**CREDIT CARD FRAUD DETECTION: A REALISTIC MODELLING AND A NOVEL LEARNING STRATEGY**

**ABSTRACT**

Detecting frauds in credit card transactions is perhaps one of the best test-beds for computational intelligence algorithms. In fact, this problem involves a number of relevant challenges, namely: concept drift (customers’ habits evolve and fraudsters change their strategies over time), class imbalance (genuine transactions far outnumber frauds), and verification latency (only a small set of transactions are timely checked by investigators). However, the vast majority of learning algorithms that have been proposed for fraud detection rely on assumptions that hardly hold in a real-world fraud-detection system (FDS). This lack of realism concerns two main aspects: 1) the way and timing with which supervised information is provided and 2) the measures used to assess fraud-detection performance. This paper has three major contributions. First, we propose, with the help of our industrial partner, a formalization of the fraud-detection problem that realistically describes the operating conditions of FDSs that everyday analyzes massive streams of credit card transactions. We also illustrate the most appropriate performance measures to be used for fraud-detection purposes. Second, we design and assess a novel learning strategy that effectively addresses class imbalance, concept drift, and verification latency. Third, in our experiments, we demonstrate the impact of class unbalance and concept drift in a real-world data stream containing more than 75 million transactions, authorized over a time window of three years.

**INTRODUCTION**

Credit card fraud detection is a relevant problem that draws the attention of machine-learning and computational intelligence communities, where a large number of automatic solutions have been proposed. In fact, this problem appears to be particularly challenging from a learning perspective, since it is characterized at the same time by class imbalance, namely, genuine transactions far outnumber frauds, and concept drift, namely, transactions might change their statistical properties over time. These, however, are not the only challenges characterizing learning problems in a real-world fraud-detection system (FDS).

In a real-world FDS, the massive stream of payment requests is quickly scanned by automatic tools that determine which transactions to authorize. Classifiers are typically employed to analyze all the authorized transactions and alert the most suspicious ones. Alerts are then inspected by professional investigators that contact the cardholders to determine the true nature (either genuine or fraudulent) of each alerted transaction. By doing this, investigators provide a feedback to the system in the form of labeled transactions, which can be used to train or update the classifier, in order to preserve (or eventually improve) the fraud-detection performance over time. The vast majority of transactions cannot be verified by investigators for obvious time and cost constraints. These transactions remain unlabeled until customers discover and report frauds, or until a sufficient amount of time has elapsed such that non-disputed transactions are considered genuine.

Thus, in practice, most of supervised samples are provided with a substantial delay, a problem known as verification latency. The only recent supervised information made available to update the classifier is provided through the alert– feedback interaction. Most papers in the literature ignore the verification latency as well as the alert–feedback interaction, and unrealistically assume that the label of each transaction is regularly made available to the FDS, e.g., on a daily basis. However, these aspects have to be considered when designing a real-world FDS, since verification latency is harmful when concept drift occurs, and the alert–feedback interaction is responsible of a sort of sample selection bias (SSB) that injects further differences between the distribution of training and test data.

**MODULES**

1. **Layers of Controls in an FDS:**
2. **Terminal:** The terminal represents the first control layer in an FDS and performs conventional security checks on all the payment requests [63]. Security checks include controlling the PIN code (possible only in case of cards provided with chip), the number of attempts, the card status (either active or blocked), the balance available, and the expenditure limit. In case of online transactions, these operations have to be performed in real time (response has to be provided in a few milliseconds), during which the terminal queries a server of the card issuing company.
3. **Transaction-Blocking Rules:** Transaction-blocking rules are if-then (-else) statements meant to block transaction requests that are clearly perceived as frauds. These rules use the few information available when the payment is requested, without analyzing historical records or cardholder profile. An example of blocking rule could be “IF internet transactions AND unsecured Web site THEN deny the transaction.”
4. **Scoring Rules:** Scoring rules are also expert-driven models that are expressed as if-then (-else) statements. However, these operate on feature vectors and assign a score to each authorized transaction: the larger the score, the more likely the transaction to be a fraud. Scoring rules are manually designed by investigators, which arbitrarily define their associated scores. An example of scoring rule can be “IF previous transaction in a different continent AND less than 1 h from the previous transaction THEN fraud score = 0.95.”
5. **Data Driven Model (DDM):** This layer is purely data driven and adopts a classifier or another statistical model to estimate the probability for each feature vector being a fraud.
6. **Investigators:** Investigators are professionals experienced in analyzing credit card transactions and are responsible of the expert-driven layers of the FDS. In particular, investigators design transaction-blocking and scoring rules. Investigators call cardholders and, after having verified, assign the label “genuine” or “fraudulent” to the alerted transaction, and return this information to the FDS. In the following, we refer to these labeled transactions as feedbacks and use the term alert–feedback interaction to describe this mechanism yielding supervised information in a real-world FDS.
7. **Features Augmentation:** Any transaction request is described by few variables such as the merchant ID, cardholder ID, purchase amount, date, and time. All transaction requests passing the blocking rules are entered in a database containing all recent authorized transactions, where the feature-augmentation process starts. During feature augmentation, a specific set of aggregated features associated with each authorized transactions is computed, to provide additional information about the purchase and better discriminate frauds from genuine transactions.
8. **Supervised Information:** Investigators’ feedbacks are the most recent supervised information made available to the FDS, but represent only a small fraction of the transactions processed every day. Additional labeled transactions are provided by cardholders that directly dispute unauthorized transactions. The timing of disputed transactions can vary substantially, since cardholders have different habits when checking the transcript of credit card sent by the bank. Moreover, checking disputed transactions entails some necessary administrative procedures that might introduce substantial delays.
9. **System Update:** Customers’ spending behavior evolves and fraudsters continuously design new attacks, and thus their strategies also change over time. It is then necessary to constantly update the FDS to guarantee satisfactory performance. Expert-driven systems are regularly updated by investigators who add ad hoc (transaction-blocking or scoring) rules to counteract the onset of new fraudulent activities and remove those rules liable of too many false alerts.

**EXISTING SYSTEM**

1. **Data-Driven Approaches in Credit Card Fraud Detection:** Both supervised and unsupervised methods have been proposed for credit card fraud-detection purposes. Unsupervised methods consist in outlier/ anomaly detection techniques that consider as a fraud any transaction that does not conform with the majority. Remarkably, an unsupervised DDM in an FDS can be directly configured from unlabeled transactions. A well-known method is peer group analysis, which clusters customers according to their profile and identifies frauds as transactions departing from the typical cardholder’s behavior. The typical cardholder’s behavior has also been modeled by means of self-organizing maps. Supervised methods are by far the most popular in fraud detection, and exploit labeled transactions for training a classifier. Frauds are detected by classifying feature vectors of the authorized transactions or possibly by analyzing the posterior of the classifier.
2. **Performance Measure for Fraud Detection:** The typical performance measure for fraud-detection problems is the AUC. AUC can be estimated by means of the Mann–Whitney statistic and its value can be interpreted as the probability that a classifier ranks frauds higher than genuine transactions. Another ranking measure frequently used in fraud detection is average precision, which corresponds to the area under the precision– recall curve. While these measures are widely used in detection problems, cost-based measures have been specifically designed for fraud-detection purposes. Cost-based measures quantify the monetary loss of a fraud by means of a cost matrix that associates a cost with each entry of the confusion matrix.
3. **Major Challenges to be Addressed in a Real-World FDS:**
4. **Class Imbalance:** Class distribution is extremely unbalanced in credit card transactions, since frauds are typically less than 1% of the overall transactions in our analysis. Learning under class imbalance has lately received a lot of attention, since traditional learning methods yield classifiers that are poorly performing on the minority class, which is definitively the class of interest in detection problems. Several techniques have been proposed to deal with class imbalance, and for a comprehensive overview, we refer the reader to. The two main approaches for dealing with class imbalance are: a) sampling methods and b) cost-based methods.
5. **Concept Drift:** There are two main factors introducing changes/evolutions in the stream of credit card transactions, which in the literature are typically referred to as concept drift. At first, genuine transactions evolve because cardholders typically change their spending behaviors over time (e.g., during holidays, they purchase more and differently from the rest of the year). Second, frauds change over time, since new fraudulent activities are perpetrated. In our experiments.
6. **Alert–Feedback Interaction and Sample Selection Bias:** The majority of classifiers used for credit card fraud detection in the literature are tested in experiments where transaction labels are supposed to be available the very next day since the transaction is authorized. In a real-world FDS, the only recent supervised information is the feedbacks Ft , provided by investigators, while the vast majority of transactions authorized everyday do not receive a label in a short time (|Ft| << |Tt|).

The interaction between the FDS (raising alerts) and the investigators (providing true labels) recalls the active learning scenario, where it is possible to select few—very informative—samples and query their labels to an oracle which in the FDS would be the investigators. However, this is not feasible in a real-world FDS, since investigators have to focus on the most suspicious transactions to detect the largest number of frauds. Requests to check (possibly genuine) transactions for obtaining informative samples would be ignored. Considering the limited number of transactions investigators can check, addressing these questions would necessarily imply that some high-risk transaction is not being controlled, with the consequent loss in detection performance.

**PROPOSED SYSTEM**

Detecting frauds in credit card transactions is perhaps one of the best test-beds for computational intelligence algorithms. In fact, this problem involves a number of relevant challenges, namely: concept drift (customers’ habits evolve and fraudsters change their strategies over time), class imbalance (genuine transactions far outnumber frauds), and verification latency (only a small set of transactions are timely checked by investigators). However, the vast majority of learning algorithms that have been proposed for fraud detection rely on assumptions that hardly hold in a real-world fraud-detection system (FDS). This lack of realism concerns two main aspects: 1) the way and timing with which supervised information is provided and 2) the measures used to assess fraud-detection performance. This paper has three major contributions. First, we propose, with the help of our industrial partner, a formalization of the fraud-detection problem that realistically describes the operating conditions of FDSs that everyday analyze massive streams of credit card transactions.

**CONCLUSION**

The majority of works addressing the fraud-detection problem in credit card transactions unrealistically assume that the class of each transaction is immediately provided for training the classifier. Here we analyze in detail the real-world working conditions of FDS and provide a formal description of the articulated classification problem involved. In particular, we have described the alert–feedback interaction, which are the mechanism providing recent supervised samples to train/update the classifier. We also claim that, in contrast to traditional performance measures used in the literature, in a real-world FDS, the precision of the reported alerts is probably the most meaningful one, since investigators can check only few alerts. Our experiments on two vast data sets of real-world transactions show that, in order to get precise alerts, it is mandatory to assign larger importance to feedbacks during the learning problem. Not surprisingly, feedbacks play a central role in the proposed learning strategy, which consists in separately training a classifier on feedbacks and a classifier on delayed supervised samples, and then aggregating their posteriors to identify alerts. Our experiments also show that solutions that lower the influence of feedbacks in the learning process (e.g., classifiers that mix feedbacks and delayed supervised samples or that implement instance weighting schemes) are often returning less precise alerts. Future work concerns the study of adaptive and possibly nonlinear aggregation methods for the classifiers trained on feedbacks and delayed supervised samples. We also expect to further increase the alert precision by implementing a learning to rank approach that would be specifically designed to replace the linear aggregation of the posterior probabilities. Finally, a very promising research direction concerns semi-supervised learning methods for exploiting in the learning process also few recent unlabeled transactions.

**BIBLIOGRAPHY**

[1] A. C. Bahnsen, D. Aouada, and B. Ottersten, “Example-dependent cost-sensitive decision trees,” Expert Syst. Appl., vol. 42, no. 19, pp. 6609–6619, 2015.

[2] C. Alippi, G. Boracchi, and M. Roveri, “A just-in-time adaptive classification system based on the intersection of confidence intervals rule,” Neural Netw., vol. 24, no. 8, pp. 791–800, 2011. [3] C. Alippi, G. Boracchi, and M. Roveri, “Hierarchical change-detection tests,” IEEE Trans. Neural Netw. Learn. Syst., vol. 28, no. 2, pp. 246–258, Feb. 2016. [4] C. Alippi, G. Boracchi, and M. Roveri, “Just-in-time classifiers for recurrent concepts,” IEEE Trans. Neural Netw. Learn. Syst., vol. 24, no. 4, pp. 620–634, Apr. 2013. [5] B. Baesens, V. Van Vlasselaer, and W. Verbeke, Fraud Analytics Using Descriptive, Predictive, and Social Network Techniques: A Guide to Data Science for Fraud Detection. Hoboken, NJ, USA: Wiley, 2015.