Bayesian Analysis (2022) **17**, Number 1, pp. 261–290

Bayesian Quickest Detection of Credit Card Fraud

Bruno Buonaguidi*, Antonietta Mira†, Herbert Bucheli‡, and Viton Vitanis§

Abstract. This paper addresses the risk of fraud in credit card transactions by developing a probabilistic model for the quickest detection of illegitimate purchases. Using optimal stopping theory, the goal is to determine the moment, known as disorder or fraud time, at which the continuously monitored process of a consumer's transactions exhibits a disorder due to fraud, in order to return the best trade-off between two sources of cost: on the one hand, the disorder time should be detected as soon as possible to counteract illegal activities and minimize the loss that banks, merchants and consumers suffer; on the other hand, the frequency of false alarms should be minimized to avoid generating adverse effects for cardholders and to limit the operational and process costs for the card issuers. The proposed approach allows us to score consumers' transactions and to determine, in a rigorous, personalized and optimal manner, the threshold with which scores are compared to establish whether a purchase is fraudulent.

Keywords: Bayesian model, credit card fraud detection, optimal stopping theory.

MSC2020 subject classifications: Primary 62H30, 60G40; secondary 65C60.

1 Introduction

Payment habits have changed dramatically over the past thirty years thanks to new technologies. Nowadays a growing number of purchases, also of small amounts, are paid by credit and debit cards. While these electronic payment methods boost business and make the lives of buyers easier, they also heighten the fraud risk borne by the payments industry. As shown by the Nilson Report (2017), in 2016 worldwide fraud losses amounted to 7.15 cents per $ 100 of card transactions; the card total volume was $ 31.878 trillion, while fraud losses reached $ 22.80 billion and the latter amount has been predicted to double by 2025. The total cost is even greater when the consequences of card fraud are considered: banks and card issuers make investments in anti-fraud technologies and bear the losses incurred by their clients; merchants sustain high cost to guarantee their customers high standard of security and can be charged back by card issuers if any negligence during a transaction occurs; finally, clients, often refunded by banks when victimized by fraud, are frustrated when their cards are blocked unnecessarily. Thus, counteracting credit card fraud is in the interest of all these actors.

*Università Cattolica del Sacro Cuore, Department of Statistical Sciences and Università della Svizzera italiana, Institute of Computational Science, bruno.buonaguidi@unicatt.it

†Università della Svizzera italiana, Institute of Computational Science and Università dell'Insubria, Department of Science and High Technology, antonietta.mira@usi.ch

‡Aduno Gruppe, Viseca Card Services SA, Zurich, Switzerland and Aarhus University, Department of Economics and Business Economics, Herbert.Bucheli@aduno-gruppe.ch

§Aduno Gruppe, Viseca Card Services SA, Zurich, Switzerland, Viton.Vitanis@aduno-gruppe.ch

 https://doi.org/10.1214/20-BA1254

Credit card fraud occurs whenever a credit card is used without the consent of its legitimate owner with the aim of either making purchases or stealing money. As first defense level against fraud, card issuers have developed authentication measures, such as the check of numerical codes (like the card PIN or the cardholder's zip code), signature and fingerprint verification systems and the 3D secure scheme (an authentication method that requires a cardholder either to insert a temporary generated password for finalizing her on-line transaction or to authorize the transaction over a second channel). However, since fraudsters dynamically adapt their strategies to the latest anti-fraud technologies, authentication measures may fail. Then, as second defense level, card issuers make use of detection measures to discriminate between legitimate and fraudulent transactions; currently employed fraud detection techniques are discussed in Section 2. Here, let us recall that these techniques are supervised methods, namely they are calibrated on a training sample of transactions which are labeled as legitimate or fraudulent: when a new transaction arrives, the trained model predicts its class. In particular, a suspicion score between 0 and 1 is assigned to each transaction, which is subsequently declared as fraudulent when the score is higher than a certain threshold.

In the literature this threshold is determined empirically: for example, in Bhattacharyya et al. (2011); Mahmoudi and Duman (2015); Quah and Sriganesh (2008); Srivastava et al. (2008); Zaslavsky and Strizhak (2006) default values, such as 0.3 or 0.5, are used; in Jurgovsky et al. (2018), the threshold is fixed so that the training set is characterized by a predetermined true positive rate; in Carneiro et al. (2017), the threshold is such that either in the training set the false positive and the true positive rates are equal or a predetermined percentage of the top most rated transactions is labeled as illegal. In any case, to the best of the authors' knowledge, there is no formal theory which justifies these choices and in Carneiro et al. (2017) it is said that "it is critical to choose the score threshold for considering an order to be legitimate or fraudulent". This paper aims therefore at introducing a probabilistic model for the quickest detection of credit card fraud where for each transaction the posterior probability of being fraudulent is returned and a personalized threshold for each cardholder is optimally determined. The unobservable *disorder* or *fraud time*, at which the continuously monitored process of a consumer's transactions exhibits a disorder due to fraud, can be estimated as the first time the posterior probability process exceeds the threshold. This is the optimal stopping time, which minimizes the expected trade-off between the probability of having a false positive and the detection delay since the occurrence of the fraud.

The quickest detection problem of a change in the probabilistic features of a stochastic process has been widely studied. In Shiryaev (1978, Chap. 4.4) the early detection of a change in the drift of a Brownian motion was analyzed and was extended to a finite horizon formulation in Gapeev and Peskir (2006) and to other diffusion processes in Johnson and Peskir (2017); Gapeev and Shiryaev (2013). Partial solutions for the detection of a shift in the intensity of a Poisson process are in Davis (1976); Gal'Chuk and Rozovskii (1971), while a complete solution was provided in Peskir and Shiryaev (2002). The latter is the basis of this article, where, according to Schmittlein et al. (1987), it is assumed that the observed process of a card user's expenditures follows a compound

Poisson process, whose arrival times are the purchase times and whose jumps represent the corresponding amounts and the geographical coordinates. This process will change its intensity and jump distribution when hit by fraud, which we detect by resorting to the algorithm developed in Dayanik and Sezer (2006). We underline that this is the first time that the optimal stopping theory (see, e.g., Peskir and Shiryaev (2006); Shiryaev (1978)) and the results of the aforementioned articles are applied to credit card fraud detection. Further results on the quickest detection for compound Poisson processes were obtained in Bayraktar and Dayanik (2006); Bayraktar et al. (2005); Buonaguidi and Muliere (2015); Gapeev (2005); Herberts and Jensen (2004).

The article is organized as follows. In Section 2 we briefly recall the data mining techniques currently used in credit card fraud detection. In Section 3 we introduce three Bayesian quickest detection models and we describe how we can adapt them to credit card fraud detection. We analyze the optimal strategy to raise the alarm of fraud and we see that it is the first time a functional of the observed cardholder's expenditures pattern, known as posterior probability process or, equivalently, (generalized) odds process, exceeds a threshold; we also describe the algorithm that can be used to compute it. In Section 4, using real credit card transactions provided by one leading company in the Swiss credit card market, we estimate the pre and post-fraud expenditure distribution parameters of the cardholders and these values will be used to compute their optimal thresholds. Then, we assess how our methodology performs in classifying new legitimate and fraudulent credit card transactions (both simulated and real); performance of our models will be derived, discussed and compared with that of other methodologies. Section 5 contains a summary discussion and concluding remarks.

2 Literature review on credit card fraud detection

Data mining refers to the discovery of patterns and relationships in a huge amount of data and is widely used to screen credit card transactions to detect fraud. Detection must work in real time: when the details of a transaction are received by a credit card issuer, the latter must decide within a few milliseconds if the transaction must be authorized or not. This step is of key importance, because approving a fraudulent operation implies the loss of the corresponding amount; on the other side, rejecting a legal purchase creates disturbances for a cardholder. In this section we recall some of the most popular data mining methodologies employed in credit card fraud detection; literature reviews are also presented in Bolton and Hand (2002); Ngai et al. (2011).

Logistic regression and *rule-based methods* are well known and among the first techniques employed in fraud detection due to their simplicity. Logistic regression is just a special case of the generalized linear model; in rule-based methods, rules are either established by experts on the basis of prior analysis or extracted from *decision trees*. Applications to credit card fraud of logistic regression and rule-based methods can be found in Bahnsen et al. (2016); Bhattacharyya et al. (2011); Carneiro et al. (2017); Yeh and Lien (2009) and Bahnsen et al. (2016); Mahmoudi and Duman (2015); Yeh and Lien (2009), respectively; we also refer to Letham et al. (2015) for a Bayesian approach to rule-based methods.

Ensemble methods are used to improve the classification accuracy. They are made up of an aggregation of different classification models: a training set is used to create training subsets and on each of them a model is calibrated. When a new transaction arrives, each model returns a class prediction, which is used together with the predictions of the other models to determine the class of the ensemble. Boosting and random forests are two examples of ensemble methods. In boosting, used in Chan et al. (1999), the models are trained sequentially: the transactions which have not been correctly classified in the previous model are weighted more in the next one, in order to give more importance to the misclassified cases. The ensemble class is a weighted average of each model class, whose weight depends on how well the model performed. A Bayesian version of boosting, known as BART (Bayesian Additive Regression Trees), was proposed in Chipman et al. (2010). Random forests are an ensemble of decision trees built on sub-samples randomly drawn with replacement from the original training set. The class that a random forest assigns to a new transaction is the mode of the classes predicted by the single decision trees. Random forests were applied in Bahnsen et al. (2016); Bhattacharyya et al. (2011); Carneiro et al. (2017) and used in a Bayesian inference setting in Raynal et al. (2019). Unlike the rule-based and decision tree methods, ensemble methods are less prone to over-fitting but their interpretation is more complex.

A *hidden Markov model* is a stochastic process with two hierarchical levels: the inner one is represented by a finite number of states and is hidden, namely not observable, while the outer one is the observable outcome generated in correspondence to a given state. Probabilities governing the transition among states and probabilities with which outcomes are generated are the model parameters. Hidden Markov models were used for credit card fraud detection in Srivastava et al. (2008): for each purchase type (the hidden state), the price range (low, medium and high) is observed. The model works as follows: after the parameters estimation, a new transaction arrives and its price range is passed to the model; the latter computes the probability that the transaction is characterized by the observed price range. When this probability is too low, the transaction deviates from the normal behavior and is therefore identified as fraudulent. This methodology has a nice probabilistic interpretation, but its structure (number of states and number of outcomes for each state) needs to be carefully adapted. We refer to Ko et al. (2015) for a Bayesian approach to hidden Markov models in change-point problems.

Support vector machines are techniques used to separate data. Data can be either linearly separable, when there exists a hyperplane that divides all the data of one class from the data of the other class, or linearly inseparable, when such a hyperplane does not exist. However, in the latter case, using a non linear mapping, the original data can be mapped to a higher dimensional space where the transformed data becomes linearly separable. Support vector machines aim at searching for the best hyperplane separating the training data, namely the hyperplane with maximal margin of separation between the edges (the so called support vectors) of the two classes. Once a support vector machine has been trained on a set of credit card transactions, a new transaction is classified as fraudulent or legitimate depending on which of the two portions of the space, determined by the estimated hyperplane, the explanatory variables lie. Limits of this methodology are the choice of the function that maps linearly inseparable data to linearly separable ones and the specificity of this function to the addressed problem.

Support vector machines were investigated in Bhattacharyya et al. (2011); Carneiro et al. (2017); Mahmoudi and Duman (2015) and in Polson et al. (2015) their Bayesian version was provided.

Neural networks have become very popular among card issuers thanks to their ability to extract solutions from highly involved problems. A neural network is always characterized by a set of nodes, or neurons, connections among the neurons and a function which weights these connections. Neurons are placed in one or more layers: each neuron of one layer receives inputs from the neurons of the previous layer and combines this information with the weights of its connections. This operation propagates up to the neurons of the last layer, which return the final output. A set of labeled credit card transactions is used to train the model: an error function measures the distance between the output of the model and the true output. As this distance function depends only on the weights of the connections among neurons, the weights minimizing the error are searched by means of optimization algorithms. Despite their good performance, neural networks have some drawbacks: they are a "black box", in the sense that the function that they aim at optimizing cannot be inferred from the network structure; their topology (number of neurons and layers) strongly depends on the specific problem to be addressed; optimization algorithms do not always converge to the optimal set of weights that minimize the error function (see, e.g., Bishop (2006, Secs. 5.2.1 and 5.5) and Hastie et al. (2009, Secs. 10.7 and 11.5.4–11.5.5)). Neural networks were applied in Dorronsoro et al. (1997); Jurgovsky et al. (2018); Mahmoudi and Duman (2015); Quah and Sriganesh (2008); Yeh and Lien (2009); Zaslavsky and Strizhak (2006). A Bayesian perspective on neural networks and on their connection to statistical data reduction techniques was given in Polson and Sokolov (2017); methods of Bayesian optimization for hyperparameter selection in neural networks (as well as in logistic regression and support vector machines) were studied in Snoek et al. (2012).

3 Methodology description

A common feature of the methodologies analyzed in Section 2 is that they return a suspicion score in [0,1] on how likely a transaction is fraudulent. Then, a transaction will be labeled as fraudulent if the score exceeds a fixed threshold. However, there is no theory which explains how to compute it optimally and, as already said in Section 1, it is usually fixed empirically. In the following sections, we introduce our model, which provides a rigorous and personalized method to determine a threshold and the associated optimal strategy for each cardholder, so that the trade-off between the losses from detecting fraud too early or too late are minimized.

3.1 The model

We describe our model following the lines in Peskir and Shiryaev (2002). On the measurable space $(\Omega, \mathscr{F})$ the random variable θ is defined with respect to a family of probability measures $(\mathbb{P}^s)_{s\geq 0}$, such that $\mathbb{P}^s(\theta = s) = 1$. θ represents the so called *fraud* or *disorder time*, at which the expenditures pattern $X := (X_t)_{t\geq 0}$ of a cardholder changes its statistical features due to fraud. According to the hypothesis in Schmittlein et al. (1987)

(see also Glady et al., 2009), we assume that X is a compound Poisson process:

$$X_t := \sum_{j=1}^{N_t} Y_j, \quad \mathbb{P}^s(X_0 = 0) = 1,\ s \geq 0. \tag{3.1}$$

In the expression above, $N := (N_t)_{t\geq 0}$ is a standard Poisson process, which models the purchases time, and $\{Y_j\}_{j\geq 1}$ is a sequence of independent and identically distributed $\mathbb{R}^d$-valued random variables, representing, for example, the amount of the transactions and their geographical coordinates. At the disorder time θ, N changes its arrival rate from λ_0 to λ_1 and $\{Y_j\}_{j\geq 1}$ switch their common distribution from $v_0(\cdot)$ to $v_1(\cdot)$. It is assumed that $\lambda_i > 0$ and $v_i(\cdot)$, defined on $(\mathbb{R}^d, \mathscr{B}(\mathbb{R}^d))$, are all known, $i = 0, 1$, and that $v_1(\cdot)$ is absolutely continuous with respect to $v_0(\cdot)$.

Next, we define the probability measure

$$\mathbb{P}_\pi := \pi\mathbb{P}^0 + (1-\pi)\int_0^\infty \lambda e^{-\lambda s}\mathbb{P}^s\, ds, \quad \pi \in [0,1),\ \lambda > 0, \tag{3.2}$$

where π and λ are given and fixed. Expression (3.2) contains our prior belief about θ, that, under $\mathbb{P}_\pi$, takes value 0 with probability π and, with probability $1-\pi$, is exponentially distributed with mean $1/\lambda$. Since fraud is not directly observable, the best we can do is detecting θ through a strategy based on the continuous monitoring of X. Let $\mathscr{F}_t^X \subset \mathscr{F}$ be the sigma-algebra generated by X up to t; then our goal is to determine a stopping time τ with respect to $\mathscr{F}^X = (\mathscr{F}_t^X)_{t\geq 0}$ which is as close as possible to θ. Formally, the *Bayesian quickest detection problem* aims at computing one of the following risk functions

$$V_1(\pi) := \inf_\tau \left(\mathbb{P}_\pi(\tau < \theta) + c_1\mathbb{E}_\pi[(\tau-\theta)^+]\right), \tag{3.3}$$

$$V_2(\pi) := \inf_\tau \left(\mathbb{E}_\pi[(\theta-\tau)^+] + c_2\mathbb{E}_\pi[(\tau-\theta)^+]\right), \tag{3.4}$$

$$V_3(\pi) := \inf_\tau \left(\mathbb{P}_\pi(\tau < \theta) + c_3\mathbb{E}_\pi[e^{\alpha(\tau-\theta)^+} - 1]\right), \tag{3.5}$$

and obtaining the optimal stopping time at which the infimum on the right-hand side of (3.3)–(3.5) is achieved. In the above expressions, c_i, $i = 1, 2, 3$, and α are positive and given values and $(x)^+ := \max\{0, x\}$. In (3.3) the probability of a false positive (stopping before the fraud time) and the expected linear detection delay are combined: the longer the observation of X, the lower the probability of raising a false alarm, but the higher the delay in detecting θ; moreover, c_1 weights the importance assigned to these two sources of costs. Analogous interpretations hold for (3.4), where the expected advance in detecting θ replaces the probability of a false alarm, and for (3.5), where the expected exponential detection delay is considered (see, e.g., Beibel, 2000; Poor, 1998), with α being the internal rate of return at which the losses due to a detection delay are compounded. It will soon be evident that the structure of the solutions to (3.3)–(3.5) is similar, as already noticed in Bayraktar et al. (2005); Davis (1976).

3.2 The optimal stopping problem

We are going to see that problems (3.3)–(3.5) can be reduced to an equivalent optimal stopping problem for a one-dimensional Markov process. Let us introduce the processes $\Pi := (\pi_t)_{t\geq 0}$, $\varphi := (\varphi_t)_{t\geq 0}$ and $\Phi_\alpha := (\Phi_{\alpha,t})_{t\geq 0}$ defined by

$$\pi_t := \mathbb{P}_\pi\left(\theta \leq t | \mathscr{F}_t^X\right), \qquad \varphi_t := \frac{\pi_t}{1-\pi_t}, \qquad \Phi_{\alpha,t} := \frac{\mathbb{E}_\pi\left[e^{\alpha(t-\theta)}\mathbf{1}_{\{\theta\leq t\}}|\mathscr{F}_t^X\right]}{1-\pi_t}. \tag{3.6}$$

Since π_t is the posterior probability that fraud has already occurred at time t given all the past history of X, Π is called posterior probability process; φ is known as odds process because φ_t is the odds of π_t. Φ_α is the generalized odds process because $\Phi_\alpha = \varphi$ when $\alpha = 0$; for this reason, in the sequel we will refer to Φ_α only. Resorting to stochastic calculus and standard arguments based on Dayanik and Sezer (2006); Peskir and Shiryaev (2002), it is easy to show that Φ_α satisfies the following stochastic differential equation:

$$d\Phi_{\alpha,t} = \big(\lambda + (\alpha + \lambda - \lambda_1 + \lambda_0)\Phi_{\alpha,t}\big)dt + \Phi_{\alpha,t-}\int_{\mathbb{R}^d}\left(\frac{\lambda_1}{\lambda_0}f(x) - 1\right)\mu(dx, dt), \tag{3.7}$$

where $\Phi_{\alpha,0} = \frac{\pi}{1-\pi}$, $f(x)$ is the Radon-Nykodym derivative of $v_1(\cdot)$ with respect to $v_0(\cdot)$ and $\mu(t, A)$ is the random measure of the jumps of X in $A \in \mathscr{B}(\mathbb{R}^d)$ over the time interval $(0, t]$ and are defined by

$$f(x) := dv_1(x)/dv_0(x), \quad \mu(t, A) := \sum_{s\leq t}\mathbf{1}_{\{\Delta X_s \in A\}}. \tag{3.8}$$

The dynamic in (3.7) shows that Φ_α is a strong Markov process. Adapting the results in Johnson and Peskir (2017) to Φ_α, it is possible to show that problems (3.3)–(3.5) are equivalent to:

$$V_i(\pi) = (1-\pi)\left(h_i + q_i U_i\left(\frac{\pi}{1-\pi}\right)\right), \quad U_i(\phi) := \inf_\tau \mathbb{E}_\phi^\infty\left[\int_0^\tau e^{-\lambda t}(\Phi_{\alpha,t} - k_i)dt\right], \tag{3.9}$$

where $\alpha = 0$, when $i = 1, 2$, and

$$h_i,\ q_i,\ k_i = \begin{cases} 1,\ c_1,\ \dfrac{\lambda}{c_1}, & i = 1, \\ \dfrac{1}{\lambda},\ c_2,\ \dfrac{1}{c_2}, & i = 2, \\ 1,\ c_3\alpha,\ \dfrac{\lambda}{c_3\alpha}, & i = 3. \end{cases} \tag{3.10}$$

The infimum in (3.9) is taken over the stopping times of Φ_α, that coincide with those of X, as evident from (3.7). Further, unlike (3.3)–(3.5), where the expectation is under $\mathbb{P}_\pi$, in (3.9) the expectation is under $\mathbb{P}_\phi^\infty$, the probability measure under which fraud never occurs, namely $\theta = \infty$, conditional to the event $\{\Phi_{\alpha,0} = \phi\}$, for $\phi \geq 0$. It is immediate to

see that $U_i \leq 0$, because $\tau = 0$ is an admissible stopping time, and $U_i \geq -k_i/\lambda$, which arises from never stopping (i.e., $\tau = \infty$) and the fact that Φ_α takes positive values. We also see that it is never optimal to stop before Φ_α reaches k_i, because, before that moment, the integrand in (3.9) remains negative. Indeed, it is well known that there exists a threshold $B_i \geq k_i$ such that the optimal stopping time in (3.9) is given by

$$\tau_i^\star := \inf\{t \geq 0 : \Phi_{\alpha,t} \geq B_i\}, \quad B_i = B_i(\lambda, c_i, \alpha, \lambda_0, v_0(\cdot), \lambda_1, v_1(\cdot)), \quad i = 1, 2, 3, \tag{3.11}$$

which is the first moment at which Φ_α exceeds B_i (see Bayraktar et al., 2005; Buonaguidi and Muliere, 2015; Gapeev, 2005; Gapeev and Shiryaev, 2013; Johnson and Peskir, 2017; Peskir and Shiryaev, 2002; Shiryaev, 1978). From (3.11) we observe that the optimal threshold is independent of π, the prior probability that fraud occurs immediately, and this is consistent with the general optimal stopping theory (Peskir and Shiryaev, 2006; Shiryaev, 1978). Further, from (3.6) and the fact that $\alpha = 0$ when $i = 1, 2$, we have that (3.11) is equivalent to

$$\tau_i^\star := \inf\{t \geq 0 : \pi_t \geq C_i\}, \quad C_i := \frac{B_i}{1 + B_i}, \ i = 1, 2. \tag{3.12}$$

3.3 The algorithm

Solving the Bayesian quickest detection problems (3.3)–(3.5) boils down to computing the function U_i in (3.9) and the threshold B_i in (3.11), $i = 1, 2, 3$ (for a simpler notation, the index i will be omitted). This can be done by resorting to the iterative procedure provided in Dayanik and Sezer (2006).

When a credit card transaction is made, X from (3.1) has a jump. Denoted by $\{\sigma_n\}_{n\geq 1}$ the jumping times of X, let us notice that they coincide with the jumping times of Φ_α, as the second addend in (3.7) shows. In particular,

$$\Phi_{\alpha,\sigma_n} = \Phi_{\alpha,\sigma_n -} \frac{\lambda_1}{\lambda_0} f(Y_n), \quad n \geq 1. \tag{3.13}$$

Equation (3.7) also shows that Φ_α solves, between two successive jumps, the first order linear differential equation

$$\frac{d\Phi_{\alpha,t}}{dt} = \lambda + a\Phi_{\alpha,t}, \quad a := \alpha + \lambda - \lambda_1 + \lambda_0, \quad \text{and} \quad t \in [\sigma_n, \sigma_{n+1}), \, n \geq 1, \tag{3.14}$$

whose solution is

$$\Phi_{\alpha,t} = x(t - \sigma_n, \Phi_{\alpha,\sigma_n}), \quad x(t, \phi) := \begin{cases} -\dfrac{\lambda}{a} + e^{at}\left(\phi + \dfrac{\lambda}{a}\right), & a \neq 0, \\ \phi + \lambda t, & a = 0, \end{cases} \tag{3.15}$$

for $t \in [\sigma_n, \sigma_{n+1})$. Φ_α is therefore a piecewise deterministic Markov process: between any two subsequent jumps it follows the deterministic flow $t \mapsto x(t, \phi)$, being $\phi \geq 0$ the starting point of the process after a jump (see, e.g., Davis, 1993).

Let us consider the family of optimal stopping problems

$$U^{(n)}(\phi) := \inf_{\tau} \mathbb{E}_{\phi}^{\infty}\left[\int_0^{\tau\wedge\sigma_n} e^{-\lambda t}(\Phi_{\alpha,t}-k)dt\right], \quad n \geq 1, \tag{3.16}$$

where the infimum is taken with respect the stopping times of $\Phi_{\alpha,t}$ and we integrate up to the minimum between a stopping time τ and σ_n. In order to exploit the piecewise deterministic Markov property of Φ_α, let us define the operators $J : C_b([0,\infty))\times[0,\infty)\times[0,\infty) \to \mathbb{R}$ and $J_0 : C_b([0,\infty)) \times [0,\infty) \to \mathbb{R}$ by

$$J(g,\phi,t) := \mathbb{E}_{\phi}^{\infty}\left[\int_0^{t\wedge\sigma_1} e^{-\lambda u}(\Phi_{\alpha,u}-k)du + \mathbf{1}_{\{t\geq\sigma_1\}}e^{-\lambda\sigma_1}g(\Phi_{\alpha,\sigma_1})\right], \tag{3.17}$$

$$J_0(g,\phi) := \inf_{t\in[0,\infty)} J(g,\phi,t), \tag{3.18}$$

where $C_b([0,\infty))$ is the set of bounded and continuous functions on $[0,\infty)$. Then, we can compute sequentially the functions $u^{(n)} \in C_b([0,\infty))$, $n \geq 0$, defined by

$$u^{(0)}(\phi) := 0, \quad u^{(n)}(\phi) := J_0(u^{(n-1)},\phi), \quad \phi \geq 0,\ n \geq 1, \tag{3.19}$$

that satisfy the property $u^{(n)} = U^{(n)}$, $n \geq 1$, and $\lim_{n\to\infty} u^{(n)} = U$. Observing that σ_1 has exponential distribution with mean $1/\lambda_0$ under $\mathbb{P}^{\infty}$ and using Fubini's theorem, (3.18)–(3.19) read more explicitly as

$$u^{(n)}(\phi) = \inf_{t\in[0,\infty)} \int_0^t e^{-(\lambda+\lambda_0)u}\Big(x(u,\phi) - k + \lambda_0 S(u^{(n-1)}, x(u,\phi)\Big)\, du, \tag{3.20}$$

where $S : C_b([0,\infty)) \times [0,\infty) \to \mathbb{R}$ is the operator defined by

$$S(g,x) := \int_{\mathbb{R}^d} g\left(\frac{\lambda_1}{\lambda_0} f(y)\, x\right) v_0(dy). \tag{3.21}$$

Finally, the threshold B in (3.11) is given by

$$B = \lim_{n\to\infty} B^{(n)}, \quad B^{(n)} := \inf\{\phi \geq 0 : u^{(n)}(\phi) = 0\}. \tag{3.22}$$

Practically, the previous computations end when n is sufficiently large. The technical details on the implementation of the illustrated approximation scheme can be found in Dayanik and Sezer (2006, Sec. 5) and are also reported in the Supplementary Material (Buonaguidi et al. 2020a; 2020b).

4 Experimental setup

In this section, we calibrate the quickest detection models in (3.3)–(3.5) on a real set of credit card transactions and we test them on simulated and real datasets. Common performance measures will be computed to evaluate the predictive power of the proposed methodology.

4.1 The dataset

One of the most important Swiss credit card issuers, with more than 1.5 million issued cards and more than 100 million transactions authorized every year, provided us with a vast dataset of real credit card transactions, including Internet purchases. This dataset covers a six-month period, from June to November 2016, and contains the details of 124,770 authorized transactions, which pertain to 4,077 different cardholders. Each transaction has the following attributes: *BaseCardID*, the identification code of a cardholder, which remained the same even if she replaced her card during the considered period (cardholders had been completely anonymized); *RecordDateTime*, the date and the time at which the operation took place; *TrxAmount*, the transaction amount in the currency of the issuer; *MerchantLocation*, the location of the merchant; *isTrxFraud*, a flag indicating whether the transaction was fraudulent. The latter attribute is created by the card issuer a few days after the transaction, which is identified as fraudulent by means of the analysis of fraud experts and the confirmation of the cardholders: when the fraud team manually revises suspicious transactions, the additional information to which the team members have access, such as the merchants identification number and the merchants category (restaurant, pharmacy, ATM, etc), allows them to identify illegal purchases reliably; then, the legitimate cardholders are contacted for their confirmation on the fraudulent nature of the transactions. In the dataset 2,778 transactions were labeled as fraudulent, implying a fraud ratio of 2.23%. Actual fraud ratios are much lower than this value; however, in our dataset, fraudulent transactions have been overweighted to mitigate the problem of data skewness, occurring when the legitimate cases far outnumber the fraudulent ones.

By means of the BaseCardID, transactions were grouped and sorted in ascending order by the RecordDateTime. A new attribute *ElapsedDays* was obtained as the number of days between two consecutive operations; for the first transaction of each cardholder, ElapsedDays was set equal to 0. This attribute has been derived because when a fraudster steals a credit card or the associated sensible information, he usually attempts to make as many transactions as possible in a narrow window of time, before the fraud is detected and the card is blocked. Accordingly, the variable TrxAmount has a key role in fraud detection, because fraudsters try to maximize spending before they are discovered, as suggested in Bhattacharyya et al. (2011); Bolton and Hand (2002). The importance of the time elapsed between transactions and their amounts was also underlined in Carneiro et al. (2017, Table 6). The MerchantLocation attribute has been finally used to derive the geographical coordinates of the associated transactions: *Latitude* and *Longitude*. Indeed, fraudsters perpetrate their activities in places which are often different from the ones where cardholders make their legitimate purchases. Then, knowing where a transaction took place is relevant for a more efficient identification of fraudulent behaviors. Table 1 shows the final structure of our dataset.

Name:	BaseCardID	ElapsedDays	TrxAmounts	Latitude	Longitude	isTrxFraud
Range:	alphanumeric	$\mathbb{R}_+$	$\mathbb{R}_+$	$[-90, 90]$	$[-180, 180]$	$\{0, 1\}$

Table 1: Structure of the derived dataset.

4.2 Models calibration

Calibrating the Bayesian quickest detection models (3.3)–(3.5) means (i) establishing the parameter λ in (3.2), governing the prior distribution of the fraud time θ, (ii) determining the constants c_i, $i = 1, 2, 3$, and α in (3.3)–(3.5), (iii) estimating the quantities $(\lambda_0, v_0(\cdot))$ and $(\lambda_1, v_1(\cdot))$ for the cardholder's expenditure process (3.1) and (iv) computing the optimal threshold B_i, $i = 1, 2, 3$, in (3.11). For point (i), we relied on fraud experts prior knowledge for a reasonable value of λ. In (ii), the values c_i, $i = 1, 2, 3$, are chosen by the models user, that, according to her needs, may decide to weigh more or less heavily the detection delay; α is still chosen by the models user on the basis of the interest rate at which the losses due to detection delays are compounded. For the quantities in (iii), we adopted the following approach: for each cardholder, her legitimate transactions (the ones where the attribute isTrxFraud takes value 0) are used to estimate her own arrival rate λ_0 and jumps distribution $v_0(\cdot)$; λ_1 and $v_1(\cdot)$ are estimated only once on all the fraudulent transactions in the dataset, meaning that we assume the existence of a representative fraudster who may potentially act against all the cardholders. The latter assumption is motivated by the fact that fraudulent transactions data do not discriminate among different fraudsters and fraud represents a tiny percentage of the total number of purchases: hence, we need to aggregate all the available information to reliably estimate at least one representative fraudster behavior. More details on the estimation procedure under different "information schemes" will be given in this section. Finally, the personalized optimal threshold at point (iv) depends on the quantities in (i)–(iii), as shown by (3.11), and is determined for each cardholder through the algorithm of Section 3.3.

Observing the elapsed days only

When the date and the time of a transaction are the unique features that we can observe and, consequently, only the ElapsedDays attribute in Table 1 is available, the cardholder's expenditure process X in (3.1) becomes a simple Poisson process. This formally arises from setting $Y_n = 1$ $\mathbb{P}^s$-a.s., $n \geq 1$ and $s \geq 0$, so that $v_i(dx) = \delta_1(x)dx$, $i = 0, 1$, where $\delta_1(\cdot)$ is the Dirac measure; it implies that $f(x) = \mathbf{1}_{\{1\}}(x)$ in (3.8). Let K_1 be the set of indexes relative to the subsequent fraudulent transactions in the derived dataset and, similarly, let K_0 be the set of indexes associated to the consecutive legitimate purchases of a given cardholder. Then, since the inter-arrival times of a Poisson process are independent and follow an exponential distribution with mean $1/\lambda_0$ and $1/\lambda_1$, for the legitimate and fraudulent cases respectively, λ_i can be determined as the maximum likelihood estimate

$$\hat{\lambda}_i = \frac{|K_i|}{\sum_{j \in K_i} \text{ElapsedDays}_j}, \quad i = 0, 1, \tag{4.1}$$

where $|K_i|$ is the cardinality of the set K_i. We obtained $\hat{\lambda}_1 = 3.012032$, meaning that when a fraudster steals a credit card, he tries to make about three transactions per day on average; the maximum likelihood estimates of λ_0 for the cardholders in the dataset range in the interval [0.005769, 2.776012]. Then, legitimate transactions occur

less frequently than the fraudulent ones. This fact can also be inferred from Figure 1, where the cumulative distribution function of the elapsed days between all the pair of consecutive fraudulent transactions of our dataset is compared with that of a sample of subsequent legitimate purchases.

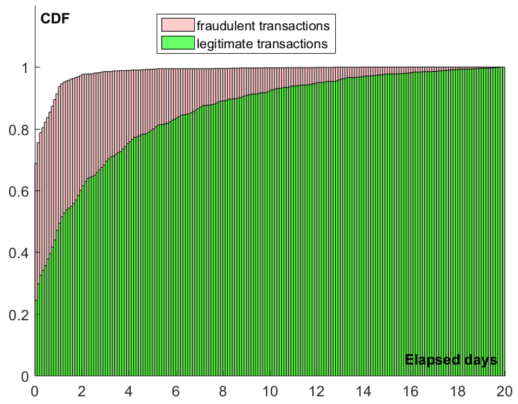

Figure 1: Comparison between the cumulative distribution function of the elapsed days between two consecutive fraudulent transactions (in red) and the one of the elapsed days in a sample of 2,000 subsequent legitimate purchases (in green).

Observing the elapsed days and the transaction amounts

When also the transaction amounts are available, in the compound Poisson process (3.1) the sequence of random variables $\{Y_j\}_{j\geq 1}$ can be used to model the purchase expenditures. Letting $\{Y_j\}_{j\geq 1}$ represent the logarithm of the variable TrxAmount, we assumed that, before and after fraud, Y_j follows a Gaussian distribution with mean and variance μ_i and σ_i^2, $i = 0, 1$, respectively. Then,

$$v_i(dx) = \mathcal{N}_1(x; \mu_i, \sigma_i^2)dx, \quad i = 0, 1, \quad \text{and} \quad f(x) = \frac{\mathcal{N}_1(x; \mu_1, \sigma_1^2)}{\mathcal{N}_1(x; \mu_0, \sigma_0^2)}, \tag{4.2}$$

where $\mathcal{N}_1(x; \mu_i, \sigma_i^2)$ is the univariate Gaussian density with mean and variance μ_i and σ_i^2 evaluated at $x \in \mathbb{R}$. Denoted by H_1 the set of indexes of all the fraudulent transactions and by H_0 the set of indexes of all the legitimate purchases for a given cardholder (let us observe that $K_i \subseteq H_i$, $i = 0, 1$), μ_i and σ_i^2 can be computed by resorting to the maximum likelihood estimators

$$\hat{\mu}_i = \frac{\sum_{j\in H_i} Y_j}{|H_i|}, \quad \hat{\sigma}_i^2 = \frac{\sum_{j\in H_i}(Y_j - \hat{\mu}_i)^2}{|H_i|}, \quad i = 0, 1. \tag{4.3}$$

For the fraudulent transactions, we obtained $\hat{\mu}_1 = 4.095233$ and $\hat{\sigma}_1^2 = 3.124095$. The left panel of Figure 2 reports the histogram of the logarithmic amounts of the fraudulent transactions and compares it to the estimated Gaussian density. The right panel plots

the estimated pairs (μ_0, σ_0) characterizing the Gaussian distribution of the logarithmic legitimate amounts of all the cardholders. The intensities λ_0 and λ_1 are estimated according to (4.1).

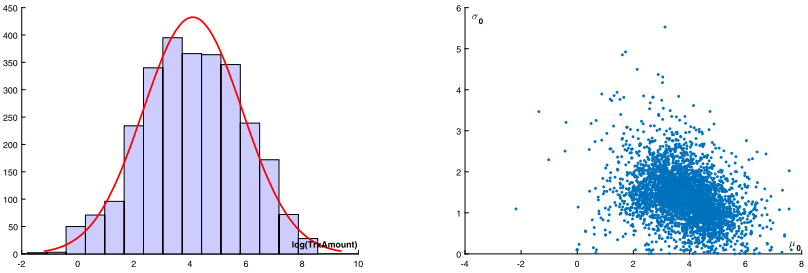

Figure 2: Left panel: histogram of the fraudulent logarithmic amounts and estimated Gaussian density with $\hat{\mu}_1 = 4.095233$ and $\hat{\sigma}_1^2 = 3.124095$. Right panel: pairs (μ_0, σ_0) of the Gaussian densities of the legitimate logarithmic amounts of all the cardholders.

Observing the elapsed days, the amounts and the geographical coordinates

The last and full informative scheme we consider is the one where the geographical coordinates are also available. The cardholder's expenditures process X in (3.1) becomes now multivariate, since the random variables $\{Y_j\}_{j\geq 1}$ are used to model the logarithm of the amounts, the longitude and latitude of the transactions and take therefore values in $\mathbb{R}^3$. Since consumers make the majority of their purchases in a few selected places, we decided to model the coordinates through mixtures of Gaussian distributions. Amounts are assumed to be independent of the coordinates; then, we extend (4.2) to

$$v_i(x,y)dxdy = \mathcal{N}_1(x;\mu_i,\sigma_i^2)dx \times \sum_{j=1}^{n_i} p_{i,j}\mathcal{N}_2(y;\eta_{i,j},\Sigma_{i,j})dy, \quad i=0,1, \quad x\in\mathbb{R},\ y\in\mathbb{R}^2, \tag{4.4}$$

where $\mathcal{N}_2(y;\eta_{i,j},\Sigma_{i,j})$ is the bivariate Gaussian density of the j-th mixture component with mean vector and covariance matrix $\eta_{i,j}$ and $\Sigma_{i,j}$, respectively. In (4.4), n_i represents the number of components, the so called clusters, of the mixture and $p_{i,j}$ is the probability that an element belongs to component j, and is such that $\sum_{j=1}^{n_i} p_{i,j} = 1$. From the expression above, we easily find that the Radon-Nykodym derivative $f(\cdot)$ in (3.8) takes the form

$$f(x,y) = \frac{\mathcal{N}_1(x;\mu_1,\sigma_1^2)}{\mathcal{N}_1(x;\mu_0,\sigma_0^2)} \times \frac{\sum_{j=1}^{n_1} p_{1,j}\mathcal{N}_2(y;\eta_{1,j},\Sigma_{1,j})}{\sum_{j=1}^{n_0} p_{0,j}\mathcal{N}_2(y;\eta_{0,j},\Sigma_{0,j})}, \quad x\in\mathbb{R},\ y\in\mathbb{R}^2. \tag{4.5}$$

The maximum likelihood estimators of λ_i, μ_i and σ_i^2, $i=0,1$, are still given by (4.1) and (4.3). Once the numbers of mixture components, n_0 and n_1, are chosen, $\eta_{i,j}$, $\Sigma_{i,j}$ and $p_{i,j}$, $j=1,\ldots,n_i$, $i=0,1$, can be estimated by resorting to the EM-algorithm (Dempster et al., 1977). For the longitude and the latitude of the fraudulent purchases,

we used a bivariate Gaussian mixture model with $n_1 = 6$ components, whose induced clusters are shown in Figure 3. For each cardholder, a bivariate Gaussian mixture model was estimated on the coordinates of her legitimate transactions. We initially fixed $n_0 = 3$ components; if the algorithm failed to converge (because, for example, transactions were concentrated in a very few or just one region), n_0 was decreased to 2 or 1.

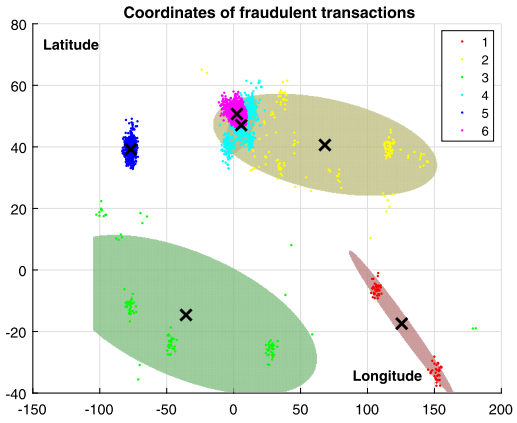

Figure 3: Clusters of the fraudulent transactions obtained by estimating a Gaussian mixture model with six components on their geographical coordinates. Cluster 1 (red) is centered in Australia and its weight is $p_{1,1} = 0.027$. Cluster 2 (yellow) embraces Asian countries with weight $p_{1,2} = 0.057$; Cluster 3 (green) comprises south American and south African countries and its weight is $p_{1,3} = 0.041$. Cluster 4 (light blue) includes central and south European Countries, with weight $p_{1,4} = 0.202$. Cluster 5 (blue) mainly refers to the United States and has weight $p_{1,5} = 0.279$. Cluster 6 (violet) mainly refers to the United Kingdom and has weight $p_{1,6} = 0.393$.

Some notes on the thresholds computation

We set $\lambda = 1/365$ in (3.2), namely in our prior belief, as elicited by experts, a cardholder suffers, on average, an attempt of fraud once per year. The algorithm of Section 3.3 for the computation of the optimal threshold in (3.11) was applied for each of the previous information schemes, for each of the models in (3.3)–(3.5) and for each cardholder. For each of the models (3.3)–(3.5), which we refer to as "linear", "expected miss" and "exponential", respectively, different values of c_i, $i = 1, 2, 3$ were used. For example, in the linear problem (3.1), the cases of $c_1 = 0.1$ and $c_1 = 0.2$ were considered; in the expected miss problem (3.4), we first set $c_2 = 10$ and then $c_2 = 50$; in the exponential case (3.5), we first considered $c_3 = 1{,}000$ and then $c_3 = 2{,}000$. In the exponential case, α was set equal to 1.3367×10^{-4}, which is equivalent to an annual internal rate of return of 5%. Information on these parameters is usually available to a credit card issuer and in any case can be obtained on the basis of statistics of previous months/years.

Let us finally make four remarks: (1) when $\{Y_j\}_{j\geq 1}$ contain more features than

what are considered in our analysis, the assumption of independence in part of these explanatory variables could ease the estimation of $v_0(\cdot)$ and $v_1(\cdot)$; (2) for a given model and cardholder, the algorithm complexity increases with the informative scheme: when only the elapsed days are considered, the integral in (3.21) disappears (because $v_0(\cdot)$ concentrates all its mass on 1), so that the optimization problem in (3.20) can be quickly solved; instead, when the amounts or both the amounts and the coordinates are considered, the integral in (3.21) is one-dimensional or three-dimensional, respectively, and this slows down the solution procedure of (3.20); (3) we wrote the code in Matlab and we used a standard 2017 laptop for the computations: the estimation of $(\lambda_0, v_0(\cdot))$ took about 0.002, 0.002 and 0.009 seconds on average for each cardholder when the elapsed days, the elapsed days and the amounts, and the elapsed days, the amounts and the coordinates are observed, respectively. These times rose to 21, 146 and 456 seconds when also the cardholder specific optimal threshold from (3.11) was computed; (4) at first sight, the just reported execution times are relevant. However, they can be easily managed if we consider that the algorithm of Section 3.3 applies independently to each cardholder and, therefore, can be parallelised among cardholders for faster and more efficient computations, as well as the fact that in practice fraud models would need to be implemented in high-performance computing languages and are usually re-trained less than once a month.

4.3 Models testing on simulated transactions

In order to assess the goodness of the models (3.3)–(3.5), 20 datasets of transactions were simulated. Each of them has the same structure reported in Table 1 and was generated in the following way: for each cardholder, 50 transactions were simulated and the flag indicating their legal or illegal nature was extracted from a Bernoulli distribution having parameter 0.1 (i.e., about 10% of the dataset transactions are fraudulent); the fraudulent transactions always occur later than the set of the legitimate ones. According to the representation in (3.1), for any legitimate cardholder transaction, the attribute ElapsedDays was extracted from an exponential distribution with mean $1/\hat{\lambda}_0$, the variable TrxAmounts was taken to be the exponential of a Gaussian random number with mean and variance $\hat{\mu}_0$ and $\hat{\sigma}_0^2$, and the Longitude and Latitude attributes were generated from a mixture of bivariate Gaussian densities. The parameters characterizing all these distributions are cardholder specific and were obtained during the calibration step as discussed in Section 4.2. For all the fraudulent transactions, the ElapsedDays variable was drawn from an exponential distribution with mean $1/\hat{\lambda}_1$, the logarithm of TrxAmounts was simulated from a Gaussian density with mean and variance $\hat{\mu}_1$ and $\hat{\sigma}_1^2$ and the Longitude and Latitude variables were simulated according to the mixture of bivariate Gaussian densities of Figure 3; let us recall that these fraudulent distributions are not cardholder specific.

Scoring

In each of the 20 simulated datasets, we computed, for each cardholder and each informative scheme, the dynamic given by (3.13)–(3.15) of the (generalized) odds process

Φ_α (we remind that $\alpha = 0$ for the linear and expected miss models, where $\varphi = \Phi_0$). As shown in Sections 3.2–3.3, its dynamic only depends on π from (3.2) (we always fixed $\pi = 0$), λ, α, $(\lambda_0, v_0(\cdot))$ and $(\lambda_1, v_1(\cdot))$. All the transactions characterized by a value of Φ_α greater than the cardholder specific optimal threshold were labeled as fraudulent, according to (3.11)–(3.12). Because of its optimality, the adopted detection strategy minimizes the trade off between early and unjustified credit card blocks and late interventions in disclosing fraudulent transactions and so, under the evaluation measures (3.3)–(3.5), outperforms any other strategy.

Performance measures

By comparing the actual nature of a transaction (variable isTrxFraud in the simulated datasets) and the corresponding model prediction, performance measures commonly used in the literature were computed. As reported in Table 2, transactions identified correctly as fraudulent are said *true positives*, while those classified correctly as legitimate are the *true negatives*; we may also have *false positives*, when legitimate transactions are identified as fraudulent, and *false negatives*, when fraudulent transactions are predicted as legitimate.

	Predicted fraudulent	Predicted legitimate
Actual fraudulent	true positive	false negative
Actual legitimate	false positive	true negative

Table 2: Confusion matrix.

Let us denote by TP, FN, TN, FP the number of true positives, false negatives, true negatives and false positives in a dataset. Then, we considered seven standard metrics: the *accuracy* (Acc), which is the proportion of correct predictions (it could be misleading because, for example, if all the transactions were predicted as legitimate in our datasets, were the percentage of fraud is about 0.1, the accuracy would be around 0.9); the *false positive rate* (FPR), also known as fallout, which is the proportion of predicted fraudulent transactions among the legitimate ones; the *true positive rate* (TPR), also called sensitivity or recall, which expresses the proportion of predicted fraudulent transactions among the fraudulent ones; the *negative predicted value* (NPV), which returns the proportion of actual legitimate transactions among those predicted as such; the *precision* (Pr), which is the proportion of actual fraudulent transactions among those predicted as such; the *Matthews correlation coefficient* (MCC), which represents the correlation between the actual and predicted nature of the transactions. Their expressions are reported for completeness in the Supplementary Material. We also derived the *area under the ROC curve* (AUC), being the ROC (receiver operating characteristic) curve defined as the set of all the pairs of points (FPR, TPR) obtained by letting the cardholders threshold varies.

Results

In the next table the values of the metrics discussed above are shown and are also reported in Figure 4 for a better visualization. They are obtained as the average of the corresponding metrics computed for each of the 20 simulated datasets; in the brackets the standard errors are reported. The first two blocks of Table 3 show the results for the linear model (3.3) when c_1 is 0.1 and 0.2, respectively, across the information schemes of Section 4.2. The abbreviations ED, TrxAm and Coo stand for elapsed days, transactions amounts and geographical coordinates. We see that the results improve as more attributes are considered: the FPR decreases and all the other metrics increase, as expected. Overall the obtained performance measures are very satisfactory both in absolute terms and when compared to the literature: for example, in Bhattacharyya et al. (2011, Tables 6a–6c) the best values of the Acc, FPR, TPR, Pr and AUC are 0.996, 0.001, 0.812, 0.613 and 0.934, respectively; in Carneiro et al. (2017, Table 5) the FPR is 0.019, the TPR is 0.587 and the Pr is 0.407.

	Acc	FPR	TPR	NPV	Pr	MCC	AUC
Linear, $c_1 = 0.1$							
ED	0.94513 (6.5e-05)	0.00182 (3.0e-05)	0.45696 (9.0e-04)	0.94418 (6.8e-05)	0.96454 (5.6e-04)	0.64311 (6.7e-04)	0.93986 (2.9e-04)
ED + TrxAm	0.95397 (8.1e-05)	0.00050 (1.5e-05)	0.53502 (9.5e-04)	0.95188 (8.3e-05)	0.99141 (2.7e-04)	0.71007 (6.8e-04)	0.95141 (2.2e-04)
ED + TrxAm + Coo	0.98591 (5.8e-05)	0.00008 (4.1e-06)	0.85700 (6.2e-04)	0.98470 (6.2e-05)	0.99909 (4.4e-05)	0.91816 (3.6e-04)	0.98052 (1.7e-04)
Linear, $c_1 = 0.2$							
ED	0.94803 (6.9e-05)	0.00654 (6.9e-05)	0.53002 (9.1e-04)	0.95111 (6.7e-05)	0.89792 (1.0e-03)	0.66666 (7.6e-04)	0.93986 (2.9e-04)
ED + TrxAm	0.95838 (7.3e-05)	0.00123 (2.3e-05)	0.58673 (7.8e-04)	0.95695 (7.4e-05)	0.98106 (3.6e-04)	0.74093 (5.5e-04)	0.95141 (2.2e-04)
ED + TrxAm + Coo	0.98695 (5.4e-05)	0.00019 (8.1e-06)	0.86861 (5.6e-04)	0.98593 (5.7e-05)	0.99795 (8.5e-05)	0.92434 (3.3e-04)	0.98052 (1.7e-04)
Exp. miss, $c_2 = 10$							
ED	0.93749 (7.2e-05)	0.00016 (9.6e-06)	0.36382 (9.4e-04)	0.93533 (7.5e-05)	0.99578 (2.3e-04)	0.58188 (7.4e-04)	0.93986 (2.9e-04)
ED + TrxAm	0.94668 (8.3e-05)	0.00011 (7.1e-06)	0.45706 (9.6e-04)	0.94428 (8.5e-05)	0.99763 (1.4e-04)	0.65604 (7.1e-04)	0.95141 (2.2e-04)
ED + TrxAm + Coo	0.98427 (6.8e-05)	**0.00002 (2.4e-06)**	0.83969 (7.0e-04)	0.98288 (7.3e-05)	**0.99975 (2.6e-05)**	0.90834 (4.1e-04)	0.98052 (1.7e-04)
Exp. miss, $c_2 = 50$							
ED	0.94663 (6.8e-05)	0.00340 (4.8e-05)	0.48690 (9.6e-04)	0.94702 (7.9e-05)	0.93949 (8.1e-04)	0.65468 (7.2e-04)	0.93986 (2.9e-04)
ED + TrxAm	0.95578 (7.7e-05)	0.00088 (2.5e-05)	0.55705 (8.2e-04)	0.95404 (7.5e-05)	0.98556 (4.1e-04)	0.72289 (6.2e-04)	0.95141 (2.2e-04)
ED + TrxAm + Coo	0.98636 (5.5e-05)	0.00011 (5.8e-06)	0.86187 (5.8e-04)	0.98522 (5.9e-05)	0.99872 (6.2e-05)	0.92082 (3.4e-04)	0.98052 (1.7e-04)
Expon., $c_3 = 1{,}000$							
ED	0.94653 (7.0e-05)	0.00326 (4.5e-05)	0.48452 (9.6e-04)	0.94679 (7.9e-05)	0.94154 (7.7e-04)	0.65384 (7.2e-04)	0.93987 (2.9e-04)
ED + TrxAm	0.95555 (7.5e-05)	0.00086 (2.3e-05)	0.55452 (8.4e-04)	0.95379 (7.5e-05)	0.98582 (3.9e-04)	0.72126 (6.3e-04)	0.95141 (2.2e-04)
ED + TrxAm + Coo	0.98631 (5.5e-05)	0.00011 (5.5e-06)	0.86134 (5.9e-04)	0.98516 (5.9e-05)	0.99876 (5.8e-05)	0.92053 (3.4e-04)	0.98052 (1.7e-04)
Expon., $c_3 = 2{,}000$							
ED	0.94844 (8.1e-05)	0.01060 (8.5e-05)	0.57154 (8.8e-04)	0.95506 (7.0e-05)	0.85423 (1.0e-03)	0.67376 (7.8e-04)	0.93987 (2.9e-04)
ED + TrxAm	0.96012 (8.5e-05)	0.00246 (3.1e-05)	0.61589 (8.0e-04)	0.95984 (8.1e-05)	0.96444 (4.4e-04)	0.75297 (6.0e-04)	0.95141 (2.2e-04)
ED + TrxAm + Coo	**0.98748 (5.2e-05)**	0.00032 (1.0e-05)	**0.87521 (5.5e-04)**	**0.98662 (5.4e-05)**	0.99663 (1.0e-04)	**0.92749 (3.3e-04)**	0.98052 (1.7e-04)

Table 3: Simulated data: metrics for linear model (3.3) with $c_1 = \{0.1, 0.2\}$, expected miss model (3.4) with $c_2 = \{10, 50\}$, exponential model (3.5) with $c_3 = \{1{,}000, 2{,}000\}$ and $\alpha = 1.3367 \times 10^{-4}$, across different information schemes. For the third information scheme, the bold font is used to highlight the best value of each metric across the different models.

We may notice from Table 3 that an increase of the parameter c_1 implies, for a given information scheme, an increase of the FPR, TPR and NPV and a decrease of the Pr. It is intuitively explained by the fact that when more importance is given to the losses due to a detection delay, each cardholder threshold shifts downwards; then, since the score process is independent of c_1 (as we can see from the AUC which is the same across the two values of c_1), more fraudulent transactions are predicted. This leads to an increment of the numerators of the FPR and the TPR, while their denominators, corresponding to

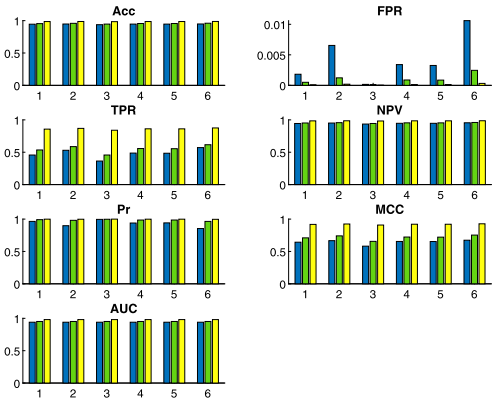

Figure 4: Metrics from Table 3. For each metric, the corresponding values across the three information schemes are shown: blue bars are for ED, green bars are for ED + TrxAm and yellow bars are for ED + TrxAm + Coo. The numbering $1, \ldots, 6$ refers to the models considered in Table 3: 1 and 2 are for the linear model when c_1 is 0.1 and 0.2, respectively; 3 and 4 are for the expected miss model when c_2 is 10 and 50, respectively; 5 and 6 are for the exponential model with $\alpha = 1.3367 \times 10^{-4}$ when c_3 is 1,000 and 2,000, respectively.

the actually legitimate and actually fraudulent transactions, remain unchanged. Lower values of the cardholder thresholds also imply that, given that a purchase has been labeled as fraudulent (resp. legitimate), there is a higher chance that it is legitimate, causing a lower Pr (resp. higher NPV).

The third and fourth block of Tables 3 show the metrics for the expected miss model (3.4) with $c_2 = 10$ and $c_2 = 50$, respectively. The fifth and sixth block of Table 3 contain the results for the exponential model (3.5) with $c_3 = 1{,}000$ and $c_3 = 2{,}000$, respectively, when $\alpha = 1.3367 \times 10^{-4}$. Considerations analogous to the ones of the linear model apply to these cases as well.

Comparative performance analysis on simulated datasets

Models associated to the data mining techniques discussed in Section 2 can be used as benchmark for the results of Table 3. These models have been trained for each of the three information schemes on the initial dataset of Section 4.1 by using appropriate built-in Matlab functions. For the logistic regression we used the *glmfit* function by specifying the binomial distribution for the response variable; for the rule-based methods we constructed decision trees based on the CART algorithm (see, e.g., Han et al., 2012) via the *fitctree* function; for boosting, in order to mitigate the problem of imbalanced data, we applied the RUSBoost (random undersampling boosting) algorithm (Seiffert et al., 2008) by means of the *fitcensemble* function; for BART we used the *pbart* function (from the BART *R*-package), setting the number of posterior

draws for each transaction to 500; for random forests we employed the Breiman's algorithm (Breiman, 2001) via the *TreeBagger* function and we adopted 50 classification trees; for the hidden Markov model we treated the dichotomous variable isTrxFraud as the hidden state and the elapsed times, amounts and geographical coordinates as the observable outcomes and we recovered the maximum likelihood estimates of the transitions and outcomes probabilities through the function *hmmestimate*; for support vector machines the ISD (iterative single data) algorithm (Kecman et al., 2005) has been used together with a Gaussian kernel for data separation via the *fitcsvm* function; for neural networks we trained a feedforward network (a special type of neural networks where there are not cycles among neurons, but information moves forward from the input neurons, through the hidden layers, up to the output neurons) with one hidden layer consisting of 10 neurons by means of the *patternnet* function.

Let us recall that the just cited classification methodologies may underperform when the training data are skewed, like in our case where fraudulent transactions are 2.23%. To overcome this problem and have more meaningful results, firstly data have been balanced by drawing from the initial dataset random sub-samples characterized by a fraud ratio of 10%; apart from boosting (where the RUSBoost algorithm balances data), the algorithms have been subsequently calibrated on these sub-samples.

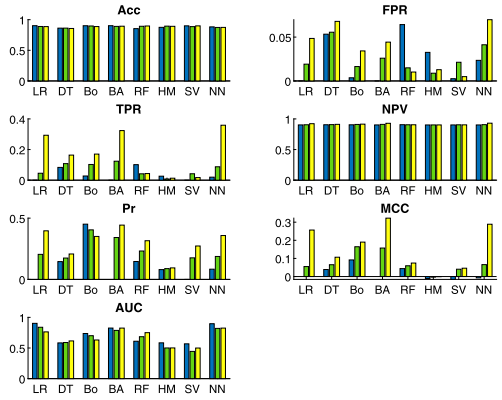

Figure 5: Metrics from Table 4. For each metric, the corresponding values across the three information schemes are shown: blue bars are for ED, green bars are for ED + TrxAm and yellow bars are for ED + TrxAm + Coo. LR, DT, Bo, BA, RF, HM, SV and NN stand for logistic regression, decision trees, boosting, BART, random forests, hidden Markov model, support vector machines and neural networks, respectively.

Table 4 shows the metrics across the three information schemes of the previous classification models applied to the 20 simulated datasets described at the beginning of Section 4.3. The sign "-" refers to the metrics that, according to their definition in the Supplementary Material, were not computable. The content of Table 4 can be visualized in Figure 5. We may notice that random forests, the hidden Markov model and support vector machines keep the FPR low, while logistic regression, decision trees, boosting,

	Acc	FPR	TPR	NPV	Pr	MCC	AUC
Logistic regression							
ED	0.90198 (1.1e-04)	0 (0)	0 (0)	0.90198 (1.1e-04)	–	–	0.90255 (1.2e-04)
ED + TrxAm	0.88915 (1.3e-04)	0.01915 (5.4e-05)	0.04529 (4.3e-04)	0.90447 (1.1e-04)	0.20443 (1.6e-03)	0.05502 (7.9e-04)	0.83843 (1.7e-04)
ED + TrxAm + Coo	0.88703 (1.4e-04)	0.04849 (1.0e-04)	0.29338 (6.2e-04)	0.92199 (1.0e-04)	0.39657 (9.7e-04)	0.25667 (8.4e-04)	0.76312 (3.2e-04)
Decision trees							
ED	0.86205 (1.8e-04)	0.05333 (1.4e-04)	0.08340 (4.7e-04)	0.90480 (1.2e-04)	0.14526 (7.3e-04)	0.03879 (6.0e-04)	0.58334 (4.3e-04)
ED + TrxAm	0.86244 (1.5e-04)	0.05557 (1.0e-04)	0.10797 (4.2e-04)	0.90691 (1.2e-04)	0.17433 (6.9e-04)	0.06524 (5.7e-04)	0.58933 (5.2e-04)
ED + TrxAm + Coo	0.85678 (1.5e-04)	0.06793 (1.4e-04)	0.16375 (4.6e-04)	0.91120 (1.2e-04)	0.20751 (8.3e-04)	0.10665 (6.5e-04)	0.61562 (4.2e-04)
Boosting							
ED	0.90140 (1.2e-04)	0.00361 (3.7e-05)	0.02731 (2.5e-04)	0.90409 (1.1e-04)	0.45087 (2.9e-03)	0.09169 (7.9e-04)	0.73675 (3.8e-04)
ED + TrxAm	0.89722 (1.3e-04)	0.01642 (4.8e-05)	0.10258 (4.7e-04)	0.90979 (1.2e-04)	0.40435 (1.5e-03)	0.16452 (8.8e-04)	0.70093 (3.8e-04)
ED + TrxAm + Coo	0.88781 (1.4e-04)	0.03425 (8.4e-05)	0.17032 (5.1e-04)	0.91465 (1.1e-04)	0.35069 (9.6e-04)	0.19001 (6.8e-04)	0.63085 (6.2e-04)
BART							
ED	0.90181 (1.2e-04)	0 (0)	0 (0)	0.90181 (1.2e-04)	–	–	0.82557 (9.2e-05)
ED + TrxAm	0.89056 (1.7e-04)	0.02595 (1.4e-04)	0.12388 (5.4e-04)	0.91080 (1.2e-04)	0.34198 (1.6e-03)	0.15732 (9.3e-04)	0.78743 (2.7e-04)
ED + TrxAm + Coo	0.89366 (1.8e-04)	0.04428 (1.1e-04)	0.32373 (1.3e-03)	0.92846 (1.5e-04)	**0.44321 (1.4e-03)**	**0.32228 (1.3e-03)**	**0.82511 (5.3e-04)**
Random forests							
ED	0.85395 (1.4e-04)	0.06426 (1.3e-04)	0.10132 (5.5e-04)	0.90560 (1.2e-04)	0.14626 (6.4e-04)	0.04380 (5.4e-04)	0.61043 (4.4e-04)
ED + TrxAm	0.89256 (1.1e-04)	0.01495 (5.3e-05)	0.04153 (3.2e-04)	0.90444 (1.2e-04)	0.23184 (1.4e-03)	0.05945 (7.0e-04)	0.68437 (3.0e-04)
ED + TrxAm + Coo	0.89708 (1.4e-04)	0.01017 (6.0e-05)	0.04322 (4.1e-04)	0.90443 (1.4e-04)	0.31572 (2.5e-03)	0.07447 (9.5e-04)	0.75034 (3.9e-04)
Hidden Markov model							
ED	0.87514 (8.3e-04)	0.03259 (1.0e-03)	0.02606 (8.3e-04)	0.90138 (1.0e-04)	0.08001 (9.6e-04)	−0.0109 (6.0e-04)	0.58427 (1.2e-03)
ED + TrxAm	0.89478 (6.6e-04)	0.00883 (7.9e-04)	0.00784 (7.7e-04)	0.90189 (1.1e-04)	0.08798 (3.0e-03)	−0.0031 (8.3e-04)	0.50079 (4.0e-04)
ED + TrxAm + Coo	0.89170 (2.2e-03)	0.01284 (2.8e-03)	0.01288 (3.1e-03)	0.90203 (1.5e-04)	0.09401 (2.5e-03)	−0.0006 (9.4e-04)	0.50135 (4.9e-04)
Support vector machines							
ED	0.89966 (1.1e-04)	0.00257 (2.6e-05)	0 (0)	0.90175 (1.1e-04)	0 (0)	−0.0158 (8.1e-05)	0.56803 (4.4e-04)
ED + TrxAm	0.88686 (1.3e-04)	0.02131 (9.3e-05)	0.04186 (2.8e-04)	0.90384 (1.2e-04)	0.17594 (1.1e-03)	0.04049 (5.8.e-04)	0.44465 (3.2e-04)
ED + TrxAm + Coo	**0.89925 (1.4e-04)**	**0.00492 (3.7e-05)**	0.01698 (1.7e-04)	0.90309 (1.4e-04)	0.27276 (2.7e-03)	0.04605 (7.0e-04)	0.49969 (5.9e-04)
Neural networks							
ED	0.88271 (1.2e-04)	0.02350 (2.1e-05)	0.01970 (2.2e-04)	0.90164 (1.2e-04)	0.08348 (8.7e-04)	−0.0075 (4.5e-04)	0.89651 (1.3e-04)
ED + TrxAm	0.87334 (1.5e-04)	0.04126 (1.0e-04)	0.08752 (4.6e-04)	0.90627 (1.3e-04)	0.18730 (8.8e-04)	0.06578 (6.6e-04)	0.82018 (1.9e-04)
ED + TrxAm + Coo	0.87416 (1.5e-04)	0.06986 (9.4e-05)	**0.35884 (8.5e-04)**	**0.93034 (1.2e-04)**	0.35810 (7.9e-04)	0.28870 (8.3e-04)	0.82494 (3.1e-04)

Table 4: Simulated data: metrics for logistic regression, decision trees, boosting, BART, random forests, hidden Markov model, support vector machines and neural networks across different information schemes. For the third information scheme, the bold font is used to highlight the best value of each metric across the different models.

BART and neural networks have good results when the TPR is considered. Overall, when the NPV, Pr, MCC and the AUC are also taken into account, the best results are given by BART and the neural networks. When Tables 3 and 4 are compared, we observe that the metrics of our proposed models are more satisfactory than those of these classification methods.

Robustness

A calibrated model can also be assessed on its robustness to correctly identify "noisy" transactions. To this aim, we considered three perturbed scenarios where transactions were simulated by increasing the values of the cardholders specific attributes: the intensity $\hat{\lambda}_0$, the mean $\hat{\mu}_0$ and standard deviation $\hat{\sigma}_0$ of the logarithmic amounts and the elements of the covariance matrices of the mixture of Gaussian distributions relative to the transactions coordinates. We also considered the case where the underlying distributional assumptions of the observed quantities are modified. Our conclusion is that our model performs sufficiently well also with very noisy transactions; moreover, among the other methodologies of Table 4, BART and neural networks show the best performance also under stressed situations, even though their results are not as good as those of our models. We refer to the Supplementary Material for a thorough analysis.

Factors affecting the metrics on data simulation

At the beginning of Section 4.3 we discussed how transactions have been simulated. Both for the cardholders and the fraudster, these transactions share the following features with the real training dataset: (i) the average number of daily purchases; (ii) the mean and the variance of the logarithmic expenditures; (iii) the mean vector and the covariance matrix of the geographical coordinates of the merchants' stores. However, it is important to underline that in the simulated data: (iv) the fraud ratio is about 10%, while in the training dataset it is about 2.23%; (v) no legitimate transactions occur once a cardholder is hit by fraud, while in the training dataset we observe cases where there are regular purchases between two fraudulent transactions.

In order to understand the possible bias induced by the last two factors and to assess the impact on the final metrics, the simulation has been repeated. 20 new datasets with the same size as the training dataset have been generated under three settings. *Setting 1:* the fraud ratio is reduced to 2.23%; *setting 2:* when a cardholder is hit by fraud, legitimate purchases may occur after fraudulent transactions; *setting 3:* setting 1 and 2 are combined. Our trained models (3.3)–(3.5) have been subsequently used to score and classify the transactions in each of these settings when all the attributes are observed. Then, by means of ANOVA, the results have been compared with those of Table 3, which we refer to as *setting 0*. The analysis is summarized in Figure 6.

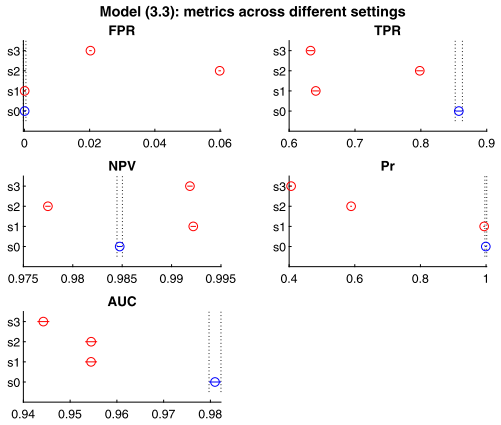

Figure 6: FPR, TPR, NPV, Pr and AUC across the four different settings for the model (3.3) with $c_1 = 0.1$. For a given metric, the corresponding plot reports on the x-axis its range and on the y-axis the four settings. For each setting a circle is placed on the metric average, whose confidence interval is represented by an horizontal line passing from the center of the circle; the averages of two settings are significantly different if their intervals are disjoint. The circle associated with setting 0 (third line of the first section in Table 3) is blue, while the other circles are red.

Figure 6 shows the metrics FPR, TPR, NPV, Pr and AUC across the different settings for the trained model (3.3) with $c_1 = 0.1$. We observe that we always reject the

hypothesis that a metric mean remains equal across the settings. We also see that the FPR increases from 8×10^{-5} to about 0.06 when we move from setting 0 to setting 2 and this can be explained by the fact that the score process Φ_α does not immediately fall below the cardholder's optimal threshold when legitimate transactions occur after fraudulent transactions, so that the former are misclassified. The TPR decreases from 0.857 to about 0.64 when we move from setting 0 to setting 1 and 3 and this is due to the fact that a small number of fraudulent transactions keeps Φ_α lower, so that their identification is more difficult. The NPV increases from 0.98 to 0.99 when we move from setting 0 to setting 1 and 3, because a lower number of fraudulent transactions makes more likely that a purchase identified as legitimate is actually as such. The Pr decreases from 0.99 to 0.59 and 0.41 when we move from setting 0 to setting 2 and 3, respectively, because when Φ_α exceeds the optimal threshold, it may take a while before coming below the threshold in the presence of legitimate purchases occurring after fraudulent transactions. The AUC gets slightly worse when moving away from setting 0, because the correct identification of transactions becomes more difficult as explained for the previous metrics. Analogous results hold true for the other models (3.4)–(3.5).

Generally speaking we can state that the since Φ_α is a Markov process and therefore depends on its past values, a different fraud ratio and/or a different mix between fraudulent and legitimate transactions have non negligible impacts on the final metrics. This finding is not true for the other methodologies of Table 4, where transactions are treated as independent, in the sense that their temporal order is irrelevant. Indeed, Figures 7 and 8 show that the FPR, TPR and AUC of boosting and neural networks remain pretty stable across the four different settings, while the NPV and Pr increases and decreases, respectively. Similar considerations hold true for the other analyzed data mining techniques.

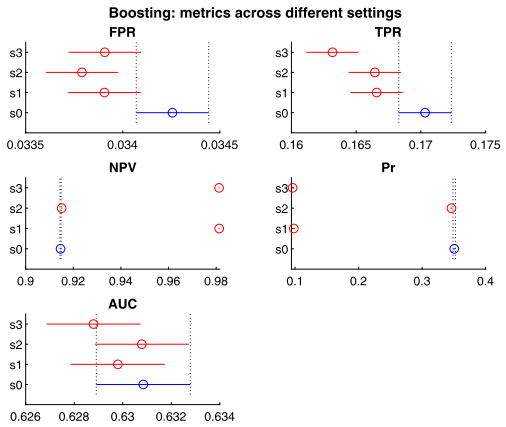

Figure 7: FPR, TPR, NPV, Pr and AUC across the four different settings for boosting. The circle associated with setting 0 (third line of the "Boosting" section in Table 4) is blue, while the other circles are red.

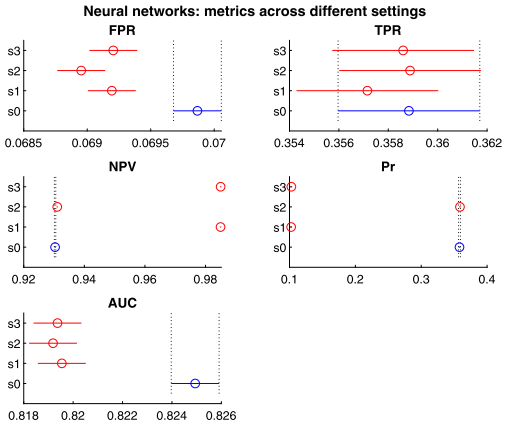

Figure 8: FPR, TPR, NPV, Pr and AUC across the four different settings for neural networks. The circle associated with setting 0 (third line of the "Neural Networks" section in Table 4) is blue, while the other circles are red.

Additional simulations have been performed by fixing a lower fraud ratio (from 1.5% to 0.05%) in setting 3. The results confirm the previous tendency: in our model, the FPR falls below 1% and the TPR decreases up to 0.54; the NPV increases above 0.99 while the Pr falls to 0.31; the AUC decreases to 0.93. For boosting and neural networks (with similar conclusions for the other techniques), the FPR, TPR and AUC remains almost unchanged as in Figures 7–8; the NPV raises above 0.99, while the Pr decreases to 0.002.

4.4 Models testing on real transactions

We tested our calibrated models also on real credit card transactions. We used the transactions occurred between the 1st and the 7th of December 2016. Since not all the 4,077 cardholders of the initial dataset (which covers June – November 2016, see Section 4.1) made purchases during this period, our testing dataset refers to 1,441 of them and contains 4,237 transactions, of which 150 are fraudulent. Similarly to the analysis of Table 3, we studied the performance of our models under the three information schemes of Section 4.2; as benchmark we used the classification models of Table 4.

Table 5 and Figure 9 report the obtained results. For example, for the logistic regression we see that when only the elapsed time between two consecutive transactions is observed, all the legitimate transactions are labeled correctly, but all the fraudulent transactions are not detected (the TPR is zero); however, when also the amounts and the geographical coordinates are considered, the FPR rises to about 5.7%, but the TPR increases to 32%. Similar considerations hold true for the other classification models, for which the TPR usually increases as more information becomes available. If we concentrate on the most complete information scheme, we see that random forests, support vector machines and BART are the most conservative in terms of the FPR (1.7%, 2.5%

	Acc	FPR	TPR	NPV	Pr	MCC	AUC
Logistic regression							
ED	0.96479	0	0	0.96479	–	–	0.37633
ED + TrxAm	0.95845	0.00827	0.04667	0.96632	0.17073	0.08125	0.60627
ED + TrxAm + Coo	0.92113	0.05693	0.32000	0.97330	0.17021	0.18370	0.66308
Decision trees							
ED	0.92089	0.04841	0.08000	0.96592	0.05687	0.02682	0.40920
ED + TrxAm	0.90117	0.06910	0.08667	0.96543	0.04377	0.01271	0.59454
ED + TrxAm + Coo	0.92535	0.04622	0.14667	0.96838	0.10377	0.08512	0.59933
Boosting							
ED	0.89131	0.07761	0.04000	0.96341	0.01846	−0.0261	0.62067
ED + TrxAm	0.91714	0.05158	0.06000	0.96509	0.04072	0.00699	0.62108
ED + TrxAm + Coo	0.92887	0.04355	0.17333	0.96942	0.12683	0.11176	0.63701
BART							
ED	0.96479	0	0	0.96479	–	–	0.56752
ED + TrxAm	0.94765	0.01946	0.04667	0.96573	0.08046	0.03544	0.61129
ED + TrxAm + Coo	0.94601	0.03041	0.30000	0.97433	0.26471	0.25385	0.76036
Random forests							
ED	0.91925	0.05206	0.13333	0.96667	0.08547	0.03916	0.52379
ED + TrxAm	0.92700	0.04038	0.03333	0.96466	0.02924	−0.0035	0.62094
ED + TrxAm + Coo	**0.95047**	**0.01678**	0.05333	0.96575	0.10390	0.04504	0.77078
Hidden Markov model							
ED	0.94531	0.02092	0.02000	0.96484	0.03370	0.00159	0.57551
ED + TrxAm	0.90423	0.06666	0.10667	0.96625	0.05517	0.02927	0.64705
ED + TrxAm + Coo	0.90258	0.06861	0.11333	0.96642	0.05685	0.03226	0.64707
Support vector machines							
ED	0.96479	0	0	0.96479	–	–	0.30394
ED + TrxAm	0.91643	0.05474	0.12667	0.96738	0.07786	0.05704	0.45921
ED + TrxAm + Coo	0.94155	0.02481	0.02000	0.96462	0.02857	−0.0057	0.91391
Neural networks							
ED	0.96479	0	0	0.96479	–	–	0.37633
ED + TrxAm	0.94390	0.02433	0.07333	0.96650	0.09909	0.05669	0.60823
ED + TrxAm + Coo	0.91385	0.06593	0.36000	0.97560	0.16615	0.20417	0.69502
Proposed model							
ED	0.93132	0.04624	0.32000	0.97450	0.20253	0.22014	0.86351
ED + TrxAm	0.94147	0.04135	0.47333	0.98024	0.29583	0.34534	0.91630
ED + TrxAm + Coo	0.93114	0.06564	**0.84138**	**0.99396**	**0.31443**	**0.48911**	**0.95955**

Table 5: Real data: metrics for logistic regression, decision trees, boosting, BART, random forests, hidden Markov model, support vector machines, neural networks and the linear model (3.3) with $c_1 = 0.1$, across different information schemes. For the third information scheme, the bold font is used to highlight the best value of each metric across the different models.

and 3%, respectively), while logistic regression, BART and neural networks are more prone to detect fraud as their TPRs (higher than 30%) suggest. Let us specify that the thresholds with which the fraudulent probabilities have been compared were fixed to 0.1 for the logistic regression, 0.07 for BART, 0.08 for the hidden Markov model

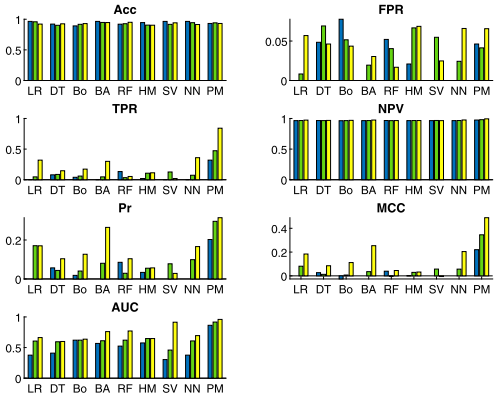

Figure 9: Metrics from Table 5. For each metric, the corresponding values across the three information schemes are shown: blue bars are for ED, green bars are for ED + TrxAm and yellow bars are for ED + TrxAm + Coo. LR, DT, Bo, BA, RF, HM, SV, NN and PM stand for logistic regression, decision trees, boosting, BART, random forests, hidden Markov model, support vector machines, neural networks and our proposed model (3.3) with $c_1 = 0.1$, respectively.

and 0.2 for the neural network (several threshold values have been tried, but those just reported seem to return the best results). The linear model (3.3) with $c_1 = 0.1$ shows FPR values which are similar to those of the majority of the benchmark models (about 4% and 6.5% depending on the considered information scheme), but is also characterized by a much higher TPR (from about 32% to 84%), which denotes its good ability to detect fraud. The good performance of our model is also confirmed by the value of the AUC that for each information scheme far exceeds the ones of the benchmarks. Moreover, our model is also fast: it takes 0.0014 seconds on average to score and classify a transaction. Similar conclusions can be drawn when the other models of Table 3 are considered.

5 Conclusions

In this work we addressed the problem of fraud detection in credit card transactions. Our main contributions are: **(i)** the application of a new detection methodology based on a Bayesian formulated optimal stopping problem, where the trade-off between an early false detection and a late fraud discovery is taken into account and where the cardholders' expenditures process are assumed to evolve according to a univariate or multivariate compound Poisson process. The Bayesian character of the problem rests on the prior exponential distribution of the fraud time and on the use of posterior probability process Π in (3.6) (or, equivalently, the generalized odds process Φ_α) as sufficient statistics for the optimal detection strategy (3.11)–(3.12); **(ii)** the computation of cardholders specific optimal thresholds with which posterior probabilities are compared to discriminate between legitimate and fraudulent transactions. This is a direct

consequence of the employed optimal stopping approach and allowed us to overcome the hurdle of how to determine a decision threshold. The latter represents one of the main critical issues in fraud detection problems, usually addressed in the available literature by fixing an exogenous, cardholder independent and thus not personalized threshold.

The proposed models have been calibrated on a set of real credit card transactions, under different information schemes involving part or all of the transactions attributes at our disposal: elapsed days, amounts and geographical coordinates. Then, the models have been applied to score simulated and real transactions and the results have been compared with those of other data mining approaches. The following are our findings: **(iii)** on simulated data with a high fraud ratio of 10% and all the fraudulent transactions occurring after the legitimate ones, our models have superior performance than that of the existing methodologies; **(iv)** under noisy simulated scenarios of legitimate cardholders' behavior, our method is robust enough to perform better than the existing methodologies; **(v)** when data are simulated by weakening the conditions of point (iii), the metrics returned by our models suffer a statistically significant worsening. This fact is due to their Markovian nature, for which the temporal order of the transactions has an important impact; instead, this limit does not affect the other classification techniques, which treat data as independent; **(vi)** when real data are used for testing, the FPR returned by our models is similar to that returned by the other methods, even though the TPR, Pr and AUC metrics beat the benchmarks.

Let us observe that, unlike other methodologies used in fraud detection, our approach is not a black box, since the target functions (3.3)–(3.5) are clearly stated and can be computationally determined. The proposed models are also flexible, in the sense that new attributes of a transaction can be incorporated, and very general, because they could be applied to other frameworks, such as intrusions detection in government or private network systems. Our models must be calibrated for each cardholder and this is an advantage in that decisions are personalized, but, at the same time, also presents three drawbacks: **(a)** an adequate computational power to speed up the training phase is required; **(b)** the payment history of a cardholder needs to be sufficiently long for a meaningful estimate of her behavior and, accordingly, **(c)** transactions of new cardholders not present in the training dataset cannot be scored. Then, we believe that future research in the area of fraud detection could be devoted to the development of hybrid models that mix the existing data mining techniques with our proposal, in order to fully exploit their potential. For example, a "two-steps" procedure could be adopted: in the first step a standard technique is used; in the second step the proposed method could ease the identification of the fraudulent transactions, among those to which a high suspicious score has been previously assigned.

Supplementary Material

Supplementary Material for "Bayesian Quickest Detection of Credit Card Fraud" (DOI: 10.1214/20-BA1254SUPPA; .pdf).

Supplementary Material for "Bayesian Quickest Detection of Credit Card Fraud" (DOI: 10.1214/20-BA1254SUPPB; .zip).

References

Bahnsen, A. C., Aouada, D., Stojanovic, A., and Ottersten, B. (2016). "Feature engineering strategies for credit card fraud detection." *Expert Systems with Applications*, 51: 134–142. doi: https://doi.org/10.1016/j.eswa.2015.12.030. 263, 264

Bayraktar, E. and Dayanik, S. (2006). "Poisson disorder problem with exponential penalty for delay." *Mathematics of Operations Research*, 31: 217–233. MR2233993. doi: https://doi.org/10.1287/moor.1060.0190. 263

Bayraktar, E., Dayanik, S., and Karatzas, I. (2005). "The standard Poisson disorder problem revisited." *Stochastic Processes and Their Applications*, 115: 1437–1450. MR2158013. doi: https://doi.org/10.1016/j.spa.2005.04.011. 263, 266, 268

Beibel, M. (2000). "A note on sequential detection with exponential penalty for the delay." *Annals of Statistics*, 28: 1696–1701. MR1835037. doi: https://doi.org/10.1214/aos/1015957476. 266

Bhattacharyya, S., Jha, S., Tharakunnel, K., and Westland, J. C. (2011). "Data mining for credit card fraud: a comparative study." *Decision Support Systems*, 50: 602–613. doi: https://doi.org/10.1016/j.dss.2010.08.008. 262, 263, 264, 265, 270, 277

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Singapore: Springer. MR2247587. doi: https://doi.org/10.1007/978-0-387-45528-0. 265

Bolton, R. J. and Hand, D. J. (2002). "Statistical fraud detection: a review." *Statistical Science*, 17: 235–255. MR1963313. doi: https://doi.org/10.1214/ss/1042727940. 263, 270

Breiman, L. (2001). "Random forests." *Machine Learning*, 45: 5–32. MR3874153. doi: https://doi.org/10.1023/A:1010933404324. 279

Buonaguidi, B. and Muliere, P. (2015). "On the disorder problem for a negative binomial process." *Journal of Applied Probability*, 52: 167–179. MR3336853. doi: https://doi.org/10.1239/jap/1429282613. 263, 268

Buonaguidi, B., Mira, A., Bucheli, H. and Vitanis, V. (2020a). "Supplementary Material of "Bayesian Quickest Detection of Credit Card Fraud"." *Bayesian Analysis*. doi: https://doi.org/10.1214/20-BA1254SUPPA. 269

Buonaguidi, B., Mira, A., Bucheli, H. and Vitanis, V. (2020b). "Supplementary Material of "Bayesian Quickest Detection of Credit Card Fraud"." *Bayesian Analysis*. doi: https://doi.org/10.1214/20-BA1254SUPPB. 269

Carneiro, N., Figueira, G., and Costa, M. (2017). "A data mining based system for credit card fraud detection in e-tail." *Decision Support Systems*, 95: 91–101. doi: https://doi.org/10.1016/j.dss.2017.01.002. 262, 263, 264, 265, 270, 277

Chan, P. K., Fan, W., Prodromidis, A. L., and Stolfo, S. J. (1999). "Distributed data mining in credit card fraud detection." *IEEE Intelligent Systems*, 14: 67–74. doi: https://doi.org/10.1109/5254.809570. 264

Chipman, H. A., George, E. I., and McCulloch, R. E. (2010). "BART: Bayesian additive regression trees." *Annals of Applied Statistics*, 4: 266-298. MR2758172. doi: https://doi.org/10.1214/09-AOAS285. 264

Dayanik, S. and Sezer, S. O. (2006). "Compound Poisson disorder problem." *Mathematics of Operations Research*, 31: 649–672. MR2281222. doi: https://doi.org/10.1287/moor.1060.0223. 263, 267, 268, 269

Davis, M. H. A. (1976). "A note on the Poisson disorder problem." *Banach Center Publications*, 1: 65–72. doi: https://doi.org/10.4064/-1-1-65-72. 262, 266

Davis, M. H. A. (1993). *Markov Models and Optimization*. Monograph on Statistics and Applied Probability 49, London: Chapman & All. MR1283589. doi: https://doi.org/10.1007/978-1-4899-4483-2. 268

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). "Maximum likelihood from incomplete data via the EM algorithm." *Journal of the Royal Statistical Society. Series B*, 39: 1–38. MR0501537. 273

Dorronsoro, J. R., Ginel, F., Sanchez, C., and Cruz, C. S. (1997). "Neural fraud detection in credit card operations." *IEEE Transactions on Neural Networks*, 8: 827–834. doi: https://doi.org/10.1109/72.595879. 265

Gal'Chuk, L. I. and Rozovskii, B. L. (1971). "The "disorder" problem for a Poisson process." *Theory of Probability and Its Applications*, 16: 712–716. MR0297028. doi: https://doi.org/10.1137/1116081. 262

Gapeev, P. V. (2005). "The disorder problem for compound Poisson processes with exponential jumps." *Annals of Applied Probability*, 15: 487–499. MR2115049. doi: https://doi.org/10.1214/105051604000000981. 263, 268

Gapeev, P. V. and Peskir, G. (2006). "The Wiener disorder problem with finite horizon." *Stochastic Processes and Their Applications*, 116: 1770–1791. MR2307058. doi: https://doi.org/10.1016/j.spa.2006.04.005. 262

Gapeev, P. V. and Shiryaev, A. N. (2013). "Bayesian quickest detection problems for some diffusion processes." *Advances in Applied Probability*, 45: 164–185. MR3077545. doi: https://doi.org/10.1017/S0001867800006236. 262, 268

Glady, N., Baesens, B., and Croux, C. (2009). "A modified Pareto/NBD approach for predicting customer lifetime value." *Expert Systems with Applications*, 36: 2062–2071. doi: https://doi.org/10.1016/j.eswa.2007.12.049. 266

Han, J., Kamber, M., and Pei, J. (2012). *Data Mining: Concepts and Techniques*. Waltham: Elsevier. MR2911453. doi: https://doi.org/10.1016/j.entcs.2011.03.007. 278

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2nd edition. MR2722294. doi: https://doi.org/10.1007/978-0-387-84858-7. 265

Herberts, T. and Jensen, U. (2004). "Optimal detection of a change point in a Poisson process for different observation schemes." *Scandinavian Journal of Statistics*, 31:

347–366. MR2087830. doi: https://doi.org/10.1111/j.1467-9469.2004.02-102.x. 263

Johnson, P. and Peskir, G. (2017). "Quickest detection problems for Bessel processes." *Annals of Applied Probability*, 27: 1003–1056. MR3655860. doi: https://doi.org/10.1214/16-AAP1223. 262, 267, 268

Jurgovsky, J., Granitzer, M., Ziegler, K., Calabretto, S., Portier, P., He-Guelton, L., and Caelen, O. (2018). "Sequence classification for credit-card fraud detection." *Expert Systems with Applications*, 100: 234–245. doi: https://doi.org/10.1016/j.eswa.2018.01.037. 262, 265

Kecman V., Huang, T. M., and Vogt, M. (2005). "Iterative single data algorithm for training kernel machines from huge data sets: theory and performance." *Support Vector Machines: Theory and Applications*, L. Wang, eds., pp. 255–274, Berlin: Springer-Verlag. doi: https://doi.org/10.1007/10984697_12. 279

Ko, S. I. M., Chong, T. T. L., and Ghosh, P. (2015). "Dirichlet process hidden Markov multiple change-point model." *Bayesian Analysis*, 10: 275–296. MR3420883. doi: https://doi.org/10.1214/14-BA910. 264

Letham, B., Rudin, C., McCormick, T. H., and Madigan, D. (2015). "Interpretable classifiers using rules and Bayesian analysis: building a better stroke prediction model." *Annals of Applied Statistics*, 9: 1350–1371. MR3418726. doi: https://doi.org/10.1214/15-AOAS848. 263

Mahmoudi, N. and Duman E. (2015). "Detecting credit card fraud by modified Fisher discriminant analysis." *Expert Systems with Applications*, 42: 2510–2516. doi: https://doi.org/10.1016/j.eswa.2014.10.037. 262, 263, 265

Ngai, E. W. T., Hu, Y., Wong, Y. H., Chen, Y., and Sun, X. (2011). "The application of data mining techniques in financial fraud detection: a classification framework and an academic review of literature." *Decision Support Systems*, 50: 559–569. doi: https://doi.org/10.1016/j.dss.2010.08.006. 263

The Nilson Report (2017). *Card Fraud Losses Reach $ 22.80 Billion*, issue 1118 – October. 261

Peskir, G. and Shiryaev, A. N. (2002). "Solving the Poisson disorder problem." *Advances in Finance and Stochastics. Essays in Honour of Dieter Sondermann*, K. Sandmann and P. Schönbucher, eds., pp. 295–312, Berlin: Springer. 262, 265, 267, 268

Peskir, G. and Shiryaev, A. N. (2006). *Optimal Stopping and Free-Boundary Problems*, Lectures in Mathematics ETH Zürich, Basel: Birkhäuser Verlag. 263, 268

Polson, N. G., Scott, J. C., and Willard, B. T. (2015). "Proximal algorithms in statistics and machine learning." *Statistical Science*, 30: 559–581. 265

Polson, N. G. and Sokolov, V. (2017). "Deep learning: a Bayesian perspective." *Bayesian Analysis*, 12: 1275–1304. 265

Poor, H. V. (1998). "Quickest detection with exponential penalty for delay." *Annals of Statistics*, 26: 2179–2205. 266

Quah, J. T. S. and Sriganesh, M. (2008). "Real-time credit card fraud detection using computational intelligence." *Expert Systems with Applications*, 35: 1721–1732. 262, 265

Raynal, L., Marin, J., Pudlo, P., Ribatet, M., Robert, C. P., and Estoup, A. (2019). "ABC random forests for Bayesian parameter inference." *Bioinformatics*, 35: 1720–1728. 264

Schmittlein, D. C., Morrison, D. G., and Colombo, R. (1987). "Counting your customers: who are they and what will they do next?" *Management Science*, 33: 1–24. 262, 265

Seiffert, C., Khoshgoftaar, T. M., Hulse, J. V., and Napolitano, A. (2008). "RUSBoost: improving classification performance when training data is skewed." 19^{th} *International Conference on Pattern Recognition*, 1—4. 278

Shiryaev, A. N. (1978). *Optimal Stopping Rules*. New York: Springer-Verlag. 262, 263, 268

Snoek, J., Larochelle, H., and Adams, R. P. (2012). "Practical Bayesian optimization of machine learning algorithms." *Advances in Neural Information Processing Systems*, 25: 2951–2959. 265

Srivastava, A., Kundu, A., Sural, S., Majumdar, A. K. (2008). "Credit card fraud detection using hidden Markov model." *IEEE Transactions on Dependable and Secure Computing*, 5: 37–48. 262, 264

Yeh, I-C. and Lien, C. (2009). "The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients." *Expert Systems with Applications*, 36: 2473–2480. 263, 265

Zaslavsky, V. and Strizhak, A. (2006). "Credit card fraud detection using self-organizing maps." *Information and Security*, 18: 48–63. 262, 265

Acknowledgments

The authors are grateful to the Editor, the Associate Editor and the referees for their valuable suggestions, which contributed to improve the presentation of the paper. The first author acknowledges the support from the Axa Research Fund, project nr. 16-AXA-PDOC-213.