2019 IEEE International Conference on Blockchain (Blockchain)

Cascading Machine Learning to Attack Bitcoin Anonymity

Francesco Zola*, Maria Eguimendia*, Jan Lukas Bruse*, Raul Orduna Urrutia*
* Dept. Data Intelligence for Energy and Industrial Processes, Vicomtech
Paseo Mikeletegi 57, 20009 Donostia/San Sebastian, Spain
{fzola, meguimendia, jbruse, rorduna}@vicomtech.org

***Abstract*— Bitcoin is a decentralized, pseudonymous cryptocurrency that is one of the most used digital assets to date. Its unregulated nature and inherent anonymity of users have led to a dramatic increase in its use for illicit activities. This calls for the development of novel methods capable of characterizing different entities in the Bitcoin network.**

In this paper, a method to attack Bitcoin anonymity is presented, leveraging a novel cascading machine learning approach that requires only a few features directly extracted from Bitcoin blockchain data. Cascading, used to enrich entities information with data from previous classifications, led to considerably improved multi-class classification performance with excellent values of Precision close to 1.0 for each considered class. Final models were implemented and compared using different machine learning models and showed significantly higher accuracy compared to their baseline implementation. Our approach can contribute to the development of effective tools for Bitcoin entity characterization, which may assist in uncovering illegal activities.

***Index Terms*— Bitcoin analysis, Bitcoin anonymity, cascading classifiers, entities classification, graph model, blockchain**

I. Introduction

Bitcoin was born in 2009 and since then its value and popularity has been rapidly increasing until its current state, in which it is the most used, assessed and priced cryptocurrency of all. Bitcoin is a pure peer-to-peer cryptocurrency [25] where all transactions are stored in a public shared ledger called blockchain that cannot be manipulated or changed [6]. Bitcoin is decentralized, which means that it is not controlled by any financial institution but it is regulated by everyone in the Bitcoin network: its blockchain architecture maintains the system without ambiguity [26].

While transactions within the Bitcoin network are openly available, Bitcoin user identity is non-transparent and protected by anonymity. This circumstance, combined with the unregulated nature of the Bitcoin market, has brought a lot of new actors to the Bitcoin network using cryptocurrency for illicit operations. Approximately one-quarter of Bitcoin users and half of all Bitcoin transactions are associated with illegal activity [9], accounting for an annual amount of around $72 billion (report 2018).

Conventional law-enforcement strategies tackling illegal financial operations such as money laundering or transactions funding criminal operations are typically based on complete knowledge of each actor's identity, while details about financial transactions are controlled by banks and thus unknown [24]. Within the Bitcoin network, these circumstances are reversed - incomplete knowledge of identities restricts traceability and transparency of operations, in turn promoting further increase of illegal activities. This calls for novel methods to attack anonymity within the Bitcoin network, aiming to uncover Bitcoin entity categories.

Among the most active categories of entities is the exchange, which represents a digital marketplace where traders can buy and sell cryptocurrencies using different fiat (money made legal tender by a government decree) or other digital currencies. Exchanges thus constitute the "front and exit doors" to the cryptocurrency world and are ideal to hide illicit operations, as documented in [22]. Another category is the darknet market. These markets are e-commerce platforms where users can find drugs, weapons and any kind of goods or services that are illegal in most countries. These cryptomarkets use electronic currencies to facilitate licit and illicit transactions among their users [5]. Further, so-called mixers represent services that allow users to obscure operations, as presented in [23]. At the same time mixed transactions increase the privacy of the users, and they can be used for money laundering of illegal funds.

Being able to classify anonymous Bitcoin entities according to such categories would increase transparency and would facilitate linking blockchain information with real actors to uncover illegal activities. Current techniques attacking anonymity often try to cluster addresses and apply heuristic assumptions combined with labelled data from external sources like markets, forums or social media in order to determine address owners in the real world [20]. However, gathering external data and combining them with Bitcoin information is tedious and could be limited due to privacy restrictions. This motivates the implementation of a model able to characterize different behaviours in the Bitcoin network by analyzing the pure blockchain information only by extracting transactions and by recognizing patterns using machine learning approaches.

In this paper, we present a novel approach to decrease Bitcoin anonymity based on a cascading machine learning model, using entity, address and motifs data as inputs. We apply a "cascade" of classifiers, performing a first entity classification based on address, 1_motif and 2_motif data, which is then used as input for a second classification step, which combines those classification results with entity information from the blockchain. Notably, our approach only requires a few features that can be directly extracted from Bitcoin blockchain data.

978-1-7281-4693-5/19/$31.00 ©2019 IEEE
DOI 10.1109/Blockchain.2019.00011

In order to compare benefits and limits of the proposed approach, two experiments are presented: firstly, a simple classifier is trained based on pure entity information gathered from the blockchain. In the second experiment, a final classifier is trained using the enriched data set generated by our cascading approach. We aimed to detect six different types of Bitcoin entity behaviours. Overall, three classifier models are tested and compared: Adaboost, Random Forest and Gradient Boosting.

The rest of the paper is organized as follows. Section II describes the related work. After that, Section III presents the graph model used and Section IV shows an overview of the used data sets. Section V describes the implemented machine learning models and Section VI presents the obtained results. Finally, in Section VII, we draw conclusions and provide guidelines for future work.

II. Related work

User anonymity has probably been the key factor for the success of cryptocurrencies and has promoted illegal activities within the Bitcoin network. Yet, several studies determine that current measures adopted by the Bitcoin protocol are not sufficient to protect the privacy of its users [19], [1], opening up possibilities to attack Bitcoin anonymity. One of the first transaction analysis is documented in [30] where typical behavior of Bitcoin users are detected based on how they spend cryptocurrencies, how they keep the balance in their accounts, and how they move Bitcoins between their various accounts. Herrera-Joancomartí [12] presents a review on Bitcoin anonymity, concluding that anonymity can be reduced by address clustering or by gathering information from various peer-to-peer networks. This technique is also advocated in [15], where conservative constraints (patterns) are applied for address clustering, and in [17] where information gathered from online forums is used to characterize the CryptoLocker, a family of ransomware. Similarly, in [8], information scraped from online forums and social media is determinant to simulate an attacker and to summarize activity of both known and unknown Bitcoin users. In [3], a generic method to deanonymize a significant fraction of Bitcoin users by correlating their pseudonyms with public IP addresses is described. Reid et al. [29] demonstrates how it is possible to associate many public-keys with each other, using a map of the topological network and external identifying information in order to investigate a large theft of Bitcoins.

Several recent studies have exploited machine learning algorithms for Bitcoin analysis. In [13], an unsupervised learning model is presented with the aim to identify atypical transactions related to money laundering. Monamo et al. [21] introduce a k-means classifier for object clustering and fraudulent activity detection in Bitcoin transactions. Another study on detection of anomalous behavior, suspicious users and transactions is presented in [27], where three unsupervised learning methods are applied to two graphs generated by the Bitcoin transaction network. Further, a supervised machine learning algorithm is used by [11] to uncover Bitcoin anonymity using a method for predicting the type of yet-unidentified entities. In [2], data mining techniques are used to implement and train a classifier to identify Ponzi schemes in the Bitcoin blockchain and in [18] a Bayesian optimized recurrent neural network (RNN) and a Long Short Term Memory (LSTM) are implemented to predict the direction of Bitcoin price in USD.

Recently, an interesting approach is given in [28], where the concept of motifs is introduced to blockchain analysis. Authors performed an analysis of the transaction directed hypergraph in order to identify several distinct statistical properties of exchange addresses. They were able to predict if an address is owned by an exchange with $> 80\%$ accuracy. The introduction of hypergraphs (or dirhypergraphs) proved beneficial due to their significant advantages over a complex graph structure typically derived from Bitcoin networks. In [14], the motif concept is further developed and is combined with multiple features (entity, address, temporal, centrality) to obtain a comprehensive entity classification into five categories: Exchange, Service, Gambling, Mining Pool and DarkNet marketplace. Using a total of 315 features, a global accuracy of 0.92 could be achieved.

Inspired by the good classification results presented in [14], we present here a novel machine-learning-based approach to attack Bitcoin anonymity, making use of motifs as introduced by Ranshous et al. and allowing for multi-class classification of Bitcoin entities as in [14], yet aiming to provide a straightforward methodology that relies on fewer, well-defined features. To achieve this, we introduce a novel cascading machine learning model for Bitcoin data analysis. The main idea is to implement a cascade of classifiers, so that outgoing classification results can be joined and can be used to enrich a final classification.

III. Graph Model

A. Blockchain Graph Model

Bitcoin transactions have a natural graph structure, with a fundamental example being the address-transaction graph (Figure 1). This graph is directly obtained by using the information gathered from the blockchain and provides an estimation of the flow of Bitcoins linking public key addresses over time. The vertices represent the addresses $(a_1, a_2, ..., a_{\mathrm{N}})$ and the transactions $(tx_1, tx_2, ..., tx_{\mathrm{M}})$. The directed edges (arrows) between entities and transactions indicate the incoming relations, while directed edges between transactions and entities correspond to outgoing relations. Each directed edge can also include additional features such as values, time-stamps, etc.

To improve anonymity in the network, users are encouraged to generate a new Bitcoin address for each new transaction, which is a common advice for the correct usage of Bitcoin[1]. Due to this procedure, several addresses belong to the same logical user, so that a simplification is possible by introducing the concept of *entities*. An entity is defined as a person or organization that controls or can control

[1] https://bitcoin.org/en/protect-your-privacy

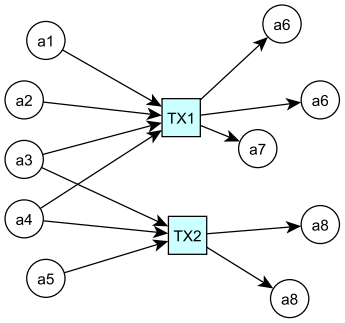

Fig. 1: *Example of address-transaction graph*

multiple public key addresses. This definition allows us to transform the address-transaction graph into the entity-transaction graph (Figure 2).

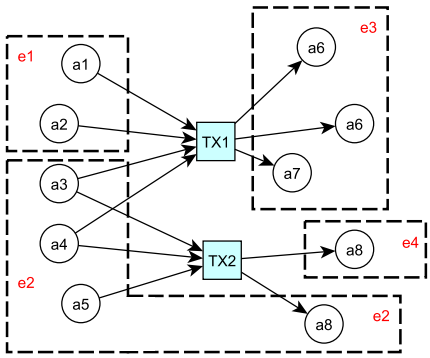

Fig. 2: *Example of entity-transaction graph obtained by address clustering*

The new graph is obtained by grouping addresses belonging to the same user into entities (address clustering). This operation is not intuitive, however several heuristic properties have already been presented with the aim to help the clusterization process, for example in [1], [15] and [7]. In the obtained graph, vertices represent the entities $(e_1, e_2, ..., e_K)$ and the transactions $(tx_1, tx_2, ..., tx_M)$. Similar to the address-transaction graph, directed edges between entities and transactions indicate the incoming relations, while directed edges between transactions and entities correspond to outgoing relations. The entity-transaction graph (Figure 2) summarizes the network well and constitutes an easily understandable representation of the money flow within the network.

B. Motif Graph Model

Motif graphs were introduced in [16] and were motivated by applications in bioinformatics, specifically in metabolic network analysis. However, as shown in Section II, prior studies such as [28] have introduced the concept of motifs to Bitcoin analysis. In this paper, a definition of N_motif is used, starting from the generalized concept introduced in [14].

***Definition** 1:* A N_motif is a path from the entity-transaction graph with length $2N$ that starts and ends with an entity. Let $(e_1, .., e_M) \in E$ be a class of entities and $(t_1, .., t_N) \in T$ be a class of transactions, with $M \leq N + 1$, then:

$$N\_motif = (e_1, t_1, ..., t_N, e_M)$$

in which at least one output from each transaction must be an input to the next transaction.

The term *branch* is used here to refer to a path in the motif graph that begins and ends with an entity passing through exactly one transaction. If a single branch of the graph has the same entity as input and output ($e_j = e_{j+1}$), the branch is called Direct Loop, otherwise it is called Direct Distinct, as shown in Figure 3.

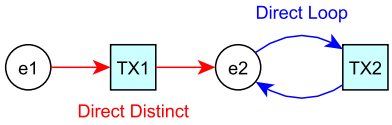

Fig. 3: *2_motif graph with Direct Loop and Direct Distinct path definition*

From the motif definition it is clear that all transactions are ordered in time, which means that $\tau(t_1) < \tau(t_2) < .. < \tau(t_N)$, where τ represents a transaction time.

Here, we use the 1_motif and 2_motif concepts. The 1_motif represents the relation between two entities (at least one distinct), while the 2_motif is the relation between three entities (at least one distinct) involved in two consecutive transactions.

IV. Data Overview

We considered the whole Bitcoin blockchain data created until February 5th 2019, 08:13:31 AM, corresponding to 561,620 blocks, which contain about 380,000,000 transactions and involve more than 1,000,000,000 addresses. This data was then combined with information available on the WalletExplorer[2], a benchmark platform for entities detection, which represents a collection of information about different known entities that have been detected until today. The data set is thus composed of 311 different samples, divided into six classes (see Table I):

- *Exchange*: entities that allow their customers to trade fiat currencies for Bitcoins (or vice versa)
- *Service*: entities that offer Bitcoin payment methods as solutions to their business (financial services, trading, lending, etc.)
- *Gambling*: entities that offer gambling services (casino, betting, roulette, etc.)
- *Mining Pool*: entities composed of a group of miners that work together sharing their resources in order to reduce the volatility of their returns
- *Mixer*: entities that offer a service to obscure the traceability of their clients' transactions

[2]https://www.walletexplorer.com/

- *Marketplace*: entities allowing to buy any kind of goods or services that are illegal in most countries paying with Bitcoin

Class	Abbreviation	# Entities	# Address	% Address
Exchange	Ex	137	9,943,512	61.63
Gambling	Gmb	76	3,054,238	18.93
Marketplace	Mrk	20	2,349,210	14.56
Mining Pool	Pool	25	76,104	0.47
Mixer	Mxr	37	475,714	2.95
Service	Serv	16	235,629	1.46
Total		**311**	**16,134,407**	**100**

TABLE I: *Overview of WalletExplorer data used for this study*

As shown in Table I, the *Exchange* is the top class represented by more than 60% of samples, while the *Mining Pool* class is the least represented with just 0.47% (even though it has more distinct entities than the *Marketplace* and the *Service*).

Cross-references between Bitcoin blockchain data and labelled data from the WalletExplorer allow us to re-size the original data set by removing all the unlabelled and unusable data. As such, we focus our analysis on known entities only. From this new data set, four dataframes (2-dimensional labelled data structure or data table with samples as rows and extracted features as columns) were extracted for the proposed analysis:

- *Entity dataframe* contains all features related to an entity that can be directly extracted from the blockchain. They are: the amount of BTC received/sent, the balance of the entity, the number of transactions in which this entity is the receiver/sender, and the number of addresses belonging to this entity used for receiving/sending money. (This dataframe was composed of 311 samples and 7 features)
- *Address dataframe* contains all features related to Bitcoin addresses. Features are: the number of transactions in which a certain address is detected such as receiver/sender, the amount of BTC received/sent from/to this address, the balance, uniqueness (if this address is just used in one transaction) and siblings. (This dataframe was composed of $16{,}134{,}407$ samples and 7 features)
- *1_motif dataframe* contains the information directly extracted from the 1_motif graph. In this case, each row contains: the amount received/sent in the transaction, number of distinct addresses used for receiving/sending money, number of similar received/sent transactions between the entities in the branch, the fee, and if the branch realizes a Direct Loop or Direct Distinct path. (This dataframe was composed of $58{,}076{,}963$ samples and 9 features)
- *2_motif dataframe* contains information gathered from the 2_motif graph. The features analyzed are: the number of addresses as input/output for the first and second path in 2_motif graph, the amount received/sent in the first and second branch, the fee of both considered transactions, number of similar sent transactions between the entities in the first and second branch, Direct Loop or Direct Distinct path for the first and the second branch and Direct Loop or Direct Distinct path considering the whole 2_motif path, see Figure 4. (This dataframe was composed of $83{,}443{,}055$ samples and 18 features)

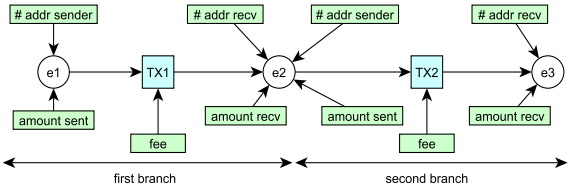

Fig. 4: *2_motif representation with extracted features highlighted*

V. Machine Learning

A. Classifier Models

To demonstrate benefits and limits of our approach, we conducted two different experiments. Firstly, we created a simple classifier, called *C_entity* (Figure 5), merely based on the samples stored in the entity dataframe, containing (seven) entity-related features that can be directly extracted from the blockchain. This classifier was evaluated via a cross-validation process (see Section V-B). Results from cross-validation were considered as our baseline classification. The simple classifier was implemented in three versions applying Adaboost, Random Forest and Gradient Boosting models as those previously yielded good classification results for Bitcoin data [28].

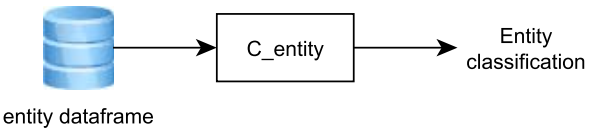

Fig. 5: *First experiment: simple entities classifier*

In the second experiment, prior to entity classification according to the six classes (Table I), we built three separate classifiers, based on the additionally available address, 1_motif and 2_motif dataframes and their respective features ($7 + 9 + 18 = 34$ features). Outgoing information from these classifications was processed, as shown in Figure 7, in order to create a set of six new features for each classifier, which were then used to enrich (extend) the entity dataframe. Finally, a new classifier *C_final* was generated to obtain final entity classification based on this enriched entity dataframe and its 25 features (7 belonged to the entity dataframe and 6x3 were generated from the three classifiers *C_address*, *C_motif1*, *C_motif2*). With this cascading approach, new entity-related characteristics were added to

the entity dataframe, ultimately improving the classification as demonstrated in the following sections.

The first step was to split the address, 1_motif and 2_motif dataframes into two parts called A-data set (for training) and B-data set (for testing) with a proportion of 70/30. The A-data set was used to compute cross-validation of the three *C_address*, *C_motif1*, *C_motif2* classifier models (Figure 6). After that, the B-data set was used as input for the trained classifiers *C_address*, *C_motif1*, *C_motif2* in order to obtain classification results based on completely new, unseen data.

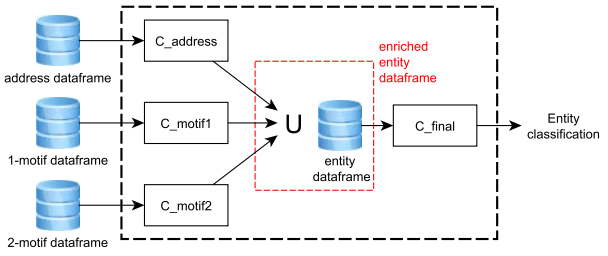

Fig. 6: *Second experiment: cascading entities classifiers*

Classification results essentially assign one of the six possible output classes to each entry in the input dataframe. As each entry has its original (ground truth) label obtained from the WalletExplorer, we can join input label and computed output class and perform a group-by and count operation as illustrated in Figure 7: we count how many times a sample belonging to a particular entity has been detected in each of the considered classes. This value is then normalized as indicated in the following formula:

$$\forall \xi \in E \qquad \frac{\parallel P_{\xi \mid \mathrm{j}} \parallel}{\sum_{i=1}^{N} \parallel P_{\xi \mid \mathrm{i}} \parallel} * 100 \qquad with \quad j \in N$$

where E is the entities set and N represents the number of considered classes ($N = 6$ in this study). The term $\parallel P_{\xi \mid \mathrm{j}} \parallel$ represents how many times a sample originally labelled with entity ξ generates a prediction belonging to the class j, while the term $\sum_{i=1}^{N} \parallel P_{\xi \mid \mathrm{i}} \parallel$ counts all the predictions generated from samples with labelled input belonging to entity ξ.

These normalized values form a dataframe containing 311 samples (one for each known entity as in the entity dataframe) and six new features, representing the percentage of being classified as belonging to one of the six classes. These features were added to the entity dataframe for data enrichment, constituting our cascading machine learning system. The elements of the enriched entity dataframe were used to implement and evaluate the final classifier, called *C_final*, and a cross-validation process (Section V-B) was applied to compute its performance.

To allow a better comparison between experiments, we implemented all classifier models *C_address*, *C_motif1*, *C_motif2* and *C_final* with Adaboost, Random Forest and Gradient Boosting models. Specifically, all Adaboost classifiers were generated with the number of estimators set to 50 and the learning rate set to 1. All Random Forest models

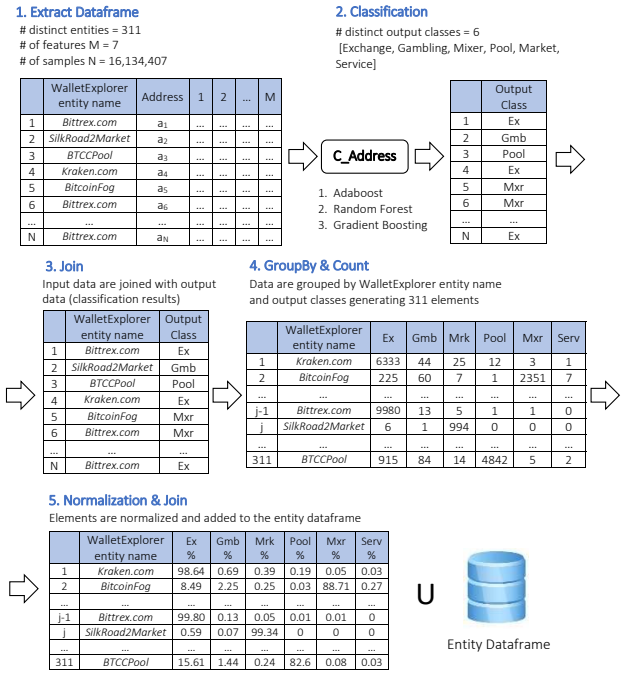

Fig. 7: *Steps to create the enriched entity dataframe applied to an example address dataframe*

were implemented with the number of estimators set to 10, a Gini function to measure the quality of the split and without a maximum depth of the tree. All Gradient Boosting models were implemented with the number of estimators set to 100, the learning rate set to 0.1 and the maximum depth for limiting the number of nodes set to 3.

B. Evaluation Metrics

All classification models were evaluated by extracting and comparing classification metrics via a cross-validation process. The goal of cross-validation is to analyze the prediction capabilities of the model in order to detect problems such as over-fitting or selection bias [4]. Here, we used stratified K-fold cross-validation, with a value of K equal to 5. This method involves dividing the whole data set into K equal partitions or folds.

Each fold is composed of data ensuring a good representative sample of the whole population by keeping the same proportion of classes present in the original data set (stratification). Then, K-1 folds are used to train the model and the one left-out fold is used to evaluate the predictions obtained by the trained model. The entire process is repeated K times, until each fold has been left out once, testing all possible combinations. During this process, the following metrics were computed:

- *Accuracy* or *Score* is defined as the number of correct predictions divided by the total number of predictions and is given as percentage
- *Precision* is the number of positive predictions divided by the total number of the positive class values predicted. It represents a measure of a classifier's exactness

given as a value between 0 and 1, with 1 relating to high precision

- *Recall* represents a measure of a classifier's completeness given as a value between 0 and 1
- *F_1-score* is the harmonic mean of Precision and Recall. It takes values between 0 and 1, with 1 relating to perfect Precision and Recall

$$F_1 score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

- *Matthews Correlation Coefficient (MCC)* is a metric yielding easy comparison with respect to a random baseline, suitable for unbalanced classes. It takes values between -1 and $+1$. A coefficient of $+1$ represents a perfect prediction, 0 an average random prediction and -1 an inverse prediction. As shown in [10], let K be the number of classes and C be a confusion matrix with dims $K \times K$, the MCC can be calculated as:

$$MCC\_part1 = \sqrt{\sum_k (\sum_l C_{\text{kl}})(\sum_{f,g|f\neq g} C_{\text{gf}})}$$

$$MCC\_part2 = \sqrt{\sum_k (\sum_l C_{\text{lk}})(\sum_{f,g|f\neq g} C_{\text{fg}})}$$

$$MCC = \frac{\sum_k \sum_l \sum_m C_{\text{kk}} C_{\text{lm}} - C_{\text{kl}} C_{\text{mk}}}{MCC\_part1 * MCC\_part2}$$

In Section VI results for the baseline model (*C_entity*) and for the final model (*C_final*) obtained after cross-validation using the enriched dataframe are presented and compared. We report global metric values for Accuracy/Score and *MCC* averaged over the K=5 cross-validation runs and per-class values for Precision, Recall and F1-score when evaluating the final models.

C. Hardware and Software Configuration

All analyses were run on a cluster of three virtual machines, each one with 16 CPUs Intel(R) Xeon(R) Silver 4114 CPU @ 2.20 GHz, 64 GB RAM DDR4 memory with 2,666 MHz, and 500 GB of Hard Disk SATA. Apache Spark[3] v2.4.0, set in cluster mode was used to manage stored data using Apache Hadoop[4]. The various classifier models were implemented and evaluated using Python's Scikit-learn[5] library. All scripts were executed within the Jupyter-notebook[6] environment.

VI. Results

Considering the simple classifier *C_entity* from the first experiment, the Gradient Boosting model yielded a better average score (61.90% accuracy) and *MCC* (0.44) than Random Forest and Adaboost classifiers, as shown in Table II (upper section). However, with overall low *MCC* for all classifiers (between 0.22 and 0.44), these scores were not sufficient to achieve reliable entities characterization. This led to introducing our cascading machine learning approach, enriching the initial entity dataframe with information gathered from prior classifications in the second experiment.

Model	Classifier	Score %	Std %	MCC
Adaboost	*C_entity*	45.63	6.34	0.22
Random Forest	*C_entity*	59.71	1.82	0.41
Gradient Boosting	*C_entity*	61.90	1.36	0.44
Adaboost	*C_final*	78.84	1.76	0.76
Random Forest	*C_final*	98.04	1.22	0.97
Gradient Boosting	*C_final*	99.68	0.63	0.99

TABLE II: *Average performance of classifiers over five cross-validation repetitions for simple C_entity model (above) and for final model after data enrichment via cascading machine learning C_final*

Analyzing the *C_address*, *C_motif1* and *C_motif2* classifiers separately for entity characterization, Table III shows that outgoing information from the Random Forest classifier resulted to be more accurate than information from Gradient Boosting and Adaboost classifiers (accuracy scores $>90\%$ for Random Forest). Notably, only using information from the address dataframe, the Random Forest classifier *C_address* could already achieve an average global accuracy of $\sim 96\%$. Due to these results, we only used results obtained from Random Forest classifiers for the subsequent entities dataframe enrichment. Random Forest classifiers not only proved to be the best in terms of accuracy, but also performed with highest speed among the considered classification models.

Model	*C_address* %	*C_motif1* %	*C_motif2* %
Adaboost	61.54	72.69	78.27
Random Forest	95.73	94.14	90.88
Gradient Boosting	83.23	83.52	83.54

TABLE III: *Average global C_address, C_motif1 and C_motif2 classifier accuracy calculated via 5-fold cross-validation*

The final classifiers *C_final* were fed with the enriched entity dataframe, which comprised the original features from the entity dataframe and included as new features the class predictions obtained from *C_address*, *C_motif1* and *C_motif2* for the respective test data sets. From Table II (lower section) it is obvious that the average score result improved significantly by exploiting the information obtained via our cascading approach. Random Forest and Gradient Boosting classifiers again performed better than the Adaboost model, reaching a score of more than 98% (respectively $\sim 39\%$ and $\sim 38\%$ percentage points higher than the baseline accuracy from *C_entity*). Furthermore, classification results were more stable during cross-validation, generating low standard deviations between 0.63% and 1.76% and the *MCC* reached values close to 1.0, relating to close-to-perfect class prediction.

In Table IV, we present per-class Precision, Recall and F_1-scores calculated for *C_entity* (baseline) and *C_final* (enriched) classifiers for each classification model. Results

[3]https://spark.apache.org/
[4]https://hadoop.apache.org/
[5]https://scikit-learn.org/
[6]https://jupyter.org/

		C_entity model			*C_final* model		
Class	Model	Precision	Recall	F1-score	Precision	Recall	F1-score
Exchange	Adaboost	0.51	0.68	0.57	0.77	0.78	0.77
Gambling	Adaboost	0.22	0.14	0.17	0.75	1.00	0.85
Market	Adaboost	0.05	0.15	0.08	0.40	0.30	0.33
Mining Pool	Adaboost	0.20	0.16	0.17	0.11	0.2	0.14
Mixer	Adaboost	0.69	0.78	0.71	1.00	0.98	0.99
Service	Adaboost	0.20	0.10	0.13	0.95	0.95	0.95
Exchange	Random Forest	0.60	0.77	0.67	0.96	1.00	0.98
Gambling	Random Forest	0.54	0.50	0.51	1.00	1.00	1.00
Market	Random Forest	0	0	0	1.00	0.85	0.91
Mining Pool	Random Forest	0.68	0.50	0.56	1.00	0.92	0.96
Mixer	Random Forest	0.89	0.78	0.82	1.00	1.00	1.00
Service	Random Forest	0	0	0	1.00	0.93	0.96
Exchange	Gradient Boosting	0.61	0.80	0.69	1.00	1.00	1.00
Gambling	Gradient Boosting	0.59	0.53	0.55	0.99	1.00	0.99
Market	Gradient Boosting	0.10	0.05	0.06	1.00	1.00	1.00
Mining Pool	Gradient Boosting	0.38	0.40	0.38	1.00	1.00	1.00
Mixer	Gradient Boosting	0.92	0.84	0.87	1.00	1.00	1.00
Service	Gradient Boosting	0	0	0	1.00	0.93	0.96

TABLE IV: *Average Precision, Recall and F_1-score calculated in each model implementation for each class*

demonstrate that the simple classifier *C_entity* - independently of the classification model used - had problems detecting *Service* and *Market* entities (calculated metrics are 0 or have very low values). However, it is to be noted that these two classes are least represented in terms of distinct entities in the original data set. Random Forest and Gradient Boosting classifiers showed overall good performance in detecting *Mixer* entities for the *C_entity* approach (F_1-scores >0.8).

By exploiting the cascading machine learning implementation however, all classifiers improved their classification performance for each class, with most values being close to 1.0. Only the Adaboost model kept having problems with the classification of *Mining Pool* and *Market* entities. Random Forest and the Gradient Boosting models instead yielded excellent values for Precision, Recall and F_1-score for each class.

Overall best classification scores were achieved by the *C_final* implementation with Gradient Boosting models. Data enrichment through prior classification and cascading thus clearly had a highly beneficial impact on classification ability of Gradient Boosting, motivating a further analysis of the importance of individual features from the enriched entity dataframe. We therefore calculated in a next step a feature *importance score* for the enriched entity dataframe.

Generally, the feature importance score provides a score that indicates how useful or valuable each feature was in the construction of the model. The more often an attribute is used to make key decisions, the greater will be its relative importance score. Importance was explicitly calculated through Python's Scikit-learn library for each attribute in the data set, allowing features to be ranked and compared to each other.

Figure 8 shows a list of the top fifteen features for the *C_final* Gradient Boosting classifier. All fifteen important features were created during the prior classifications taking into account *C_address*, *C_motif1* and *C_motif2*. These features represent how address, 1_motif, or 2_motif data,

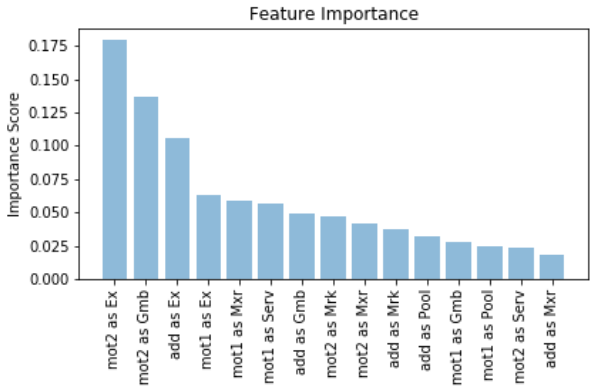

Fig. 8: *Top 15 important features from the GB classifier*

related to a certain entity, were previously classified. This highlights again that the information brought in from prior classifications (first step of the cascade) clearly contributes to much improved entities characterization.

VII. Conclusion and Future Work

In this paper, we present a novel approach of how to attack Bitcoin anonymity through entity characterization. Specifically, we demonstrate how a cascading machine learning model combined with an adequate set of input features directly derived from Bitcoin blockchain data (entity and address data) as well as derived via 1_motif and 2_motif concepts introduced by Ranshous et al. [28] can lead to impressive classification performance for a number of relevant Bitcoin entity classes. In fact, we were able to obtain an average global accuracy score of 99.68% with low standard deviation of 0.63% and a Matthews Correlation Coefficient (*MCC*) of 0.99 over 5-fold cross validation for a Gradient Boosting model using our cascading approach.

These final models were indeed able to predict each of the six entity classes used (Exchange, Gambling, Market, Mining Pool, Mixer, Service) with Precision, Recall and

F_1-score values close to 1.0. Ranshous et al. [28] obtained similar results using Random Forest and Adaboost classifiers, however their study was limited to exchange address classification. Jourdan et al. [14] generally obtained lower values for per-class F_1-score and Precision ranging between 0.67 and 1.0 using Gradient Boosting and their approach involved a complex step of model hyper-parameter calibration and required a total number of 315 input features.

Our approach applies one more classification step in the "classification cascade" generating a set of new entity-related features used for the final classification, but we do not require extensive parameter tuning. Most importantly, we only use 34 features for the initial classification step (involving address, 1_motif and 2_motif) and finally 7 features from the entities data set plus 3 x 6 = 18 new features obtained as outgoing information from the initial classification step. The final classification is thus based on only 25 features, which equals to less than 10% of features compared to Jourdan et al. Our future work will focus on investigating deeper the matter of feature importance, in order to further reduce the number of relevant features required for obtaining high entity classification performance. This will facilitate the process of attacking Bitcoin anonymity further.

One major drawback of our approach is that we were not able to characterize entities behaving as normal users as this information is not currently available as ground truth data in the WalletExplorer. We had to remove all entities that have not yet been classified in the WalletExplorer from our analysis. Nevertheless, we were able to detect six classes of key Bitcoin services that have previously been associated with illicit financial operations with very high classification scores. We therefore believe that our study can contribute to improving crime investigation and may form a base for developing effective tools assisting law enforcement agencies in uncovering illegal activities within the Bitcoin network.

ACKNOWLEDGMENT

This work was partially funded by the European Commission through the Horizon 2020 research and innovation program, as part of the "TITANIUM" project (grant agreement No 740558).

REFERENCES

[1] Androulaki, E., Karame, G.O., Roeschlin, M., Scherer, T., Capkun, S.: Evaluating user privacy in bitcoin. In: International Conference on Financial Cryptography and Data Security. pp. 34–51. Springer (2013)

[2] Bartoletti, M., Pes, B., Serusi, S.: Data mining for detecting bitcoin ponzi schemes. In: 2018 Crypto Valley Conference on Blockchain Technology (CVCBT). pp. 75–84. IEEE (2018)

[3] Biryukov, A., Khovratovich, D., Pustogarov, I.: Deanonymisation of clients in bitcoin p2p network. In: Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security. pp. 15–29. ACM (2014)

[4] Cawley, G.C., Talbot, N.L.: On over-fitting in model selection and subsequent selection bias in performance evaluation. Journal of Machine Learning Research **11**(Jul), 2079–2107 (2010)

[5] Christin, N.: Traveling the silk road: A measurement analysis of a large anonymous online marketplace. In: Proceedings of the 22nd international conference on World Wide Web. pp. 213–224. ACM (2013)

[6] Crosby, M., Pattanayak, P., Verma, S., Kalyanaraman, V., et al.: Blockchain technology: Beyond bitcoin. Applied Innovation **2**(6-10), 71 (2016)

[7] Ermilov, D., Panov, M., Yanovich, Y.: Automatic bitcoin address clustering. In: 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA). pp. 461–466. IEEE (2017)

[8] Fleder, M., Kester, M.S., Pillai, S.: Bitcoin transaction graph analysis. arXiv preprint arXiv:1502.01657 (2015)

[9] Foley, S., Karlsen, J., Putniņš, T.J.: Sex, drugs, and bitcoin: How much illegal activity is financed through cryptocurrencies? (2018)

[10] Gorodkin, J.: Comparing two k-category assignments by a k-category correlation coefficient. Computational biology and chemistry **28**(5-6), 367–374 (2004)

[11] Harlev, M.A., Sun Yin, H., Langenheldt, K.C., Mukkamala, R., Vatrapu, R.: Breaking bad: De-anonymising entity types on the bitcoin blockchain using supervised machine learning. In: Proceedings of the 51st Hawaii International Conference on System Sciences (2018)

[12] Herrera-Joancomarti, J.: Research and challenges on bitcoin anonymity. In: Data Privacy Management, Autonomous Spontaneous Security, and Security Assurance, pp. 3–16. Springer (2015)

[13] Hirshman, J., Huang, Y., Macke, S.: Unsupervised approaches to detecting anomalous behavior in the bitcoin transaction network. 3rd ed. Technical report, Stanford University (2013)

[14] Jourdan, M., Blandin, S., Wynter, L., Deshpande, P.: Characterizing entities in the bitcoin blockchain. arXiv preprint arXiv:1810.11956 (2018)

[15] Koshy, P., Koshy, D., McDaniel, P.: An analysis of anonymity in bitcoin using p2p network traffic. In: International Conference on Financial Cryptography and Data Security. pp. 469–485. Springer (2014)

[16] Lacroix, V., Fernandes, C.G., Sagot, M.F.: Motif search in graphs: application to metabolic networks. IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB) **3**(4), 360–368 (2006)

[17] Liao, K., Zhao, Z., Doupé, A., Ahn, G.J.: Behind closed doors: measurement and analysis of cryptolocker ransoms in bitcoin. In: 2016 APWG Symposium on Electronic Crime Research (eCrime). pp. 1–13. IEEE (2016)

[18] McNally, S., Roche, J., Caton, S.: Predicting the price of bitcoin using machine learning. In: 2018 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP). pp. 339–343. IEEE (2018)

[19] Meiklejohn, S., Orlandi, C.: Privacy-enhancing overlays in bitcoin. In: International Conference on Financial Cryptography and Data Security. pp. 127–141. Springer (2015)

[20] Meiklejohn, S., Pomarole, M., Jordan, G., Levchenko, K., McCoy, D., Voelker, G.M., Savage, S.: A fistful of bitcoins: characterizing payments among men with no names. In: Proceedings of the 2013 conference on Internet measurement conference. pp. 127–140. ACM (2013)

[21] Monamo, P., Marivate, V., Twala, B.: Unsupervised learning for robust bitcoin fraud detection. In: 2016 Information Security for South Africa (ISSA). pp. 129–134. IEEE (2016)

[22] Moore, T., Christin, N.: Beware the middleman: Empirical analysis of bitcoin-exchange risk. In: International Conference on Financial Cryptography and Data Security. pp. 25–33. Springer (2013)

[23] Moser, M.: Anonymity of bitcoin transactions (2013)

[24] Möser, M., Böhme, R., Breuker, D.: Towards risk scoring of bitcoin transactions. In: International Conference on Financial Cryptography and Data Security. pp. 16–32. Springer (2014)

[25] Nakamoto, S., et al.: Bitcoin: A peer-to-peer electronic cash system (2008)

[26] Narayanan, A., Bonneau, J., Felten, E., Miller, A., Goldfeder, S.: Bitcoin and cryptocurrency technologies (2016)

[27] Pham, T., Lee, S.: Anomaly detection in bitcoin network using unsupervised learning methods. arXiv preprint arXiv:1611.03941 (2016)

[28] Ranshous, S., Joslyn, C.A., Kreyling, S., Nowak, K., Samatova, N.F., West, C.L., Winters, S.: Exchange pattern mining in the bitcoin transaction directed hypergraph. In: International Conference on Financial Cryptography and Data Security. pp. 248–263. Springer (2017)

[29] Reid, F., Harrigan, M.: An analysis of anonymity in the bitcoin system. In: Security and privacy in social networks, pp. 197–223. Springer (2013)

[30] Ron, D., Shamir, A.: Quantitative analysis of the full bitcoin transaction graph. In: International Conference on Financial Cryptography and Data Security. pp. 6–24. Springer (2013)