Litecoin Cryptocurrency Address Classification

Krish Suchak

William Ansehl

Rahul Rangwani

Michael Carrion

1. Introduction

a. Background

Cryptocurrency is a global phenomenon which has received significant attention in recent years. Unlike typical currencies, cryptocurrency has been lauded for its absence of centralised control and assumed high degree of anonymity [1]. Central to most cryptocurrencies is the concept of the “blockchain”, or the record-keeping technology underlying such cryptocurrencies as Bitcoin, Litecoin, and Ethereum. A blockchain is an immutable ledger that sequentially records a series of “blocks.” Within each block exists a series of transactions. Each transaction contains certain attributes (structured data) like transaction value (in satoshis), fees (in satoshis), size (in bytes), is_coinbase (boolean value reflecting whether or not the transaction is a block reward - what a miner receives for contributing hashpower to secure the network), the input addresses (sender), output addresses (recipient), and more [2].

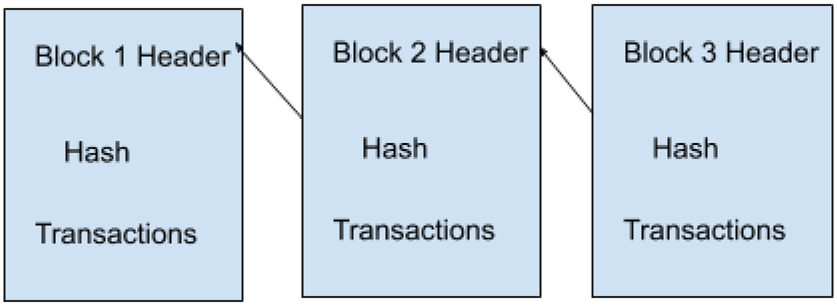


Figure 1: Blockchain blocks connected in a linked list

Due to the speculative nature of cryptocurrency, different types of institutions or entities have cropped up. Namely, the entity types (classes) we will focus on for the purpose of classification of Litecoin input addresses are: exchanges, gambling sites, mining pools, and other (none of the above) (Table 1).

SERVICE	DESCRIPTION
Exchange	Platform where currencies can be bought and sold
Gambling	Online gambling sites (virtual dice and card games)
Mining Pool	Collective group of miners who share computational resources to discover blocks
Other	Independent users, smaller mining pools, etc.

Table 1: Cryptocurrency classification groups

Using techniques from two different types of literature (that which focuses on cryptocurrency address clustering and that which focuses on feature construction), we wish to create a multi-class classifier that can predict what kind of entity to which an input address belongs [3, 4].

In order to identify these clusters, we used a heuristic that was defined in “A Fistful of Bitcoin” which allowed us to develop a script that could cluster the addresses into related groups [1]. It’s important to note, however, that beyond multi-class classification (i.e. determining whether an address belongs to an exchange, a mining pool, etc.), there is currently no way of unearthing the specific *identity* of the entity in question beyond the biggest exchanges and pools, due to their wallet addresses being publicly known [5]. Our project seeks to extend these findings to the world of Litecoin. Namely, we sought to develop a model capable of identifying a given user’s service or purpose by leveraging their transaction history. In this study, we decided to focus on classification into the four groups listed above: Exchange, Gambling, Mining Pool, and Other.

2. Motivation

a. Address Clustering Research

In “A Fistful of Bitcoins,” Sarah Meiklejohn and her colleagues discuss a heuristic of determining which input addresses (across different transactions) actually belong to the same user [1]. This is because input addresses are computed from a user’s private key (or password). For multiple users (like members of a single organization) to have access to the same address would imply that all parties involved are sharing passwords (a caveat that Meiklejohn acknowledges). The simple way to understand this concept is the example in the paper: Imagine 2 input addresses A and B contribute to a transaction. Later in the blockchain, we find a transaction with two input addresses: B and C. Since B is common to both transactions and one must possess the corresponding wallet’s private key to send a valid transaction from a certain address, we can reason that all three input addresses A, B, and C all belong to the same individual / entity or “cluster.” This means that if we can label the entity type for A, then we can label every other input address in A’s cluster (B, C, etc). Note that for our non-ML purposes, a “cluster” corresponds to a user or entity (like Coinbase exchange) not an entity type or class (like exchanges in general).

b. Entity Classification Research

Previous efforts on entity classification (namely Google’s Kaggle notebook to showcase their BigQuery product) have focused on binary classifications such as mining pool vs not mining pool with features like total value of all transactions where a given address is the input address (sender) [6].

c. Improvements

We wish to use this address clustering heuristic to label our unlabeled input addresses (into EXCHANGE, MINING POOL, GAMBLING, and OTHER classes) and to improve on features from previous model by including new features like *average fee rate* (satoshi / byte) across all transactions where a given address is an input address of the transactions.

3. Empirical Strategy

a. Methodologies

Data Collection

The Litecoin ledger is publicly available via Google's BigQuery. We queried the data using a series of SQL queries. The goal was to look at enough data such that we would be able to observe a significant amount of clusters of addresses, but not so much that we couldn't computationally process the data efficiently. The query was for a week's worth of data, resulting in approximately one million rows of data.

Data Cleaning

The transaction table was then split into two tables based on input vs output transactions, which allowed for input or output specific feature construction. Then, the tables were left merged together on the input address. This action was done to connect any input addresses with output addresses to create a more clear transaction roadmap and identify addresses with high volumes of activity. To clean the data, we filled any NA values with 0 and converted a constructed feature total_is_coinbase to an integer.

Feature Construction

To construct our features, we considered elements of cryptocurrency transactions that may help differentiate types of users. One such feature is fee rate, or the total fee divided by the value of the transaction. This feature is informative because certain exchanges offer different fees for different sized transactions, and can be identified by their fee rates. Other such features were constructed as well to help create a more informative model.

Feature Labeling

To take advantage of the address clustering heuristic detailed above, we first group by individual transactions (as there are multiple rows of data per transaction) and consolidate all the input addresses of each transaction in a series of sets. Intuitively, we would iterate the list of sets (preliminary clusters) and if any two sets intersect (at least one input address is common to both), then we consolidate the two sets / clusters (take the union). In practice, we use a graph optimization library (networkx) to treat each input address as a vertex and form edges between every combination of pairs of vertices in a cluster. After we add all the preliminary clusters to the graph, we can simply take the list of connected components to be the final clusters (fully connected vertices).

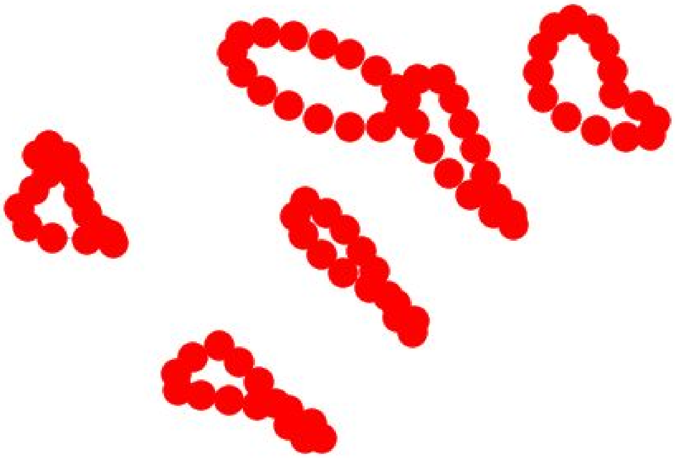
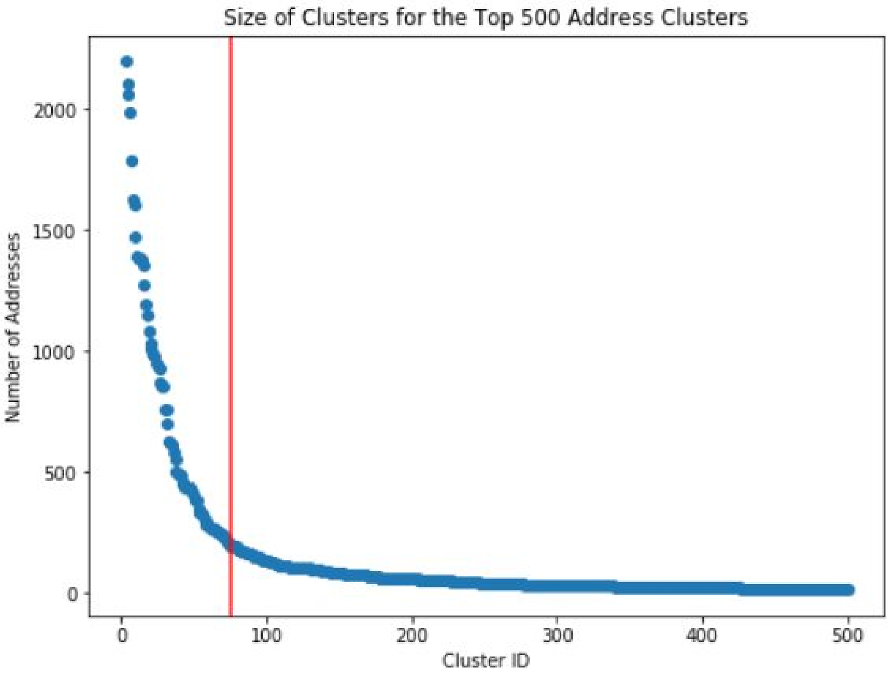


Figure 2: Address clusters, with some clusters overlapping

Next, we plot the largest clusters by size (number of input addresses) and estimate the number of clusters we need to label in order to sufficiently label (presumably) institutional clusters. The elbow of the plot falls around the 75th largest cluster. If we label the top 75 clusters, that accounts for 30% of the input labels. So, how do we label the data exactly? We plug in a single address from a given cluster into https://chainz.cryptoid.info/ltc/ (which has already implemented the heuristic and has labels as well) and note the result (look for a highlighted entity - for example querying the address La8sJjnSNRF6qzTWTwdNKZUYgjmdb7qnoS yields the gambling site FortuneJack). We can label every address in the FortuneJack cluster now (over 8000 addresses out of 200,000+ addresses in the whole dataset) with that single lookup. We perform 75-100 such lookups manually, and are able to successfully label 56 such clusters (18.5% of the dataset). (The alternative is writing a web scraper with BeautifulSoup, but we would still need to manually verify FortuneJack is a gambling site as opposed to an exchange.)



```
The top 75 clusters (in terms of size) contain 29.62% of the addresses.
```

Figure 3: Cluster sizes, with the red line separating the top 75 clusters from the rest

To choose the best model, we first considered a few different models, which were suggested to work well in this kind of prediction [7] according to past research. Once each model was run, we analyzed the results to see which model performed best and the conclusion was Random Forest.

b. The Data

As mentioned in section (a), the dataset analyzed in this project is queried from Google BigQuery, a service that enables interactive analysis of large datasets. The dataset can be viewed on Kaggle [7]. The dataset is originally divided into 4 main tables:

1. Blocks: Contains 13 columns with information pertaining to the blocks that comprise the blockchain. Features consist of a Block Hash, Block Size, Timestamp and Transaction Count among others.
2. Inputs: Contains 14 columns with information pertaining to the source of transactions that comprise a block. Features include a Transaction Hash, Block Hash, Block Timestamp and addresses among others.
3. Outputs: Contains 11 columns with information pertaining to the receiving address of transactions that comprise a block. Features include a Transaction Hash, Block Hash, Block Timestamp and addresses among others.
4. Transactions: Contains 17 columns with information pertaining to each transaction as they comprise the block. Features include Transaction Hash, Transaction Size, Block Hash, Block Timestamp, input_count, output_count among others.

This project utilized only the Transactions table, which is a combination of information pertaining to both the Inputs and Outputs tables, respectively.

c. Preprocessing

The quantity of available data on litecoin transactions is immense. This analysis limited the examined timespan of transactions to the week of January 22-29, 2019. This decision was simply due to the limitations on processing power and storage of our home laptops. This timespan filter resulted in a table with 1,167,611 rows of transactions. However, some of these rows detailed transactions tied to the same address. The original dataset downloaded from the Google

BigQuery Notebook was a csv file Since transactions are a form of structured data and the table is an atomic database, preprocessing simply involved filling missing numerical values with zeros.

d. Feature Construction

Following past literature [1, 3], we sought to construct a series of features that were relevant to the identification and classification of litecoin entities. These features are as follows:

Feature	**Definition**
total_fee_in	The total fee paid for all transactions for a unique address conditioned on that unique address appearing as an input address on the transactions
avg_feerate_in	The average fee rate (summation of fees paid by a unique address / summation of the values of transactions tied to that unique address), conditioned on that unique address appearing as an input address on the transactions
total_tx_inputs_val	The summation of the values of all transactions tied to a unique address, conditioned on that unique address appearing as an input address on the transactions
total_tx_input_count	The total number of addresses across all transactions for which a unique address appears in, conditioned on that unique address appearing as an input address on the transactions
total_input_tx	The total count of transactions for which a unique address appears in, conditioned on that unique address appearing as an input address on the transactions
total_is_coinbase_in	The total number of coinbase transactions for a unique address, conditioned on that unique address appearing as an input address on that transactions

** Note: For each feature, there is a second feature for which the metric is conditioned on the unique address appearing as an output address. For example, total_fee_out is the total fee paid for all transactions for a unique address, conditioned on that unique address appearing as an output address on the transactions. As a result, we constructed a total of 12 features to feed into our supervised machine learning algorithms later on.

4. Analysis & Results

For the analysis of Litecoin data, we used the following supervised machine learning algorithms, which have been suggested in past literature [7]:

- K-Nearest Neighbors
    - Classifies an observation as belonging to the majority class of the k most similar observations
- Decision Tree
    - Decision Trees are easier to interpret than more complex models such as Neural Networks
- Random Forest
    - Random forests have the power to handle large datasets with high dimensionality without overfitting. Suitable for classification problems. Typically provides higher accuracy scores than decision trees.
- Neural Network
    - Multilayer perceptron neural network: the “traditional” type of neural network. A MLP Neural network is suitable because the training set is assigned an entity label/classification prior to passing through the input layer.
- Adaptive Boosting

- AdaBoost is great as a starting point among boosting algorithms to implement in classification problems. This project utilized AdaBoost on the decision tree algorithm.

- Naive Bayes
  - A simple and commonly used algorithm to acquire the base accuracy of the dataset

The model accuracies and corresponding run-times can be visualized below (Figures 4 and 5). 5-fold cross-validation was performed on the top three performing models to ensure our models were able to generalize to other test sets.

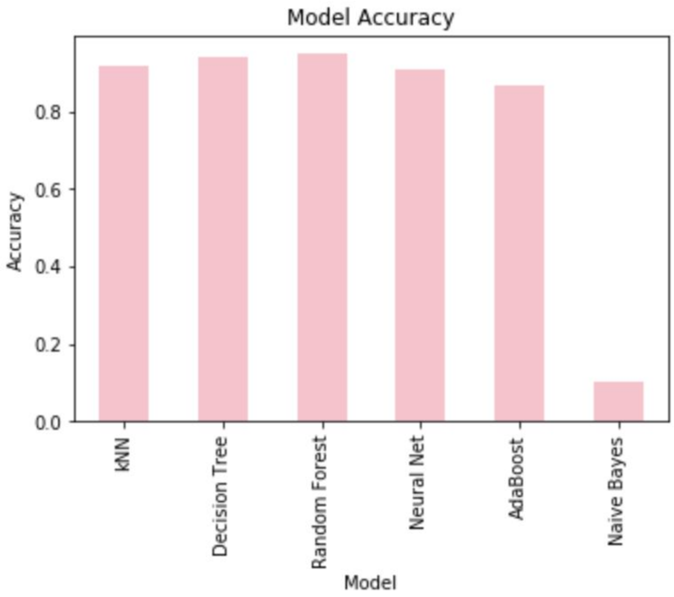


Figure 4: Model accuracy

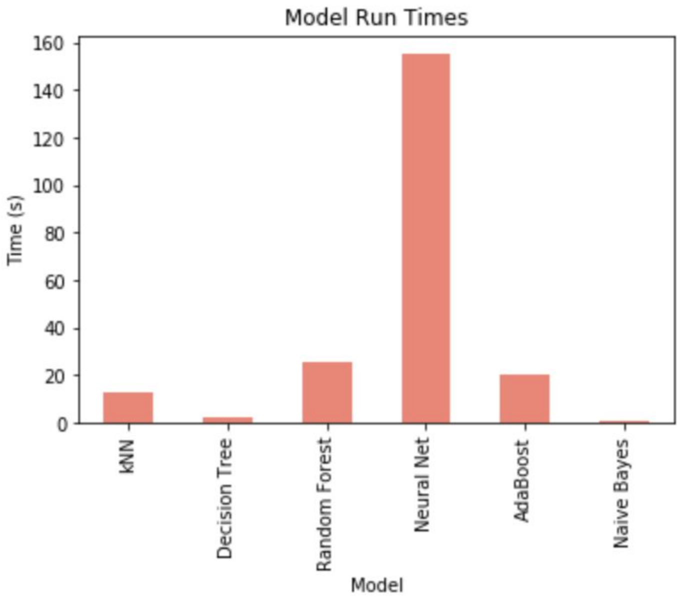


Figure 5: Model run times

As can be seen above, the Random Forest model performed the best, optimally classifying 95% of the observations in approximately 30 seconds. With the exception of Naive Bayes, all models performed very well, achieving about about 90% accuracy. Notably, the Neural Net took significantly longer than the other models, likely a result of its complexity.

We next assessed the confusion matrix of our optimal Random Forest model, to assess the distribution of misclassifications. As can be seen below, most misclassifications were observations which our model incorrectly classified as "Other." This can likely be attributed to the large proportion of "Other" observations in relation to the other classes (e.g. ~80% of our training set belonged to the "Other" class). Thus, the "Other" observations likely spanned a larger range of input counts, fee rates, transaction values, etc., and, conceptually, when the model encountered an observation that didn't fully coincide with another class, it would "default" to the "Other" classification.

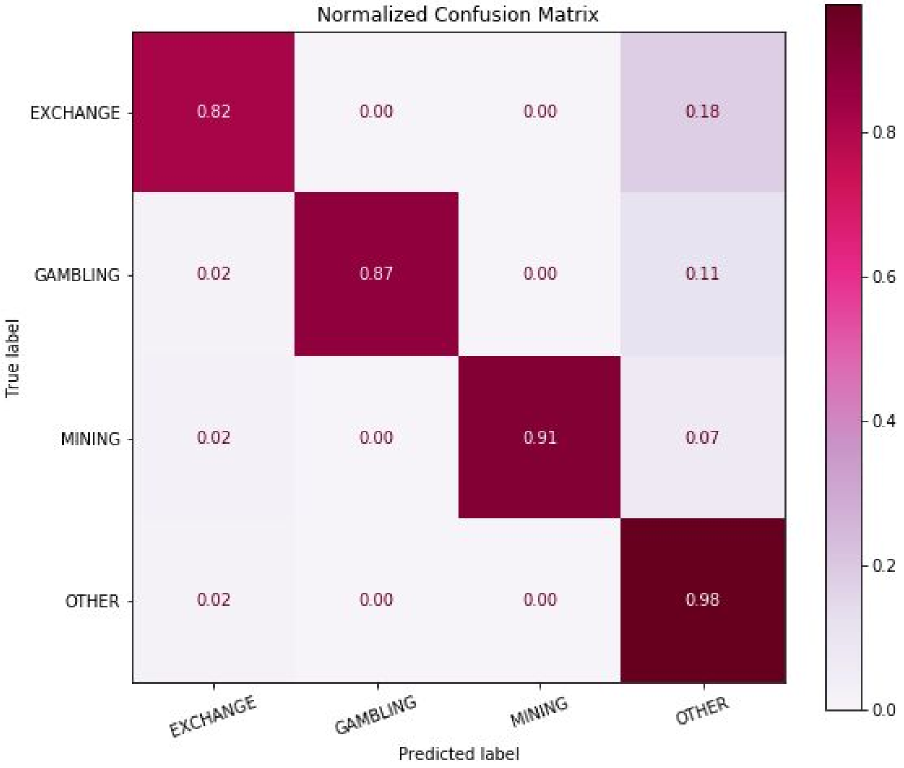


Figure 6: Confusion Matrix for Random Forest Model

Further, 10 of the 12 features were found to contribute to entity classification in our optimal Random Forest model. Most notably, total input fee, average fee rate, and total transaction value played the top three most significant contributions to entity classification. Intuitively, this makes sense, as most of the larger entities examined in this study transact much more than a single individual or a smaller entity such as a faucet or investment program, most of whom comprise the “Other” category. Thus, after aggregating across all transactions for a given exchange, the transaction values and total input fees are likely to be much larger than those of a mining pool or gambling site, which is likely to total more fees and input values than an average user.

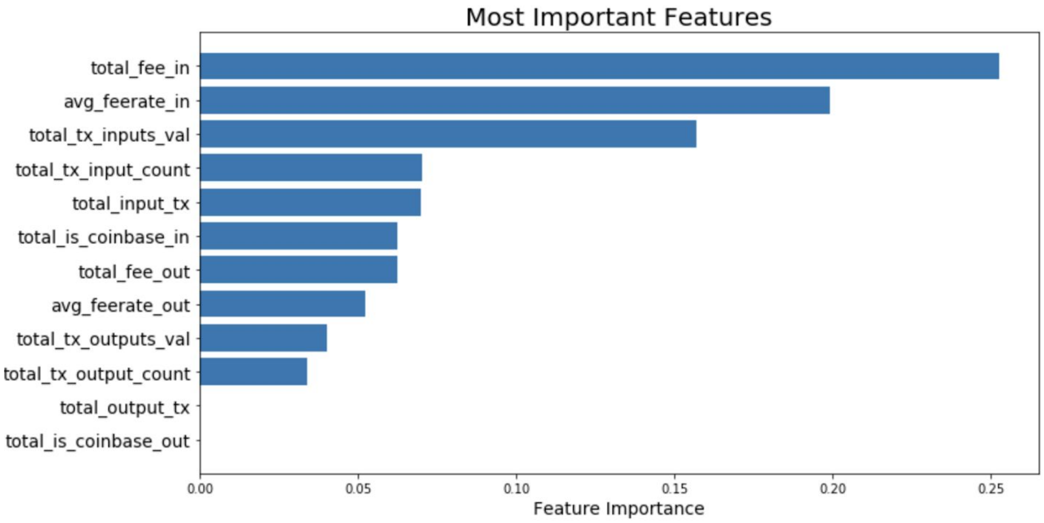


Figure 7: Feature Importance for Random Forest Model

5. Conclusion

a. Applications

1. The blockchain is not as anonymous as people commonly believe. Given a user’s transaction history, we’ve developed a model capable of classifying an address as belonging to an exchange, gambling site, mining pool, or other. However, it’s important to note that individual identification remains impossible.

2. With the addition of more entities, especially those known for illicit activities such as "Dark Nets," our findings can be extended to flag suspicious and possibly illegal activity to aid in law enforcement and prevent cybercrimes such as theft.
3. As an extension of past research, our model is relatively unique in that it maintains the high accuracy score associated with past binary classification efforts [8], while allowing for multiclass classification. A review of past literature suggests a (perhaps unsurprising) tradeoff between model accuracy and the number of attempted classes. For example, [5] developed a model capable of classifying 7 entities with an accuracy of about 77%. Our model and classification method provides a promising result in the realm of cryptocurrency address classification.

b. Limitations

1. Entity Stratification: By sourcing entity types ourselves using https://chainz.cryptoid.info/btc/, it was difficult to label enough data to accurately classify enough of our dataset. Most of the clusters were conducted by unknown entities (that is, entities not labeled in the website above), and, as such, approximately 80% of our dataset was labeled "Other". As displayed in the confusion matrix above, this unbalanced dataset had minor implications in our model, specifically with regard to misclassified observations. In the future, it would be interesting to scavenge other websites/resources to accurately label a larger proportion of the dataset.
2. Data Acquisition: Initially, we were planning on working with Bitcoin; however, querying one day of Bitcoin transactions yielded a 1.5 GB file, which was too large to process efficiently. As such, we instead queried one week's worth of Litecoin transactions, yielding a more manageably 0.75 GB file. However, because we used only one week's worth of data, it's possible that the queried transactions are not representative of the transactions

certain entities usually partake in (i.e. certain entities may be misrepresented in this short time frame). For example, though we ensured weekly volume and Litecoin price were relatively stable during the week in question, it's possible that the week represented an anomaly in trading for a certain exchange or individual.

c. Future Investigations

1. Additional Entities: While our model was successful in classifying an observation as belonging to "Exchange", "Gambling", "Mining Pool", or "Other," it would be interesting to see if we could extend our findings to additional entities as well. For example, future investigations should seek to further classify the "Other" category into investment sites, faucets, and average users, to see if these platforms too are not as anonymous as once believed.
2. Additional Features: In order to discern between additional entities (especially less distinct entities as mentioned above), a more nuanced model may be required. If so, it would be interesting to examine temporal features, such as the time of transaction, duration between transactions, etc.
3. Additional Models: Future work should investigate additional models, especially when considering the different entities mentioned above. For example, though we had considered Adaptive Boosting, previous literature also suggests looking into other boosting methods, namely stochastic gradient boosting, which tend to be simpler, more efficient, and more accurate.
4. Hyperparameter Tuning: While cross validation was performed to ensure generalizability of our results, many models examined have hyperparameters which can be tuned for increased classification accuracy. Especially when considering more nuanced entities such as investment platforms and faucets, hyperparameter tuning may need to be

performed. Although the random forest model performed quite well, tuning hyperparameters may significantly reduce the number of false positives that occured. For example, as can be seen in the confusion matrix above, 18% of “Exchange” observations were incorrectly classified as “Other”; Hyperparameter tuning might lower this value.

6. Contribution Statement

Krish - Responsible for getting the data, suggesting features, labeling the “address clusters,” model selection, k-fold cross validation, attempting k-means clustering, most of the visualizations (except model accuracy and runtime), and final report writeup.
Michael - Responsible for model construction, several visualizations, powerpoint development, and final report writeup.
Will - Responsible for data cleaning, feature research, feature construction/engineering, model construction/selection, presentation creation, final report writeup, interpreting results.
Rahul - Responsible for data processing and model selection. Rahul was responsible for writing scripts that helped in consolidating the clusters such that any overlapping clusters are combined, for setting up and running each model and interpreting the results, and final report writeup.

7. References

[1] Meiklejohn, Sarah, et al. *A Fistful of Bitcoins: Characterizing Payments among Men with No Names*. University of California, San Diego & George Mason University, Oct. 2013, www.researchgate.net/publication/262357109_A_fistful_of_bitcoins_characterizing_payments_among_men_with_no_names.

[2] Day, Allen, et al. “Introducing Six New Cryptocurrencies in BigQuery Public Datasets-and How to Analyze Them | Google Cloud Blog.” *Google*, Google, 5 Feb. 2019, cloud.google.com/blog/products/data-analytics/introducing-six-new-cryptocurrencies-in-bigquery-public-datasets-and-how-to-analyze-them.

[3] Lin, Yu-Jing, et al. *An Evaluation of Bitcoin Address Classificationbased on Transaction History Summarization*. Department of Computer Science, National Taiwan University & Institute of Statistical Science, Academia Sinica, 19 Mar. 2019, arxiv.org/pdf/1903.07994.pdf.

[4] Toyoda, Kentaroh, et al. *Multi-Class Bitcoin-Enabled Service Identification Based on Transaction History Summarization*. Dept. of Information and Computer Science, Keio University & National and Kapodistrian, University of Athens, July 2018, www.researchgate.net/publication/333599819_Multi-Class_Bitcoin-Enabled_Service_Identification_Based_on_Transaction_History_Summarization.

[5] Harlev, Mikkel Alexander, et al. *Breaking Bad: De-Anonymising Entity Types on the Bitcoin Blockchain Using Supervised Machine Learning*. Centre for Business Data Analytics, Copenhagen Business School & Westerdals Oslo School of Arts, Comm & Tech, 2018, core.ac.uk/download/pdf/143481278.pdf.

[6] Price, Will. “Bitcoin Mining Pool Classifier.” *Kaggle*, Kaggle, 31 Jan. 2019, www.kaggle.com/wprice/bitcoin-mining-pool-classifier.

[7] Google. “Bitcoin Blockchain.” *Kaggle*, 12 Feb. 2019, www.kaggle.com/bigquery/bitcoin-blockchain.

[8] K. Toyoda, T. Ohtsuki, and P. T. Mathiopoulos, “Identification of High Yielding Investment Programs in Bitcoin via Transactions Pattern Analysis,” in Proc. of Global Communications Conference (GLOBECOM). IEEE, 2017.

8. Appendix

- BigQuery SQL Query

```
SELECT
        `hash` as tx_hash,
        size as tx_size,
        virtual_size,
        version,
        block_hash,
        block_timestamp,
        input_count,
        input_value,
        output_count,
        output_value,
        is_coinbase,
        fee,
        inputs.spent_transaction_hash as `inputs_spent_transaction_hash`,
        inputs.required_signatures as `inputs_required_signatures`,
        inputs.type as `inputs_type`,
        inputs.addresses[OFFSET(0)] as `inputs_addresses`,
        inputs.value as `inputs_value`,
        outputs.required_signatures as `outputs_required_signatures`,
        outputs.type as `outputs_type`,
        outputs.addresses[OFFSET(0)] as `outputs_addresses`,
        outputs.value as `outputs_value`
FROM `bigquery-public-data.crypto_litecoin.transactions` txs
LEFT JOIN UNNEST (inputs) AS inputs
LEFT JOIN UNNEST (outputs) AS outputs
WHERE
         EXTRACT(YEAR FROM block_timestamp) = 2019 AND
         EXTRACT(MONTH FROM block_timestamp) = 01 AND
         EXTRACT(WEEK(TUESDAY) FROM block_timestamp) = 4;
```