2+) inputs, it is likely the change. [28, 30]	Only applies to transactions with 2+ inputs. We use two variants, one ignoring and one accounting for the fee.	✓
Address type: The change likely uses the same address type as the inputs. [19, 30]	False positives possible due to protocol upgrades or obfuscation.	✓
Power of ten: As purchase amounts may be rounded, and change amounts depend on the input values and fee, it is more likely to have fewer trailing zeros. [19, 30]	We use six different variants, which are partially redundant.	✓
Shadow address: Many clients automatically generate fresh change addresses, whereas spend addresses may be more easily reused. [1, 25]	Modern wallets discourage reuse of receiving addresses. We do not use the heuristic as our ground truth is filtered based on address freshness.	×
Consistent fingerprint: The transaction spending a change output should share the same characteristics [6, 30]. We use 17 variants based on the following characteristics: – input/output counts and order – version – locktime – serialization format (SegWit) – replace-by-fee (RBF) – transaction fee – input coin age (zero-conf) – address and script types	False positives are possible when a wallet implementation or the protocol change. We only consider characteristics after they are available in the protocol. Appendix C describes the characteristics we use in more detail.	✓

我们看到了三个重要的趋势:首先，普遍启发法随着时间的推移而下降，这可能是因为圆角值变得越来越不常见。第二，一致的指纹启发式可以看到正确投票的数量稳步上升，这是由于随着时间的推移，比特币中可用的协议功能越来越多。最后，从2017年底开始，正确和不正确的指纹投票都出现了上升，当时钱包实现开始切换到SegWit序列化和地址格式(例如，[5,34])。

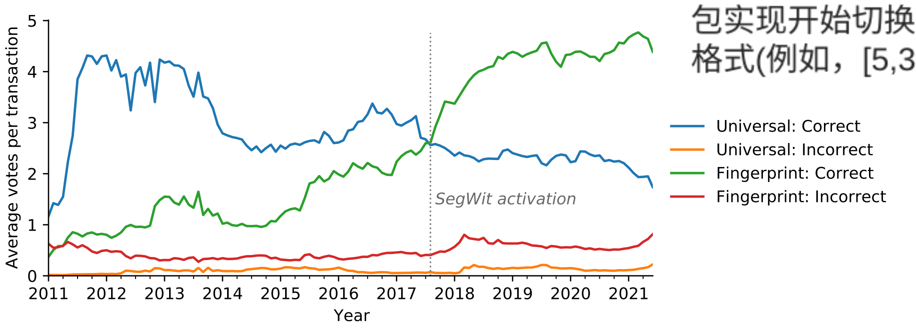


Fig. 5. Average number of correct and uncorrect votes per transaction and type of heuristic in the ground truth data set, over time

by the increasing variety of protocol features available in Bitcoin over time. Finally, there's an uptick in both correct and incorrect fingerprint votes starting in late 2017, when wallet implementations started to switch to SegWit serialization and address formats (e.g., [5, 34]).

For 858 582 transactions no heuristic returned a change output, we remove these from the subsequent analyses. When we later predict change outputs for the remaining standard transactions, we will also exclude transactions where no heuristic determined a potential change output.

While most individual heuristics have high precision, they only cover a subset of transactions each. Furthermore, some heuristics may be more applicable during certain epochs of Bitcoin's history than others. Given the variety of heuristics available to us compared to previous studies (e.g., an evaluation of three change heuristics in [28]), we now consider new ways of combining them.

3.2 Threshold vote

Figure 5 suggest that a majority of heuristics should generally identify the correct output. However, the number of heuristics returning an potential output varies among transactions, and individual heuristics could be incorrect. We thus compute a threshold vote: if at least t more heuristics vote for output a than for output b, then output a is considered the change. Increasing the threshold t thus allows the analyst to require higher degrees of confidence and thereby lower the risk of cluster collapse.

We apply the threshold vote to our ground truth data set and plot the resulting ROC curve in Figure 6a (for comparison, we also show the FPR and TPR of each individual heuristic). We achieve an ROC AUC of 0.94, and, for example, a 37.0 % true positive rate (TPR) below a false positive rate (FPR) of 0.1 % with a threshold of $t = 7$.

Using a threshold vote may not be ideal as the individual heuristics have varying true positive and false positive rates, and some might be more or less

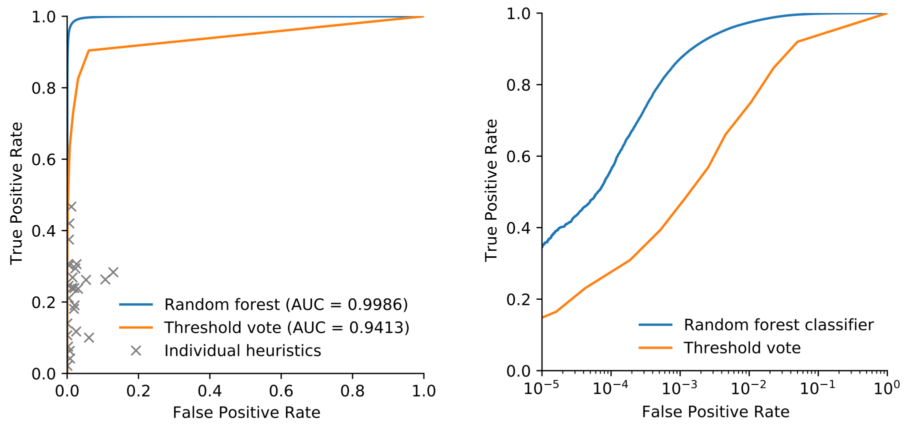
对于没有返回更改输出的858 582个事务，我们从随后的分析中删除这些事务。当我们稍后预测剩余标准事务的更改输出时，我们还将排除没有启发式方法确定潜在更改输出的事务。

虽然大多数启发式算法都具有很高的精度，但它们只覆盖每个事务的一个子集。此外，一些启发法在比特币历史的某些时期可能比其他时期更适用。与之前的研究相比，我们可以使用多种启发法(例如，[28]中对三种改变启发法的评估)，我们现在考虑将它们结合起来的新方法。

图5表明，大多数试探法通常应该识别正确的输出。但是，返回潜在输出的启发式方法的数量因事务而异，个别启发式方法可能不正确。因此，我们计算一个阈值投票:如果对输出a的启发式投票比对输出b的启发式投票至少多t，那么输出a被认为是变化。因此，增加阈值t允许分析人员要求更高的置信度，从而降低集群崩溃的风险。

我们将阈值投票应用于我们的地面真实数据集，并在图6a中绘制得到的ROC曲线(为了进行比较，我们还显示了每个启发式的FPR和TPR)。我们获得的ROC AUC为0.94，例如，37.0%的真阳性率(TPR)低于0.1%的假阳性率(FPR)，阈值为t = 7。

使用阈值投票可能不是理想的，因为单个启发式有不同的真正和假正率，有些可能在比特币历史的不同时期或多或少是可靠的。相反，启发式的一个特定子集可能提供更好的分类准确性。我们选择使用监督学习分类器，而不是手动尝试不同的组合



(a) The random forest classifier outperforms the threshold vote and the individual heuristics.

(b) The log scale highlights the difference between the classifiers for low false positive rates (on the same test set).

Fig. 6. ROC curves for predicting change in the ground truth data set using the threshold vote and the random forest classifier, compared to individual heuristics.

reliable during different periods of Bitcoin's history. Rather, a specific subset of heuristics may provide better classification accuracy. Instead of manually trying different combinations, we opt to use a supervised learning classifier.

我们决定使用随机森林分类器来预测事务的变化输出，而不是手动尝试不同的组合。随机森林是一个集合分类器
训练和聚集单个决策树的结果。它本质上非常适合我们的数据集，因为它可以根据协议特征或时间段将数据集划
为同构子集。在对我们的数据进行监督分类器的初步比较中，它也获得了最高的ROC AUC评分(参见附录E)。

3.3 Random forest classifier

We decide to use a random forest classifier to predict a transaction's change output. A random forest is an ensemble classifier that trains and aggregates the results of individual decision trees. It is inherently well suited for our data set as it can divide it into homogeneous subsets, for example, based on protocol characteristics or time periods. In an initial comparison of supervised classifiers on our data it also achieved the highest ROC AUC score (cf. Appendix E).

We model an output-based binary classification problem, where every output is either a change (1) or spend (0) output. An individual heuristic may produce one of three outcomes: vote for the output, against the output, or not be able to discern between the outputs. We further add characteristics about each output and corresponding transaction that may allow the random forest to differentiate between distinct types of transactions, or wallets (see Appendix E for details).

As we consider an analyst who works with a static snapshot of the blockchain, we randomly split our data set into 80 % training and 20 % test set. We use the training set for hyperparameter tuning using 4-fold cross-validation, using the ROC AUC as our scoring metric. To account for the fact that transactions in the same base cluster may be highly similar, we explicitly ensure that all outputs of a base cluster remain in the same set and fold.

将随机森林模型(RF-1)应用到测试集，我们获得了0.9986的AUC(图6a)。与阈值投票相比，该模型能够检测出更高比例的输出，特别是在低误报率的情况下。

Applying the random forest model (RF-1) to the test set, we achieve an AUC of 0.9986 (Figure 6a). The model is able to detect a higher share of outputs, especially at low false positive rates, compared to the threshold vote.

In Figure 6b we show the ROC curves of both the threshold vote and the random forest on the same test set, log-transforming the x-axis to highlight the important difference in low false positive rates (to prevent cluster collapse). The random forest achieves much higher true positive rates at low false positive rates, meaning that it correctly identifies the change output in a larger number of transactions. For example, if we target a false positive rate below 0.1 %, the threshold vote achieves a TPR of around 39 % at a FPR of 0.06 %. For the same FPR, the random forest achieves a TPR of 82 %, more than twice as high.

在图6b中，我们展示了阈值投票和随机森林在同一测试集上的ROC曲线，用log转换x轴来突出低误报率的重要差异(以防止群集崩溃)。在较低的误报率下，随机森林的真阳性率要高得多，这意味着它能正确识别大量事务中的变更输出。例如，如果我们将误报率设定在0.1%以下，门槛投票的TPR将达到39%左右，FPR为0.06%。对于相同的FPR，随机森林的TPR达到了82%，是前者的两倍多

We train a second random forest model (RF-2) without the fingerprint heuristics on transactions that contain predictions from the universal heuristics to later predict change in transactions with unspent outputs. Using a similar evaluation strategy as for the full model, the ROC AUC of this model is 0.9981.

To ensure that the performance of our model is not dependent on the particular split and to determine its variance, we repeatedly split our ground truth data set into 80 % training and 20 % test set 20 times and train the random forest classifier using the previously determined hyperparameters. The average ROC AUC score on the test sets is 0.9974 (SD = 0.0016) for RF-1, and 0.9965 (SD = 0.0036) for RF-2.

我们训练第二个随机森林模型(RF-2)，不使用交易的指纹启发式，该模型包含来自通用启发式的预测，以预测具有未使用输出的交易中的变化。使用与完整模型相似的评估策略，该模型的ROC AUC为0.9981。为了保证我们的模型的性能不依赖于特定的分割，并确定其方差，我们重复将我们的ground truth数据集分割为80%训练集和20%测试集20次，并使用之前确定的超参数训练随机森林分类器。RF-1测试集的平均ROC AUC评分为0.9974 (SD = 0.0016)，RF-2为0.9965 (SD = 0.0036)。

We note one caveat: because the base clustering is incomplete, grouping transactions by their base cluster may not fully prevent homogeneous transactions from the same entity to appear in both sets. Yet, some of the variability we see comes from unusual clusters that do not appear in the respective training sets. Other researchers with private, more heterogeneous ground truth may be able to evaluate the degree to which this affects the overall performance of the model.

我们注意到一个警告: 由于基本集群是不完整的，按其基本集群分组事务可能不能完全防止来自同一实体的同质事务出现在两个集合中。然而，我们看到的一些可变性来自于不寻常的集群，它们没有出现在各自的训练集。其他拥有私人的、更加异质的事实的研究人员可能能够评估这在多大程度上影响了模型的整体性能。

3.4 Additional model validation

We use two data sets to assess the performance of the random forest model trained on the entire ground truth data. First, we use 16 764 transactions identified by Huang et al. [17] as ransom payments related to the Locky and Cerber ransomware. Those payments were identified through clustering, transaction graph analysis and known characteristics of the ransom amounts. After removing non-standard transactions and those with revealed change output, we predict the change output for 11 196 transactions and achieve an AUC of 0.996.

Our second data set is constructed using a GraphSense tagpack [12] that contains 382 tags for addresses of 273 distinct entities (such as exchanges or gambling services) extracted from WalletExplorer.com. We identify each associated cluster and then extract up to 1000 transactions occurring between the individual clusters, assuming that the output belonging to a different cluster is the spend output. After removing transactions with no predictions as well as those with revealed change output, we predict the change output for 268 774 transactions and achieve an AUC of 0.976.

在这里插入文字

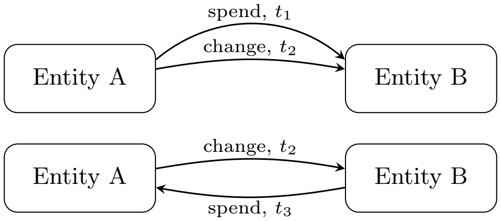


Fig. 7. In the pictured scenarios our constrained clustering prevents the merging of clusters A and B due to conflicting types of payments between them.

现在，我们使用随机森林模型通过对更改输出进行聚类来增强基本聚类。为此，我们预测3.1亿个标准交易的变化产出，但变化未知。我们排除了1050万个没有单独启发式识别变更输出的事务，并对1930万个具有未使用输出的事务使用RF-2。为了降低误报的可能性，我们使用p change = 0.99.3的保守概率阈值，这给我们带来了1.5556亿个变更输出(交易的50.24%)。然后，我们按照事务在区块链上出现的顺序，将输入的基本集群与更改地址的基本集群合并，从而增强基本集群。

4 Clustering Change Outputs

We now use our random forest models to enhance the base clustering by clustering change outputs. To this end, we predict the change outputs for 310 million standard transactions with yet unknown change. We exclude 10.5 million transactions where no individual heuristic identified a change output and use RF-2 for 19.3 million transactions with unspent outputs.

To keep the likelihood of false positives low, we use a conservative probability threshold of $p_{change} = 0.99$.[3] This gives us 155.56 million change outputs (for 50.24 % of transactions). We then enhance the base clustering by merging the base cluster of the inputs with the base cluster of the change address in the order that the transactions appear on the blockchain.

4.1 Naive merging leads to cluster collapse

Naively clustering the identified change outputs reduces 184.3 million affected base clusters into 39.8 million enhanced clusters. However, it leads to severe cluster collapse: there is one large supercluster, containing the prior Mt. Gox supercluster, that contains 223.9 million addresses (a 1596 % increase) and 108.2 million transactions (a 2500 % increase). Inspecting the 273 clusters labeled by the Graphsense tag pack, we find that 113 have been merged into the supercluster.

4.2 Constraints prevent cluster collapse

The majority of cluster merges involve address clusters from which only a single transaction originated. Here, the impact of a single misclassification is low unless a sequence of such merges collapses multiple larger clusters. At the same time, we observe a small number of merges that combine two large clusters. Imagine two large exchanges whose users frequently interact with each other. A single, misidentified change output could collapse their clusters.

[3] This corresponds to a false positive rate of 0.044 % for RF-1. We use a threshold of 0.997 for RF-2 to match the FPR.

Approach. We use this intuition to constrain which clusters we merge. While change outputs predicted by our model should be clustered, we can use predicted spend outputs to prevent cluster merges: the input cluster should not be clustered into the cluster of the spend. Given the probability p_i returned by the random forest model for output i, we define two thresholds p_{change} and p_{spend} such that if $p_i > p_{change}$ the clusters should be merged (as before), but if $p_i < p_{spend}$ then the clusters should not be merged. In many cases, these constraints will prevent the spend and change output of a transaction to end up in the same cluster (cf. Figure 7).

This approach is comparable to that by Ermilov, Panov, and Yanovich [10] to use address tags in combination with a probabilistic model to reduce the number of conflicting tags in the final clustering. However, public sources of address tags contain information on a limited number of intermediaries only. Our approach, instead, potentially covers all clusters appearing in the 310 million standard transactions, including those that may be hard to interact with (and identify) manually. Due to the size of our data set we only consider the binary case of preventing any potential conflict, accepting that we may prevent some valid merges in the process.

We implement a constrained union-find algorithm that prevents merging two clusters related by a predicted spend output. For every spend from cluster c_m to cluster c_n, predicted with $p_i < p_{spend}$, we add a constraint to cluster c_m that it must not be merged with cluster c_n. Before merging two clusters, we the check the constraints of both clusters and skip the merge if it would violate them.

Results. Using the same $p_{change} = 0.99$ and setting $p_{spend} = 0.01$, the constrained clustering prevents 413 608 merges that would have violated constraints and retains 231 340 more individual clusters than the unconstrained clustering.

We find that the constraints prevent the previously observed severe cluster collapse. For example, the constrained clustering does not produce the large Mt. Gox supercluster: the cluster contains only 4.4 million transactions (a 6 % increase) and 14.5 million addresses (a 10 % increase). Assessing the 273 labeled clusters, there are seven instances where two labeled clusters were merged. We suspect that unusual types of payouts from these services might have triggered the collapse.

The largest cluster in the constrained clustering contains 20.4 million transactions and 40.5 million addresses. Inspecting its composition, we find that it is the result of merging many small clusters (including 9 421 343 single-transaction clusters).

Overall, in at least 90 % of merges the smaller cluster created at most one outgoing transaction, which highlights the usefulness of change address clustering to merge small clusters that are missed by multi-input clustering. The constrained clustering specifically prevents some of the largest merges observed in the naive clustering, thereby preventing cluster collapse.

Varying thresholds. We chose conservative thresholds in order to reduce the possibility of cluster collapse. At the same time, this means that fewer change outputs are being clustered than with lower thresholds. To assess the impact of varying thresholds, we create two additional constrained clusterings, one with a threshold corresponding to a 0.1 % FPR and one corresponding to a 1 % FPR. At 0.1 %, the number of collapsed clusters identified by the Graphsense tag pack increases to 12. At 1 %, however, there are already 60 instances of cluster collapse. This highlights the importance of using conservative thresholds to prevent cluster collapse.

5 Impact on Blockchain Analyses

Address clustering is a common preprocessing step before analyzing activity of entities on the blockchain. Using different change heuristics (or none at all) thus affects the outcome of these analyses.

5.1 Increased cashout flows from darknet markets to exchanges

We evaluate the impact of our enhanced clustering on analysing payment flows from darknet markets to exchanges. Such analyses are potentially relevant for cybercrime researchers, economists, regulators or law enforcement, highlighting the importance of address clustering for a variety of use cases. To identify relevant intermediaries, we use address tags in the GraphSense tag pack for 117 exchanges and 15 darknet markets.

We extract the value of all outputs in transactions initiated by a darknet market that are sending bitcoins to an exchange, comparing the transaction volume calculated using our base clustering to that of our enhanced clustering. The median increase in value sent across all 15 markets amounts to 11.5 %. The total amount of bitcoins flowing from the darknet markets to exchanges increases from BTC 823 839 to BTC 937 330 (a 13.8 % increase). We provide each individual market's increase in transaction volume in Appendix F.

5.2 Improved estimate of velocity

We replicate the analysis of velocity conducted by Kalodner et al. [19], an example for a longitudinal analysis of economic activity occurring on the Bitcoin blockchain. For this analysis, clustering is used to remove self-payments of users (such as change outputs), which would artificially inflate estimates of economic activity. The better and more complete our clustering, the more self-payments are removed and hence the lower the estimate will be.

Our refined clustering reduces their estimate of bitcoins moved per day between January 2017 to June 2021 by about 11.9 %. We notice that the magnitude is quite similar to the impact on cash-out flows.

5.3 Comparison to the Meiklejohn et al. heuristic

Finally, we compare our clustering to one created naively using the address reuse-based heuristic presented by Meiklejohn et al. [25], which has subsequently been used in other studies (e.g., [8, 29]). While the authors highlight the need for manual intervention to prevent cluster collapse, this is likely infeasible for analysts without in-depth domain knowledge or the right set of tools. The heuristic considers an output to be the change if its address has only been used a single time, based on common wallet behavior to not reuse change addresses.

Applying the heuristic to standard transactions with unknown change produces a supercluster containing 133.1 million transactions and 298.4 million addresses, with 177 tagged clusters ending up in the supercluster. The probability of two addresses being clustered together increases by a factor of 40 compared to our constrained clustering. Looking at the individual predictions, the heuristic differs on 1.9 million transactions out of an overlapping 81.1 million. The total pairwise difference in output values between those predictions amounts to BTC 4.1 million, or USD 38.7 billion, a significant difference in economic activity that might be misattributed due to clustering.

6 Conclusion

Address clustering is an important cornerstone of many blockchain analyses. In this paper, we've taken a first step towards building better models that allow analysts to identify change outputs in transactions, enabled by a new ground truth data set extracted from the Bitcoin blockchain. Evaluating this data set, we find that for most transactions identifying the change address is feasible with high precision. Crucially, our work is the first to apply machine learning to the problem of change identification. We find that our random forest model outperforms a baseline voting mechanism, detecting twice as many change outputs when targeting low false positive rates. Turning to the subsequent clustering of change addresses, we've demonstrated that constraints based on our model's predictions can prevent cluster collapse. Finally, we've explored the impact of our clustering on the outcome of economic analyses. We hope that our work will encourage and enable further research into address clustering.

Acknowledgements

We thank Rainer Böhme and Kevin Lee for their feedback on an earlier draft of this paper. This work is supported by NSF Award CNS-1651938 and a grant from the Ripple University Blockchain Research Initiative.

References

[1] Elli Androulaki, Ghassan O Karame, Marc Roeschlin, Tobias Scherer, and Srdjan Capkun. "Evaluating user privacy in Bitcoin". In: *International Conference on Financial Cryptography and Data Security*. Springer. 2013, pp. 34–51.

[2] Kristov Atlas. *Lexicographical Indexing of Transaction Inputs and Outputs*. URL: `https://github.com/bitcoin/bips/blob/master/bip-0069.mediawiki` (visited on 01/20/2020).

[3] Massimo Bartoletti, Barbara Pes, and Sergio Serusi. "Data mining for detecting Bitcoin Ponzi schemes". In: *2018 Crypto Valley Conference on Blockchain Technology (CVCBT)*. IEEE. 2018, pp. 75–84.

[4] Massimo Bartoletti and Livio Pompianu. "An analysis of Bitcoin OP_RETURN metadata". In: *Financial Cryptography and Data Security: FC 2017 International Workshops, WAHC, BITCOIN, VOTING, WTSC, and TA, Sliema, Malta, April 7, 2017, Revised Selected Papers*. Springer. 2017, pp. 218–230.

[5] *Bitcoin Core 0.16.0*. URL: `https://bitcoincore.org/en/releases/0.16.0/` (visited on 02/25/2021).

[6] *Blockchair.com API v.2.0.76 Documentation: Privacy-o-meter*. URL: `https://blockchair.com/api/docs#link_M6` (visited on 02/22/2021).

[7] Tao-Hung Chang and Davor Svetinovic. "Improving bitcoin ownership identification using transaction patterns analysis". In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 50.1 (2018), pp. 9–20.

[8] Mauro Conti, Ankit Gangwal, and Sushmita Ruj. "On the economic significance of ransomware campaigns: A Bitcoin transactions perspective". In: *Computers & Security* 79 (2018), pp. 162–189.

[9] Nicolas Dorier. *A Simple Payjoin Proposal*. URL: `https://github.com/bitcoin/bips/blob/master/bip-0078.mediawiki` (visited on 01/20/2020).

[10] Dmitry Ermilov, Maxim Panov, and Yury Yanovich. "Automatic Bitcoin address clustering". In: *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE. 2017, pp. 461–466.

[11] Mark Friedenbach, BtcDrak, Nicolas Dorier, and kinoshitajona. *Relative lock-time using consensus-enforced sequence numbers*. URL: `https://github.com/bitcoin/bips/blob/master/bip-0068.mediawiki` (visited on 01/20/2020).

[12] *GraphSense Public TagPacks*. URL: `https://github.com/graphsense/graphsense-tagpacks` (visited on 04/01/2021).

[13] David A. Harding and Peter Todd. *Opt-in Full Replace-by-Fee Signaling*. URL: `https://github.com/bitcoin/bips/blob/master/bip-0125.mediawiki` (visited on 01/20/2020).

[14] Mikkel Alexander Harlev, Haohua Sun Yin, Klaus Christian Langenheldt, Raghava Mukkamala, and Ravi Vatrapu. "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.

[15] Martin Harrigan and Christoph Fretter. "The unreasonable effectiveness of address clustering". In: *2016 Intl IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld)*. IEEE. 2016, pp. 368–373.

[16] Yining Hu, Suranga Seneviratne, Kanchana Thilakarathna, Kensuke Fukuda, and Aruna Seneviratne. "Characterizing and Detecting Money Laundering Activities on the Bitcoin Network". In: *arXiv preprint arXiv:1912.12060* (2019).

[17] Danny Yuxing Huang, Maxwell Matthaios Aliapoulios, Vector Guo Li, Luca Invernizzi, Elie Bursztein, Kylie McRoberts, Jonathan Levin, Kirill Levchenko, Alex C Snoeren, and Damon McCoy. "Tracking ransomware end-to-end". In: *IEEE Symposium on Security and Privacy*. IEEE. 2018, pp. 618–631.

[18] Marc Jourdan, Sebastien Blandin, Laura Wynter, and Pralhad Deshpande. "Characterizing entities in the Bitcoin blockchain". In: *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*. IEEE. 2018, pp. 55–62.

[19] Harry Kalodner, Malte Möser, Kevin Lee, Steven Goldfeder, Martin Plattner, Alishah Chator, and Arvind Narayanan. "Blocksci: Design and applications of a blockchain analysis platform". In: *29th USENIX Security Symposium*. 2020, pp. 2721–2738.

[20] Yu-Jing Lin, Po-Wei Wu, Cheng-Han Hsu, I-Ping Tu, and Shih-wei Liao. "An evaluation of Bitcoin address classification based on transaction history summarization". In: *2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*. IEEE. 2019, pp. 302–310.

[21] Eric Lombrozo, Johnson Lau, and Pieter Wuille. *Segregated Witness (Consensus layer)*. URL: `https://github.com/bitcoin/bips/blob/master/bip-0141.mediawiki` (visited on 01/20/2020).

[22] Damiano Di Francesco Maesa, Andrea Marino, and Laura Ricci. "Data-driven analysis of bitcoin properties: exploiting the users graph". In: *International Journal of Data Science and Analytics* 6.1 (2018), pp. 63–80.

[23] Gregory Maxwell. *CoinJoin: Bitcoin Privacy for the Real World*. 2013. URL: `https://bitcointalk.org/index.php?topic=279249.0` (visited on 06/30/2019).

[24] Sarah Meiklejohn and Claudio Orlandi. "Privacy-enhancing overlays in Bitcoin". In: *International Conference on Financial Cryptography and Data Security*. Springer. 2015, pp. 127–141.

[25] Sarah Meiklejohn, Marjori Pomarole, Grant Jordan, Kirill Levchenko, Damon McCoy, Geoffrey M Voelker, and Stefan Savage. "A fistful of bitcoins: characterizing payments among men with no names". In: *Internet Measurement Conference*. ACM. 2013, pp. 127–140.

[26] Malte Möser and Rainer Böhme. “The price of anonymity: empirical evidence from a market for Bitcoin anonymization”. In: *Journal of Cybersecurity* 3.2 (2017), pp. 127–135.

[27] Malte Möser, Rainer Böhme, and Dominic Breuker. “An inquiry into money laundering tools in the Bitcoin ecosystem”. In: *eCrime Researchers Summit (eCRS), 2013*. IEEE. 2013, pp. 1–14.

[28] Jonas David Nick. “Data-driven de-anonymization in Bitcoin”. MA thesis. ETH-Zürich, 2015.

[29] Francesco Parino, Mariano G Beiró, and Laetitia Gauvin. “Analysis of the Bitcoin blockchain: socio-economic factors behind the adoption”. In: *EPJ Data Science* 7.1 (2018), p. 38.

[30] *Privacy - Bitcoin Wiki*. URL: `https://en.bitcoin.it/Privacy` (visited on 12/15/2020).

[31] Fergal Reid and Martin Harrigan. “An Analysis of Anonymity in the Bitcoin System”. In: *Security and Privacy in Social Networks*. Springer, 2013, pp. 197–223.

[32] Dorit Ron and Adi Shamir. “Quantitative analysis of the full Bitcoin transaction graph”. In: *International Conference on Financial Cryptography and Data Security*. Springer. 2013, pp. 6–24.

[33] Jürgen E Schatzmann and Bernhard Haslhofer. “Bitcoin Trading is Irrational! An Analysis of the Disposition Effect in Bitcoin”. In: *arXiv preprint arXiv:2010.12415* (2020).

[34] *SegWit FAQ*. URL: `https://help.coinbase.com/en/pro/getting-started/general-crypto-education/segwit-faq` (visited on 04/27/2021).

[35] Peter Todd. *Discourage fee sniping with nLockTime #2340*. 2014. URL: `https://github.com/bitcoin/bitcoin/pull/2340` (visited on 08/11/2015).

[36] Kentaroh Toyoda, Tomoaki Ohtsuki, and P Takis Mathiopoulos. “Multiclass Bitcoin-enabled service identification based on transaction history summarization”. In: *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. IEEE. 2018, pp. 1153–1160.

[37] Mark Weber, Giacomo Domeniconi, Jie Chen, Daniel Karl I Weidele, Claudio Bellei, Tom Robinson, and Charles E Leiserson. “Anti-money laundering in Bitcoin: Experimenting with graph convolutional networks for financial forensics”. In: *arXiv preprint arXiv:1908.02591* (2019).

[38] Yuhang Zhang, Jun Wang, and Jie Luo. “Heuristic-Based Address Clustering in Bitcoin”. In: *IEEE Access* 8 (2020), pp. 210582–210591.

A Additional details: Filtering the ground truth data set

Selecting transactions with two outputs, no OP_RETURN outputs, where no input address has been directly reused in the outputs and where at least one output is in the same base cluster as the inputs yields a total of 53.41 million transactions. We first exclude 1.08 million transactions with unspent outputs, as our subsequent analyses rely upon the spending transactions being known.

Transactions with two change candidates. Out of the 52.33 million transactions with at least one change candidate, for 0.97 million transactions *both* outputs are in the same base cluster. This can happen when a user transfers funds to an address in their own wallet, an online service restructures their funds, or cluster collapse leads to merging of both outputs' addresses. In a first step, we exclude all transactions with two change candidates.

However, it is possible that there are yet unidentified transactions (due to an incomplete base clustering) where both outputs do belong to the same entity. This should occur only in rare cases, but there may be specific intermediaries that create such transactions more frequently. We therefore exclude *all* transactions from base clusters where more than 10 % of transactions exhibit such behavior. This removes an additional 480 845 transactions in 9967 base clusters from our ground truth.

Potential false positives. A risk of using the base clustering to extract ground truth is that the multi-input heuristic could have produced false positives. For example, if a user Alice makes a payment to merchant Bob and their wallet addresses are incorrectly clustered together, her spend output could appear to be the change.

To this end, we first remove 366 926 transactions belonging to the Mt. Gox supercluster (cf. [15]). Next, we spot-check our base clustering against the website WalletExplorer.com. For the 100 largest base clusters in our ground truth we select 25 addresses at random and collect the tag (which is either explicitly named or pseudo-random) that WalletExplorer assigns to the address. In five instances, the addresses yield multiple tags. Four of these return only additional pseudo-random tags, which upon manual inspection we believe to be the result of a heuristic to not link addresses in transactions with large numbers of inputs. Only one base cluster contains addresses with two different named tags: "LocalBitcoins.com-old" and "AnxPro.com". This could be a result of cluster collapse, or an instance of mislabeling on the side of WalletExplorer. We remove the 87 947 transactions from this base cluster from our ground truth. Overall, this check gives us confidence that our base clustering does not already include widespread cluster collapse.

Change address reuse. Our initial selection removed transactions where the change address appeared in an input of the transaction. Yet, we find many instances where the change address did not appear in the inputs but had been seen before. For example, a base cluster labeled by WalletExplorer as the gambling service "SatoshiDice", contains 5.77 million transactions that use only 50 different change addresses. Similarly, there are 1.27 million transactions from a base cluster tagged as "LuckyB.it" that all use a single change address.[4] In many of these cases, the change address could have already been revealed (before the transaction took place) through the multi-input heuristic.

If the change is known at the time the transaction is created, applying change heuristics is unnecessary. In contrast, whenever a transaction uses a fresh address for change, it cannot possibly be revealed as the change at the time the transaction was created. With this intuition, we remove transactions with change addresses that were not freshly generated if, at the time they were included in the blockchain, the change had already been revealed by the multi-input heuristic. This removes a total of 15.17 million transactions (90.80 % of transactions with reused change addresses). Table 2 provides an overview of whether the change and spend addresses are fresh in our ground truth data.

Table 2. Number of transactions (in million) in our ground truth data set with fresh or reused spend and change outputs.

	Spend		
Change	Reused	Fresh	Total
Reused	0.73	0.81	1.54
Fresh	19.38	14.34	33.71
Total	20.11	15.15	35.26

B Additional details: Assessing the final set of ground truth transactions

Scale and time frame. Figure 8 shows the share of ground truth transactions of all transactions and standard transactions over time. Overall, the distribution is relatively stable.

Transaction composition and use of protocol features. Table 3 compares characteristics of transactions in our ground truth data to those of standard transactions with yet unknown change, including the number of inputs as well as a number of important protocol features (an overview and description of the protocol characteristics used in this paper is available in Appendix C).

C Protocol characteristics used for fingerprinting

- Input/output count: the number of inputs and/or outputs may indicate a wallet software's behavior of creating transactions. While the number of

[4] `1NxaBCFQwejSZbQfWcYNwgqML5wWoE3rK4`

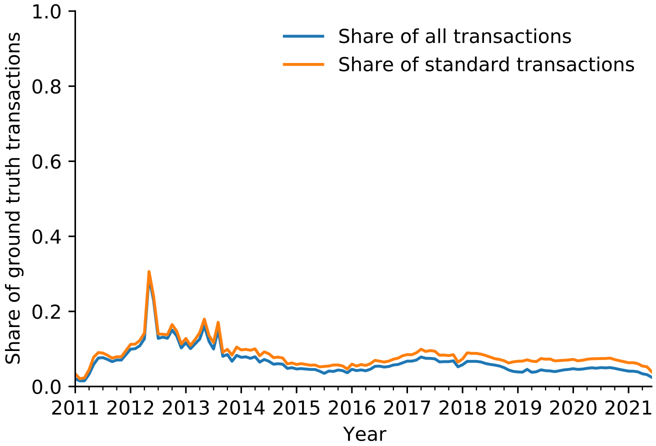


Fig. 8. Share of ground truth transactions of all and standard transactions over time.

Table 3. Comparison of transaction characteristics between ground truth transactions and transactions with 2 outputs for which change is unknown.

Characteristic	Ground truth (%)	Remaining (%)
1 Input	38.99	78.76
2 Inputs	22.09	13.61
3+ Inputs	38.92	7.63
Version = 1	79.83	80.68
Locktime > 0	25.25	24.60
RBF	3.57	6.22
SegWit	18.30	27.10
n (in million)	35.26	309.65

inputs depends on the UTXOs available to the user, some commonly occurring patterns such as peeling chains ([25, 27]) have consistent input and output counts.

- Version: BIP 68 [11] introduced relative timelocks for transactions, which requires transaction to set the transaction version to 2.
- Locktime: Transactions can set a timelock such that they are valid only after the tip of the chain has passed a specific block height or timestamp. Some clients (e.g., Bitcoin Core) produce timelocked transactions by default to prevent fee-sniping [35].
- Replace-by-fee (RBF): Transactions opting into the replace-by-fee policy can be replaced by a similar transaction paying a higher fee [13].
- SegWit: Segregated witness [21] is a protocol update that enabled storing the inputs' signatures outside of the transaction, thereby increasing available space in blocks. As the upgrade is backwards-comptabile, not all wallets produce SegWit transactions. A wallet might also be able to produce SegWit transaction, but may be required to use non-SegWit serialization if none of the inputs use SegWit. We call this behavior SegWit-conform.
- Ordered inputs/outputs: BIP 69 [2] defines non-binding rules (i.e. not enforced by the consensus mechanism) for lexicographically sorting inputs and outputs in a transaction. (A limitation of our implementation is that it does not compare the raw `scriptPubKey` in case the output values are equal, as they are not available in BlockSci).
- Zero-conf: Bitcoin user's are encouraged to wait for up to six confirmations (about an hour) before accepting a payment, as there is a risk that funds might be double-spent. A transaction spending inputs without any confirmations indicates willingness to accept the double-spending risk, which could be specific to certain intermediaries.
- Transaction fee: Bitcoin users pay transaction fees for their transactions to be included into the blockchain by miners. Some clients may pay the same exact fee (either absolute, or relative to the transaction's size) for every transaction.
- Multisignature: Multisignature scripts allow to specify a list of public keys and a threshold m such that the redeemer must provide valid signatures for m out of n of these keys. They aren't typically used by normal end-user wallets.
- Address types: Bitcoin Core defines a number of standardized output scripts types including Pay-to-Pubkey-Hash (P2PKH), Pay-to-Script-Hash (P2SH) as well as their respective SegWit variants (P2WPKH and P2WSH). Often, a wallet consistently uses a specific address type. (Compared to the normal address type heuristic, the fingerprint checks for overlap with the address types of all inputs of the spending transaction).

D True and false positive rate of individual heuristics

Table 4. True and false positive rates of each individual heuristic applied to transactions in our ground truth data set.

	Ground Truth		Remaining
Heuristic	TPR	FPR	Coverage*
Universal heuristics			
Optimal change	0.306	0.026	0.133
• incl. fee	0.239	0.020	0.096
Address type	0.237	0.031	0.369
Power of ten			
• $n = 2$	0.467	0.012	0.383
• $n = 3$	0.420	0.006	0.311
• $n = 4$	0.375	0.005	0.253
• $n = 5$	0.302	0.006	0.173
• $n = 6$	0.211	0.005	0.104
• $n = 7$	0.107	0.001	0.048
Consistent fingerprint			
Output count	0.283	0.129	0.445
Input/output count	0.263	0.107	0.568
Version	0.245	0.004	0.320
Locktime	0.307	0.003	0.363
RBF	0.075	0.003	0.114
SegWit	0.191	0.021	0.260
SegWit-conform	0.021	0.001	0.028
Ordered ins/outs	0.262	0.053	0.443
Zero-conf	0.100	0.061	0.214
Absolute fee	0.117	0.025	0.305
Relative fee	0.042	0.008	0.204
Multisignature	0.140	0.001	0.154
Address type			
• P2PKH	0.239	0.014	0.312
• P2SH	0.269	0.015	0.334
• P2WPKH	0.181	0.019	0.256
• P2WSH	0.063	0.007	0.082
All address types	0.294	0.023	0.392

[*] Coverage denotes share of standard transactions with yet unidentified change where the heuristic returned exactly one output.

E Additional details: Random forest model

Encoding. We transform our transaction-based predictions into an output-based binary classification problem. Every output is either a change (1) or spend (0) output. An individual heuristic may produce one of three outcomes: vote for the output, against the output, or not be able to discern between the outputs. Due to the large size of the data set, we forgo one-hot encoding and instead use the following ordinal encoding for the heuristics:

1 the heuristic votes for the output
0 the heuristic votes neither for nor against the output
-1 the heuristic votes against the output

Additional characteristics. We add the following characteristics about each output and corresponding transaction that may help the classifier differentiate between distinct types of transactions, or wallets.

- Ratio of output's value to total transaction value
- Output index
- Total transaction value
- Transaction fee paid per byte
- Version number
- Non-zero locktime
- SegWit serialization
- Number of inputs
- Time of inclusion (as epochs of 1008 blocks, about one week)

Baseline ROC AUC scores for different classifiers. Initial runs were done for a baseline comparison without hyperparameter tuning. We note that our encoding may not be ideal for some classifiers, specifically for attributes that allow to subdivide behavior between different clients and epochs. This is a major limitation of linear models and one of the primary reasons we choose a random forest model, as it is able to split the data set along those attributes.

- Logistic regression (l2 penalty): 0.9933
- Support Vector Machine (linear kernel): 0.9931
- Adaboost: 0.9926
- Random forest: 0.9982

Hyperparameter tuning for the random forest classifier. Our hyperparameter grid search returns the following parameters:

- All heuristics / full model
  - `max_features: 7`
  - `min_samples_leaf: 10`
  - `min_samples_split: 20`
- Universal heuristics only
  - `max_features: 6`
  - `min_samples_leaf: 10`
  - `min_samples_split: 20`

F Change in transaction volume between darknet markets and exchanges

Table 5. Change in outgoing transaction volumes of darknet markets (most of which were active between 2013 and 2016) to exchanges using the base clustering (before) and our enhanced clustering (after).

	Volume (BTC)		
Tag name	Before	After	Change (%)
abraxasmarket	21 925	23 368	6.58
agoramarket	158 360	170 970	7.96
alphabaymarket	35 496	41 573	17.12
babylonmarket	222	283	27.13
blackbankmarket	8292	9245	11.49
blueskymarket	2520	3333	32.3
cannabisroadmarket	6	7	25.15
doctordmarket	224	277	23.92
evolutionmarket	49 891	84 637	69.64
middleearthmarket	11 793	12 021	1.93
nucleusmarket	45 265	47 006	3.85
pandoraopenmarket	8708	9461	8.64
sheepmarket	12 104	13 309	9.96
silkroad2market	47 292	49 559	4.79
silkroadmarket	421 741	472 282	11.98
Total	823 839	937 330	13.78