Computational Statistics and Data Analysis 181 (2023) 107687

Contents lists available at ScienceDirect

Computational Statistics and Data Analysis

journal homepage: www.elsevier.com/locate/csda

Functional classification of bitcoin addresses

Manuel Febrero-Bande [a], Wenceslao González-Manteiga [a], Brenda Prallon [b], Yuri F. Saporito [c,*]

[a] Departamento de Estadística, Análisis Matemático y Optimización, Universidade de Santiago de Compostela, Spain
[b] Department of Economics, Cornell University, United States
[c] Escola de Matemática Aplicada, Fundação Getulio Vargas, Brazil

ARTICLE INFO

Article history:
Received 28 June 2021
Received in revised form 26 December 2022
Accepted 27 December 2022
Available online 10 January 2023

Keywords:
Bitcoin market
Darknet market
Functional data analysis
Functional classification
Functional principal components

ABSTRACT

A classification model for predicting the main activity of bitcoin addresses based on their balances is proposed. Since the balances are functions of time, methods from functional data analysis are applied; more specifically, the features of the proposed classification model are the functional principal components of the data. Classifying bitcoin addresses is a relevant problem for two main reasons: to understand the composition of the bitcoin market, and to identify addresses used for illicit activities. Although other bitcoin classifiers have been proposed, they focus primarily on network analysis rather than curve behavior. The proposed approach, on the other hand, does not require any network information for prediction. Furthermore, functional features have the advantage of being straightforward to build, unlike expert-built features. Results show improvement when combining functional features with scalar features, and similar accuracy for the models using those features separately, which points to the functional model being a good alternative when domain-specific knowledge is not available.

1. Introduction

The question of building a model to classify the main activity of bitcoin addresses is considered. Bitcoin was the first decentralized cryptocurrency to be created, and it is also the most popular. Anonymity is a central characteristic of the bitcoin protocol, and one of the reasons for its success. This implies, nevertheless, that the market composition of this cryptocurrency is not obvious: the purposes for which bitcoins are spent are obscure. It is known, however, that illicit services account form a relevant portion of that market – 46% of bitcoin transactions, as estimated by Foley et al. (2019) – which follows from anonymity certainly being an attractive for law-breakers; for instance, the Silk Road darknet market, closed by the FBI in 2013 and used mainly for commercializing illegal drugs, moved approximately fifteen million dollars annually in transactions, Christin (2013). Thus, identifying the main activity of a bitcoin address is a relevant issue, since it both aids law enforcement and sheds some light on the bitcoin market organization.

* Corresponding author at: Praia de Botafogo, 190, Rio de Janeiro, RJ 22250-900, Brazil.
E-mail addresses: manuel.febrero@usc.es (M. Febrero-Bande), wenceslao.gonzalez@usc.es (W. González-Manteiga), bq45@cornell.edu (B. Prallon), yuri.saporito@fgv.br (Y.F. Saporito).

https://doi.org/10.1016/j.csda.2022.107687
0167-9473/

1.1. Literature review on bitcoin addresses' classification

Regarding the classification of bitcoin data, most of the work focuses on identifying illicit activities, and uses some type of aggregation of addresses, linking them to one common entity. Furthermore, the literature tends to utilize features from network relations, instead of the functional behavior of transactions. Meiklejohn et al. (2013) propose a method to cluster bitcoin addresses to user-level; that is, multiple addresses are assigned to each user (also called entity). They use a small number of transactions labeled through their own empirical interactions with various entities, and then identify major institutions and their interactions.

Foley et al. (2019) study the bitcoin market of illicit activities. They find that illegal users of bitcoin tend to transact more, in transactions involving fewer addresses, and they tend to hold smaller amounts of the cryptocurrency. To make this analysis, they first cluster addresses to user-level using the approach from Meiklejohn et al. (2013). Then, two models are developed: the first one used information on the addresses network - that is, which addresses communicated with each other - to cluster legal/illegal activities. The second method is detection-controlled estimation. The covariates used account for transaction frequency, USD value, and addresses' lifespan, but there are also other variables with information of the network, such as how many users are involved in the transactions, and information of external shocks.

Graph neighborhood features were proposed by Jourdan et al. (2018) to classify users with five different labels, taken from `WalletExplorer` (walletexplorer.com): exchange, gambling, mining, service and darknet. They build a total of 315 features, but obtain great accuracy results using only 15. They show that graph features improve the accuracy significantly over features considering only the entities addresses. Logistic regression and gradient boosting were employed, and results show that the darknet is perfectly classified, and other classes, apart from mining, also score well. Hu et al. (2019) also propose graph features to identify money laundering activities occurring across the bitcoin network. Their binary model classifies transactions, with high accuracy.

1.2. Literature review on FDA

There are many specific methods for classifying functional data. In this article, the focus is on adapting classifiers from the multivariate framework. This is achieved by projecting the curves of credits and debits onto a functional basis, and then using the coefficients as functional features.

The proposed basis is the eigenbasis of the Karhunen-Loève expansion, the functional principal components (FPCs) basis. Besides being widely explored in the literature, this basis maximizes the variance of the observed curves, analogously to classical multivariate principal components; this means that, typically, few components are necessary to explain most of the differences of the functions; Hall et al. (2001) argue that the FPCs capture the greatest part of the curves, in a L^2 sense, since they are maximizing variance. They apply this transformation to radar signals curves, and estimate the coefficients density to compute the posterior probabilities in order to classify the signals in eight different groups. Their model performs significantly better than the multivariate counterpart.

In Müller et al. (2005) it is shown that, when using the principal components basis to represent the functions and truncating the expansion, the functional logistic model becomes the equivalent of a multiple logistic regression with FPCs. Escabias et al. (2004) compare two different approaches for a functional logistic model. The first one consists of smoothing the curves on an arbitrary basis, and then performing regular PCA on the matrix defined by the multiplication of the coefficients matrix by the inner product matrix. The second one consists on estimating the FPCs by the method described in Section 8.4.2 of Ramsay and Silverman (2005). The resulting principal components are inputs for the multiple logistic regression. In Escabias et al. (2005), the first method is applied with B-splines to predict risk of drought across different areas in Canada.

A functional logistic model is also used to classify gene expression data in Leng and Müller (2006). On the same subject, Song et al. (2008) develop a model to classify gene expression that consists of applying multivariate classification algorithms to FPCs estimated from the data. They first smooth the data with a pre-defined basis and then compute the principal components. The pre-defined basis is chosen as a cubic B-spline basis, since it is flexible and computationally efficient.

Following the same idea, Li et al. (2013) attempt to classify hyperspectral images by treating their pixels as curves instead of high-dimensional discrete vectors. As in Song et al. (2008), the curves were first smoothed using cubic B-splines, with the difference that, instead of choosing the number of basis, they use a smoothing parameter for penalizing the second derivative; functional principal components were computed in the same fashion, and the number of FPCs was chosen by five-fold cross-validation. We use the same procedures in our work. The method was compared to other methods, including Support-Vector Machine (SVM) applied to regular principal components, on three popular hyperspectral data sets. Results show that their approach performs consistently better. Functional principal components as features for SVMs were also employed in Lee (2004).

Furthering the topic of FPCs, Hall and Hosseini-Nasab (2006) derive some asymptotic bounds for the truncated estimators of eigenfunctions and eigenvalues. Yao et al. (2005) propose a nonparametric method to perform functional principal components analysis (FPCA) with sparse logitudinal data. Finally, since FPCA is very sensitive to outliers, some tools to identify and remove those based on notions of functional data depth have been discussed in López-Pintado and Romo (2007) and Cuevas et al. (2007).

Additionally to the classical reference Ramsay and Silverman (2005), we forward the reader to some recent reviews and new methods on functional classification methods using FDA: Febrero-Bande and González-Manteiga (2013), Jiang and Chen (2020), Aneiros et al. (2021), Piotr et al. (2017), Ling and Vieu (2021) and Li et al. (2022), Wang et al. (2016).

1.3. Contribution

We propose a solution to the classification of bitcoin addresses considering information of addresses' balances. More precisely, we use the addresses' transactions, which are the credits and debits made throughout a period of time. This is essentially a task of classification by analyzing the behavior of curves. Although, as we have aforementioned, there have been a few works with the same ultimate goal - to classify bitcoin addresses by activity -, the classifiers in the literature focus on arbitrary features, that come from field specific knowledge - also known as the process of 'feature engineering' - or network analysis. Our approach differs from these because we use the fact that the account movements are a function of time, or, more specifically, of the address' life span, here understood as the time elapsed between the first credit, at $t = 0$, up to a pre-defined limit, $t = T$. That is, we aim to find patterns in the shapes of the credits and debits curves, and to classify the addresses based on them. This is done by employing Functional Data Analysis (FDA) tools. The advantage of this strategy is, beyond its straightforwardness, that we do not need to have any information about which addresses are communicating with each other on our training set in order to make predictions, which is very useful for classifying addresses mainly connecting to unlabeled vertices of the network; this is our main contribution.

The arbitrary features normally contain implicit information of some aspects of curve behavior; for example, the number of credits is a measure for frequency, and the total amount of credit is a measure of level. FDA, however, accomplishes summarizing this information in a more direct way, meaning that the methods do not require as much domain-specific knowledge. This is particularly useful when the process of feature engineering is complicated.

This paper is organized as follows: Section 2 discusses data acquisition and treatment, including details on how to represent the data as functional. Section 3 discusses the different types of features considered, both scalar and functional. Section 4 presents the results, and Section 5 concludes the paper.

2. Empirical strategy

2.1. Data acquisition

A bitcoin address is a unique string of number and letters stored in the bitcoin's blockchain that can be the recipient or sender of bitcoins. A bitcoin transaction can have multiple addresses as inputs and outputs. One could think the address as the number of a bank account, but with some very distinct properties as public visibility of balance and all transactions, anonymity, cannot hold negative quantities of bitcoins (debt), and each user in this market usually has a large number of addresses. The user here, though, is not an actual individual, but a company/website/organization that typically holds multiple addresses; we refer to it as "entity". The balance of an address is the amount (possibly zero) of bitcoins in that address at a given time. The entity that a given address belongs to is usually unknown but some companies make some of its addresses public for various reasons. Based on this public information, one can identify more addresses as belonging to this same entity if they were inputs to the same transaction with one or more inputs from this entity.

The data used in this project includes both information on the address's class and their balances over time. This allows for the development of a supervised classification model in which the balance values are used to predict the class of a certain address, that is, a model which predicts the main activity of a given address based on its transactions. Thus, the data was collected in essentially two stages: classifying the addresses by their major purpose and fetching their balances over time.

Through the scraping of `WalletExplorer` we obtained information linking addresses with entities (for example, address `1PkJRQaKStcmCeh CJUjxYiQejFDm3w4yV` belongs to 999Dice.com). We then used the classes of `WalletExplorer`: exchange, gambling, pools, services/others, and old/historic; however, we only kept the addresses of the old/historic category related to illegal activities, and renamed the class "darknet". This work focuses on linking addresses to these classes rather than entities. The addresses gathered were active at some time between April 2011 and April 2017. Furthermore, we used Google's BigQuery public dataset on bitcoins (available at https://cloud.google.com/blog/products/data-analytics/introducing-six-new-cryptocurrencies-in-bigquery-public-datasets-and-how-to-analyze-them) to obtain the addresses' balances.

The initial data set had the descriptions shown in Table 1.

2.2. Data treatment

In this section, we describe all the treatment necessary to pose a well-defined classification problem and build the functional features. We choose a time frame to observe the addresses, creating a clear rule of how much time is necessary to wait after the first observed transaction before classifying them. We also limited the sample to a minimum of observations, in order to properly capture curve behavior. Measures of level (shape of credits and debits), variation (the derivative of credits and debits) and frequency (Poisson rate in which transactions occur) are created using FDA methods, and their FPCs

Table 1
Original Number of Addresses and Entities, by Category.

Category	Addresses	Entities
Darknet Marketplace	106,274	8
Exchanges	259,102	50
Gambling	45,986	17
Pools	56,811	1
Services/others	287,160	22
Total obs.	755,333	98

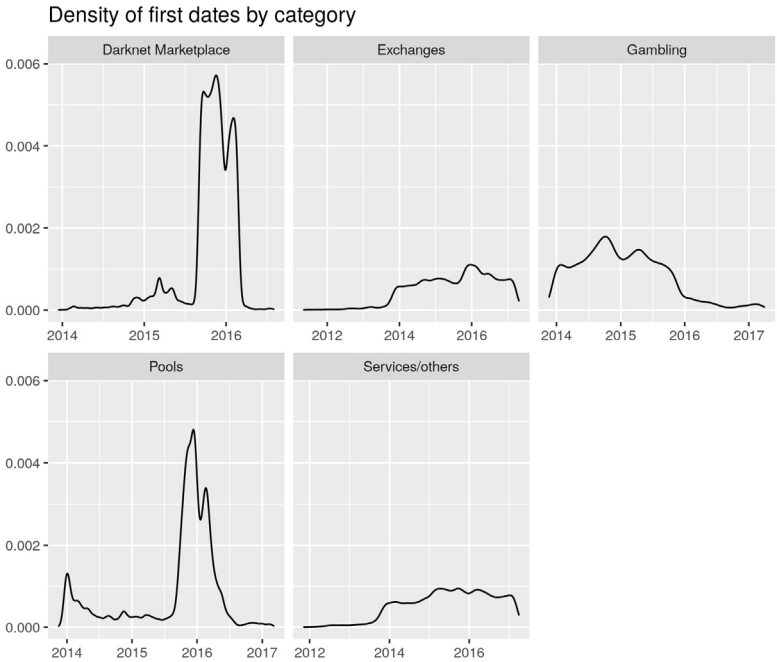

Fig. 1. Distribution of starting dates by category.

are computed. We used the R software, with packages Febrero-Bande and Oviedo de la Fuente (2012) and Ramsay et al. (2020).

2.2.1. A sampling issue

The classification of bitcoin addresses by their balances over time depends on the underlying hypothesis that an address' behavior is invariant over calendar time. That is, for addresses of the same class, if one is observed in a year and another on the next, they should still present the same characteristics of movements and amounts. We argue that this amount should be measured in dollars. Since the bitcoin price is very volatile, if the balances were measured in bitcoins, they might impose a change in level through time. Again, for two addresses observed through different time frames, if the bitcoin price is higher in one period, they will probably be receiving smaller amounts of bitcoins for providing the same service.

This difference in level of bitcoin amounts could possibly add a timestamp bias: the model may be classifying better the category with addresses observed in similar dates. If the samples of a certain category are concentrated around a period of time, say, beginning in March 2015, the bitcoin price volatility would affect them equally. If instead the samples of another category are addresses whose start is somewhere in the range from 2011 to 2017, the amount of bitcoins will most likely vary greatly between them, even if they present the same behavior. Greater variability due to differences in the start date of the samples of a category could make defining a pattern harder, thus harming the classification of that category.

Indeed, the addresses are not observed uniformly over time: in Fig. 1, it is evident that the darknet and pool samples are much more concentrated than their counterparts. This is due to the sampling nature: for darknet addresses to be identified and labeled, a police operation was most likely necessary - as was the case with the Silk Road website, shut down by the FBI in 2013. This poses another matter, that was not addressed in this work: services that put a lot of effort into anonymity, such as the darknet, will most likely have few entities discovered. This is in fact the case and Table 1 shows the imbalance among the categories for the classification task.

However, even with the values in dollars, there could still be an implicit timestamp in volatility: addresses in a time of high (or low) volatility could be grouped together. Furthermore, volatility introduces noise between one transaction and

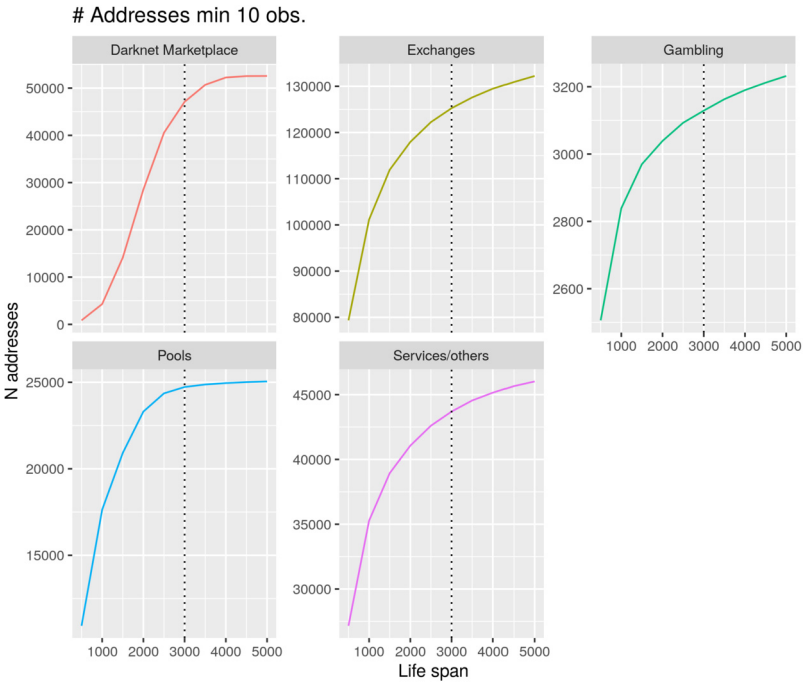

Fig. 2. Number of addresses with at least 10 obs. by observed hours.

another. To avoid these phenomena, instead of transforming each amount by the bitcoin price at end of the day, the mean of the bitcoin price is taken over the period that the addresses are observed and then used to level the units. Therefore, the addresses balances are represented in dollars, instead of bitcoins. More specifically, the transformed balances are the balances in bitcoins multiplied by a fixed amount, which is the mean of the bitcoin value in dollars over the address lifespan.

2.2.2. Functional data: additional treatment

It was required to limit the sample for addresses with a minimum number of observations - otherwise there is not enough points to treat the data as functional. This was tested using thresholds of 10 and 20 transactions. No higher threshold was tested because, for a minimum of 30 transactions, we only have fewer than 500 addresses classified as darknet, and we would have to keep a very small sample.

Another important remark is the fact that one cannot really know if an address' life has ended, because there is nothing to prevent its use after a long time of inactivity. With that in mind, it was necessary to choose a time window to consider for prediction. The value was 3000 hours. Fig. 2 shows this as a reasonable value, for that are not a lot of addresses to gain after this span (to clarify, we first limit the observations to the first 3000 hours, and then filter the addresses with at least 10 or 20 account transactions). For addresses where a last transaction occurred before the end of this time frame, we know, since no other transaction was made, that the balances remain the same, so they were extended as constants until the end of the interval. This interval was then normalized, so that time extends between $[0, 1]$.

The original data of the addresses' balances were very stiff; a lot of credits were immediately followed by debits of the same amount, which accounted for a very erratic behavior. Here, we show some examples for darknet addresses in Fig. 3; other categories can be viewed in the extended version of this paper Febrero-Bande et al. (2022).

The solution to this problem was to split the balances in order to form two different curves: one consisting of credits and another of debits. They were then integrated by calculating the accumulated sum (Fig. 4).

2.3. Functional data modeling

Even with a more regular behavior, the curves were still step functions. We choose to smooth them because we want more regularity on our level measure. Even though there is no observational noise, in the sense that we know exactly when and how much was credited/debited at each address, since the underlying classification assumption is that each class follows a pattern of transactions, smoothing aids in capturing the more relevant trends in the amounts. Through smoothing, one can represent the possibly infinite dimensional underlying function as an approximate finite dimensional basis expansion. To do so, we fit the data with a roughness penalty.

We consider the penalized functional regression framework over $L^2([0, 1])$. The ingredients of this regression are the predefined basis of $L^2([0, 1])$, denoted by $(\phi_k)_{k\in\mathbb{N}}$, the number $K \in \mathbb{N}$ of basis elements being considered in the approximation

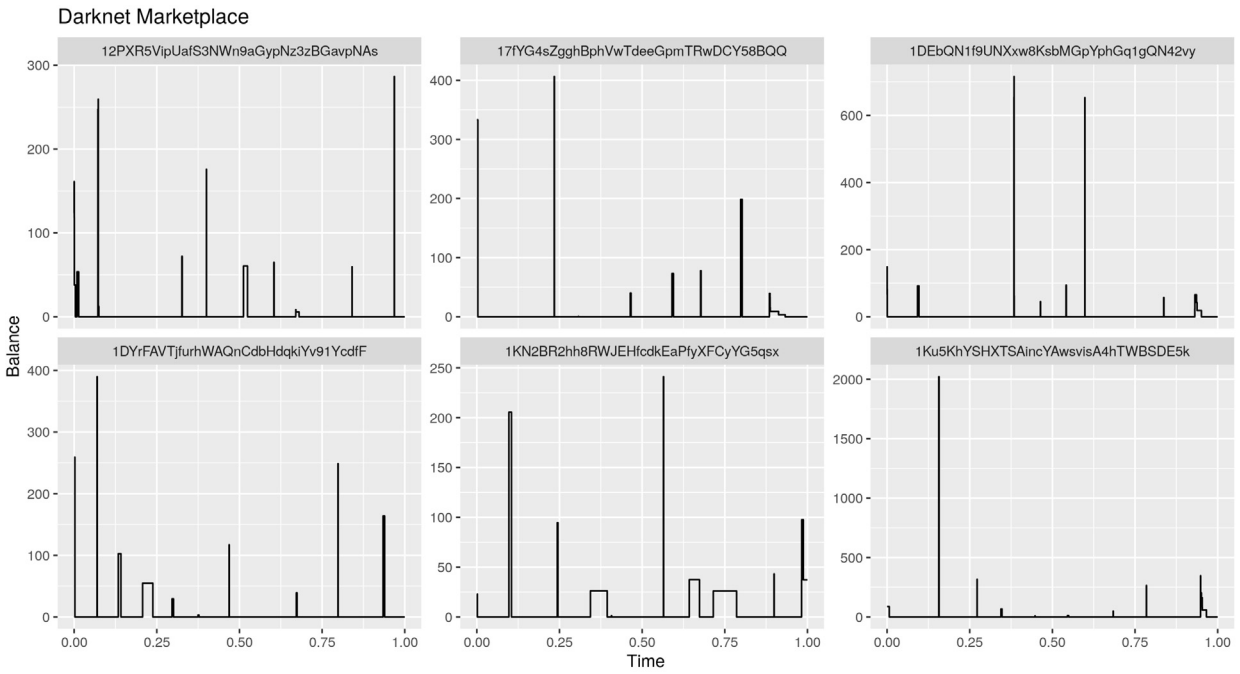

Fig. 3. Original account balances for six darknet addresses, in dollars. We will use these same addresses in the following figures.

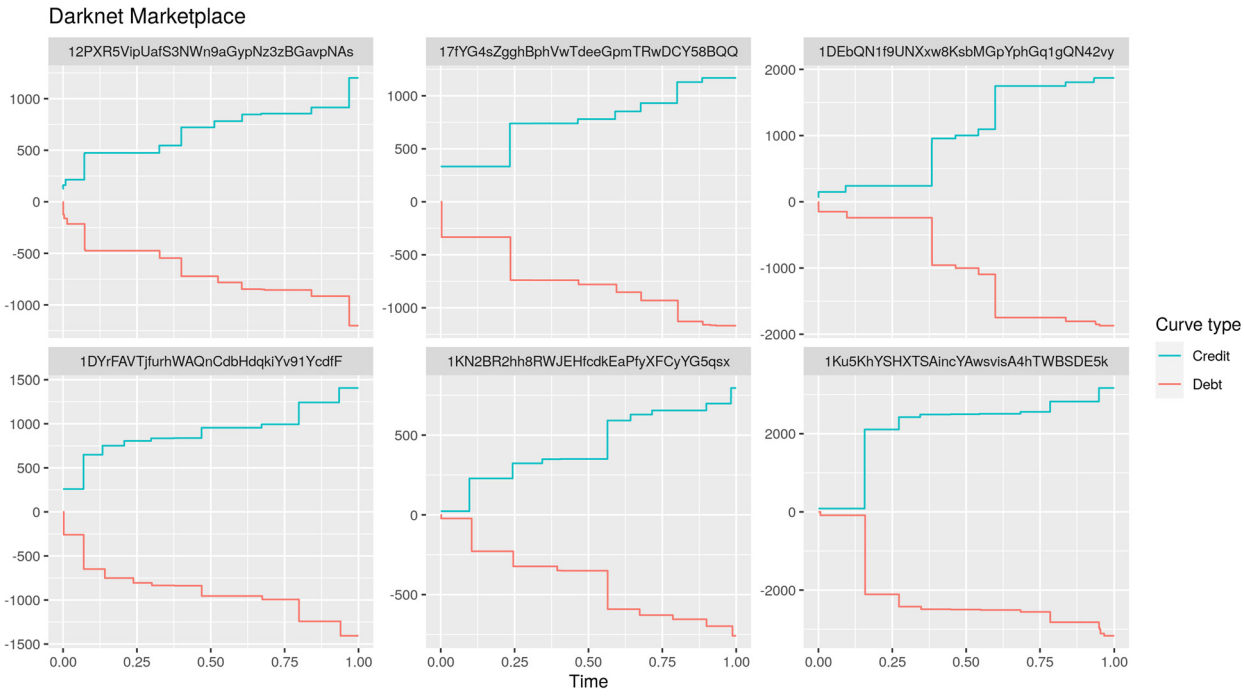

Fig. 4. Accumulated sum of credits and debits for six darknet addresses, in dollars.

and the penalization, considered here as the L^2 norm of the m-th derivative of the approximation. The design matrix of regression, which is defined as the first K elements of the pre-defined basis evaluated at the observation times, will be denoted by $\mathbf{\Phi}_{n\times K}$ and the penalization matrix $\mathbf{R}_{K\times K} = \int_0^1 D^m(\boldsymbol{\phi}(s))D^m(\boldsymbol{\phi}^T(s))ds$, where $\boldsymbol{\phi} = [\phi_1, \ldots, \phi_K]^T$.

Because of the great variability in scale of the accumulated credits and debits across different addresses, the logarithm was also taken. It is important to realize that it is possible to, regardless of the number of transactions, obtain an arbitrarily large number of observations through time for these curves - since it is known that, if no other transaction was made,

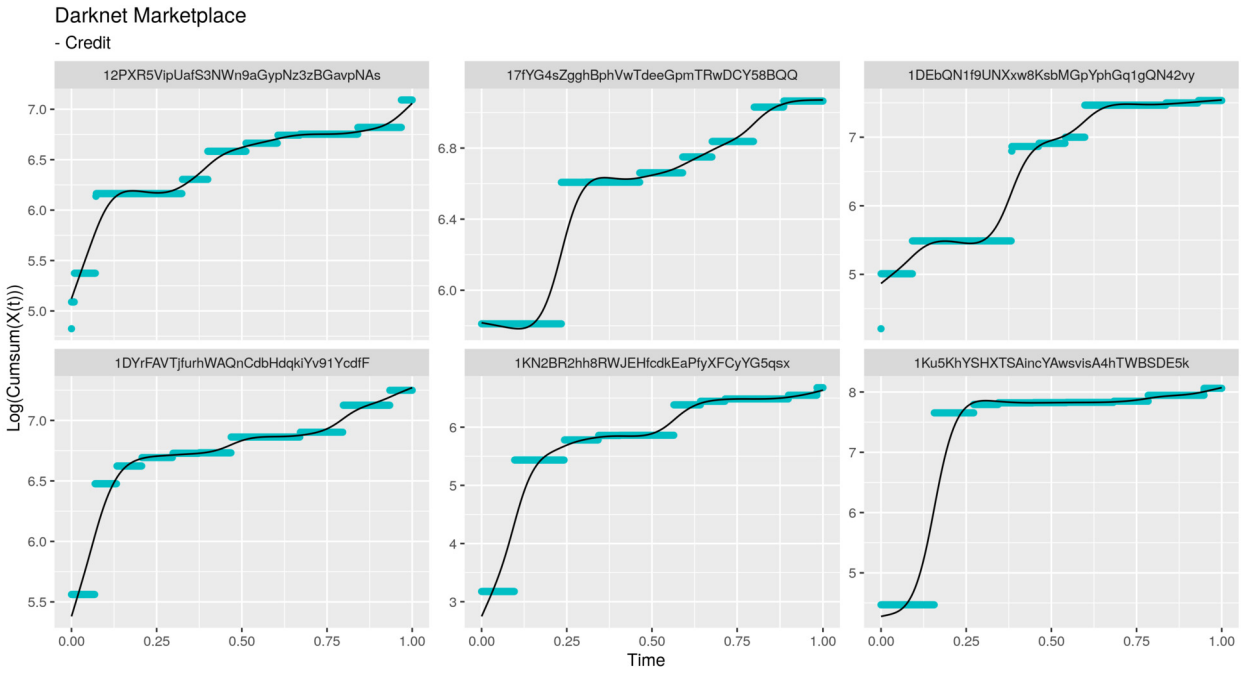

Fig. 5. Smoothed log-accumulated sum of credits for six darknet addresses.

the account value in bitcoins remains constant. In order to provide more data for better curve fitting, we used this fact to sample 501 evenly spaced points throughout the first 3000 hours additionally to the credit/debit times.

We choose the *cubic* B-spline basis system, since it is the most common choice for polynomial functions. We penalized the second derivative, a natural measure of curvature ($m = 2$). One important aspect of using B-splines is choosing where to place the knots. We chose to place at almost every data point (including the original sampling points and the 501 evenly spaced points). It was necessary to remove points that were too close by (less than 3 minutes apart), because otherwise the steep nature of the data would make the estimation of derivatives, particularly the second derivatives, too large; which, in turn, would make the values of matrix $\mathbf{R}$ too high in magnitude, risking to overwhelm $\mathbf{\Phi}^T\mathbf{\Phi}$; $\mathbf{R}$ itself may not have full rank, so it may not be invertible; therefore, attempting to invert $(\mathbf{\Phi}^T\mathbf{\Phi} + \lambda\mathbf{R})$ will yield an error message or inaccurate results.

The removal of close data points does not, however, imply loss of information, because the frequency is accounted for in the estimated Poisson rate curves, as will be explained in a couple of paragraphs. The smoothing parameter was chosen to be $\lambda = 0.001$. Although we did experiment with the Generalized Cross Validation (GCV) criterion to choose λ, we found the resulting curves to be under-smoothed, even after attempting to further discount the degrees of freedom. We wanted a reduced level of roughness in the approximation, so we followed the heuristic approach and our visual explorations of fit suggested $\lambda = 0.001$ to be a good choice.

Furthermore, using the same grid (evenly 501 points) as before, the functional principal component analysis was performed; it is convenient that these functions are regularly measured. Some examples for credit curves can be seen in Fig. 5.

In order to better account for the variations in the curves, their first derivatives were also estimated. For a B-spline, the derivatives are calculated analytically after smoothing, but the details of such computations are not of particular interest to this paper. Since smoothness is a necessary hypothesis for the existence of derivatives, we must use the already smoothed curves $\hat{\mathbf{x}}(t) = \hat{\mathbf{c}}^T\boldsymbol{\phi}(t)$ rather than the step curves as our data. To estimate the curve's derivative of order m, the smoothing process is done by penalizing the derivative of order $m+2$ to ensure that the derivative itself will be smooth. We have then used an order 5 B-spline system to re-smooth the curves applying a penalization of the third derivative (5 is the minimum order of a B-spline with a square-integrable third derivative). The derivatives were then evaluated at the same 501 equally spaced points. We chose $\lambda = 0.0001$ in the same fashion as the smoothing parameters for the curves themselves. The results for credits are displayed in Fig. 6.

Furthermore, the curves can also be modeled by a Marked Point Process, see for instance Jacobsen and Gani (2006), in which the credits and debits are viewed as arrivals. We simplify the hypothesis so that it becomes an inhomogeneous Poisson process. The assumption made for the functional model is that the rate is not constant over time, but a function of it instead.

Denoting this inhomogeneous Poisson process by $(N(t))_{t\in[0,1]}$, its arrival times by T_n, $T_0 = 0$, and functional rate by $\mu : [0,1] \longrightarrow [0,+\infty)$, we have that, for instance, from Theorem 10.1 of Thompson (1988),

$$T_1, \ldots, T_n \mid N(1) = n \sim f,$$

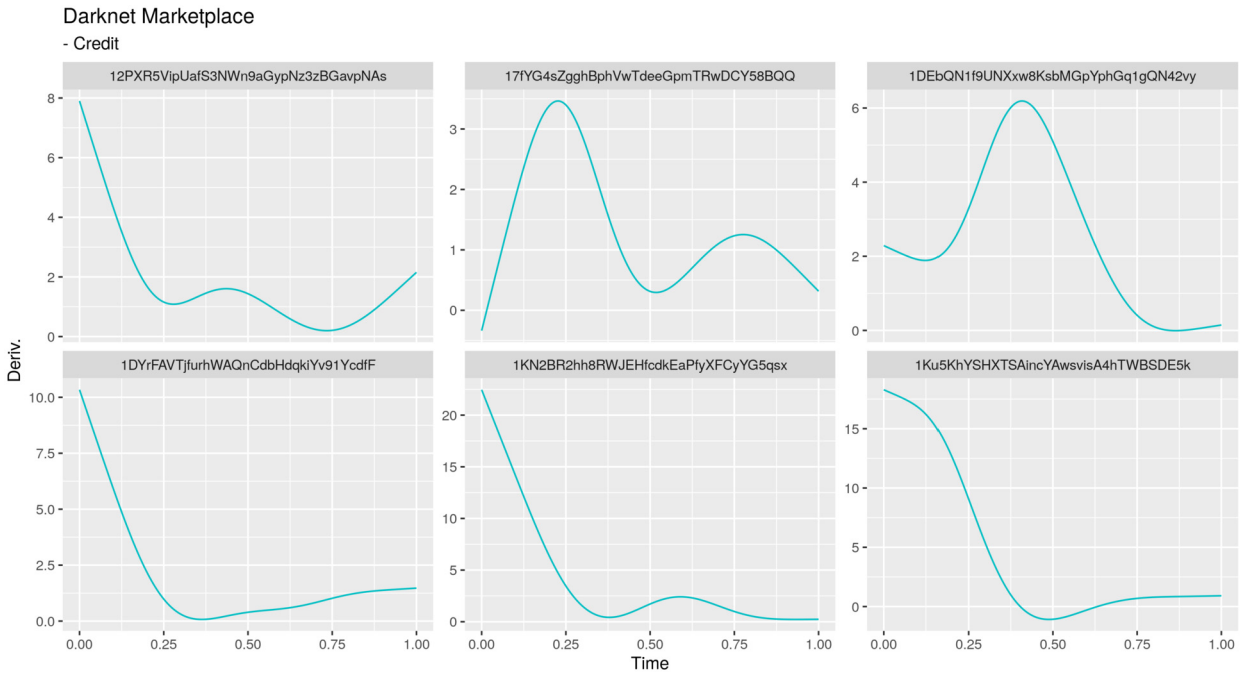

Fig. 6. Derivatives of the smoothed log-accumulated sum of credits for six darknet addresses.

where $N(1)$ is the number of arrivals in $[0, 1]$ and

$$f(t_1, \ldots, t_n) = \frac{n!}{M^n} \prod_{i=1}^{n} \mu(t_i), \quad t_1 \leq t_2 \leq \cdots \leq t_n, \quad t_i \in [0, 1], \quad M = \int_0^1 \mu(s)ds,$$

where $t_1, \ldots t_n$ are the observed arrival times up to the n-th arrival. Therefore, if we observe the time of arrival of the $N(1) = n$ events that occurred in $[0, 1]$, the likelihood becomes, where we notice the trivial equality $\mathbb{P}(T_1 \leq t_1, \ldots, T_n \leq t_n, N(1) = n) = \mathbb{P}(T_1 \leq t_1, \ldots, T_n \leq t_n \mid N(1) = n) \times \mathbb{P}(N(1) = n)$,

$$\begin{aligned} L(t_1, \ldots, t_n \mid \mu) &= f(t_1, \ldots, t_n) \times \mathbb{P}(N(1) = n) \\ &= \frac{n!}{M^n} \prod_{i=1}^{n} \mu(t_i) \times e^{-M(1)} \frac{M^n}{n!} = e^{-M} \prod_{i=1}^{n} \mu(t_i). \end{aligned}$$

Then, the log-likelihood function is given by

$$\log[L(t_1, \ldots, t_n \mid \mu)] = \sum_{i=1}^{n} \log \mu(t_i) - \int_0^1 \mu(s)ds.$$

Following the penalized functional regression framework, we can write μ in a basis expansion. However, this function has a new constraint: $\mu(t) > 0$. To account for that, we take the exponential:

$$\mu(t) = \exp[\mathbf{c}^T \boldsymbol{\phi}(t)].$$

Since the simplified hypothesis is that μ is constant, we penalize the log-likelihood maximization with the first derivative ($m = 1$) of $\boldsymbol{\phi}$. Then, the following problem must be numerically solved:

$$\underset{\mathbf{c}}{\operatorname{argmax}} \sum_{i=1}^{n} \mathbf{c}^T \boldsymbol{\phi}(t_i) - \int_\tau \exp[\mathbf{c}^T \boldsymbol{\phi}(s)]ds - \lambda \times \mathbf{c}^T \mathbf{R}\mathbf{c}.$$

We use the method described above, with a smoothing parameter of $\lambda = 0.1$ and penalization on the first derivative ($m = 1$), for each credit and debit curve - recalling that the observations are considered for the first 3000 hours. Results can be seen in Fig. 7. The resulting rate curves are less smooth than the other types of curves, which is natural given that the first derivative is used as penalization.

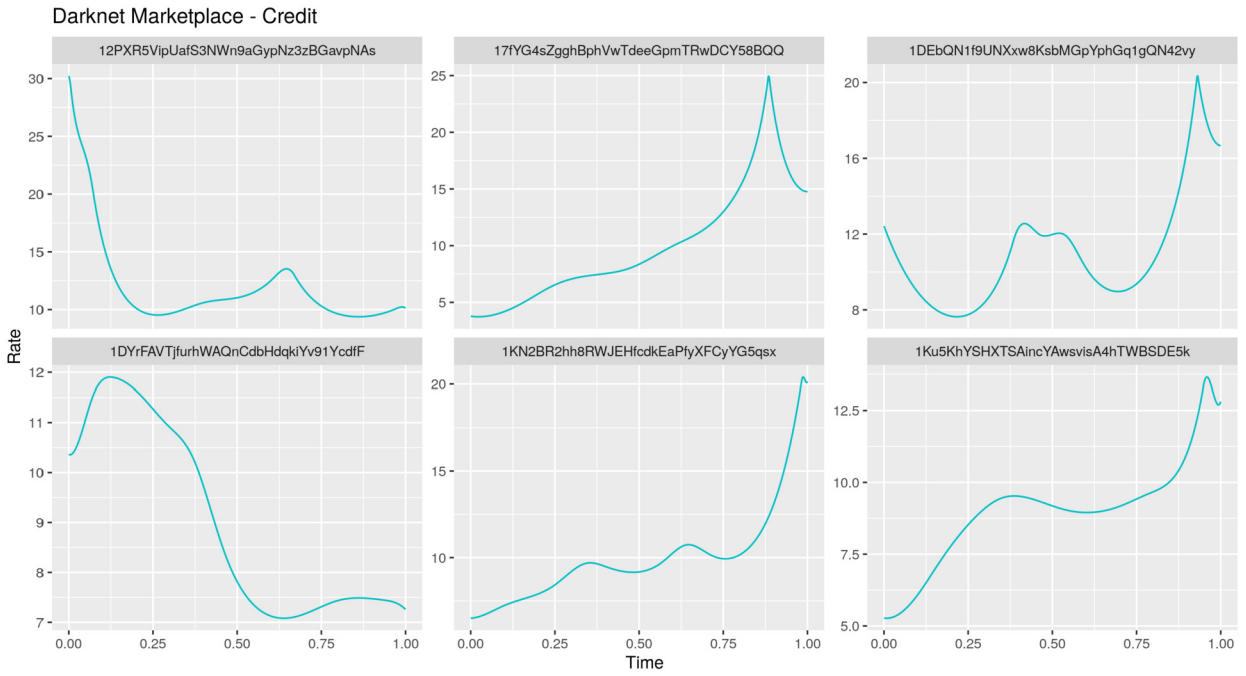

Fig. 7. Poisson rate of credits for six darknet addresses.

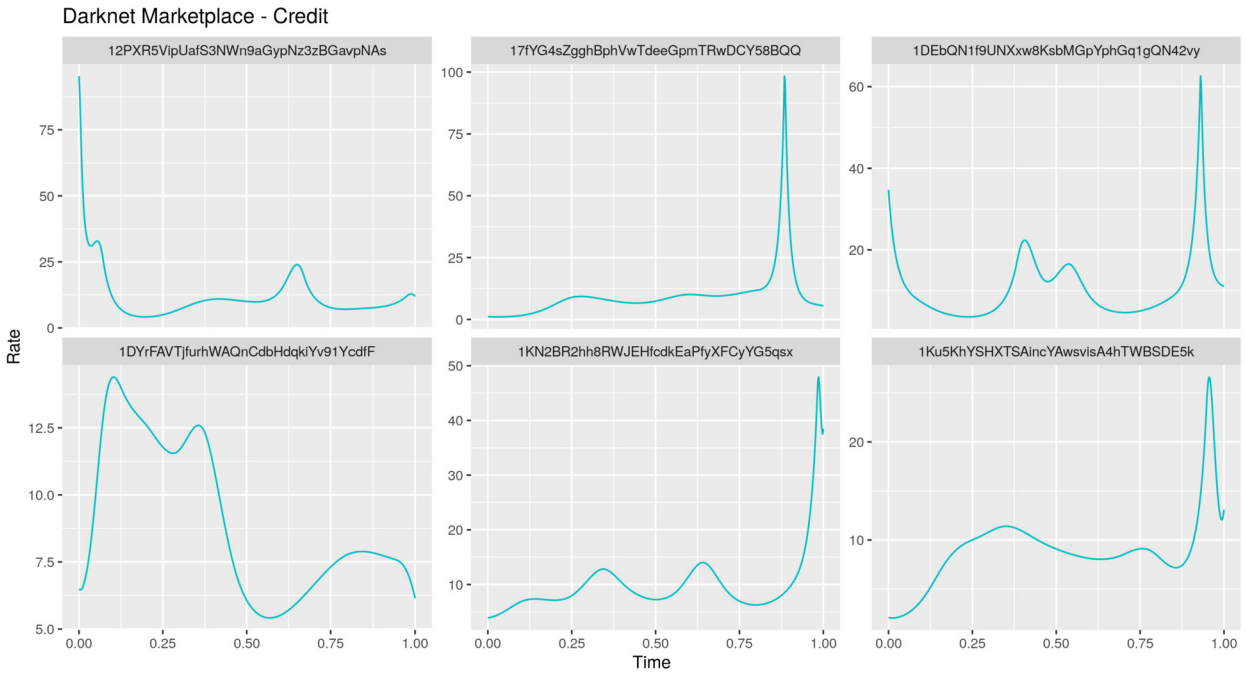

Fig. 8. Poisson rate of credits for six darknet addresses, using $\lambda = 0.01$.

As before, GCV criterion was used to investigate an appropriate value for λ, but results were also under-smoothed, and the heuristic approach was preferred, setting $\lambda = 0.1$. Smaller values of λ yielded estimations that we believed were too irregular for our classification purpose; Fig. 8 shows the estimation with $\lambda = 0.01$.

3. Classification models

The prediction of the addresses' classes is a supervised classification problem, since the classes are known. Four different algorithms were used to compare models: Multinomial Logistic Regression, Gradient Boosting, Support-Vector Machine and Random Forest. Details of each algorithm can be found in Friedman et al. (2001).

There are essentially two types of features to build models in this case: scalar and functional. The scalar features are chosen according to the problem at hand, and have no higher wisdom behind their engineering than the domain knowledge of the person responsible for developing them. They are present in the vector model described below. Functional features, on the other hand, are less arbitrary in the sense that they are representations of the function being analyzed. This is a result of the basis regression minimization problem expressed in Section 2.3. The domain specific choices, in this case, are the basis system, the number of basis functions and the smoothing parameter.

3.1. Vector model

The simplest model attempted is a vector model with scalar features built from credits and debts transactions. There is no functional treatment in this approach. The features considered here are: credit/debt count; credit/debt sum; credit/debt minimum; credit/debt maximum; credit/debt median; credit/debt first quantile; credit/debt third quantile; difference of third and first quantile values for credits/debts; credit/debt interval time (time elapsed between first and last credits/debts). The features regarding the difference of quantiles were not used with the multinomial logit model in order to avoid multicollinearity.

Additionally, constant Poisson rates (number of credits divided by interval time/number of debts divided by interval time) were added in order to compare the gains of a functional Poisson rate.

3.2. Functional model

We choose to adapt the multivariate classification algorithms to the functional context for one main reason: we want to be able to compare and combine models using features accounting for different curves and the vector approach.

As previously stated, the functional features are representation of the functions at hand; more specifically, they are the coefficients of the basis expansion of each curve. For this work, we choose the functional principal components (FPCs) basis, as it usually allows for the representation of the majority of the function's variance with few basis functions, thus preventing overfitting. In this case, the coefficients are the FPCs. We estimate them for the training set as described in Section 8.4.2 of Ramsay and Silverman (2005), for each curve type (accumulated balances, first derivative and Poisson rates for each credit and debt). This classical method requires the data to be sampled at the same timestamps. Then, the equally sampled data must be written on a previously defined basis: similarly to the smoothing procedure described in Section 2.3, we use B-splines to re-smooth the data, with penalization on the second derivative and knots at equally spaced times. Different re-smoothing parameters are tested in terms of classification performance; more details are given in Section 4.

Notice that the coefficients of the re-smoothing procedure are different from the first smoothing procedure described in Section 2, in which each function is smoothed individually, with different timestamps, in a pre-defined basis. However, smoothing data on the FPC basis cannot be done individually, because the FPC basis is an empirical basis; that is, smoothing on this basis is dependent on the set of observed curves. Since we are building a prediction model, it is important to differentiate how the principal components are computed for the training and test sets: the FPC basis functions are estimated on the training set, and then the test data set is smoothed over it. The validation data set is also re-smoothed in the same fashion.

However, to obtain the FPCs for the test set, we need to find the coefficients of each curve corresponding to the expansion on the eigenbasis estimated on the training set. Let $\boldsymbol{\Xi}$ be the $n \times L$ matrix containing the values $\hat{\xi}_i(t_j)$, where $\hat{\xi}_i$ is the approximation of the ith eigenfunction of the covariance operator, $\hat{\xi}_i(t) = \mathbf{b}_i^T \boldsymbol{\phi}(t)$, and we are considering the first L eigenfunctions. Hence $\boldsymbol{\Xi} = \boldsymbol{\Phi}\mathbf{B}$, where $\mathbf{B}$ is the $K \times L$ matrix whose columns are the coefficients of each eigenfunction in the given basis ϕ and $\boldsymbol{\Phi}$ as defined in Section 2.3. If $\mathbf{Z} = [Z_1, \ldots, Z_K]^T$ is the vector of FPCs, we can estimate it as in Section 2.3:

$$\hat{\mathbf{Z}} = \left(\boldsymbol{\Xi}^T \boldsymbol{\Xi} + \lambda \mathbf{R}^*\right)^{-1} \boldsymbol{\Xi}^T \mathbf{y}_{\mathbf{cen}},$$

with $\mathbf{y}_{\mathbf{cen}}$ being the centered observation $\mathbf{y} - \hat{\boldsymbol{\mu}}$, where $\hat{\boldsymbol{\mu}} = [\hat{\mu}(t_1), \ldots, \hat{\mu}(t_n)]$ and $\hat{\mu}$ is the mean of the smoothed curves in the training set at every t_j. Moreover, $\mathbf{R}^*$ is the penalization matrix of $\boldsymbol{\Xi}$ and it is given by $\mathbf{R}^* = \mathbf{B}^T\mathbf{R}\mathbf{B}$, where $\hat{\boldsymbol{\xi}} = [\hat{\xi}_1, \ldots, \hat{\xi}_L]^T$.

4. Results

Models such as the Multinomial Logistic Regression, Support-Vector Machine, Gradient Boosting and Random Forest accept multiple functional objects and also vector objects, which means that these features can be combined to build more

accurate classifiers. The R code and data for replicating this paper can be found at https://github.com/brendaprallon/FDA-bitcoin-paper. In the following section, the results for different combinations will be presented. The combinations are:

- Vector Model (only);
- Functional Model (with all types of curves);
- Vector Model combined with Functional Model;
- Vector model with constant Poisson rates;
- Vector model with functional Poisson rates;
- Vector model with functional derivatives and functional Poisson rates.

Regarding the prediction of the classes, undersampling was employed with the purpose of getting a balanced dataset (see Table 1): for a minimum of 10 transactions, we select 3129 samples of each class (the minimum number of addresses in one category - gambling); and for 20, 1741 samples (also the minimum of gambling addresses). 20% of the samples were set apart for testing the best model, and the best model was chosen by 5-fold cross-validation in the training set (80% of samples).

Multiple values for different parameters were tested: re-smoothing parameters of 10^{-10} and 10^{-1}; and, for the number of principal components to use to describe each functional attribute, 1, 3, 5, and 7 FPCs were used; however, they were tested for all the functions at a time, that is, one model considered 1 principal component for credit curves, 1 for debit curves, 1 for credit derivatives, 1 for debit derivatives, 1 for credit rates, 1 for debit rates; another model considered 3 principal components for credit curves, 3 for debit curves, 3 for credit derivatives... and so on.

Because we have performed undersampling to ensure the model was balanced - that is, we have the same number of observations for each class - evaluation of the models was done by computing their accuracy. In the validation process, overall, the random forest classification algorithm performed better, regardless of the number of principal components, re-smoothing parameter, or minimum number of transactions (10 or 20). Few exceptions were the vector model (only) or functional model, which for some combination of number of principal components and smoothing parameters were outperformed by the Gradient Boosting algorithm with both functional and vector features. The re-smoothing parameter did not seem to make a significant difference on the result – this makes sense, considering the curves were already smooth and the re-smoothing process is just part of the FPC estimation; neither did the number of FPCs (in Fig. 9, we can see that the number of principal components does not greatly affect the accuracy). Instead, what seems to be the most relevant factor for better accuracy are the combined functional and scalar features. The vector model combined with functional model, vector model with functional Poisson rates, and vector model with functional derivatives and functional Poisson rates consistently outperform the vector model (only), functional model (with all types of curves), and vector model with constant Poisson rates.

To properly access the accuracy of the estimation, the models were run with the best parameters for each algorithm using a test set. We report the results for the random forest algorithm, given that it was consistently better during validation. For a minimum of 10 observations, the best model was the vector model with functional derivatives and functional Poisson rates, re-smoothing parameter of 10^{-10} and 5 FPCs; for a minimum of 20 observations, this was the vector model combined with functional model, re-smoothing parameter of 0.1 and only 1 FPC.

Results for a minimum of 10 and 20 observations can be seen in Tables 2 and 3, respectively. Contrary to what was expected, there were no significant improvements when further restricting the sample to a minimum of 20 observations: the accuracy of the best models are around 0.72. However, as was observed during validation, combining scalar and functional features has the potential to improve accuracy between $3-5\%$, and it seems that functional features of derivatives and rates contribute more than the functional features of the curves themselves. Furthermore, it is worth noticing the functional model achieves similar results to the vector model - outperforming it by 3% in the case of a minimum of 10 observations, and being short of no more than 2% when using the 20 observations threshold. This shows that in a setting where the feature engineering process is too complex or where specific domain knowledge is scarce, representing the data with functional principal components could be a very good alternative, given that it is a straightforward method.

It does not appear that increasing the number of FPCs would aid in capturing more information. Fig. 9 show that 3 components explain the variance almost entirely, and that the effect of the number of components on accuracy is very limited. However, even though it is evident that the scalar features are important for classification, the information contained on the empirical basis of the curves mimics them quite well, even with few components.

The fact that the darknet and pool addresses are consistently better classified (Figs. 10 and 11) could mean that the sampling issue described in Section 2.2.1 is still present. The behavior of the addresses might change along with the bitcoin price, or the behavior is more consistent along the same entity, and categories with fewer entities would be easier to classify. This is corroborated by the low accuracy of the exchange category, which has 50 entities.

The feature importance plot of the random forest model, for a minimum of 10 observations (Fig. 12), points to the first FPC of the credit rates curves as being the most relevant. It is followed, with a significant difference, by the first FPC of the debit rates curves, and then the first FPC of the debit derivative; the other variables are much closer in terms of aiding prediction. For a minimum of 20 observations (Fig. 13), the first FPC of credit and debit rates curves continue to be among the three more important, but the number of credits gain more importance. The relative importance of the three most

Table 2
Accuracy of Random Forest models - Min 10 Obs.

	Vec.	Vec. + Const. Rate	Vec. + Fun. Rate	Vec. + Fun.	Fun.	Vec. + Deriv. + Fun. Rate
In Sample	1.000	1.000	1.000	1.000	1.000	1.000
Out of Sample	0.671	0.680	0.702	0.724	0.702	0.720

Table 3
Accuracy of Random Forest models - Min 20 Obs.

	Vec.	Vec. + Const. Rate	Vec. + Fun. Rate	Vec. + Fun.	Fun.	Vec. + Deriv. + Fun. Rate
In Sample	1.000	1.000	1.000	1.000	1.000	1.000
Out of Sample	0.688	0.691	0.713	0.716	0.673	0.723

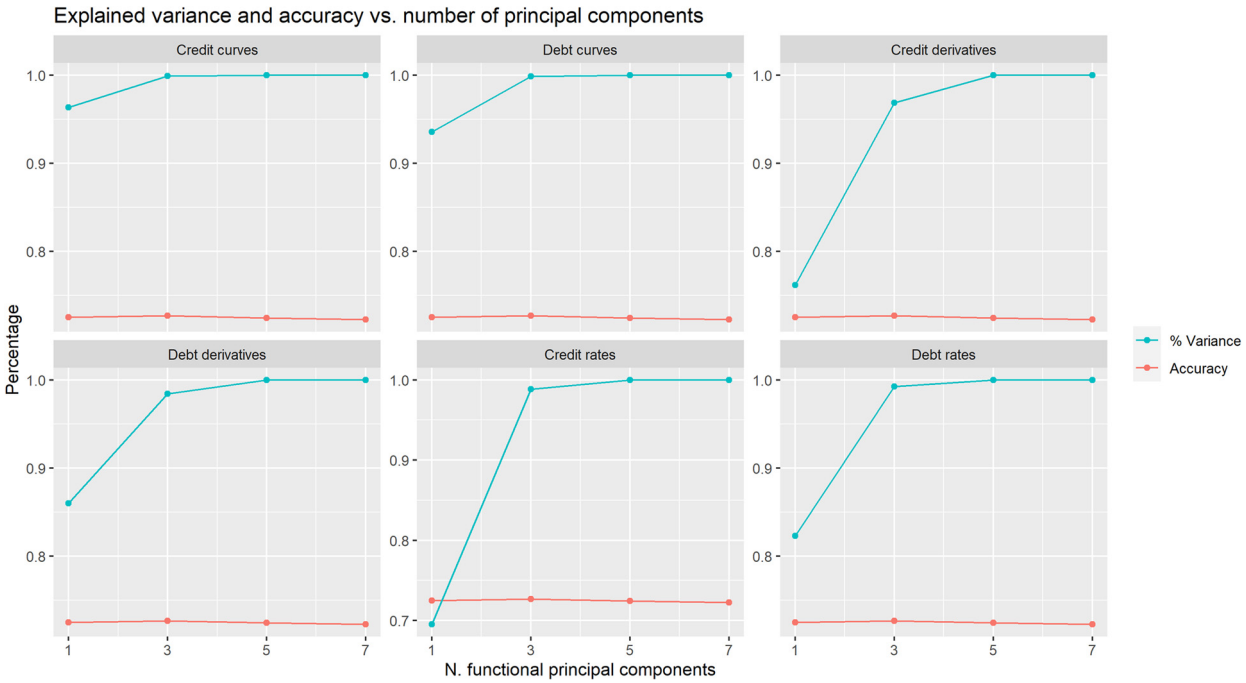

Fig. 9. Accuracy and proportion of explained accumulated variance for the winning model with different numbers of principal components, min. 20 obs.

important features to the followings is also large. It is clear that functional frequency measures are the most helpful in prediction, specially when considering addresses with fewer transactions.

4.1. Functional principal components

Fig. 14 shows resulting plots of adding and subtracting eigenfunctions (weighted by their standard deviations) around the mean curve. The plots are made with the balanced sample with a minimum of 20 observations; the equivalent plots for a minimum of 10 observations are very similar. The modes of variation of credits and debits are very alike for the log curves and derivatives, but vary a bit for the rates, even though they preserve the general behavior. Rates are also an exception when it comes to the smoothness of eigenfunctions; they are significantly rougher than their counterparts. This is expected since the rate curves are themselves rougher because penalization is on the first derivative, and no smoothness constraints were imposed for the resulting eigenfunctions.

Regarding the curves, the first mode of variation represents differences in level: this accounts for more than 95% of variance of accumulated credits, and over 90% of accumulated debits. The next eigenfunctions represent shifts around the mean: curves that start with more (less) volume than the mean, and, at some point in time, invert this relation. For derivatives, the most important mode of variation captures the behavior of curves that are faster (slower) than the mean at the beginning, shift their trend after about one quarter of the addresses life and then stabilize around the mean. Finally, the variability of rate curves is maximized by weighting heavily the initial instants of the addresses life. This makes sense since many addresses have transactions up until before 3000 hours, which concentrates the frequencies in the beginning.

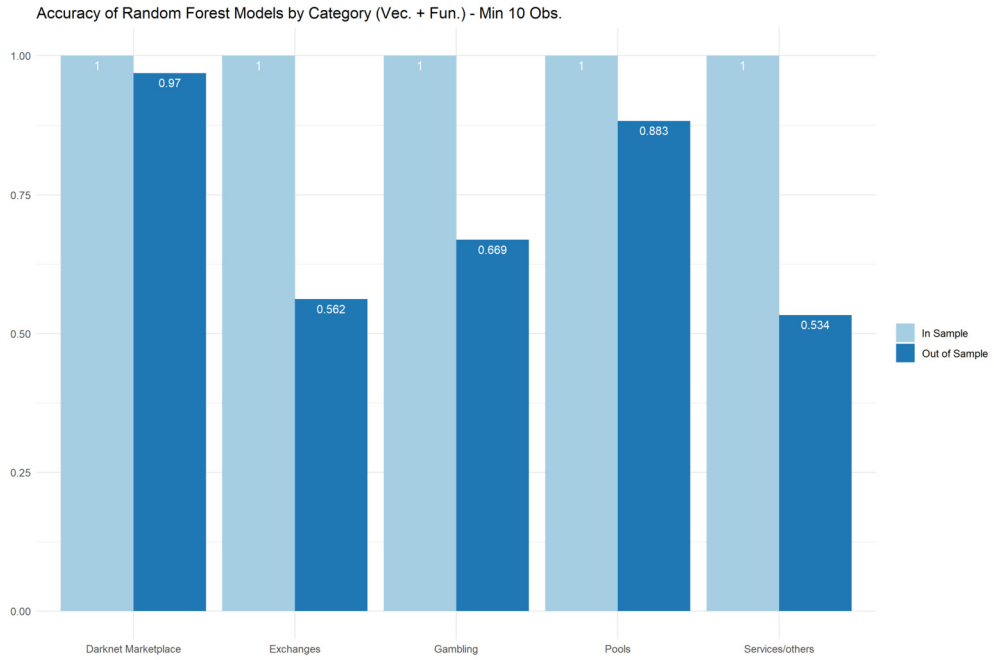

Fig. 10. Accuracy by category of the combining model using all scalar and functional features, min. 10 obs.

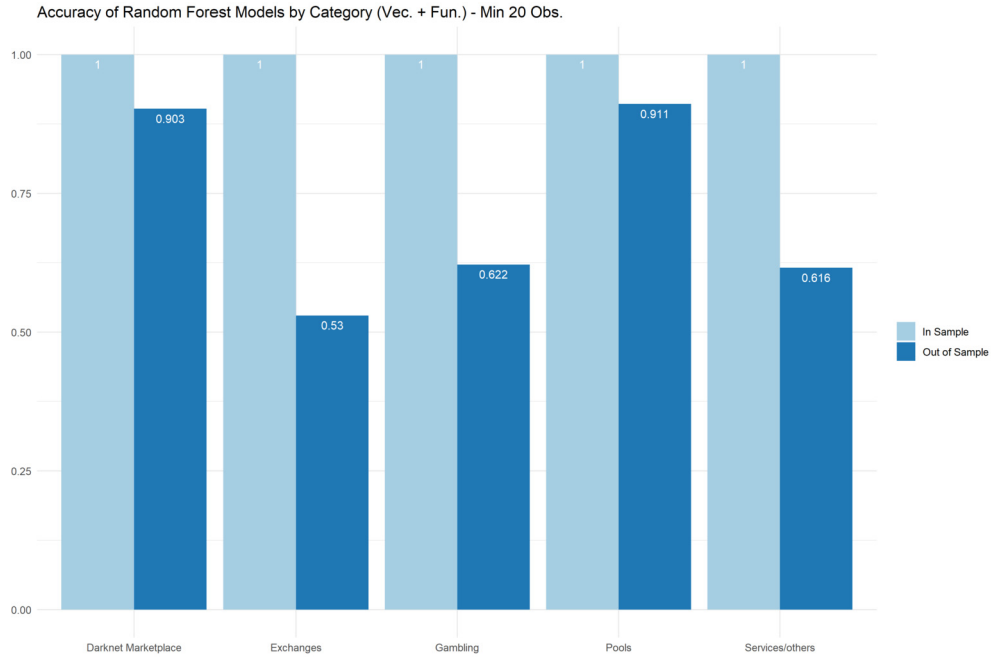

Fig. 11. Accuracy by category of the combining model using all scalar and functional features, min. 20 obs.

5. Conclusion

FDA methods were employed to properly represent the addresses' balances as functions; some treatment was required, such as removing addresses with few observations, and smoothing was applied to obtain the log curves, the derivatives and the Poisson rates. The functional principal component basis was used. At last, different models were estimated using functional, scalar and a combination of both variables. The functional variables were the FPCs given by smoothing procedures.

It has been shown that, for the available data, the random forest algorithm yields better results in general than the other tested algorithms (multinomial logit, SVM and gradient boosting). The best results are given by models combining scalar and

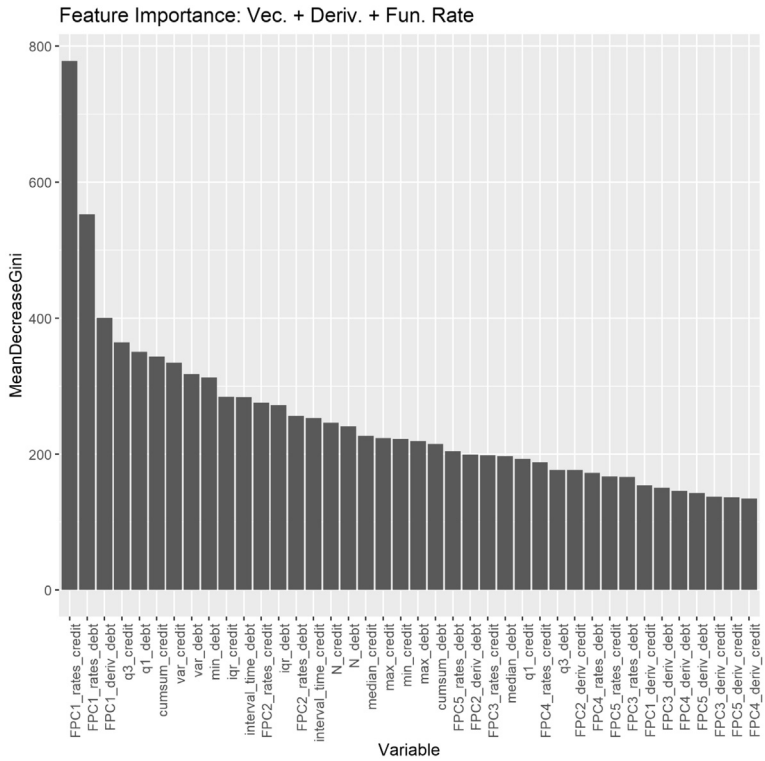

Fig. 12. Feature importance of the combining model using scalar, functional rates and derivatives, min. 10 obs.

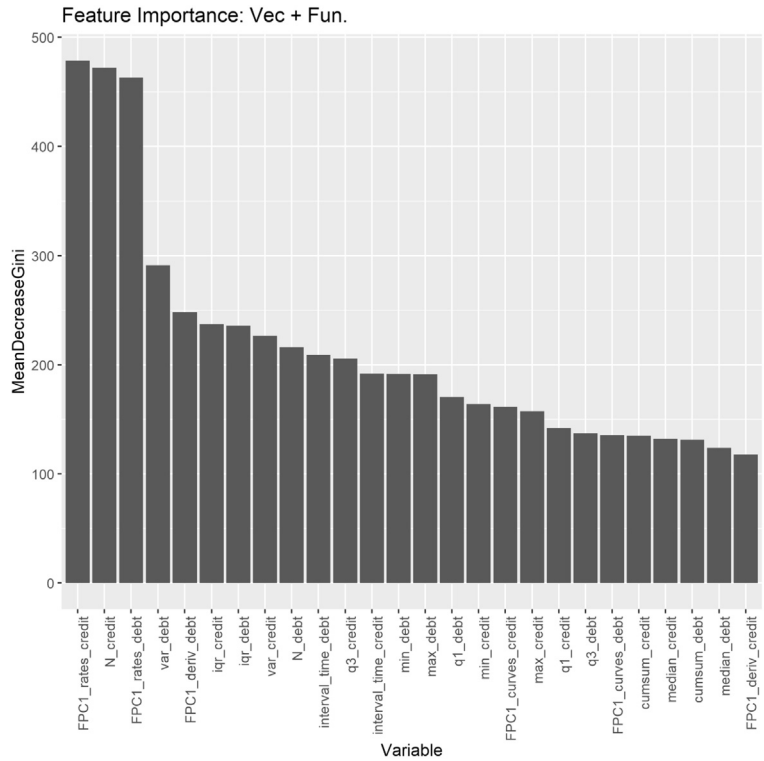

Fig. 13. Feature importance of the combining model using all scalar and functional features, min. 20 obs.

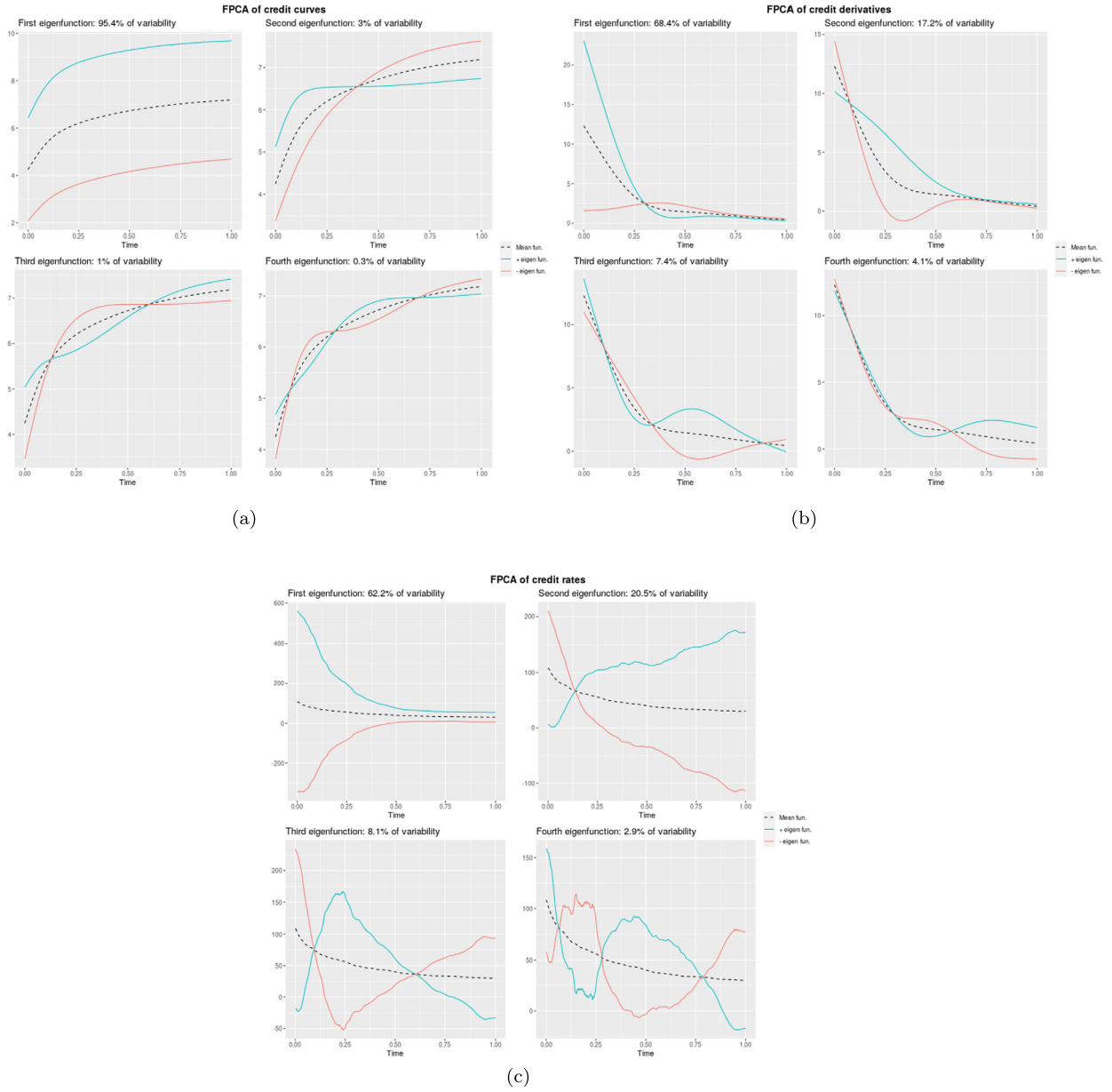

Fig. 14. (a) FPCA of credit curves, min. 20 obs. (b) FPCA of credit derivatives, min. 20 obs. (c) FPCA of credit rates, min. 20 obs.

functional variables: improvements are around 5% for a threshold of 10 observations. This points out that variance might not be the best way to separate the data in this case. However, the fact that the accuracy of the functional and vector model are very close highlights an important point: the principal components basis is capable of providing automatic features that are enough to replace domain-specific features. This can be specially useful in contexts where domain-specific knowledge is not available, or typical feature engineering is too costly. An additional highlight is that the model is capable of predicting classes without network information: for example, it is possible to predict addresses of new entities, which have possibly never interacted with any other address on the training set. This is not the common case in the literature and is our main contribution.

Furthermore, the darknet category has the best accuracy, while the exchange category has the worst. Given the distribution of their addresses over time, this could mean that there is still an implicit timestamp bias. The number of entities associated with each category may also contribute to the difference in accuracy across groups. One possible solution is to gather the addresses per entity and classify them instead, but more samples would be necessary. Another is to explore notions of depth for classification of more heterogeneous data. These are interesting proposals for future work.

Acknowledgements

YFS and BP were partly supported by Silicon Valley Community Foundation, grant number 199610, and by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001. The research by Manuel Febrero–Bande and Wenceslao González–Manteiga has been partially supported by MCIN/AEI/10.13039/501100011033, grant number PID2020-116587GB-I00, through the European Regional Development Funds (ERDF). We thank the Associate Editor and two anonymous referees for their insightful comments, which improved the quality and clarity of our paper substantially.

References

Aneiros, G., Novo, S., Vieu, P., 2021. Variable selection in functional regression models: a review. J. Multivar. Anal. 104871.
Christin, N., 2013. 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.
Cuevas, A., Febrero, M., Fraiman, R., 2007. Robust estimation and classification for functional data via projection-based depth notions. Comput. Stat. 22 (3), 481–496.
Escabias, M., Aguilera, A., Valderrama, M., 2004. Principal component estimation of functional logistic regression: discussion of two different approaches. J. Nonparametr. Stat. 16 (3–4), 365–384.
Escabias, M., Aguilera, A., Valderrama, M., 2005. Modeling environmental data by functional principal component logistic regression. Environmetrics: The Official Journal of the International Environmetrics Society 16 (1), 95–107.
Febrero-Bande, M., González-Manteiga, W., 2013. Generalized additive models for functional data. Test 22 (2), 278–292.
Febrero-Bande, M., González-Manteiga, W., Prallon, B., Saporito, Y., 2022. Functional classification of bitcoin addresses. Preprint. Available at: arXiv:2202.12019.
Febrero-Bande, M., Oviedo de la Fuente, M., 2012. Statistical computing in functional data analysis: the R package fda.usc. J. Stat. Softw. 51 (4), 1–28.
Foley, S., Karlsen, J.R., Putniņš, T.J., 2019. Sex, drugs, and bitcoin: how much illegal activity is financed through cryptocurrencies? Rev. Financ. Stud. 32 (5), 1798–1853.
Friedman, J., Hastie, T., Tibshirani, R., 2001. The Elements of Statistical Learning. Springer Series in Statistics. Springer New York, NY.
Hall, P., Hosseini-Nasab, M., 2006. On properties of functional principal components analysis. J. R. Stat. Soc., Ser. B, Stat. Methodol. 68 (1), 109–126.
Hall, P., Poskitt, D.S., Presnell, B., 2001. A functional data—analytic approach to signal discrimination. Technometrics 43 (1), 1–9.
Hu, Y., Seneviratne, S., Thilakarathna, K., Fukuda, K., Seneviratne, A., 2019. Characterizing and detecting money laundering activities on the bitcoin network. Preprint. arXiv:1912.12060.
Jacobsen, M., Gani, J., 2006. Point Process Theory and Applications: Marked Point and Piecewise Deterministic Processes. Springer.
Jiang, C.-R., Chen, L.-H., 2020. Filtering-based approaches for functional data classification. Wiley Interdiscip. Rev.: Comput. Stat. 12 (4), e1490.
Jourdan, M., Blandin, S., Wynter, L., Deshpande, P., 2018. Characterizing entities in the bitcoin blockchain. In: 2018 IEEE International Conference on Data Mining Workshops (ICDMW). IEEE, pp. 55–62.
Lee, H.-J., 2004. Functional Data Analysis: Classification and Regression. Texas A&M University.
Leng, X., Müller, H.-G., 2006. Classification using functional data analysis for temporal gene expression data. Bioinformatics 22 (1), 68–76.
Li, H., Xiao, G., Xia, T., Tang, Y.Y., Li, L., 2013. Hyperspectral image classification using functional data analysis. IEEE Trans. Cybern. 44 (9), 1544–1555.
Li, Y., Qiu, Y., Xu, Y., 2022. From multivariate to functional data analysis: fundamentals, recent developments, and emerging areas. J. Multivar. Anal. 188, 104806.
Ling, N., Vieu, P., 2021. On semiparametric regression in functional data analysis. Wiley Interdiscip. Rev.: Comput. Stat. 13 (6), e1538.
López-Pintado, S., Romo, J., 2007. Depth-based inference for functional data. Comput. Stat. Data Anal. 51 (10), 4957–4968.
Meiklejohn, S., Pomarole, M., Jordan, G., Levchenko, K., McCoy, D., Voelker, G.M., Savage, S., 2013. A fistful of bitcoins: characterizing payments among men with no names. In: Proceedings of the 2013 Conference on Internet Measurement Conference, pp. 127–140.
Müller, H.-G., Stadtmüller, U., et al., 2005. Generalized functional linear models. Ann. Stat. 33 (2), 774–805.
Piotr, K., Hanny, O., Byeong, P., Sangalli, L.M., 2017. Special issue on functional data analysis.
Ramsay, J., Wickham, H., Ramsay, M.J., deSolve, S., 2020. Package 'fda'.
Ramsay, J.O., Silverman, B.W., 2005. Functional Data Analysis, 2nd ed. Springer.
Song, J.J., Deng, W., Lee, H.-J., Kwon, D., 2008. Optimal classification for time-course gene expression data using functional data analysis. Comput. Biol. Chem. 32 (6), 426–432.
Thompson, W., 1988. Point Process Models with Applications to Safety and Reliability. Springer Science & Business Media.
Wang, J.-L., Chiou, J.-M., Müller, H.-G., 2016. Functional data analysis. Annu. Rev. Stat. Appl. 3, 257–295.
Yao, F., Müller, H.-G., Wang, J.-L., 2005. Functional data analysis for sparse longitudinal data. J. Am. Stat. Assoc. 100 (470), 577–590.