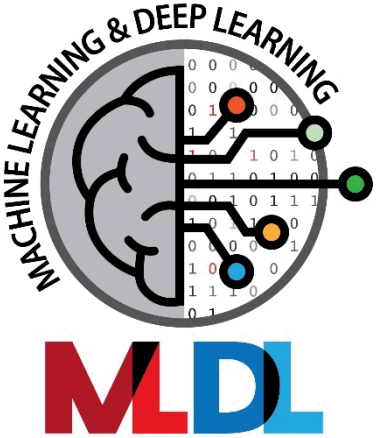
SAND2017-8407C



MLDL

Machine Learning and Deep Learning Conference 2017

Bitcoin Address Classification

Lynne Burks (8716)

Andrew Cox (8716), Kiran Lakkaraju (1463),

Mark Boyd (8762), and Ethan Chan (8754)

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc. for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Outline

- Background & Motivation
- Classification
  - Collect data
  - Compute features
  - Apply supervised learning classification
- Application for Law Enforcement
- Conclusions

What is Bitcoin?

- A digital currency and underlying software created in 2009 by Satoshi Nakamoto (probably not a single person)
- Decentralized operation: Network of users, rather than a bank, verifies validity of transactions

Highly anonymous: Difficult to know who is sending and receiving bitcoins

Challenges for law enforcement

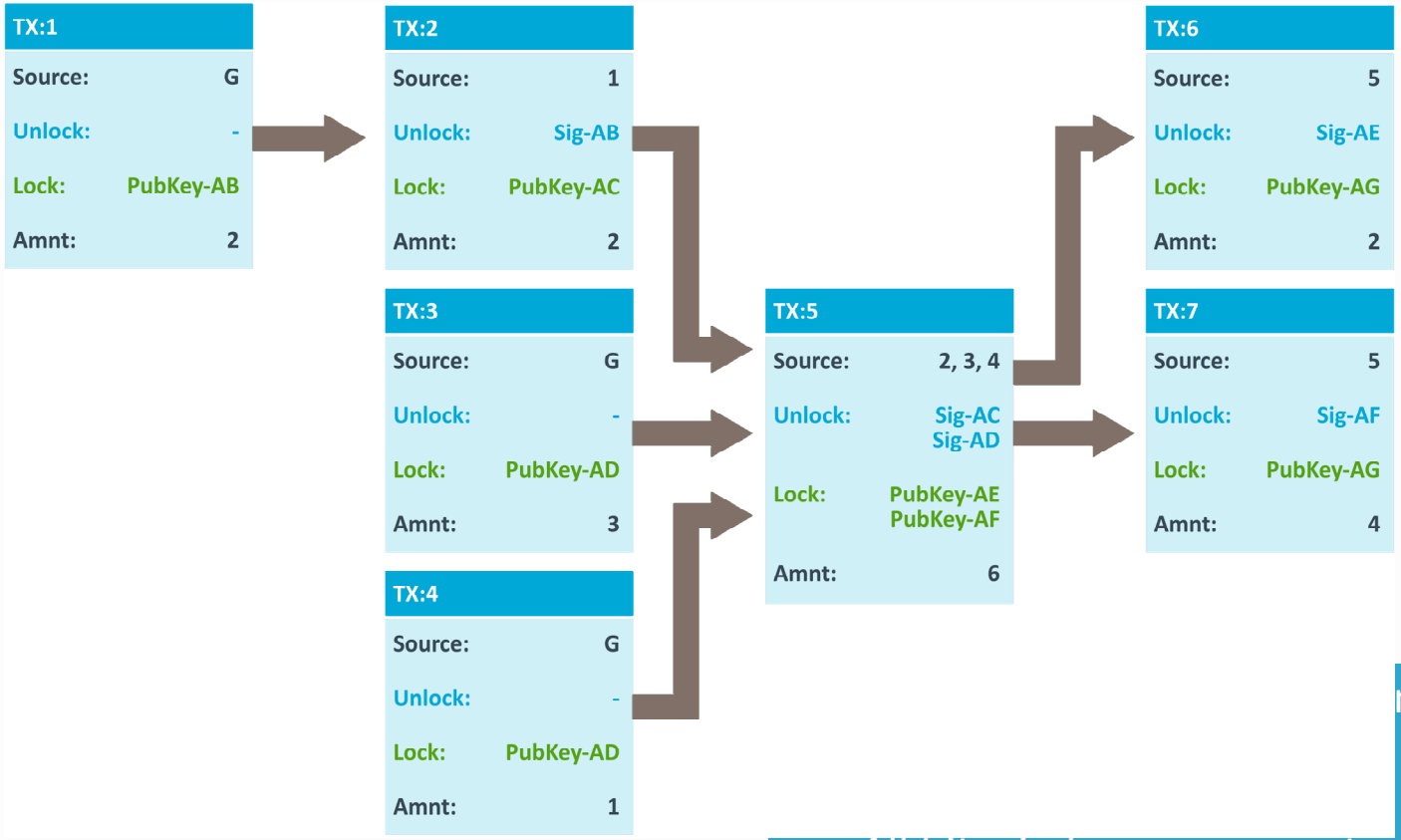
- Significant time and resources needed to identify users—and identification is not always possible
- Bitcoin likely to spawn innovations that will enable new forms of legitimate and illicit commerce
- Authorities have few battle-tested legal, policy, and technical tools to counter illicit uses of Bitcoin

Real-world illicit commerce examples

Activity	Example	Law Enforcement Need	Law Enforcement Potential Tactics
Sale/purchase of an illicit good using bitcoin	• Agora • Haven Marketplace • East India Company • Silk Road • OpenBazaar	• Link a Bitcoin public key to a specific, verifiable identity • Seize bitcoins used in illicit commerce	• Use of undercover agents • Analysis of open webpages • Use of cybercrime investigators to locate servers
Hiding source of funds obtained from illicit commerce and converting them to assets or cash	• Charlie Shrem • Robert Faiella • Pascal Reid • Michel Espinoza	• Track Bitcoin transactions across multiple public keys • Link a Bitcoin public key to a specific, verifiable identity • Link transactions across identities	• Use of undercover agents • Verification of registration of exchange sites with proper authorities

Patterns: Tracking bitcoin flow

Tracking the flow of bitcoins is an important law enforcement tool



must come
(or from

- All bitcoin from a source transaction must be used in related outcome transaction

Background

- Many different classes of entities use Bitcoin, for example:
  - Mixing services, such as Bitcoin Fog and Shared Coin
  - Gambling sites, such as Satoshi Dice and Betcoin
  - Exchanges, such as Coinbase and BTC-e
  - Tor markets, such as Silk Road and Agora Market
- Key question: given a bitcoin address, can we predict what class of entity it belongs to?
- For example, the address *19oztMBBL519s22iYba8BQo28Wnba4m19M* has participated in 2 transactions and sent a total of 0.121 bitcoin. Is this address most likely owned by a mixing service, gambling site, exchange, or tor market?

Motivation

- This work is part of a larger project whose primary goal is entity identification on the Bitcoin blockchain for use by Law Enforcement.
- Classifying Bitcoin addresses by entity class helps achieve this goal in two ways:
    1. If two addresses are classified as different entity classes, then it is unlikely they are owned by the same entity.
    2. Mixers are designed to obfuscate entities, so when an address is classified as a mixer, standard entity identification techniques cannot be applied.

Classification Process

1. Collect data
   - Aggregate a set of bitcoin addresses with known owners.
   - Label each address with the class of entity that owns it, for example, mixing, gambling, exchange, etc.
2. Compute features
   - For each address, compute interesting features about it's behavior.
   - For example, the number of transactions it has participated in, the total amount of bitcoins it has sent, etc.
3. Apply supervised learning classification algorithms
   - Try several algorithms to compare performance, including Support Vector Machines (SVM) and Random Forest.
   - Identify important features.

Collect Data

- Downloaded a labeled dataset from Chainalysis
- Total of 10 classes, 16,651,820 addresses, and 176 entities

Class	Num. Addresses	Num. Entities	Example Entity
Mixing	4,753,891	4	Bitcoin Fog
Exchange	3,698,795	27	Coinbase
Merchant Services	2,335,030	4	BitPay
Unknown	2,016,683	44	1JNoitCVT46D…
Gambling	1,791,795	81	Satoshi Dice
Tor Market	1,443,584	7	Silk Road
Hosted Wallet	257,500	2	Instawallet
Mining Pool	172,455	5	BTTC Pool
Scam	113,280	1	MMMGlobal
Other	68,807	1	BTC Jam

Compute Features

- For each address in our dataset, loop through the blockchain, identify transactions that address has participated in and compute features.
- There are 37 features per address.
  - Number of transactions
  - The maximum bitcoin balance, and total and average sent and received bitcoin
  - Lifetime (in blocks)
  - Total, average, minimum and maximum number of transaction inputs and outputs, when the address is a source and when the address is either a source or destination
  - Total and average number of each output type (e.g., public key, public key hash, multi-signature, script hash, etc.)

Example Features

- Address: 1Lwnrk7bmt4hc4JB7DTbchWwbzcSczBc5Q
- Class: Mixing (Bitcoin Fog)

1. nnulldata_outputs_avg: 0.0
2. maxbtcbalance: 0.06962401
3. max_ninputs_issrc: 4
4. npubkey_outputs_avg: 0.0
5. min_ninputs_issrc: 4
6. min_ninputs_all: 1
7. lifetime: 14,
8. nnulldata_outputs: 0
9. npubkeyhash_outputs: 8
10. noutputs_all: 9
11. nmultisig_outputs: 0
12. ninputs_all_avg: 2.5
13. min_noutputs_issrc: 8
14. ninputs_issrc_avg: 4.0
15. min_noutputs_all: 1
16. noutputs_issrc_avg: 8.0
17. ntrans: 2
18. npubkey_outputs: 0
19. sentbtc_avg: 0.06962401
20. noutputs_all_avg: 4.5
21. npubkeyhash_outputs_avg: 8.0
22. nscripthash_outputs_avg: 0.0
23. ninputs_issrc: 4
24. n_issrc: 1
25. sentbtc: 0.06962401
26. receivedbtc: 0.06962401
27. receivedbtc_avg: 0.06962401
28. nscripthash_outputs: 0
29. max_noutputs_issrc: 8
30. max_ninputs_all: 4
31. noutputs_issrc: 8
32. ninputs_all: 5
33. nnonstandard_outputs: 0
34. n_isdes: 1
35. max_noutputs_all: 8
36. nnonstandard_outputs_avg: 0.0
37. nmultisig_outputs_avg: 0.0

Apply Supervised Learning Classifier

1. Support Vector Machine (SVM)
   - Uses a kernel to transform the data and then optimally separates it based on classes. The kernel can be linear or non-linear, allowing for complex transformations when data is not easily separable.
   - However, results are computationally expensive to compute and often difficult to interpret.
2. Random Forest Classifier
   - Fits a set of decision tree classifiers to the data and uses averaging on the results to improve accuracy and reduce over-fitting.
   - Each decision tree learns simple decision rules based on the gini coefficient (a measure of statistical dispersion) of the features.
   - Results are easier to interpret and require relatively less computation.

SVM Implementation

- Full dataset of over 16M addresses caused performance issues, so built a sample dataset by randomly sampling addresses from each class.
- Normalized features by subtracting mean and dividing by standard deviation.
- Tested multiple kernels and penalty parameters.
- Performed 5-fold cross validation: split sampled dataset into 5 groups, trained on 4 then tested on 1, and repeated 5 times.
- Trained a multi-class and single-class classifier. Multi-class predicts one of 10 classes, where single-class predicts mixer vs. non-mixer.

SVM Results

- Multi-class SVM
  - Sampled 1,000 addresses from each class.
  - The best average accuracy was **40.7%**, with a linear kernel and penalty parameter of 1 (compared to an average accuracy of 10% for random guessing).

- Single-class SVM
  - Sampled 10,000 addresses from each class (mixer vs. non-mixer).
  - The best average accuracy was **95.1%**, with an rbf kernel and penalty parameter of 1 (compared to an average accuracy of 50% for random guessing).

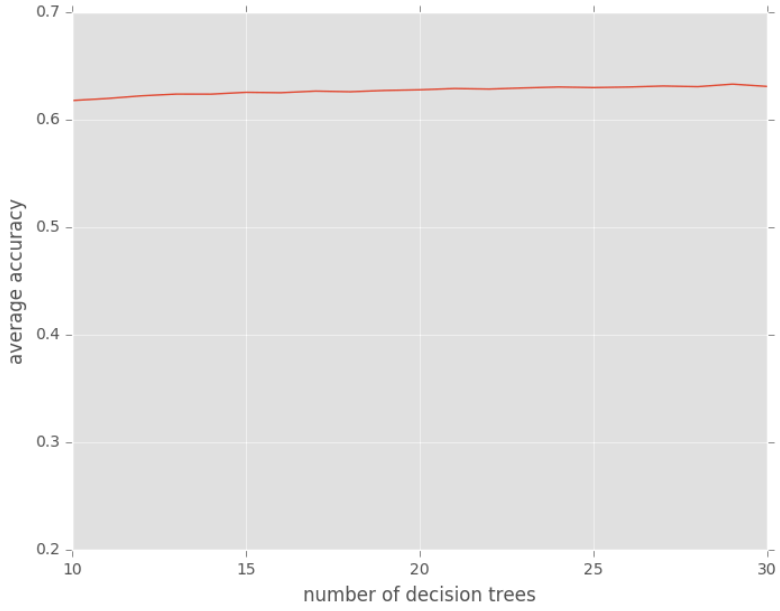
Random Forest Implementation

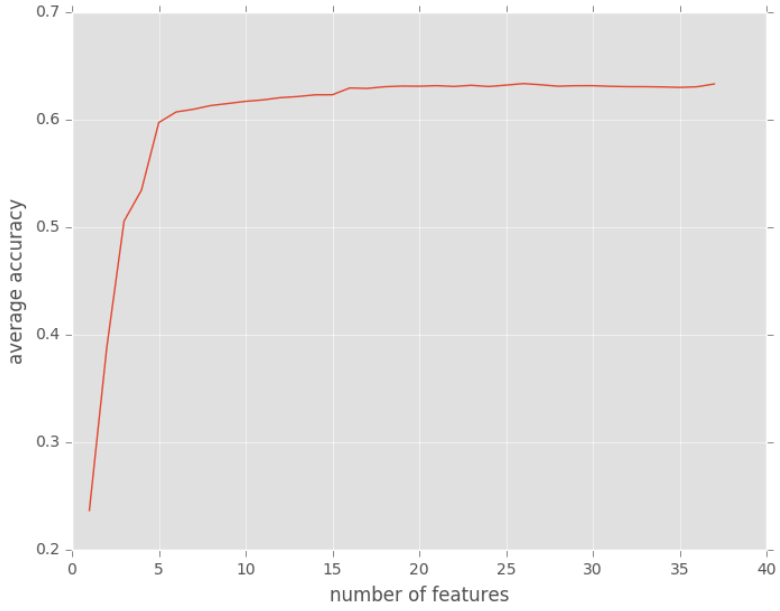
- Full dataset of over 16M addresses caused performance issues, so built a sample dataset by randomly sampling addresses from each class.
- Tested variable number of trees in the random forest.
- Performed 10-fold cross validation: split sampled dataset into 10 groups, trained on 9 then tested on 1, and repeated 10 times.
- Trained a multi-class and single-class classifier. Multi-class predicts one of 10 classes, where single-class predicts mixer vs. non-mixer.
- Identify important features by looking at features at the top of trees.

Random Forest Results

Multi-class Random Forest

- Sampled 10,000 addresses from each class.
- Average accuracy around **65%** (compared to 40% for SVM and 10% for random guessing).
- Most important features: (1) lifetime, (2) average received btc, (3) average sent btc, (4) max btc balance, (5) average number of pubkeyhash outputs

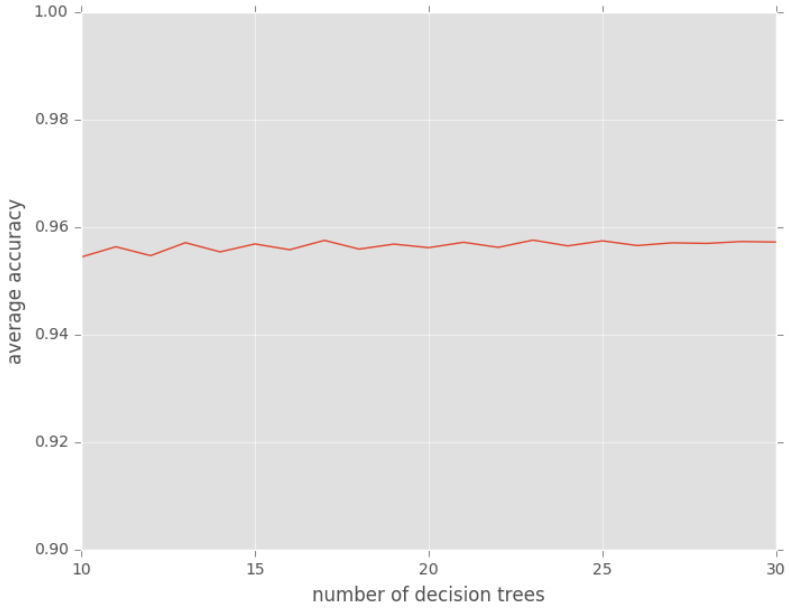


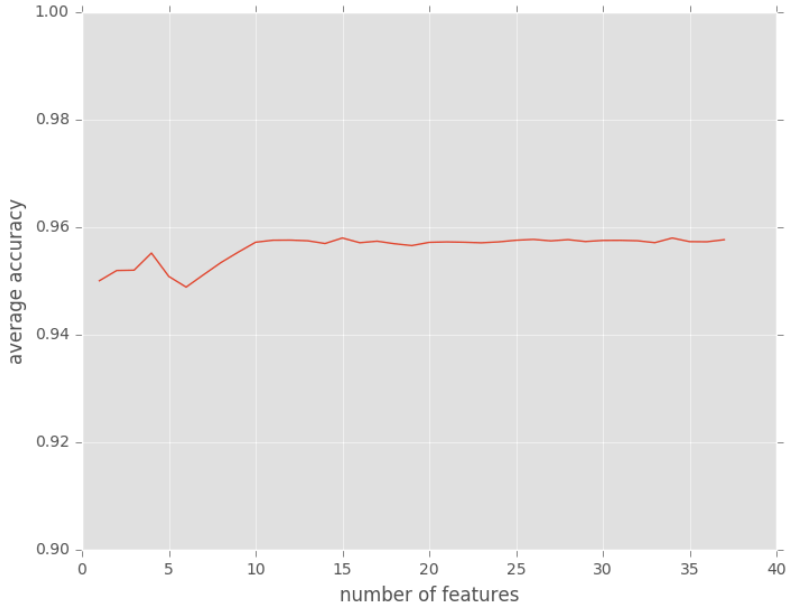


Random Forest Results

Single-class Random Forest

- Sampled 100,000 addresses from each class (mixer vs. non-mixer).
- Average accuracy around **95.7%** (compared to 95.1% for SVM and 50% for random guessing).
- Most important features: (1) max number of outputs, (2) min number of outputs when address is a source, (3) average number of pubkeyhash outputs, (4) max number of outputs when address is a source, (5) total number of pubkeyhash outputs

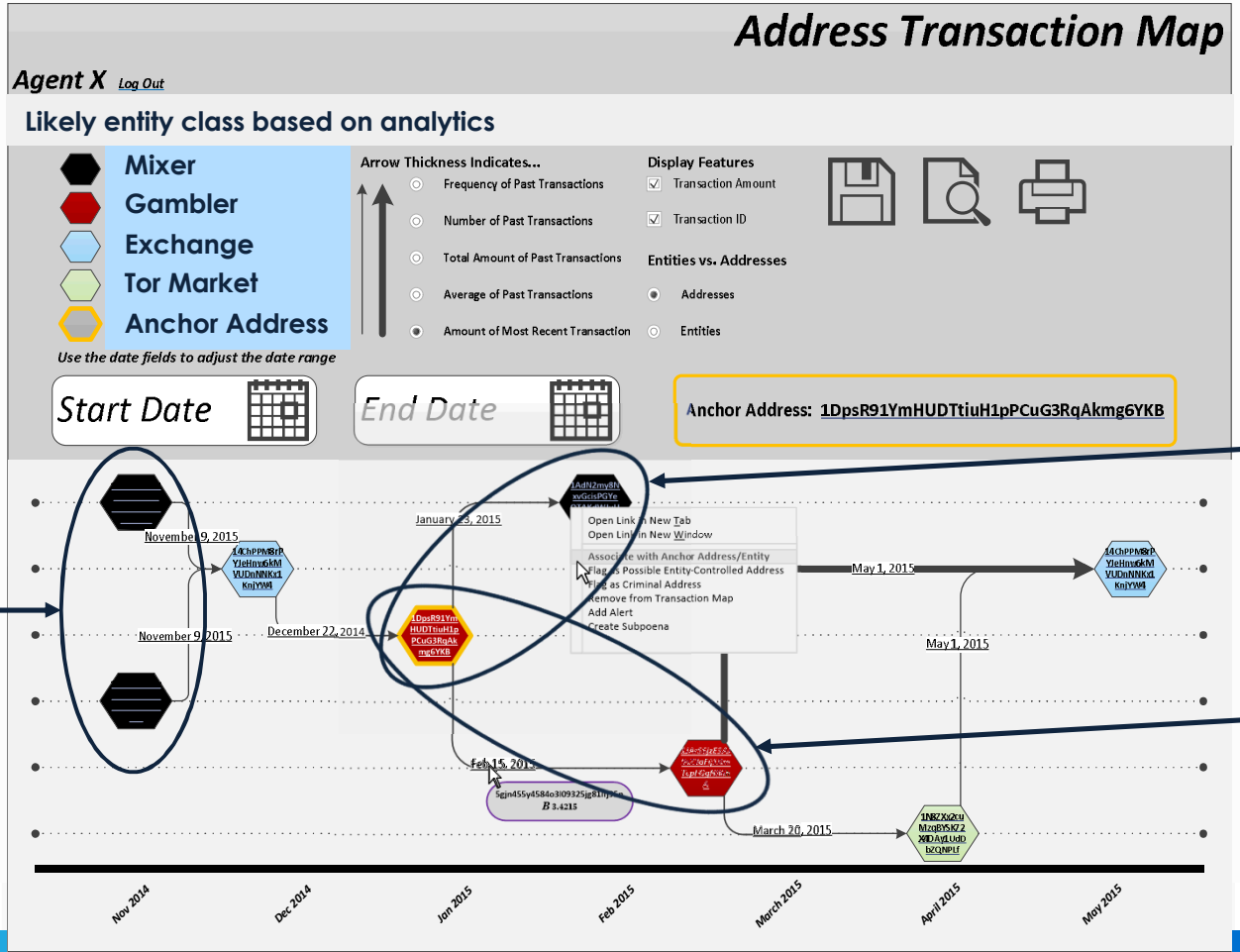




Application for Law Enforcement

This work can be used to classify any bitcoin addresses of interest to law enforcement, and can be visualized in a bitcoin analytics tool (such as the example below).

These addresses are classified as likely mixers, and may therefore not be the same entity since the multi-input heuristic does not apply.



These addresses are classified as a likely gambler and mixer, and are therefore not likely owned by the same entity.

These addresses are classified as likely gamblers, and may therefore be owned by the same entity.

Conclusions

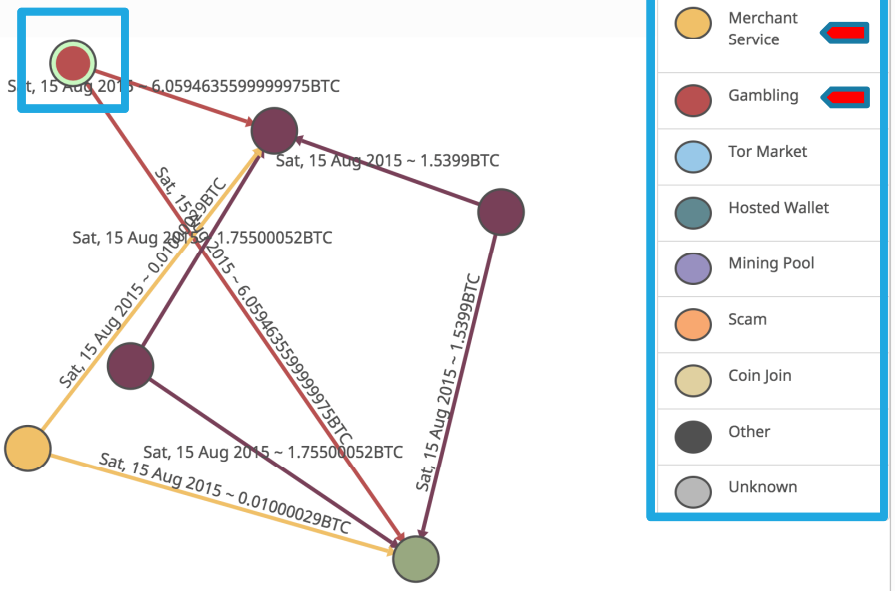
- Multi-class classifiers (both SVM and Random Forest) are significantly more accurate than random guessing.
- The random forest classifier tends to be more accurate than SVM for multiple classes.
- Classifiers can identify mixers with very high accuracy (around 95%). This may be partially due to the Shared Coin mixer, which owns significantly more addresses than other mixers in the data set.
- The number of outputs is important for identifying mixers, and the amount of bitcoin sent and received is important to differentiate classes of entities.

Thank you!

Appendix

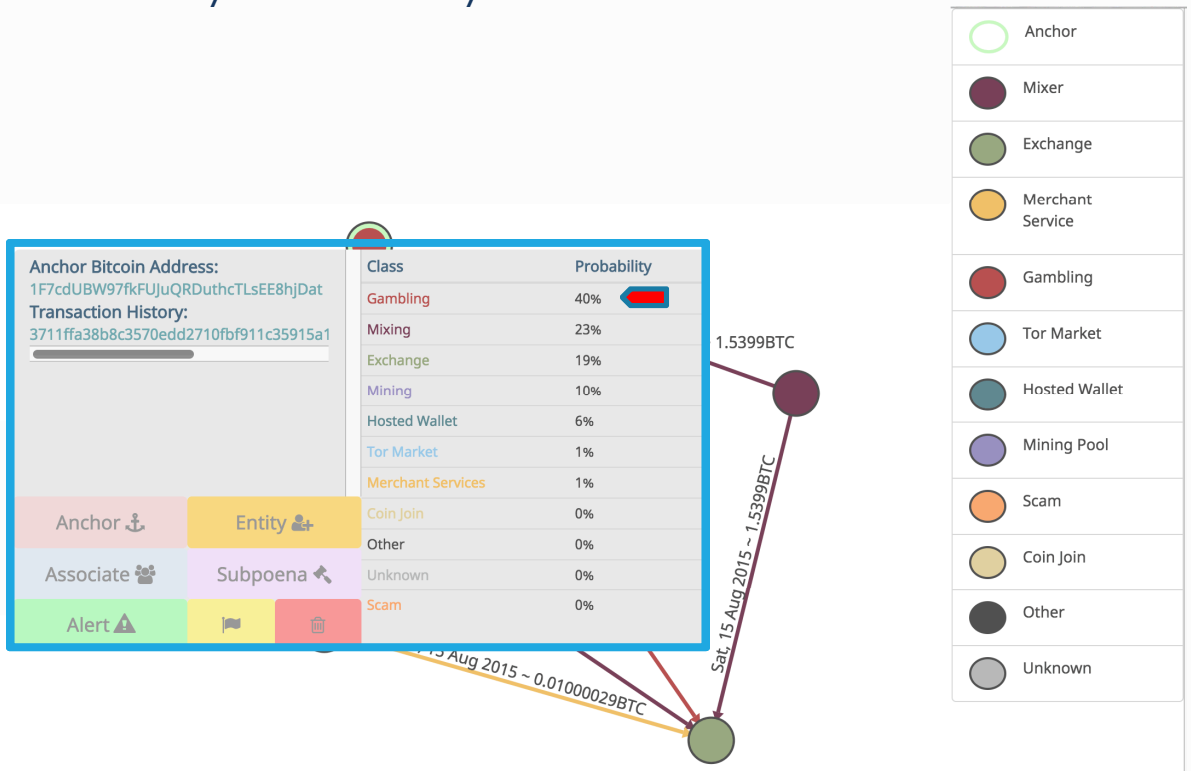
NeoNode: Categorization

- Agent investigating *1F7cdUBW97fkFUJuQRDuthcTLsEE8hjDat* can easily see the most probable categorization
  - Categorization analysis done by custom tool



NeoNode: Categorization

- Agent investigating *1F7cdUBW97fkFUJuQRDuthcTLsEE8hjDat* can easily see the most probable categorization
  - Categorization analysis done by custom tool



Glossary of Terms

- Kernel = function that enables the classification algorithm to operate in a high-dimensional and implicit feature space. Some example kernel functions are:
  - Linear = resulting feature space is defined by a linear combination of the original features.
  - Polynomial = resulting feature space is defined by polynomial functions of the original features.
  - Radial Basis Function (RBF) = resulting feature space is defined by a Gaussian function of the original features.
- Penalty Parameter, C = regularization parameter that defines the penalty for misclassified observations. Large C values can result in over-fitting while small C-values can result in lower accuracy.
- Average Accuracy = fraction of observations in the test set that were classified correctly.

SVM Detailed Results

Multi-class SVM

- Sampled 1,000 addresses from each class.

Average Accuracy

kernel \ C (penalty)	0.3	0.5	0.7	1.0
linear	38.7%	40.0%	40.4%	40.7%
poly	24.5%	24.9%	24.9%	25.4%
rbf	35.2%	36.6%	37.0%	37.6%

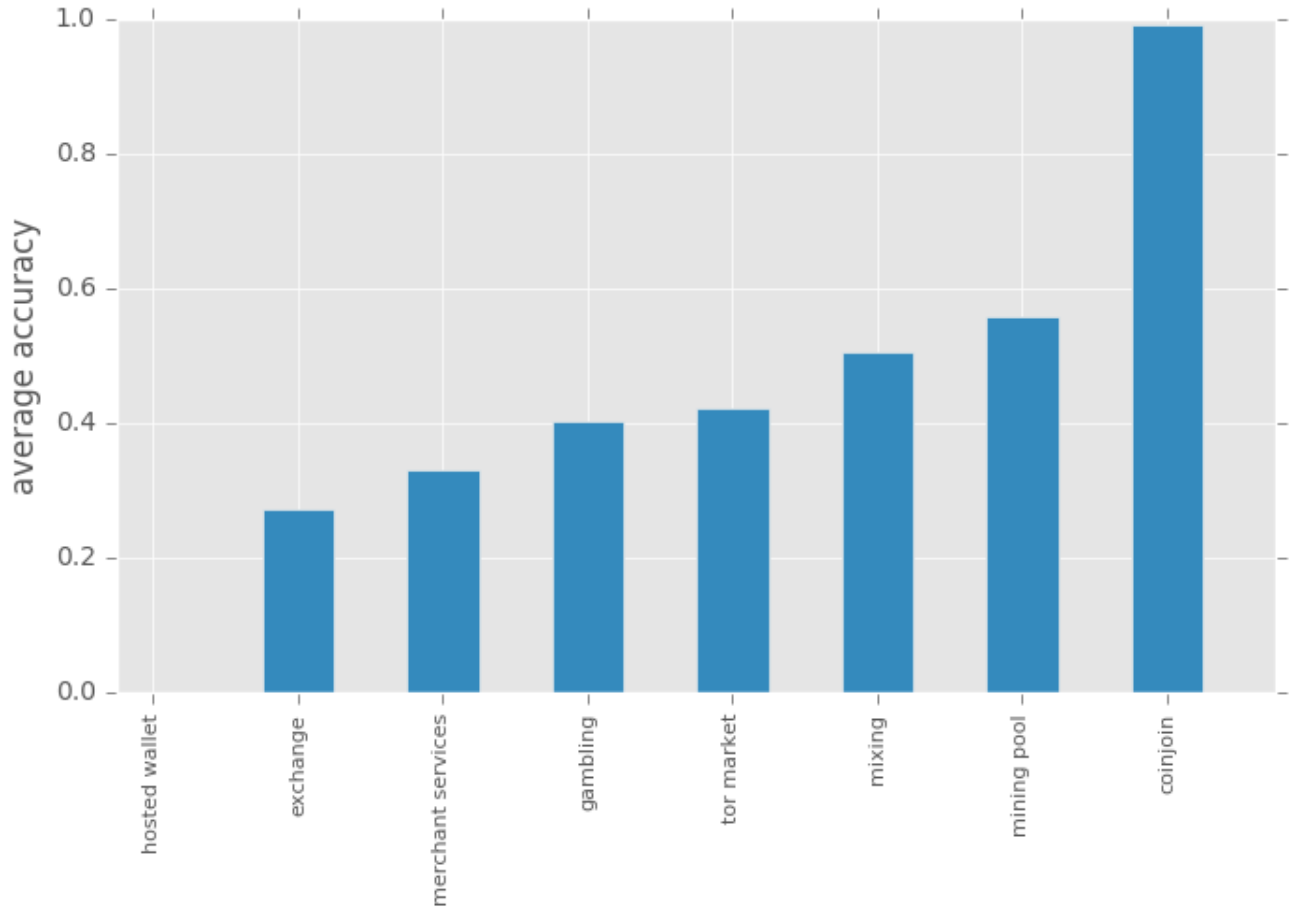
Single-class SVM

- Sampled 10,000 addresses from each class (mixer vs. non-mixer).

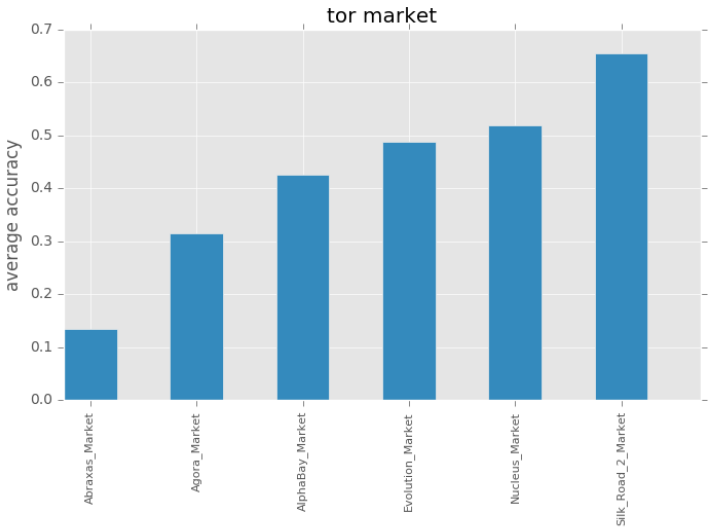
Average Accuracy

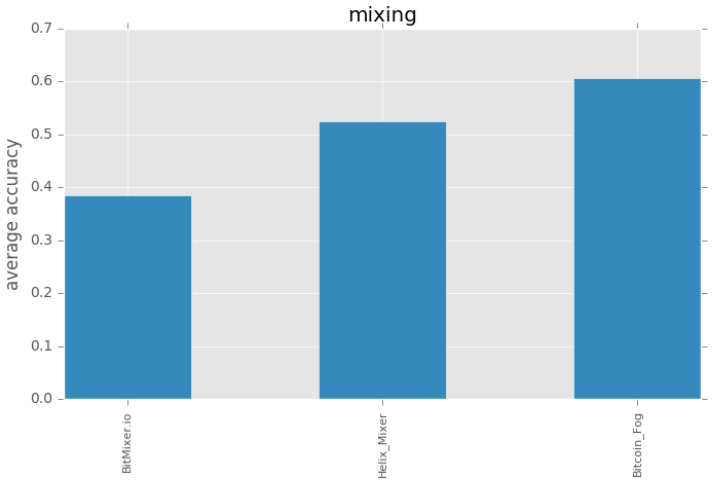
kernel \ C (penalty)	0.3	0.5	0.7	1.0
linear	94.9%	94.9%	94.9%	94.9%
poly	64.1%	68.5%	87.3%	92.5%
rbf	95.1%	95.1%	95.1%	95.1%

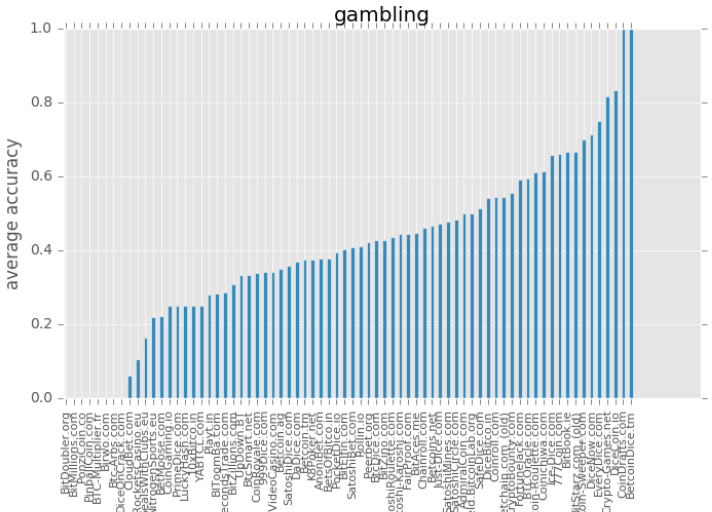
Average Accuracy by Class (RF)

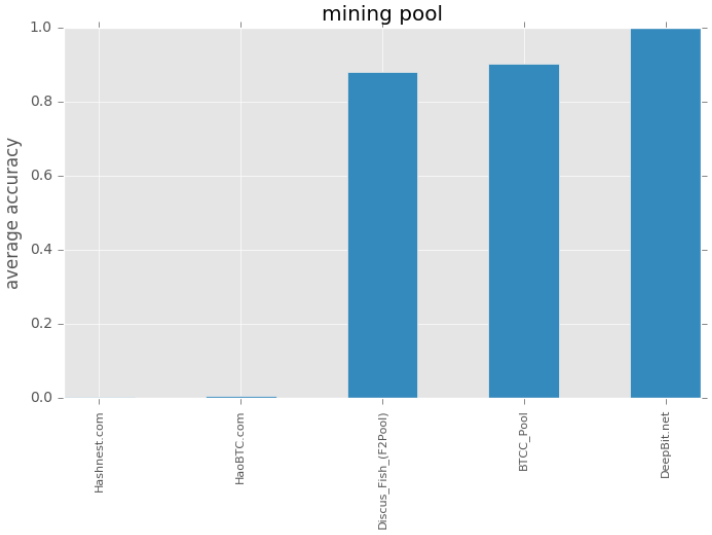


Average Accuracy by Class (RF)

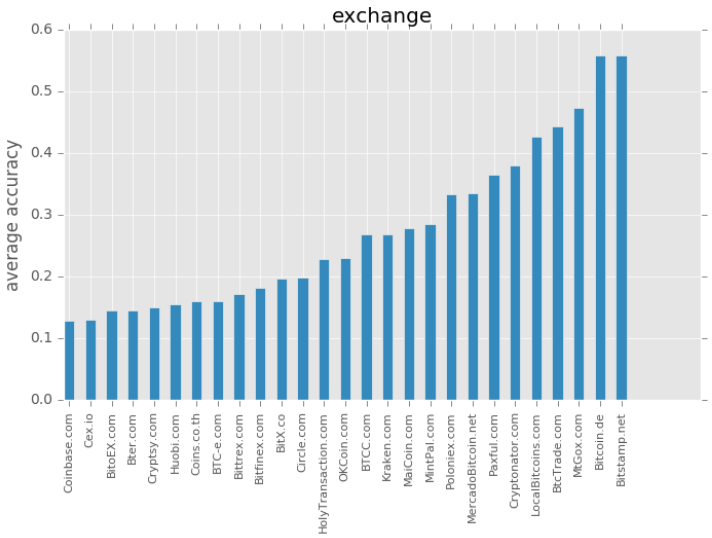


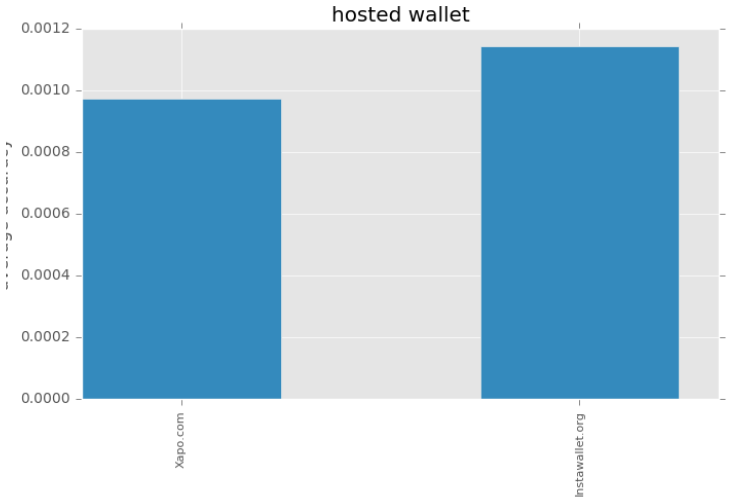


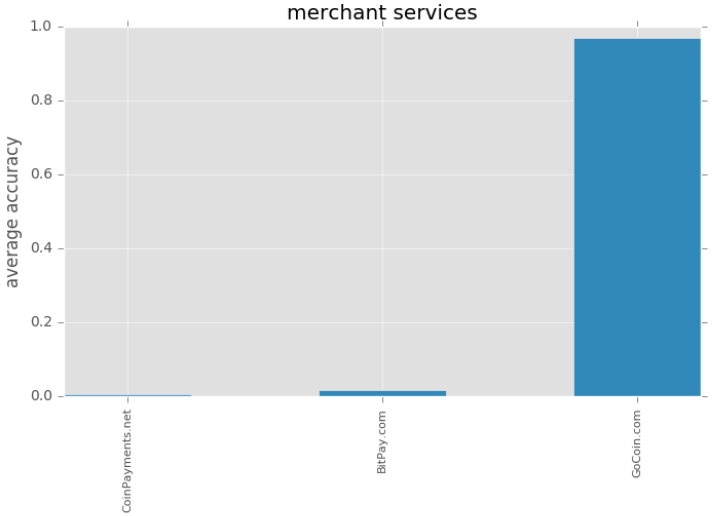


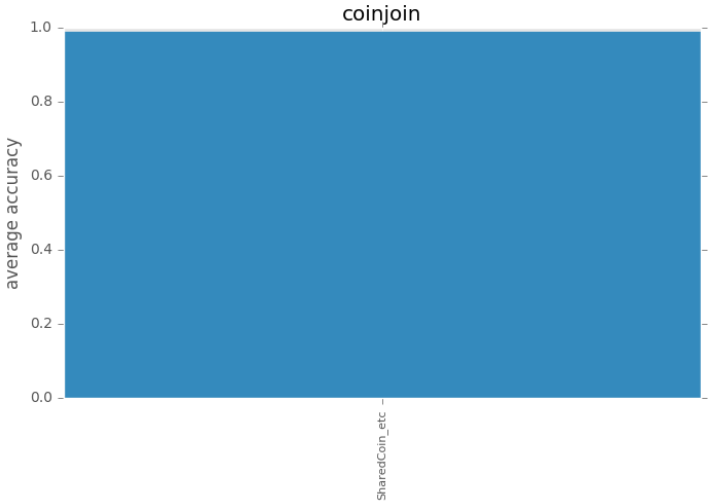


Average Accuracy by Class (RF)

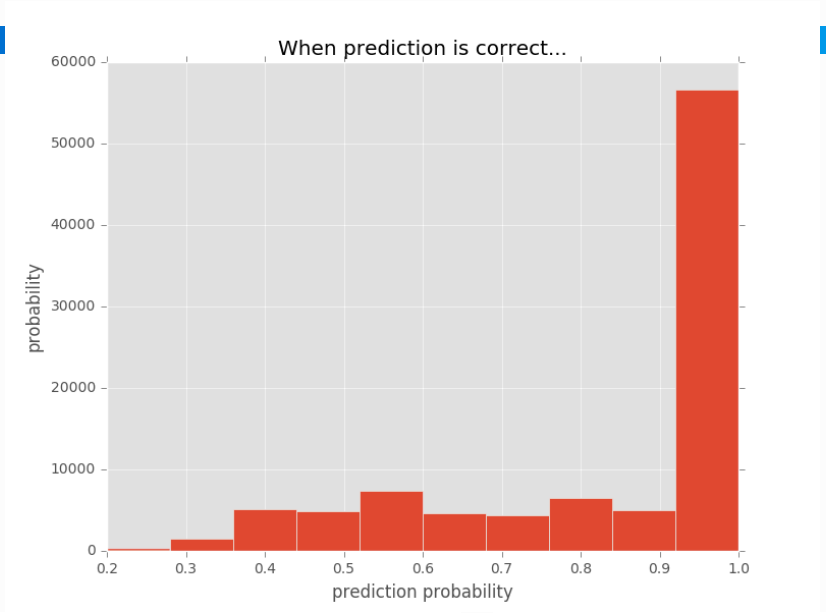


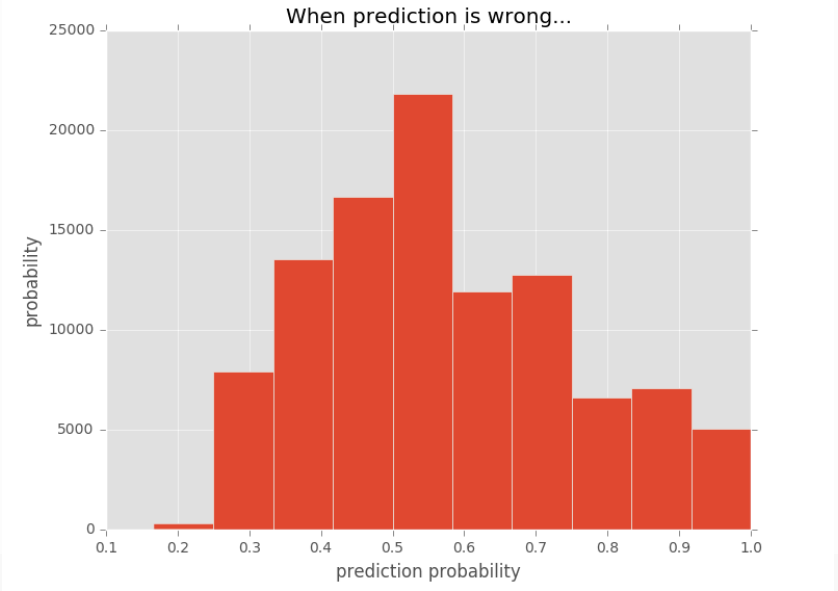


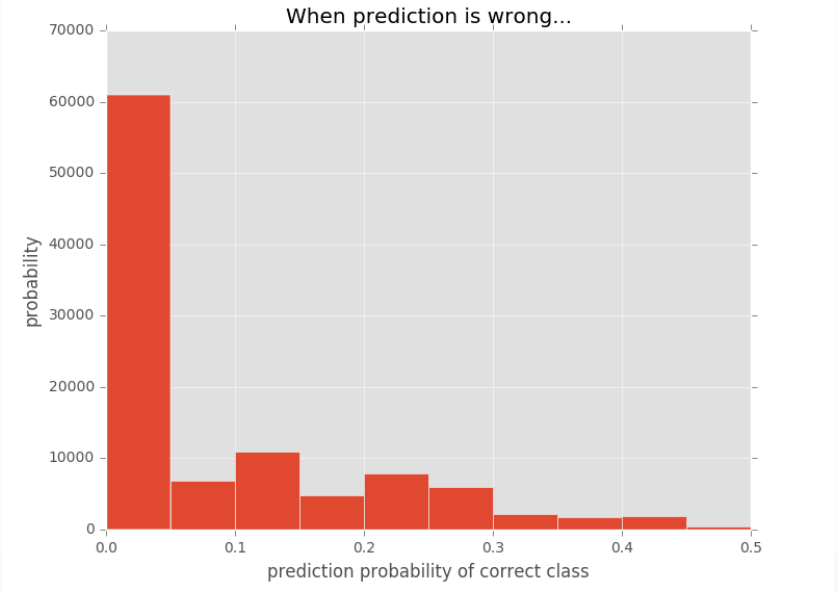




Prediction Probabilities







Abstract

Bitcoin is currently the most widespread cryptocurrency in use. Whatever its legitimate benefits, as the acceptance of Bitcoin spreads, its use as a means of conducting illicit commerce is likely to increase as well. Although anonymization of Bitcoin transactions is far from fool-proof, it has proved to be a non-trivial barrier to authorities slowing the growth of electronic illicit commerce. This research aims to assist law enforcement officials de-anonymize Bitcoin transactions by linking Bitcoin addresses to real-world entities through a suite of forensics tools. One proposed tool is the automatic characterization of Bitcoin addresses into one of several entity types. For example, some Bitcoin addresses are owned by gambling sites while others are owned by mixers. Knowing the type of entity that owns a Bitcoin address can help law enforcement identify the actual entity, and therefore assist in their successful arrest for illegal activity. This characterization is done using the Random Forest algorithm and has an accuracy rate of 65% for multi-class characterization (compared to 10% for random guessing) and an accuracy of 95% for discriminating between a mixer and non-mixer (compared to 50% for random guessing).

Problem you are trying to solve

- Many different classes of entities use Bitcoin, for example:
  - Mixing services, such as Bitcoin Fog and Shared Coin
  - Gambling sites, such as Satoshi Dice and Betcoin
  - Exchanges, such as Coinbase and BTC-e
  - Tor markets, such as Silk Road and Agora Market
- Key question: given a bitcoin address, can we predict what class of entity it belongs to?
- For example, the address *19oztMBBL519s22iYba8BQo28Wnba4m19M* has participated in 2 transactions and sent a total of 0.121 bitcoin. Is this address most likely owned by a mixing service, gambling site, exchange, or tor market?
- This work is part of a larger project whose primary goal is entity identification on the Bitcoin blockchain for use by Law Enforcement.
- Classifying Bitcoin addresses by entity class helps achieve this goal in two ways:
  1. If two addresses are classified as different entity classes, then it is unlikely they are owned by the same entity.
  2. Mixers are designed to obfuscate entities, so when an address is classified as a mixer, standard entity identification techniques cannot be applied.

Algorithmic approach of your solution

1. Collect data
   - Aggregate a set of bitcoin addresses with known owners.
   - Label each address with the class of entity that owns it, for example, mixing, gambling, exchange, etc.
2. Compute features
   - For each address, compute interesting features about it's behavior.
   - For example, the number of transactions it has participated in, the total amount of bitcoins it has sent, etc.
3. Apply supervised learning classification algorithms
   - Try several algorithms to compare performance, including Support Vector Machines (SVM) and Random Forest.
   - Identify important features.

Description of the data used

- Entire history of all transactions on the bitcoin blockchain from the beginning (Jan. 2009) to the end of 2015, from open sources.
- Labeled dataset of bitcoin addresses from Chainalysis, a Bitcoin forensics company, containing a total of 10 classes, 16,651,820 addresses, and 176 entities.

Class	Num. Addresses	Num. Entities	Example Entity
Mixing	4,753,891	4	Bitcoin Fog
Exchange	3,698,795	27	Coinbase
Merchant Services	2,335,030	4	BitPay
Unknown	2,016,683	44	1JNoitCVT46D…
Gambling	1,791,795	81	Satoshi Dice
Tor Market	1,443,584	7	Silk Road
Hosted Wallet	257,500	2	Instawallet
Mining Pool	172,455	5	BTTC Pool
Scam	113,280	1	MMMGlobal
Other	68,807	1	BTC Jam

Results

Using SVM and Random Forest implementations in Python's scikit-learn, we get accuracies of 40% to 65%, respectively, when attempting to classify a bitcoin address as one of 10 classes.

When attempting to classify a bitcoin address as a mixer or non-mixer, we get accuracies of about 95% for both algorithms.

Conclusions

What is the one-sentence summary of your R&D that you would want a technical person to remember?

When trying to classify Bitcoin addresses as one of many classes, the random forest algorithm had much higher accuracy than SVM, and both algorithms were easily able to identify a new class that we did not know existed before: coin join transactions that are a type of mixer.

What is the one-sentence summary of your R&D that you would want a manager or program developer to remember?

Applying machine learning techniques to Bitcoin addresses allowed us to discover a new class of transaction that we did not know existed before: a coin join transaction, which is a type of mixer that can be identified with great accuracy and has a large impact on law enforcement investigations.

Do you prefer an oral or a poster presentation?

Oral presentation

If oral, indicate presentation time between 15 and 30 minutes or a 5-minute spotlight.
15 to 20 minute presentation

Notes

- There are no published results on the dataset we used because the dataset is proprietary.
- Chainalysis validates labels by performing transactions with the entities and reviewing the blockchain.
- However, they also cluster addresses using basic de-anonymization metrics, including the multi-input heuristic and the change heuristic. I am not aware of any studies on the accuracy of their data.