

RESEARCH

Open Access



# Online payment fraud: from anomaly detection to risk management

Paolo Vanini<sup>1\*</sup> , Sebastiano Rossi<sup>2</sup>, Ermin Zvizdic<sup>3</sup> and Thomas Domenig<sup>4</sup>

\*Correspondence:  
paolo.vanini@unibas.ch

<sup>1</sup> University of Basel, Basel, Switzerland

<sup>2</sup> Novartis AG, Basel, Switzerland

<sup>3</sup> swissQuant Group, Zurich, Switzerland

<sup>4</sup> IT Couture, Zurich, Switzerland

## Abstract

Online banking fraud occurs whenever a criminal can seize accounts and transfer funds from an individual's online bank account. Successfully preventing this requires the detection of as many fraudsters as possible, without producing too many false alarms. This is a challenge for machine learning owing to the extremely imbalanced data and complexity of fraud. In addition, classical machine learning methods must be extended, minimizing expected financial losses. Finally, fraud can only be combated systematically and economically if the risks and costs in payment channels are known. We define three models that overcome these challenges: machine learning-based fraud detection, economic optimization of machine learning results, and a risk model to predict the risk of fraud while considering countermeasures. The models were tested utilizing real data. Our machine learning model alone reduces the expected and unexpected losses in the three aggregated payment channels by 15% compared to a benchmark consisting of static if-then rules. Optimizing the machine-learning model further reduces the expected losses by 52%. These results hold with a low false positive rate of 0.4%. Thus, the risk framework of the three models is viable from a business and risk perspective.

**Keywords:** Payment fraud risk management, Anomaly detection, Ensemble models, Integration of machine learning and statistical risk modelling, Economic optimization machine learning outputs

## Introduction

Fraud arises in the financial industry via numerous channels, such as credit cards, e-commerce, phone banking, checks, and online banking. Juniper Research (2020) reports that e-commerce, airline ticketing, money transfer, and banking services will cumulatively lose over \$ 200 billion due to online payment fraud between 2020 and 2024. The increased sophistication of fraud attempts and the increasing number of attack vectors have driven these results. We focus on online and mobile payment channels and identity theft fraud (i.e., stealing an individual's personal information to conduct fraud) (Amiri and Hekmat 2021). The aim is to identify external fraudsters who intend to initiate payments in their interests. As fraudsters gain access to the payment systems as if they were the owners of the accounts, they cannot be identified based on the account access process. However, the fraudster behaves differently during a payment transaction

than the account owner and/or the payment has unusual characteristics, such as an unusually high payment amount or transfer to an account in a jurisdiction that does not fit the life context and payment behavior of the customer. The assumption is that algorithms can detect anomalies in behavior during payment transactions.

West and Bhattacharya (2016), Abdallah et al. (2016), Hilal et al. (2021), and Ali et al. (2022) reviewed financial fraud. They found a low number of articles regarding online payment fraud. For example, Ali et al. (2022) cited 20 articles on financial statement fraud and 32 articles on credit card fraud, see Li et al. (2021) for credit card fraud detection. Online payment fraud was not listed. The reviews also clarified that many articles utilized aggregated characteristics. However, we emphasize that fraud in online payments can only be detected based on individual data, as such fraud can only be detected through the possible different behavior of the fraudster and the account holder during payments. As fraudsters learn how best to behave undetected over time, they adapt their behavior. Therefore, self-learning defense methods are expected to outperform static-rule-based algorithms. The correctness of the expectation was shown by Abdallah et al. (2016) and Hilal et al. (2021). Various machine-learning algorithms for fraud detection have been proposed in the literature, including decision trees, support vector machines, and logistic regression to neural networks. Aggarwal and Sathe (2017) discussed various methods for outlier ensembles, and Chandola et al. (2009) provided a taxonomy and overview of anomaly detection methods.

A common feature in many studies is imbalanced data (i.e., the low proportion of fraud events in the dataset, see Wei et al. 2013; Carminati et al. 2015; Zhang et al. 2022a; Singh et al. 2022). Risk detection involves detecting fraudulent transactions and stopping them before execution.

In addition to the efficiency of the algorithms, the data basis is an important reason for the differences in fraud-detection performance. While many studies have utilized either often less rich synthetic or Kaggle data, we were able to work with real data. Log files, which have substantial information content in our work, are hardly expected in Kaggle data. The difference in the data complexity is also reflected in the number of features. Singh et al. (2022) showed that the feature space consists of 31 features compared to our 147 features. Moreover, the proportion of fraudulent transactions in Singh et al. (2022) is more than a hundred times higher than in our case. Consequently, our data are much more unbalanced than any other study we know of, and the task of finding efficient fraud detection algorithms is more difficult.

However, limiting risk management to the optimal detection of anomalies does not ensure that losses caused by fraud are minimal. Optimal fraud detection can be economically suboptimal if, for example, it is efficient for small amounts of money but unsuccessful for large amounts. Thus, the machine learning outputs for risk identification must be optimized from an economic perspective. We call this optimization the triage model. Yet, neither fraud detection nor the triage model can provide an answer to the question of how large the losses in a payment channel are. Therefore, we develop a statistical risk model that considers the effects of countermeasures on loss potential. The risk model provides risk transparency and makes it possible to assess which measures in the fight against fraud in various payment channels make sense from an economic and risk perspective.

Literature on fraud risk models often refers to qualitative or assessment models for assessing fraud risk or risk assessment models (Sabu et al. 2021). We are not aware of any quantitative fraud risk management framework that explicitly considers the impact of the fraud detection process statistically in risk modelling. For organizational, procedural, and legal risk aspects, we refer to the literature. Montague's (2010) book focuses on fraud prevention in online payments but does not consider machine learning and risk management in detail. The Financial Conduct Authority's Handbook (FCA 2021) provides a full listing of the FCA's legal instruments, particularly those relating to financial crime in financial institutions. Power (2013) highlights the difference between fraud and fraud risk from historical and business perspectives. Van Liebergen (2017) looks at "regtech" applications of machine learning in online banking. Fraud risk events in cryptocurrency payment systems are different from the online banking cases under consideration; see Jung et al. (2019) for fraud acting on a decentralized infrastructure and the review article of Trozze et al. (2022).

The development and validation of the three linked models are the main contributions of our work. To our knowledge, this is the first study to develop, validate, and link components of the risk management process. The output of the anomaly detection model (i.e., the ROC curves), is the input for the triage model, which provides economically optimized ROC curves. Fraud statistics data were utilized to calibrate the various components in the risk model. With these three models, the fraud risk management process can be qualitatively implemented at the same level as the risk management of market or counterparty risks (see Bessis (2011) to describe risk management in banks).

The performance of our risk management framework is the second contribution, although the performance comparison of our fraud detection method with the literature is limited and cautious, due the use of synthetic data instead of real data, a consideration of different channels in payments with different behavioral characteristics of bank customers, and the publication of incomplete statistics. Nevertheless, we compared our work with Wei et al. (2013) and Carminati et al. (2015), both of which analyze online banking fraud based, in part, on real data. The true positive rate (TPR) at a false positive rate (FPR) of 1% was 45%. In Wei et al. the TPR is between 49% and 60%, but unfortunately, the FPR is unknown. In the relevant scenario of Carminati et al. (2015), the TPR is 70% with an FPR of 14%. This FPR is not acceptable to any bank. Processing by specialists leads to high costs. We discuss all these statements in detail in the "[Validation results](#)" section. Considering all three models, the theoretical and practical importance of our approach becomes clear. The expected losses in a scenario of CHF 2.023 million, which utilizes the results of machine learning without economic optimization in the triage model, common in the literature, are reduced to CHF 0.800 million with the triage model (i.e., a reduction in the loss potential by more than 60% follows). In addition, if fraud detection is implemented without a risk model, fraud risk can be massively overestimated. Applying our models to three different payment channels, the overestimation of risk ranged from 54% to over 700%.

The remainder of this paper is organized as follows. In "[Fraud risk management framework](#)" section, the model selection for the fraud risk management framework is motivated and described. In "[Online payment fraud anomaly detection](#)" section, we consider the anomaly-detection model. "[Fraud detection triage model](#)" section links fraud

detection to an economic perspective utilizing the triage model. "Risk model" presents the statistical risk model. "Conclusion" section concludes.

### **Fraud risk management framework**

We provide an overview of the three interrelated quantitative models in the context of risk management: online payment anomaly detection, triage model, and risk model.

#### **Online payment fraud anomaly detection**

The goal of anomaly detection is to detect fraudulent activities in e-banking systems and to maintain the number of false alarms at an acceptable level. The implementation of the model consists of three steps: pre-filter, feature extraction, and machine learning.

Non-learning pre-filters ensure that both obvious fraud and normal transactions are sorted early to reduce the false positive rate. Only transactions that pass the pre-filter step are passed on to the machine-learning model. Banks utilize non-customer-based static if-then rules, such as blacklists or whitelists. Pre-filters free the algorithms from obvious cases. The adaptability and flexibility of the machine-learning model is necessary to counter the ever-improving attacks of fraudsters with effective fraud detection.

Our data face the following general challenges in payment fraud detection (per Wei et al. 2013): large transaction volume with the need for real-time fraud detection, a highly imbalanced dataset, dynamic fraud behavior, limited forensic information, and varying customer behavior.

Given the extremely imbalanced data, fully supervised algorithms typically struggle. Aggarwal and Sathe (2017) proposed unsupervised and weakly supervised approaches based on features that encode deviations from normal behavior. For each customer participating in an e-banking session, we assess whether the agent's behavior is consistent with the account holder's normal behavior. The key information for behavioral analysis lies in the sequence of the customer's clicks during the session. We show that, unlike online e-commerce transactions (see Wei et al. 2013), transaction data, customer behavior data, account data, and booking data are also important for the performance of the algorithm. More precisely, these features are divided into behavioral, transactional, and customer-related features. Starting with nearly 800 features, 147 were extracted utilizing a Bagged Decision Tree Model (BDT). These numbers are many times higher than those for credit card fraud with one to two dozen features (see Table 8 in Hilal et al. 2022). The high dimensionality of the feature space also arises in machine learning corporate default prediction models, where several steps are needed to extract all noisy features (see Kou et al. 2021).

Our e-fraud model operates according to the following principles.

- The model learns the "normal" behavior of each customer based on historical payment data and online banking log data.
- Each new transaction is checked against the learned "normal" behavior to determine if it is an anomaly by extracting the 147 features from the data.
- If an anomaly is detected, it is flagged as suspected fraud.
- Detected transactions that are not found to be fraudulent after manual review are reported back to the model for learning purposes.

As there are very few known fraud cases, all base learners are trained on fraud-free data only in step one. Fraud cases are only utilized in step two of ensemble aggregation when the base learners are combined to form the final predictive function. The first step is to define base learners who are rich enough to detect a wide range of suspicious transactions or online user sessions. We consider three base learners: the density-based outlier detection model (unsupervised, Local Outlier Factor (LOF)), the isolation-based outlier detection model (unsupervised, Isolation Forest (IF)), and a model for normal customer behavior (supervised, Bagged Decision Trees (BDT)) as base learners (see Breunig et al. (2000), Chandola et al. (2009); Zhang et al. (2022b) for LOF, Liu et al. (2012); Tokovarov and Karczmarek (2022) for IF). We refer to individual instances of LOF, IF, or BDT as base learners. The BDT model is not only a base model, but it is also utilized for feature selection in the other two base models: LOF and IF. The LOF method is suitable for outlier detection, where each observation is assigned an outlier level based on its distance from the nearest cluster of neighboring observations. The aim is to detect outliers in inhomogeneous data, for which classical global outlier methods typically do not provide satisfactory results. Conversely, IF explicitly isolates anomalies without capturing all normal instances. These two methods consider the heterogeneity in the data.

In the second stage, the base learner's fraud score was aggregated. We consider two approaches to determine the weights in the ensembles: a simple averaging and a supervised approach, although our model largely consists of unsupervised procedures because of the limited availability of fraud cases for which we can extract all the required features. However, we introduce supervision where we utilize scarce labelled data to adjust the importance of certain base learners in the voting scheme, ultimately deciding whether an observation is fraudulent. The penalized logistic regression chosen for classification allows for a better interpretation of the model, as the weights can be utilized to identify base learners, subsets of features, and subsets of samples that have been particularly useful in detecting a particular type of fraud.

### **Triage model**

The fraud detection model calculates scores and, in comparison with a threshold value, decides whether a transaction is flagged as an anomaly. This process results in the probability of detection for a given investigation effort, as indicated by the ROC curve. By making the threshold dependent on the transaction size, we can ensure that larger transaction amounts are more likely detected than smaller ones. This gives up part of the true positive rate (TPR) to reduce overall economic losses (i.e., the TPR decreases for a given FPR). This economic optimization that leads to adjusted ROC curves defines the triage model.

To minimize expected cumulative losses, the constant fraud anomaly detection threshold becomes a function of the transaction amount. Here, the transaction amounts are random variables whose distributions are estimated. In the optimization problem, the transaction function is chosen to maximize the average cumulative sum of the detected fraudulent transactions, where the expected FPR must not exceed a certain threshold. Utilizing this optimal threshold function, the fitted ROC curves were obtained.

The optimization problem has a unique solution if the ROC curve is a concave function of the false positive rate function of the threshold and if the acceptance set of the

expected false positive function constraint is convex. With the chosen piecewise linear false positive constraint function, the assumptions regarding the existence of an optimum are satisfied. The ROC curves that result when fraud anomalies are detected serve as inputs for the optimization. However, because only a vanishingly small number of fraud cases exist, the TPR values for certain FPR levels are subject to considerable uncertainty. Hence, cubic spline functions were utilized for the ROC curve of the optimization.

The UK Finance (2019) report states that the recovery value for online and mobile banking in the UK is 18% of the potential loss. Therefore, we introduced an extension to the optimization program to include recovery.

### **Risk model**

Losses from transaction fraud are included in operational risk incurred by banks. As for other operational risks, one of the key questions from a risk-management perspective is whether the allocated resources and countermeasures are adequate. To answer this, one needs some way of quantifying the risk incurred, ideally a Value-at-Risk (VaR)-model that fits the general risk framework of the bank. Simply, the model calculates the loss  $L = E(\lambda) \times E(\tau)$  where  $\lambda$  is the expected event frequency (fraud), and  $\tau$  is the expected loss per event. The challenge is to determine the distributions of these variables in a tractable and plausible manner and define a model while having very scarce data on past events. We chose the path of an aggregated simulation of many scenarios per e-channel to account for the inherent uncertainty in the choice of these parameters.

Unlike market or credit risk, fraud risk is borne by comparatively few individuals or groups who utilize very specific strategies and technologies to overcome vulnerability in the payment process. Simultaneously, defenders analyze attack plans and update their countermeasures. In this constantly changing environment, neither the frequency of attacks nor the transaction amounts can be assumed to be statistically regular with great certainty. Therefore, we propose a simple, flexible stochastic model composed of basic building blocks. With such a model, risk managers can quickly adjust the model as needed, perform what-if analyses, or simulate changes in payment infrastructure.

The basic model structure for the e-fraud risk model consists of (i) independent models for the three channels, whose components and parameters can be flexibly assembled and adjusted, (ii) sub-models in each channel based on three model types, and (iii) a recovery model for each channel. The three model types for the three different online payment channels in this study are a Beta model (restricted distribution of transaction amounts), a Generalized Pareto Distribution (GPD, unrestricted distribution of transaction amounts), and a “mass attack model” (many simultaneous Beta-type attacks).

Countermeasures against fraud and recovery measures after fraud events play an essential role in determining risk potential. Therefore, they were integrated into the risk models. Countermeasures against online fraud can be divided into those that strengthen the general infrastructure of the payment process and those that focus on defense against actual attacks. The former is conceptually part of the risk model described above, as it affects the frequency and possibly the transaction size of the attacks. However, the latter is better understood in the context of recovery and is considered in the triage model.

## Online payment fraud anomaly detection

### Data

Raw data consisted of transaction data, interaction data between customer and e-banking interface, account, booking, and customer reference data. All users with fewer than 10 logged online sessions were removed as input for ensemble learning. The removed cases were handled separately by utilizing a case-back model.

The transaction history in our dataset consists of 140 million transactions over three years. One hundred fraud cases are reported, but only 11 cases can be linked to the recorded 900'000 online session logs: a 0.0012% fraud rate. Only 900'000 of the 140 million transactions were possible as the log files were only stored in the bank for three months. This change occurred after the project.

A feature vector is created for each e-banking session based on raw data. The interaction pattern features consist of  $n$ -grams constructed from customers' request sequences and normalized deviations from the expected duration between each pair of consecutive requests sent by a customer in an online session. Particular attention was paid to the typical time required to complete a two-step verification process during enrolment. Payment pattern characteristics were calculated for weekday, weekly, and monthly seasonality. These include normalized deviations from the expected payment amount and remaining account balance. Technical data included the IP address of the online session, the HTML agent, and the number of JavaScript scripts executed. Finally, we utilized historically observed confirmed fraudulent transaction identifiers as the ground truth for the weakly monitored part of the pipeline.

Several quality checks were performed. Consistency tests ensure that the session interaction data and transactions match, for example, that the account exists or that a recipient is listed in the transaction. We also checked for missing or non-parsable values, the latter removed.

The data are extracted from several different data sources within the bank in a two-step Python extract, transform, and load (ETL) process, and converted into features for the algorithm. First, we introduce all raw data into a standard structured format for all data sources. Then, we perform the feature engineering described in the following sections to compute the data input for the ensemble.

Our fraud rate 0.0012% is much lower than that reported in the literature. The figures in the two online banking fraud papers, Wei et al. (2013) and Carminati et al. (2015), are 0.018% and 1%, respectively. For credit card fraud, the fraud case number is larger, such as 2% in Piotr et al. (2008). Outside the banking fraud sector, anomalies account for up to 40% of the observations (see Pang et al. 2020). Similar numbers hold in Zhang et al. (2022b), who tested their ensemble-based outlier detection methods on 35 real datasets from various sectors outside the financial sector, with an average fraud rate of 26%.

### Feature extraction

For weak supervision, we utilized historically observed confirmed fraudulent transaction identifiers as the ground truth. For training and inference, we created a feature vector for each e-banking session based on the raw data. Each feature aims to encode deviation from expected ("normal") customer behavior, as observed in historical

interactions with the online banking interface and executed transactions. Three types of features are considered.

### **Behavioral features**

The underlying motivation for utilizing features derived from customers' online session logs is that a large fraction of online payment fraud involves hijacking, where a foreign agent (human or robot fraudster) takes control of the e-banking session. Consequently, it is expected that the timing and sequence of requests posted by the user in a fraudulent session will be significantly different from those in a non-fraudulent session. We utilize the information about user request types (e.g., "get account balance", "create payment") and corresponding timestamps to construct the following features:

- *Normalized n-gram frequency of user's requests within the online session.* We utilized single, pairs and triplets of consecutive requests (1-, 2- and 3-grams) to derive a fixed-size representation. We performed normalization by dividing the number of occurrences for each request with the total number of requests in each session.
- *Normalized time between consecutive user request n-grams.* For each pair of recorded consecutive n-grams, we transformed absolute time between them into deviations by computing z-scores relative to respective historical observations.
- *Technical attributes of a session* (e.g., IP address of the online session, HTML agent, screen size, and the number of executed Javascript scripts) in binary format - 0 if previously observed, otherwise 1.

### **Transactional features**

Transactional features aim to quantify how anomalous aggregate payments scheduled in an online session are compared with previously executed payments and the remaining balance on the account. They are designed to capture attempts to empty a victim's account through a single large or many small transactions, while being mindful of seasonal patterns (e.g., holidays, travel expenses, bills, etc.).

- *Normalized ratio of the payment amount relative to remaining account balance.* We normalize by computing z-scores relative to historically observed ratios.
- *Deviation of the scheduled payment amount from the seasonally expected amount.* We compute four deviations per session using z-scores relative to historical payments executed in the same hour of the day, day of the week, week of the month, and month of the year, respectively.
- *Scheduled time for payment execution* in binary format - 0 if immediate, 1 if lagged.

A short payment history and many accounts with relatively infrequent transactions proved detrimental to seasonality modelling, hence, these features were omitted from the final model.



**Customer-related features**

Customer-related features provide insight into peer groups, relationships with other customers, and the service packages they utilize. These include:

- *Sociodemographic*(e.g., age, nationality, profession, and income)
- *Seniority of client relationship*
- *Product usage* (i.e., savings, investment accounts, and mortgages)
- *Relationship to other customers* (i.e., shared accounts, spouses, and family)

These features were not considered within the scope of our study because of data limitations and time constraints.

**Functionality and structure of the fraud model**

**Base learner: bagged decision tree**

Bagged decision trees (BDT) are trained utilizing the concept of transfer learning, which assumes that distinguishing between the behaviors of different clients within their online sessions is a related problem in distinguishing between fraudulent and non-fraudulent sessions. The underlying motivation considers that a large fraction of online payment fraud involves hijacking or when a foreign agent (human or robot fraudster) takes control of the e-banking session. As fraudulent sessions are rare and non-fraudulent sessions abound, the utilization of transfer learning enables the extraction of custom patterns from a much broader dataset and the use of supervised learning. Transfer learning comprises two phases. A learning phase where one discriminates behavioral characteristics of each customer vs. non-customers and a prediction phase where discrimination between non-fraudulent and fraudulent users sessions were considered in this study. The “non-customer” class label is then attributed to fraudulent behavior.

The decision function in BDT base learners is the probability that an observation, a planned transaction  $x$ , is a “non-customer behavior.” This value is equal to the average probability observed for all decision trees in the forest.

$$\delta_{\text{BDT}}(x) = \frac{1}{M} \sum_i P_i(Y \neq c_j | X = x) \tag{1}$$

where  $M$  is the number of trees in the bagged forest,  $c_j$  is the corresponding customer behavior class, and  $P_i(Y \neq c_j | X = x)$  is the probability that observation  $x$  is a “non-customer behavior” as predicted by the  $i$ th tree. Customer behavior class  $c_j$  consists of the collected sessions and transactions, excluding potential fraud. The model was fitted as follows:

- For each customer  $c_j$ , we collect the associated sessions, excluding the potential fraud cases. This set represents the “behavior of customer  $c_j$ .”
- From the pool of all customer sessions  $C_{-j}$  (excluding  $c_j$ ) we draw a uniform sample of observations to generate a set representing the class “behavior of customer  $c_j$  not,”  $\ell_j$  for short, and equal in size to the  $c_j$  set. Equal sampling is performed to ensure that none of the other customers are overrepresented in  $\ell_j$ .

- Each bagged forest in the ensemble is trained on a matching feature subspace utilizing all these observations. The forests consist of 100 decision trees each.

BDTs provide variable selection as an additional benefit, owing to the large amount of data involved in supervised classification. Therefore, they can be utilized to estimate the importance of variables and adequacy of feature engineering. We achieve this by calculating the Gini impurity decreases at each split point within each tree for each feature. Gini impurity is a measure of the likelihood that a randomly selected observation would be incorrectly classified by a specific node  $m$ :

$$I_G(m) = 1 - \sum_{i=0}^1 p_{mi}^2 \tag{2}$$

where  $p_{mi}$  is the portion of samples classified as  $i$  at node  $m$ . These impurity decreases were averaged across all decision trees and outputs to estimate the importance of each input variable. The greater the decrease in impurity, the greater the importance. Utilizing these results, we can identify the relevant subsets of input variables.

Relying on the concept of transfer learning (if the problem described in ?? is sufficiently like fraud detection), we use BDT to select a subset of  $N = 147$  features. Particularly important were features measuring the deviation from the typical time required to complete a two-step verification process during login. Features that encode the relative time between  $n$ -grams and specific user request sequences are important. The following additional base model was built using the features selected by BDT.

**Base learner: local outlier factor**

The Local Outlier Factor (LOF) detection method assigns an outlier level to each observation based on its distance from the nearest cluster of neighboring observations (Breunig et al. 2000). The general intent of the LOF model is to identify outliers in the interior region of data, for which classical global outlier methods and the other considered algorithm, isolation forest, usually do not provide satisfactory results. The LOF decision function is as follows:

$$\delta_{\text{LOF}}(x) = \frac{\frac{1}{|K|} \sum_{j \in K} LD(x_j)}{LD(x)} \tag{3}$$

with  $K$  a  $k$ -neighborhood and  $LD(x)$  the local reachability distance density from  $x$  to its  $k$ -th nearest neighbor. We fit the model for each customer by collecting the first associated sessions or transactions, excluding potential fraud. Each LOF in the ensemble is created utilizing all these observations on a subspace of the relevant features selected by BDT and the sampled hyper-parameter. Finally, each time a new observation is available, its decision function value is computed regarding the observations from the training set.

**Base learner: isolation forest**

The IF algorithm recursively splits the data into two parts based on a random threshold, until each data point is isolated. The algorithm randomly selects a feature at each step, and then randomly selects a division value between its minimum and maximum

values. The algorithm filters out data points that require fewer steps to be isolated from the entire dataset. In our case, IF separates one observation from the rest of a randomly selected subsample of the original dataset (Fei Tony et al. 2008). Anomalies are instances with short average isolation path lengths. The IF decision function is

$$\delta_{IF}(x) = 2^{-\frac{E(H(x))}{C}} \tag{4}$$

where  $E(H(x))$  is the average number of edges traversed to isolate the node and  $C$  is the average number of edges traversed in an unsuccessful search. To fit the model, we first collected each client’s associated sessions or transactions, excluding potential fraud cases. In step two, each isolation forest in the ensemble was created utilizing all these observations in a matching feature subspace. Each forest consisted of 100 isolated trees. Finally, each time a new observation was available, its decision function value was computed regarding the isolation trees created based on the training set.

**Base learner scores combination**

The decision functions of the base learners produced by our ensembles must be combined into a single fraud score. As these are in different ranges and scales to render the decision functions comparable, we first replace the original scores by their ranks, regarding the non-fraudulent training scores. Rank normalization is more robust and numerical stable as opposed to z-scores, for example. Therefore, we replaced the original scores with their ranks regarding the non-fraudulent training scores for each base learner:

$$\delta_{Base}^{Norm}(p) = \text{rank}(\delta_{Base}(p), V), \tag{5}$$

where  $V$  is the set of all  $\delta_{Base}(p)$  over all observations  $p'$  in the learners’ training subsample, with Base being LOF, IF, or BDT.

Owing to the few fraud cases, our model largely consists of unsupervised procedures. However, we introduced supervision utilizing scarce labelled data to readjust the importance of particular base learners in the voting scheme, ultimately deciding whether an observation is fraudulent.

The following score combination procedure was established. First, a training set comprises all fraud cases in the sample, along with healthy transactions uniformly sampled over customers from the ensemble training data. Second, a logistic regression is trained to classify observations as fraudulent or not utilizing the  $6N$  normalized decision function features and the known fraud status on past transactions as the label.

The following binary-class penalized cost function is minimized:

$$\min_{w,c} \left( \frac{1}{2} w' w + R \sum_{i=1}^N \alpha_i \log \left( e^{-y_i(X'_i w + c)} + 1 \right) \right) \tag{6}$$

where  $y_i$  is the fraud label of transaction  $i$ ,  $X'_i$  is a row of  $6N$  decision functions describing transaction  $i$ ,  $R$  is the regularization factor, and  $(w, c)$  is a set of weights defining the decision boundary between the two classes: fraud and non-fraud. To account for the imbalance between fraud and non-fraud transactions in our sample, we assign asymmetric penalties for fraud misclassification, as opposed to non-fraud classification.

$$\alpha_i = \begin{cases} \frac{\#samples}{2\#frauds}, & y_i = \text{fraud} \\ \frac{\#samples}{2\#non-frauds}, & \text{else.} \end{cases} \tag{7}$$

Choosing logistic regression to optimize the weights of the base learners ensures that the final score combination  $x'w$  represents the log-odds of an observation to be fraudulent.

$$x'w = \log \left( \frac{\delta(x)}{1 - \delta(x)} \right) - c \tag{8}$$

where  $\delta(x)$  is the probability that the observation  $x$  is fraudulent. Finally, to assign a fraud label to a session  $x$ , we compare the output combined score, or equivalently, the probability  $\delta(x)$ , to a threshold  $y$ , which is chosen based on ROC curve analysis such that the defined maximum allowed false positive rate is not exceeded. The decision boundary of logistic regression is linear, where each base learner is assigned a weight  $w_i$  to determine its relative importance for fraud classification. This structure simplifies the interpretation of the model because these weights can be utilized to identify the base learners, feature subsets, and sample subsets, which are particularly useful in detecting a particular type of fraud associated with a high weight  $w_i$ . Appendix A provides a detailed description of the ensemble design.

**Normal customer behavior model**

In summary, we created an ensemble model for each client, which is re-trained with new data at regular intervals and can be described by the following steps:

- We consider the disjoint sets of behavioral features on session observations and transactional features on transaction observations.
- For each of the two features/observations-pairs, we define  $N = 1000$  learners for each of the three model categories as follows.
  - We fix  $N$  random sub-samples of features from the feature set. Each sub-sample remains fixed for all customers.
  - For each customer, we fix  $N$  random observation samples from the customer-specific sessions or transactions observations.
  - For each of the three model categories, for each customer, and for  $i = 1, \dots, N$ , a base learner is defined by applying the model algorithm to the  $i$ -th features sub-sample and  $i$ -th observations sub-sample. Thus, this results in  $6N$  base learners per customer,  $3N$  for sessions, and another  $3N$  for transaction data.
- The decisions for the three base learners are aggregated utilizing supervision, where the knowledge obtained from existing fraud cases is utilized to adjust the base learner weights.

Utilizing this representation, we train a model that (i) outputs an indicator of how likely a scheduled transaction is fraudulent, (ii) aggregates the overall provided decision functions to derive the unified hypothesis, while assigning more importance to

learners that showcase the capability to better distinguish between fraud and non-fraud, and (iii) deals with a large imbalance in class representation.

### Validation results

The training, test, and validation sets consisted of data collected from July to October 2017, as dictated by the availability of online session log files. Around 900'000 sessions formed the dataset.

Raw data were processed using ETL to derive customer-specific feature representations of each recorded online session. The data were then split into non-fraud and fraud sets. The fraud set was not utilized to train the unsupervised base learners. Non-fraudulent ("healthy") sessions were separated into training and test sets utilizing a 3-fold cross-validation split. We then sequentially trained the models on each derived training fold and computed the scores for observations in the corresponding test folds. Following, we obtained an out-of-sample decision function value for each healthy session and each base learner. We then assigned base learner scores to each fraudulent session utilizing base learners trained on all healthy data.

The out-of-sample logistic regression decision function values were aggregated by averaging within their respective ensembles (LOFs, IFs, and BDTs). This step yields a 3-dimensional representation of each customer's online session. Finally, we utilized leave-one-out cross-validation to report the ROC curve measures. Hence, the logistic regression model is consecutively trained on an all-but-one observation, followed by computing the probability of an observation that was left out. Thus, we again obtain an out-of-sample fraud probability for each observation in the sample. We opted for leave-one-out cross-validation to maximize the number of fraudulent observations in each training set, because these are particularly scarce. Once we have obtained the aforementioned out-of-sample probabilities for each observation, we construct an ROC curve to display the FPR and TPR relationship depending on the decision threshold.

Resultantly, when utilizing no transaction data, the detection rate of the machine learning model was in a realistic range of 18% true positives. These primary results can be easily optimized to increase the TPR and simultaneously reduce the FPR utilizing different measures. This led to an increase in true positives by up to 45%, see Table 1.

Overall, LOF seems to perform best over the entire dataset compared to IF and BDT. However, BDT has a slightly steeper ROC curve at the beginning, thus showing better pure outlier detection capabilities. Furthermore, because BDT seems to detect frauds, as discussed below, involving larger amounts than those detected by LOF, we cannot conclude that LOF outperforms the other approaches considered. Aggregating the decision functions of the ensembles utilizing simple means outperformed supervised aggregation. Through analysis of logistic regression weights assigned to each ensemble of learners, we determined that significantly higher weights were assigned to the LOF ensemble, most likely due to its best performance over the whole dataset. This dampened the input from the other two ensembles. However, this is not the case when the mean for aggregation is utilized. The results were affected by the small number of frauds and the size of the sample analyzed.

Different ensembles detected different types of fraud, and by observing figures depicting money saved per raised alarm, we see that different ensembles (LOF and

**Table 1** The zero percentage TPR in the benchmark model is due to the fact that the static rule based model of the bank was not able to identify a single true positive case

Model	TPR	FPR
Benchmark model	0%	1% and 2%
500 LOF	18%	1%
500 LOF	45%	2%
500 BDT	18%	1%
500 BDT	27%	2%
500 IF	0%	1%
500 IF	27%	2%
Mean decision function	36%	1%
Mean decision function	36%	2%
Supervised decision function	18%	1%
Supervised decision function	18%	2%

The results of the mean aggregation approach derived from an aggregation of three standalone ensembles consisting of 500 LOF, 500 BDT and 500 IF base learners, where aggregation was performed by taking the mean value of decision functions from three different types of base learners

**Table 2** Comparing the performance of different anomaly fraud detection models

Paper	TPR	FPR	Remark
Wei et al. (2013)	49%-60%	NA	-
Carminati et al. (2015)	98.26%	0.19%	Scenario I+II
Carminati et al. (2015)	70%	14%	Scenario III
Vanini et al. (2022)	45%	1%	-
Vanini et al. (2022)	62%	3.5%	-

TPR means True Positive Rate and FPR the False Positive Rate

BDT) detect different types of fraud cases, displayed by a large difference in saved money per trigger. Logistic regression supervision alarms were affected mainly by LOF, thus making it miss large embezzlement detected by the BDT ensemble. This motivates the triage model described in the next section. As of the restriction of the FPR to no greater than 2%, the entire ROC curve is of less interest. The ROC AUC values for the LOF ensemble is 0.93, for the BDT ensemble 0.82, and for the mean decision function ensemble 0.91.

We compared our results with those of Wei et al. (2013), and Carminati et al. (2015). These are two of the few studies dealing with online fraud detection that use real-world data, at least in part. Wei et al. (2013) utilized an unsupervised approach, whereas Carminati et al. (2015) utilized a semi-supervised approach. Table 2 compares the performance of our model with those of Wei et al. (2013) and Carminati et al. (2015). The results of this table should be interpreted with caution. First, different payment channels were considered. Second, the data of Carminati et al. (2015) were anonymized and did not include fraud cases. These are artificially added to tune 1% of the data volume, compared to 0.018% in Wei et al. (2013) and 0.0012% in our dataset. Third, Wei et al. (2013) did not report the FPR. Finally, Carminati et al. (2015) published an almost perfect error detection for scenarios I + II, but in scenario III,

the false positives are too high; they generate too much manual work for the bank. The former scenarios are simple fraud scenarios that would be blacklisted and filtered out in our data before machine learning engages.

### Fraud detection triage model

#### Formalization

We formalize the triage model and denote by  $\Omega$  the set of all transactions with  $\omega$  as a single transaction,  $T(\omega)$  as the transaction amount function, and  $\chi_F(\omega)$  as the fraud indicator function, where  $\chi_F(\omega) = 1$  represents fraud. Space  $\Omega$  is attributed to a probability distribution  $P$  with  $p(x)$  as the density function of the transaction amounts. The threshold value  $L$  of the fraud score function  $S$  is a function of the transaction amount  $x$ . We define:

$$\begin{aligned} \gamma(x) &:= \frac{1}{E(1 - \chi_F)} E((S > L(x))(1 - \chi_F) | T = x), \\ \phi(x) &:= \frac{1}{E(\chi_F)} E(S > L(x)) \chi_F | T = x). \end{aligned} \tag{9}$$

If we assume stochastic independence of the transaction amount  $T$ , score  $S$  and fraud indicator  $\chi_F$ , we obtain the following interpretation:

$$\gamma(x) = \text{FPR}(L(x)) \quad \phi(x) = \text{TPR}(L(x)). \tag{10}$$

Note that the assumptions of independence are strong, as transaction sizes are utilized as the input of the machine-learning model underlying the score. Conversely, the independence of  $\chi_f$  and  $T$  implies that the transaction amounts of fraudulent transactions have the same distribution as those of non-fraudulent transactions. In the context of the value-at-risk model in the next section, we argue that there is little evidence to support this. This considered, the assumption of independence is theoretically difficult to uphold, but in practice quite necessary to obtain our results.

We formulate our optimization problem as follows:

$$\max_L E(T (S > L(T)) \chi_F) \tag{11}$$

under the constraint of the *integrated FPR*

$$\int_0^\infty \text{FPR}(L(x)) p(x) dx \leq \text{constant}. \tag{12}$$

The expectation in (11) is the average cumulated sum of fraudulent transaction amounts detected by the detection model. By letting  $q_0 := E(\chi_F)$  and utilizing (10), we can rewrite it as

$$\max_L \int_0^\infty x q_0 \text{TPR}(L(x)) p(x) dx. \tag{13}$$

The constant  $q_0$  is irrelevant to the optimization. Setting  $g(x) = \text{FPR}(L(x))$ , we reformulate the optimization problem in terms of the ROC curve as

**Table 3** Moments and quantiles of the fitted lognormal distributions in CHF for the online and mobile banking channel transactions

	Mean	Std. Dev.	90%	99%	99.9%
Online banking	2'355	11'290	4'722	30'304	118'663
Mobile banking	774	2499	1'699	8'565	27'954

**Table 4** Support points for the cubic spline construction of the ROC curve

FPR	0	0.002	0.004	0.008	0.01	0.02	0.03	0.05	0.1	1
TPR	0	0.1	0.18	0.28	0.31	0.4	0.45	0.5	0.55	1

$$\max_g \int_0^\infty x \text{ROC}(g(x))p(x) dx \tag{14}$$

under the constraint:

$$\int_0^\infty g(x)p(x) dx \leq \text{constant}. \tag{15}$$

To account for the recovery, we introduce a recovery function  $\theta : \Omega \rightarrow [0, 1]$ . This function changes the objective function in the optimization problem, as follows:

$$E[(1 - \theta)T(S > L(T))\chi_F] \tag{16}$$

whereas this constraint does not change.

**Optimization**

To put our formal model into practice, we need to fix a distribution for transaction amounts. Utilizing approximately 12 million transactions from online banking and 1.2 million from mobile banking, we approximated the distribution for both channels utilizing lognormal distributions. Although this choice does not particularly focus on the distribution's tails, it will be seen that the optimal model still places strong emphasis on the detection of anomalies with large transaction amounts. Some basic statistics of the fitted distributions are given in Table 3.

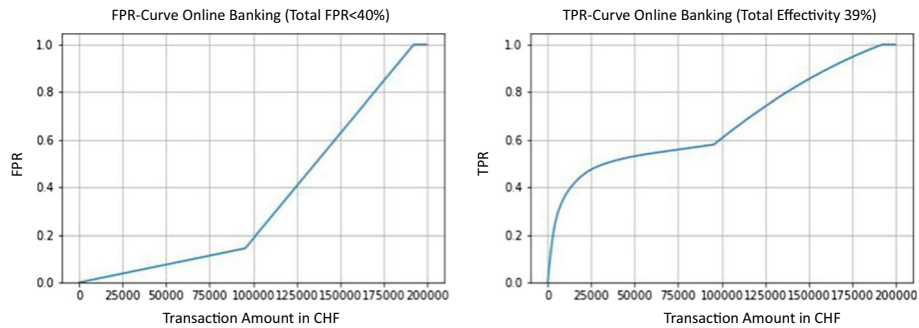
The ROC curve is conceptually the output of the detection model described in the previous section. However, owing to the limited number of actual fraud cases available, the TPR values for the given FPR levels are tainted with considerable uncertainty. The ROC curve utilized in our optimization was obtained by fitting a cubic spline function to the base points, as presented in Table 4. The support points were adjusted to avoid unwanted spikes in the cubic interpolation.

As the triage model aims to prevent large losses with a higher probability than smaller ones, the optimal FPR will be an increasing function of the transaction size. To avoid possible optimization problems, we choose a simple form for FPR as a function of transaction size, namely, a piecewise linear function satisfying  $g(0) = 0, g(T_1) = a, g(T_2) = 1$  for the parameters  $a > 0$  and  $0 < T_1 < T_2$  (see Fig. 1).



**Table 5** Optimized values

	$T_1$	$T_2$	$a_1$
Online banking	95'523	192'350	0.1437
Mobile banking	31'629	74'556	0.1568



**Fig. 1** Panel Left: False positive rate as a function of the transaction amount under the constraint that the total false positive rate is smaller than 0.4 percent. Right Panel: True positive rate as a function of the transaction amount. The total effectiveness is 39 percent

The optimization problem can be simplified by assuming equality in (12) and solving  $a$  as a function of  $T_1$  and  $T_2$ . For a target integrated FPR of 0.4%, we obtained the solutions listed in Table 5.

Figure 1 illustrates the results for an online banking channel. The concave shape of the FPR curve up to  $T_2$  shows that the optimal solution emphasizes the detection of large transaction fraud cases, accepting, in turn, the less rigorous testing of small and moderate transactions up to  $T_1$ . For transaction amounts larger than  $T_2$ , FPR and TPR are equal to 1 by construction. Hence, all such transactions are automatically flagged as anomalies.

*Total Effectiveness*

$$TE = \frac{1}{\int_0^\infty xp(x)dx} \int_0^\infty ROC(g(x))p(x) dx \tag{17}$$

is the average percentage of integrated fraudulent transaction amounts detected as anomalies. In our optimized case, the rate was 39%.

**Risk model**

**The model**

Compound Poisson processes were utilized as basic building blocks. We utilize beta marginal distributions for modelling bounded transaction amounts and generalized Pareto marginal distributions (GPD) for unbounded ones. The so-called mass-attack model is formulated as a nested compound Poisson process with a marginal beta distribution. All subprocesses are aggregated independently. Loss statistics, such as value-at-risk or other quantiles of the distribution, are obtained by running Monte Carlo simulations.

Utilizing the limited available fraud data and drawing on discussions with practitioners, we develop the following model for online banking fraud:

- Isolated attacks with a moderate transaction size of up to CHF 70'000 are modelled by a compound Poisson process with beta marginal distribution.
- Isolated attacks with transaction amounts larger than CHF 70'000 are modelled by a compound Poisson process with GPD marginal.
- “Mass attacks” are modelled as a nested compound Poisson process, where the inner Poisson process simulates the individual transactions triggered by the mass attack. The inner process has a beta marginal distribution and generates transaction amounts up to CHF 20'000.

The intensities of the Poisson processes constituting the submodels vary. In our case, isolated attacks of moderate size were by far the most frequent, followed by isolated attacks of large size. Mass attacks were the least frequent.

Mobile banking fraud is modelled analogously, albeit with transaction sizes only up to CHF 20'000, because larger amounts were inadmissible on this channel during our investigation. Hence, there is no Poisson process with GPD marginal in this case. Contrastingly, in the EBICS channel, which is an internet-based payment channel between banks, only the possibility of large fraudulent transactions was of interest. Hence, this model consists of a single compound Poisson process with GPD marginals above CHF 100'000. The details of the parametrization are given in Appendix A.

Countermeasures against fraud and recovery measures after fraud events play an essential role in determining risk potential. Therefore, they were integrated into the risk models. Countermeasures against online fraud fall into two categories: those that strengthen general infrastructure of the payment process to make it harder for attackers to find a weak spot, and those that are geared towards fighting off actual attacks. The first type is conceptually part of the base model described above, as it affects the frequency and possibly the transaction size of attacks. However, the second type is better understood in the context of recovery.

A recovery variable is introduced in the triage model, which accounts for it often being possible to recover money even after it has been transferred to another bank through fraudulent transactions. Conversely, by monitoring transactions utilizing the fraud detection and triage model, a certain percentage of attacks can be identified even before the transactions are released. The ROC curve of the detection model's ROC curve, in combination with the triage model, allows us to infer the probability of detection from the transaction size: such that this component of the recovery process is readily integrated into the stochastic framework.

Owing to the nonlinearity of the risk statistics, the aggregation of the models was performed at the level of individual scenarios. Thus, for each scenario,  $s_i$ , the loss of the overall model for one-year was calculated from the simulated loss events of the channel models:

$$\begin{aligned}
 \text{Loss}(s_i) &= \sum_{\text{Channels } C_j} \text{Loss}(C_j, s_i) \\
 &= \sum_{\text{Channels } C_j} \sum_{\text{Sub-model } U_k} \text{Loss}(C_j, U_k, s_i).
 \end{aligned}
 \tag{18}$$

**Table 6** Simulated moments and quantiles for online banking based on 100'000 Monte Carlo simulations

in CHF 1'000	Mean	Std. Dev.	90%	99%	99.9%
Total no detection	2'023	1'056	3'443	5'083	6'505
Total FPR 0.4 Pct	1'659	892	2'865	4'258	5'524
Total FPR Opt 0.4 Pct	800	535	1'518	2'403	3'114
Beta no detection	358	89	475	588	679
Beta FPR 0.4 Pct	293	81	400	504	585
Beta FPR Opt 0.4 Pct	206	65	291	376	442
GPD no detection	773	537	1'455	2'512	4'000
GPD FPR 0.4 Pct	634	484	1'251	2'248	3'450
GPD FPR Opt 0.4 Pct	81	91	207	361	489
Mass attack no detection	892	903	2'029	3'635	4'826
Mass attack FPR 0.4 Pct	732	743	1'690	2'983	3'920
Mass attack FPR Opt 0.4 Pct	514	523	1'199	2'100	2'799

Total represents the risk figures for all three distributions generalised Pareto (GPD), Beta and mass attack

For each sub-model, the loss is calculated by pulling the event frequency for the year according to the Poisson intensity, loss magnitude according to the marginal distribution, and stochastic recovery:

$$\text{Loss}(C_j, U_k, s_i) = \sum_{k=1}^{n_{\text{Poisson}}} \text{Loss}(\text{Margin Distribution}, s_i, k)(1 - \text{Rec}(s_i, k))$$

where Rec denotes recovery function. Simulated loss figures were obtained by simulating the nested overall model, from which the risk statistics could be calculated empirically. Juniper Research (2020) estimated the recovery rate as 18%.

**Results**

The simulation results for online banking are presented in Table 6. The table shows the simulation results without applying fraud detection utilizing a constant FPR level of 0.4% and the triage model for an integrated FPR of 0.4%, respectively. In this simulation, no additional recovery was applied.

The above table shows the strong mitigation of risk due to fraud detection. The triage model performs better than the constant FPR benchmark in all submodels, particularly for the GPD submodel. Recall that the triage model places strong emphasis on detecting large fraudulent transactions, even flagging all transactions larger than CHF 192'000.

As a second application, we compare the results of this risk model for the three e-channels with the bank's overall 2019 risk policy. This means that we compare the capital-at-risk (CaR) limits for market and credit risks with operational risk limits, where the e-channel part is now calculated in our model. The following allocation of CaR holds according to the annual report of the bank<sup>1</sup>: Credit Risk, 69%; operational risk,

<sup>1</sup> CaR for credit risk is VaR on the bank's quantile level and for market risk CaR was in the past chosen on an annual basis and a risk budgeting process was defined to align present risk with the annual risk budget.

11%; market risk trading, 4%; market risk treasury, 11%; market risk real estate, 2%; and investment, 4%.

Approximately 1% of operational risk capital can be attributed to these three channels. Even if we add another 4–5% of the total volume to all payment services, including corporate banking and interbank payments, less than 10% of the operational risk capital is attributed to payment systems. As payment systems account for a significant portion of operational risk, our results confirm serious doubts about the accuracy of the chosen operational risk capital in banks. Without reliable models and data, capital is determined by utilizing dubious business indicators. Our models, which represent a micro-foundation of risk, show that, at least in payment systems, trustworthy risk quantities can be derived by combining machine learning and statistics.

## Conclusion

Defense against sophisticated online banking fraud involve several resources and methods. These include risk models, algorithms, human action, knowledge, computer tools, web technology, and online business systems in the context of risk management.

We show that anomaly detection is not only useful per se, identifying a significant proportion of fraud while controlling false alarms, but that linking anomaly detection with statistical risk management methods can significantly reduce risk. A bank equipped with an anomaly detection system will be exposed to orders of magnitude of higher risks in payments than a bank implementing our end-to-end risk management framework with the three components of fraud detection, fraud detection optimization, and risk modelling.

As fraud is part of regulated operational risk, our model allows us to analytically capture these operational risks without crude benchmarking. This also provides a microeconomic foundation for capital adequacy. In the area of operational risk, these results put internal models that are not risk sensitive or difficult to verify on a solid footing.

A complicated problem, such as online payment fraud detection, requires a comprehensive understanding. A prerequisite for this is access to a large dataset. To evaluate our method, we utilized a real dataset from a private bank. Regardless of the chosen algorithm, feature extraction is an essential part of developing an effective fraud detection method. We utilized historically observed and confirmed fraudulent transaction identifiers as the ground truth. Each feature in the feature vectors for each e-banking session aims to encode deviations from normal customer behavior. Thus, behavioral, transactional, and customer-specific features are important.

Our framework opens interesting directions for future research. Roughly speaking, the framework goes in only one direction, from machine learning methods in fraud detection to statistical risk modelling. The feedback process from the risk model to the triage model and from the triage model back to the fraud detection model is a challenging task that can be addressed utilizing reinforcement-learning methods. With such a feedback loop, the entire risk-management framework becomes a learning system. Another research direction is to extend the optimization of fraud detection (trriage model) by considering transaction-dependent loss risks and other features such as customer segmentation. More emphasis is placed on segments that are known or suspected to be less

alert or more vulnerable to fraudulent attacks. This resulted in a higher-dimensional triage model.

## Appendix A

The structure of our ensemble results from several design decisions based on both insights from the field (online banking) and experience in building machine learning models. Our ensemble model was built according to the following guidelines:

**Customer-specific models** The features used in our approach encode patterns in customers' session behaviour and transactions. These patterns vary widely from client to client, which is why we chose to use a client-specific model rather than a global model.

**Global feature space** Although behaviours vary, we have chosen to use the same feature representation for each session/transaction, which allows us to assign weights to specific (model, feature) pairs based on their performance across all clients. This in turn allows for consistent scoring across all clients and information sharing between clients when fraudulent activity occurs. In other words, our approach makes it easy for learners to adjust their weights.

**Separation of models based on feature type** We have chosen to form separate ensembles, one based on behavioural features and one based on transactional features, rather than concatenating all features into a single vector and forming a single ensemble based on concatenation. This ensures better interpretability and reduces the likelihood of constructing nonsensical feature subspaces during feature bagging.

**Modified Bootstrap aggregation (Bagging)** To build an ensemble of weak learners, we use a modification of bootstrap aggregation (bagging). Bagging is a meta-algorithm for ensembles that is used to reduce the variance and improve the stability of the prediction as well as to avoid overfitting.

### Bagging Pipeline

*Observational sampling (bagging):* Bagged ensembles for classification generate additional data for training by resampling with replacement from the initial training data to produce multiple sets of the same size of initial training data, one for each base learner. This is done to reduce the prediction variance. For the two outlier detection ensembles, we used variable subsampling (without replacement) to avoid problems associated with repeated data and to mimic random selection of the neighbourhood count hyperparameter (cf. Aggarwal and Sathe 2017).

*Feature bagging:* An important task in outlier detection is to identify the appropriate features on which to base the analysis. However, these features may differ depending on the fraud mechanism. Therefore, instead of pre-selecting features, a more robust approach is to create an ensemble of models that focus on different feature sets and assign different weights to the models that use different features depending on their performance. The procedure is applied to each base learner  $b_j$  as follows:

- Randomly select a number  $r_{b_j}$  in range  $[d/10, d - 1]$ , where  $d$  denotes the feature dimension.
- Sample a subspace of features of size  $r_{b_j}$
- Train the base learner  $b_j$  on the sampled subspace.

*No Hyperparameter bagging:* Due to limited fraud, tuning the hyperparameters via validation may lead to overfitting. For this reason, similar to feature selection, we could instead randomly select a set of different hyperparameters. In our case, however, IF and BDT are not expected to be sensitive to the choice of hyperparameters, and resampling the hyperparameter from LOF would be redundant to the subsampling of the data performed. We therefore set all hyperparameters to reasonable ones found in the literature.

*Sharing bagged features and parameters across customers:* Subspaces for sampled features and parameters for local outlier factors are shared between all client models and all types of base models (manifested in the respective weak learners). This allows, for example, the introduction of supervision in the aggregation step and increased interpretability of the model, as it is easier to identify features relevant to the detection of certain types of fraud.

*Model aggregation:* Each base model provides a decision function  $\delta(x)$  for a given observation  $x$ . The base model ensemble directly aggregates (majority voting or averaging) the weak learner results based on different subsamples to form a single hypothesis that determines the class membership of an observation. Usually, this aggregation is directly extended after a normalisation step to include models of different types, parameters or feature groups. In our approach, however, this final aggregation is performed based on a monitoring step that uses knowledge of available frauds to assign different weights to each pair (model, feature set). Essentially, these weights quantify how appropriate each model and feature pair is for fraud detection.

## Appendix B

We refer to a composite Poisson process whose marginal distribution corresponds to a beta distribution as a beta model or as a GPD model if the marginal distribution corresponds to a generalised Pareto distribution. The mass attack model is a nested compound Poisson process. The outer Poisson process models the mass attack event, while an inner Poisson process models the number of affected transactions. The extent of damage of the individual affected transactions is modelled with a beta distribution.

- Online banking:
  - Beta model: Intensity 35,  $\alpha = 0.42$ ,  $\beta = 2.4$  and scale : 71.000. By shifting and scaling, explicitly by the transformation  $x \rightarrow \alpha + (\beta - \alpha)x$ , the beta distribution is shifted from  $[0, 1]$  to the interval  $[\alpha, \beta]$ . The parameter  $\alpha$  is called location, and  $\beta - \alpha$  scale.
  - GPD model: Intensity 3, shape = 0.25, location = 60.000, scale = 100.000.
  - Mass Attack model: Intensity 0.1, intensity nested model 1000, Beta model  $\alpha = 0.42$ ,  $\beta = 2.4$  and scale : 20.000.
- Mobile banking:
  - Beta model: Intensity 45,  $\alpha = 0.42$ ,  $\beta = 2.4$  and scale : 20'000.

- Mass Attack model: Intensity 0.1, intensity nested model 1000, Beta model  $\alpha = 0.42$ ,  $\beta = 2.4$  and scale : 20'000.
- Ebics:
  - GPD model:shape = 0.25, location = 100'000, scale = 300'000.
- Recovery model:
  - The recovery model models the percentage recovery in a fraud case. It has the following form:
    - With a probability  $p_1 = 65\%$ , a complete recovery is simulated, i.e. no damage remains. This resulted from the fact that in the 159 fraud cases considered, it was actually possible to reduce the loss amount to zero even in 80% of the cases.
    - With a probability  $p_2 = 18\%$ , a recovery of zero is simulated, i.e. the damage corresponds to the full amount of the offence.
    - With probability  $1 - p_1 - p_2$ , a recovery between 0 and 1 is simulated. A beta distribution is chosen as the distribution of these partial recoveries.

The beta distribution parameters for the online banking channel were fitted on the fraud cases recorded from 13/03/2013 to 13/03/2018. These are 159 fraud cases, of which both the initial fraud transaction amounts and the effective loss amount, i.e. the residual amount after recovery, were recorded. Of the 159 cases, 152 have a fraud amount between CHF 0 and 60,000, while the remaining 7 fraud amounts range between CHF 100,000 and 300,000. A beta distribution was fitted on the 152 cases with fraud amounts up to 60,000 CHF, whereby the scaling parameter, i.e. the upper limit of the distribution, was defined as a free parameter of the fitting procedure and estimated by it to be 71,000 CHF. Similar procedures apply to the marginal distribution fits of the GPD and mass attack models.

There exists significant statistical uncertainty and variability in the driving forces of the defined models. Putting the intended flexibility of the model structure into practice, we distinguish between 'easily accessible' parameters, which should be subject to discussion at any time in the context of risk assessments, and 'deeper' parameters, whose mode of action is less obvious and whose adjustment is subject to the process of model reviews. Roughly speaking, Poisson intensities, which determine the expected frequency of events, as well as upper and lower boundaries of the marginal distributions belong to the former category, while shape parameters for the Beta and GPD marginal distributions belong to the latter.

#### Abbreviations

CHF	Swiss Franc
\$	US Dollar
ROC	Receiver operating curve
AUC	Area under the ROC curve
TPR	True positive rate
FPR	False positive rate
BDT	Bagged decision tree
LOF	Local outlier factor
IF	Isolation forest
VaR	Value-at-risk

GPD	Generalized pareto distribution
EBICS	Payment channel for corporate banking clients
ETL	Extract, transform, load is a three-phase process where data is extracted, transformed and loaded into an output data container

### Acknowledgements

The authors are thankful to P. Senti, B. Zanella, A. Andreoli and R. Brun all from Zurich Cantonal Bank for the discussions and for providing us with the resources to perform his study. The authors are grateful to P. Embrechts (ETH Zurich) for the model discussions.

### Author contributions

Sebastiano Rossi and Ermin Zvizdic designed the fraud detection model and analysed the data. Thomas Domenig and Paolo Vanini designed the triage model. Thomas Domenig designed the risk model and did the calculations for the triage and risk model. Paolo Vanini wrote the manuscript with contributions from all authors. Paolo Vanini was involved in the analysis of the triage and risk model. Ermin Zvizdic led the project in the bank. All authors read and approved the final manuscript.

### Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

### Availability of data and materials

The bank provided real transaction data and data on transactions ("raw data") of the customers. The legal basis of the Swiss Federal Data Protection Act (2020) prevents the raw data from leaving the bank in any form or being accessible to any party other than the bank.

### Declarations

#### Competing interests

The authors declare that they have no competing interests.

Received: 22 March 2022 Accepted: 18 February 2023

Published online: 13 March 2023

### References

- Abdallah A, Maarof MA, Zainal A (2016) Fraud detection system: a survey. *J Netw Comput Appl* 68:90–113
- Ali A, Shukor AR, Siti HO, Abdu S (2022) Financial fraud detection based on machine learning: a systematic literature review. *Review Appl Sci* 12:9637
- Amiri M, Hekmat S (2021) Banking fraud: a customer-side overview of categories and frameworks of detection and prevention. *J Appl Intell Syst Inf Sci* 2(2):58–68
- Aggarwal CC, Sathe S (2017) *Outlier ensembles: an introduction*. Springer
- Bessis J (2011) *Risk management in banking*. Wiley, New York
- Bolton RJ, Hand DJ (2002) Statistical fraud detection: a review. *Stat Sci* 17(3):235–249
- Bolton RJ, Hand DJ (2001) Unsupervised profiling methods for fraud detection, *Credit Scoring and Credit Control VII*, pp 235–255
- Breunig MM, Kriegel H-P, Ng RT, Sander J (2000) LOF: Identifying density based local outliers. In: *Proceedings of the 2000 ACM SIGMOD international conference on management of data*, pp 93–104
- Carminati M, Caron R, Maggi F, Epifani I, Zanero S (2015) BankSealer: a decision support system for online banking fraud analysis and investigation. *Comput Secur* 53:175–186
- Chandola V, Banerjee A, Kumar V (2009) Anomaly detection: a survey. *ACM Comput Surveys* 41(3):1–58
- Embrechts P, Klüppelberg C, Mikosch T (2013) *Modelling extremal events: for insurance and finance (Vol 33)*. Springer Science & Business Media
- FCA (2021) *Financial conduct authority handbook*. [www.handbook.fca.org.uk](http://www.handbook.fca.org.uk)
- Fei Tony L, Kai T, Zhi-Hua Z (2008) Isolation forest. In: *2008 Eighth IEE E International Conference on Data Mining, IEEE*, pp. 413–422
- Haixiang G, Yijing L, Shang J, Mingyun G, Yuanyue H, Bing G (2017) Learning from class-imbalanced data: review of methods and applications. *Expert Syst Appl* 73:220–2
- Hilal W, Gadsden SA, Yawney J (2021) A review of anomaly detection techniques and applications in financial fraud. *Exp Syste Appl* 116429.
- Hilal W, Gadsden SA, Yawney J (2022) Financial fraud: a review of anomaly detection techniques and recent advances. *Expert Syst Appl* 193:11
- Jung E, Le Tilly M, Gehani A, Ge Y (2019, July) Data mining-based ethereum fraud detection. In: *2019 IEEE international conference on blockchain (Blockchain)* (pp 266–273). IEEE
- Juniper Research (2020) *Online payment fraud: Emerging threats, segment analysis and market forecasts 2020–2024*. [www.juniperresearch.com](http://www.juniperresearch.com)
- KPMG (2019) *Global banking fraud survey*, KPMG International
- Kou G, Xu Y, Peng Y, Shen F, Chen Y, Chang K, Kou S (2021) Bankruptcy prediction for SMEs using transactional data and two-stage multiobjective feature selection. *Decis Support Syst* 140:113429
- Krawczyk B (2016) Learning from imbalanced data: open challenges and future directions. *Prog Artif Intell* 5(4):221–232



- Li T, Kou G, Peng Y, Philip SY (2021) An integrated cluster detection, optimization, and interpretation approach for financial data. *IEEE Trans Cybern* 52(12):13848–13861
- Liu FT, Ting KM, Zhou ZH (2008) Isolation forest. In: 2008 eighth IEEE international conference on data mining, pp 413–422. IEEE
- Liu FT, Ting KM, Zhou ZH (2012) Isolation-based anomaly detection. *ACM Trans Knowl Discov Data (TKDD)* 6(1):1–39
- McNeil AJ, Frey R, Embrechts P (2015) *Quantitative risk management: concepts, techniques and tools-revised edition*. Princeton University Press, Princeton
- Montague DA (2010) *Essentials of online payment security and fraud prevention*, vol 54. Wiley, New York
- Molloy I, Chari S, Finkler U, Wiggerman M, Jonker C, Habeck T, Schaik RV (2016) Graph analytics for real-time scoring of cross-channel transactional fraud. In: *International conference on financial cryptography and data security*, pp 22–40. Springer, Berlin, Heidelberg
- Pang G, Shen C, Cao L, Hengel AVD (2020) Deep learning for anomaly detection: a review. *arXiv preprint arXiv:2007.02500*
- Piotr J, Niall AM, Hand JD, Whitrow C, David J (2008) Off the peg and bespoke classifiers for fraud detection. *Comput Stat Data Anal* 52:4521–4532
- Power M (2013) The apparatus of fraud risk. *Account Organ Soc* 38(6–7):525–543
- Sabu AI, Mare C, Safta IL (2021) A statistical model of fraud risk in financial statements. Case for Romania companies. *Risks* 9(6):116
- Shen H, Kurshan E (2020) Deep Q-network-based adaptive alert threshold selection policy for payment fraud systems in retail banking. *arXiv preprint arXiv:2010.11062*
- Singh A, Ranjan RK, Tiwari A (2022) Credit card fraud detection under extreme imbalanced data: a comparative study of data-level algorithms. *J Exper Theor Artif Intell* 34(4):571–598
- Tokovarov M, Karczmarek P (2022) A probabilistic generalization of isolation forest. *Inform Sci* 584:433–449
- Trozze A, Kamps J, Akartuna EA, Hetzel FJ, Kleinberg B, Davies T, Johnson SD (2022) Cryptocurrencies and future financial crime. *Crime Sci* 11(1):1–35
- Van Liebergen B (2017) Machine learning: a revolution in risk management and compliance? *J Financ Trans* 45:60–67
- Vanini P (2022) *Reinforcement Learning in Fraud Detection*, Preprint University of Basel
- Wei W, Li J, Cao L, Ou Y, Chen J (2013) Effective detection of sophisticated online banking fraud on extremely imbalanced data. *World Wide Web* 16(4):449–475
- West J, Bhattacharya M (2016) Intelligent financial fraud detection: a comprehensive review. *Comput Secur* 57:47–66
- Zhang W, Xie R, Wang Q, Yang Y, Li J (2022a) A novel approach for fraudulent reviewer detection based on weighted topic modelling and nearest neighbors with asymmetric Kullback-Leibler divergence. *Decis Support Syst* 157:113765
- Zhang G, Li Z, Huang J, Wu J, Zhou C, Yang J, Gao J (2022b) eFraudcom: An ecommerce fraud detection system via competitive graph neural networks. *ACM Trans Inform Syst (TOIS)* 40(3):1–29.
- Zhou ZH (2012) *Ensemble methods: foundations and algorithms*. CRC Press, Boca Raton

## Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

---

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)

---