Web Service for Detecting Credit Card Fraud in Near Real-Time

Alexey Tselykh
Southern Federal University
44 Nekrasovsky Street
Taganrog, 347928, Russia
+7 8634 371743
tselykh@sfedu.ru

Dmitry Petukhov
Immanuel Kant Baltic Federal University,
Quantum Art
Moscow, Russia
dpetukhov@0xcode.in

ABSTRACT

This paper focuses on the design and implementation of a distributed, highly scalable, and fault-tolerant anti-fraud service accessible via REST API. Web service works in near real-time and employs machine learning algorithms for predictive analytics. Our goal is to develop an affordable anti-fraud service, which provides a possibility for participating parties (i.e. merchants, aggregating agents, payment systems, and banks) to reduce the risks of fraudulent payments over their sites. We explore a number of approaches resulting in a significant reduction of hardware and software costs as well as the size of the team working on the project.

Categories and Subject Descriptors
D.4.6 [**Operating Systems**]: Security and Protection—*information flow controls*; H.2.8 [**Information Systems**]: Database Applications—*Data mining*; H.3.5 [**Information Systems**]: Online Information Services—*Web-based services*

General Terms
Algorithms, Security, Languages

Keywords
Credit card fraud, machine learning, data mining, cloud technologies.

1. INTRODUCTION

The rapid growth in a quantity of transactions with credit cards made over the Internet poses new challenges for the developers of payment acceptance systems due to the increasing scale and complexity of frameworks required to provide system safety and security.

A quantity of fraudulent transactions and a variety of fraud types grows no less rapidly. Russia, along with the UK, France, Germany, and Spain, is in the top 5 European countries with the highest annual volume of credit card fraud. The total volume of losses caused by credit card fraud has exceeded 1 billion euro in 2013 in Europe and out of them 110 million euro is accounted for Russia.

SIN '15, September 08-10, 2015, Sochi, Russian Federation
© 2015 ACM. ISBN 978-1-4503-3453-2/15/09…$15.00
DOI: http://dx.doi.org/10.1145/2799979.2800039

The problem of fraudulent transactions over the Internet affects the whole chain from the customer to the card-issuing bank. This involves significant financial expenses and reputational risks for all the participating parties, except for cardholders. The fraud also has tangible negative consequences for e-commerce industry that prevents a wider spread of electronic payments.

An effective way to reduce the costs on the part of the merchant is to introduce additional verifications for the client as well as to delegate (partially or completely) responsibilities for verification to another participant, e.g. 3D-Secure technology. However, additional steps that require follow-up actions may result in dramatic reduction in the amount of successfully completed transactions (up to 25% on certain categories of goods).

Section 2 describes technical and non-technical requirements in order to reduce the production and owning costs for an anti-fraud system.

Section 3 describes the service software architecture, its modular structure and key implementation details.

2. REQUIREMENTS

2.1 System Challenges

An anti-fraud system is a business-critical system and its downtime can cause the business process interruption, or (when the system operates incorrectly) increase the risks of financial and reputational losses for the company that owns the system.

That is why there are some standard requirements for such systems: (a) fault-tolerance, (b) reliability, and (c) security of data storage and data transfer. In addition to the above requirements, the following specifications exist: (d) distribution, and (e) high scalability.

We toughen the requirements for the security of data storage and transfer yet more to comply with PCI DSS[1] standard and provisions of the Russian Federal Law No. 152-FZ "On Personal Data":

- not to store PAN and CVV (in no form);

[1] PCI DSS (Payment Card Industry Data Security Standard) – a list of requirements to ensure the storage and transmission safety for payment data.

- to store the rest of the data in a secure form;
- to transfer information between the software client and anti-fraud system only via secure channels of communication;
- to work with depersonalized data only.

2.2 Business Requirements

The main reasons for transaction refusal are payment data generated improperly or a fraudulent transaction.

2.2.1 Validity check for payment data introduction

Anti-fraud service verifies payment data coming from the software client. Regardless of whether it was user's error or malicious actions, early identification of errors in payment details can save CPU resources, and prevent noise masking of a learning model.

It is necessary to check whether the name of the cardholder contains at least two characters (dash and numbers in the name are acceptable) as well as whether the card is valid (the card has a validity period), and whether the card number passes Luhn algorithm[2] checks.

2.2.2 Transaction check for fraud

There are many heuristics to identify the fraud. At the same time, the number of heuristics exceeding 100 may lead to overfitting, incorrect identification of fraudulent transactions, and overall decrease in application performance.

The most effective heuristics included in the design model are:

- many-to-one relationships (multiple online credit card transactions made with different cards on a single IP address);
- one-to-many relationships (multiple IP and email addresses);
- the name of the card holder is not the same as the name of the internet-account owner forwarding the payment;
- the country of the cardholder does not coincide with the country of the Internet account owner forwarding the payment;
- payment made late at night (according to the local time of the client).

2.2.3 Global Filters

Allow to blacklist fraudulent purchasers by blocking specific credit card numbers, specific IP addresses, specific email domains, specific countries, cities, or regions. Global filters are either static or dynamic, they deal both with business rules (e.g. to reject payments from a particular country), and with abnormal activity detection (IP address).

3. DESIGN

Web service consists of several applications running on a cloud platform:

- Anti-Fraud API Service – REST-service providing API for interaction with Fraud Predictor ML service.
- Fraud Predictor ML – fraud detection service based on machine learning algorithms.
- Transaction Log – NoSQL transaction data storage.

In addition, the service includes numerous software clients which are Web applications or JavaScript widgets calling REST services of Anti-Fraud API Service.

3.1 Infrastructure

Web service is run on a publicly available Microsoft Azure cloud platform.

3.2 Architecture

Cloud-ready architecture relies on the following design patterns [5]:

- web/worker-nodes are stateless;
- horizontal partitioning (sharding) for structured and semi-structured data (Sharding Pattern);
- asynchronous network interactions using retry policies (Retry Pattern);
- message queues for load balancing and guaranteed task processing (Queue-Based Load Leveling Pattern).

To achieve near real-time performance, we use the following approaches:

- parallel data processing algorithms (i.e. MapReduce);
- Push'n'Forget paradigm to save unit records in the transaction log (one missing record out of 10K successful records will not essentially influence the precision of machine learning algorithms);
- no locking of transaction logs (by adding the timestamp field);
- killing slow queries.

3.3 Anti-Fraud API Service

For the merchant, the service is a REST service accessible via https. Anti-Fraud API Service operates in a cluster consisting of several stateless web roles (Azure web role is an application layer operating as a web application).

The sequence diagram is as follows:

Step 1. Requesting payment information.

Step 2. Model transformation (in terms of MVC).

Step 3. Sending request to a service predicting the result of a payment.

Step 4. Reporting the result, whether the payment is successful or not.

Step 5. Saving the data.

Step 6. Returning the result to the client.

Step 7, 8. Recalculation and update of the training sample, model retraining.

Step 9-12 (optional). The client initiates a request with information on the payment result (in the case when the prediction differs from the actual payment result transmitted in a request).

[2] Luhn algorithm – an algorithm for calculating the card number check digit. It is designed for detection of errors caused by unintentional data corruption. It confirms that the card number contains no errors (with some degree of confidence only.

3.4 Transaction Log

Both transactions and supplementary data (mainly statistics) are stored in the transaction log – long-term storage based on Azure Table. Transaction log consists of two tables:

- TransactionsInfo table with transaction facts: Transaction ID (Row Key), Merchant ID, card holder name hash (if available), amount and currency of payment;
- TransactionsStatistics table with the following statistical metrics: quantity of payment made with the card (several timeframes), number of IP addresses, time intervals between payments, how long the buyer is registered with the merchant, and how many payments were successful.

The model is retrained during steps 7 and 8. The training sample is a transaction log data. Retraining is made either on schedule, by the occurrence of a fixed value of new records in the transaction log or at the threshold level of incorrect predictions.

3.5 Fraud Predictor ML

Identification of various types of fraud is a typical supervised learning task. We use Azure Machine Learning cloud service for predictive analytics.

3.5.1 Obtaining data

Data set for fraud authorization model will be a transaction log.

3.5.2 Data preparation and examinations

We create Inner Join of loaded tables in TransactionId field with String, Integer and Timestamp data types, select a column with answers (labels) and columns with predictors (features) with Nominal and Absolute data types.

We replace null values with “undefined”. We also remove the lines where cardholder, payment amount and/or the currency fields have missing values as well as the lines containing knowingly incorrect data that is a noise for the model. The next step is to get rid of unused model fields: address, cardholder name hash, RowId, and PartitionId.

We make ZScore-normalization for data containing big numeric values such as payment amount (TransactionAmount column).

3.5.3 Data splitting

We split the resulting data set into training (70%) and test (30%) sample. We flag Randomized split option when dividing into data subsets in order to avoid “distortions” in the learning sample associated with massive credit card data leakage (and, as a consequence, the abnormal activity of fraud robots in this period).

3.5.4 Building and evaluating the model

We experiment with several classification algorithms and evaluate their performance in order to select which one is the best fit for our task.

We use several algorithms for two-class classification in this study: Two-Class Logistic Regression, Two-Class Boosted Decision Tree (gradient boostimg), Two-Class Support Vector Machine, Two-Class Neural Network.

Two-Class Boosted Decision Tree algorithm accepts the following parameters: number of trees to be built and minimum/maximum number of leaves on each tree; Two-Class Neural Network algorithm accepts the number of hidden nodes and learning iterations as well as initial weights.

4. EVALUATION

The cost of undetected (missed) fraudulent attempts (False Negative) is much higher than the cost of payments mistakenly taken for fraud (False Positive). At a Threshold level of 0.28, the value of Area Under Curve (AUC) metric is 0.955 (Accuracy = 0.880, Precision = 0.710, Recall = 0.873, and F1 Score = 0.783).

It some cases it is critical to explain a reason for the decision. Many analysts still prefer decision trees to neural networks as the former are easier to interpret.

After choosing the “right” model, we publish a scoring experiment in Machine Learning Studio as a web service. API documentation includes a description of the expected input formats and output messages as well as examples of service calls in C#, Python, and R.

5. RELATED WORK

Most of the papers contribute to an extensive and growing list of data mining techniques in credit card fraud detection. Those include but are not limited to neural networks [1, 3, 6], Bayesian learning [14, 17], support vector machines (SVMs) [12], AdaBoost ensemble learning [4], decision trees [20], a hidden Markov Model [10], evolutionary algorithms [7], and self-organized maps [2]. Although banks rely heavily on neural network based solutions [18], there is no evident preference for one technique over the overs. Classification is still, by far, the most common type of data mining tasks in credit card fraud studies [15].

[4] is one of the earliest papers on distributed data mining in credit card fraud detection. The authors put an early focus on scalable techniques to analyze massive amounts of transaction data and compute efficient fraud detectors in a timely manner. More recent papers [11, 19] tend to explore real-time fraud detection solutions. A typical present-day example of a corporate anti-fraud system, namely Yandex.Money, relies on real-time transaction analysis using BRE Web Rule (now Code Effects) system and ThreatMetrix cloud-based real-time device identification. [13] examines feasibility of MapReduce mechanisms in fraud detection, e.g. for the task of scalable and efficient outlier detection in large distributed data sets. Beymani software consists of a set of Hadoop- and Storm-based tools for sequence based outlier and anomaly detection that can be used for fraud detection. [8] put an emphasis on the fact that areas such as credit card fraud detection have been highly interested in tailored Big Data solutions using MapReduce and Hadoop.

An extensive list of open research problems is given in [16]. To compensate for the lack of labeled data, an author foresees a growing need for unsupervised techniques (profiling, fraud signature analysis, outlier detection, clustering, community detection, dependency analysis, etc.) as well as adapting techniques from semi-supervised learning, deep learning, transfer learning and co-training.

6. CONCLUSIONS AND FUTURE WORK

Eventually, we end up with a cloud-based anti-fraud service with an external REST API. Leveraging the power of predictive analytics our web service reduces the initial costs for infrastructure and software virtually to zero.

At this point, we develop a series of auxiliary tools and subsystems in order to provide a truly seamless solution. Machine learning part requires parameter fine-tuning to yield optimal classifier performance.

7. ACKNOWLEDGMENTS

The reported study was supported by RFBR, research project No. 15-01-09139 A.

8. REFERENCES

[1] Aleskerov, E., Freisleben, B., and Rao, B. CARDWATCH: A neural network based database mining system for credit card fraud detection. 1997. In *Proceedings of IEEE/IAFE Conference on Computational Intelligence for Financial Engineering* (New York City, NY, USA, March 23 – 25, 1997). CIFEr’97. IEEE, 220-226. DOI=http://dx.doi.org/10.1109/CIFER.1997.618940.

[2] Bansal, M. and Suman. Credit Card Fraud Detection Using Self Organised Map. 2014. *International Journal of Information & Computation Technology*. 4, 13 (2014), 1343-1348.

[3] Brause, R., Langsdorf, T., and Hepp, M. Neural data mining for credit card fraud detection. 1999. In *Proceedings of the 11th IEEE International Conference on Tools with Artificial Intelligence* (Chicago, IL, USA, November 09 – 11, 1999). IEEE, 103-106. DOI= http://dx.doi.org/10.1109/TAI.1999.809773.

[4] Chan, P.K., Fan, W., Prodromidis, A.L., and Stolfo, S.J. Distributed data mining in credit card fraud detection. 1999. *IEEE Intell. Syst. App.* 14, 6 (Nov./Dec. 1999), 67-74. DOI= http://dx.doi.org/10.1109/5254.809570.

[5] Dean, J. and Ghemawat, S. MapReduce: Simplified data processing on large clusters. 2008. *Commun. ACM*. 51, 1 (Jan. 2008), 107-113. DOI= http://dx.doi.org/10.1145/1327452.1327492.

[6] Dorronsoro, J.R., Ginel, F., Sánchez, C., and Santa Cruz, C. Neural fraud detection in credit card operations. 1997. *IEEE T. Neural. Networ.* 8, 4 (Jul. 1997), 827-834. DOI= http://dx.doi.org/10.1109/72.595879.

[7] Duman, E. and Özçelik, M.H. Detecting credit card fraud by genetic algorithm and scatter search. 2011. *Expert Syst. Appl.* 38, 10 (Sep. 2011), 13057-13063. DOI=http://dx.doi.org/ 10.1016/j.eswa.2011.04.110.

[8] Grolinger, K., Hayes, M., Higashino, W.A., L'Heureux, A., Allison, D.S., and Capretz, M.A.M. Challenges for MapReduce in Big Data. 2014. In *Proceedings of the 2014 IEEE World Congress on Services* (Anchorage, AK, USA, June 27 – July 02, 2014). SERVICES’14. IEEE, 182-189. DOI= http://dx.doi.org/10.1109/SERVICES.2014.41.

[9] Homer, A., Sharp, J., Brader, L., Narumoto, M., and Swanson, T. 2014. *Cloud Design Patterns: Prescriptive Architecture Guidance for Cloud Applications*. Microsoft patterns & practices.

[10] Iyer, D., Mohanpurkar, A., Janardhan, S., Rathod, D., and Sardeshmukh, A. Credit card fraud detection using hidden Markov model. 2011. In *Proceedings of the 2011 World Congress on Information and Communication Technologies* (Mumbai, India, December 11 – 14, 2011). WICT’11. IEEE, 1062-1066. DOI=http://dx.doi.org/10.1109/WICT.2011.6141395.

[11] Khan, A., Akhtar, N., and Qureshi, M. Real-Time Credit-Card Fraud Detection using Artificial Neural Network Tuned by Simulated Annealing Algorithm. 2014. In *Proceedings of International Conference on Recent Trends in Information, Telecommunication and Computing* (Chandigarh, India, March 21, 2014). ITC’14. ACEEE, 113-121. DOI=http://dx.doi.org/02.ITC.2014.5.65.

[12] Lu, Q. and Ju, C. Research on credit card fraud detection model based on class weighted support vector machine. 2011. *Journal of Convergence Information Technology*. 6, 1 (Jan. 2011), 62-68. DOI= http://dx.doi.org/10.4156/jcit.vol6.issue1.8.

[13] Mardani, S., Akbari, M.K., and Sharifian, S. Fraud detection in Process Aware Information systems using MapReduce. 2014. In *Proceedings of 2014 6th Conference on Information and Knowledge Technology* (Shahrood, Iran, May 27 – 29, 2014). IKT’14. IEEE, art. no. 7030339, 88-91. DOI= http://dx.doi.org/10.1109/IKT.2014.7030339.

[14] Mukhanov, L.E. Using bayesian belief networks for credit card fraud detection. 2008. In *Proceedings of the IASTED International Conference on Artificial Intelligence and Applications* (Innsbruck, Austria, February 11 – 13, 2008). AIA’08, 221-225.

[15] Ngai, E.W.T., Hu, Y., Wong, Y.H., Chen, Y., and Sun, X. The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature. 2011. *Decis. Support Syst.* 50, 3 (Feb. 2011), 559-569. DOI= http://dx.doi.org/10.1016/j.dss.2010.08.006.

[16] Palshikar, G.K. Detecting frauds and money laundering: A tutorial. 2014. *Lect. Notes Comput. Sc.* 8883, 145-160. DOI= http://dx.doi.org/10.1007/978-3-319-13820-6_12.

[17] Panigrahi, S., Kundu, A., Sural, S., and Majumdar, A.K. Credit card fraud detection: A fusion approach using Dempster-Shafer theory and Bayesian learning. 2009. *Information Fusion*. 10, 4 (Oct. 2009), 354-363. DOI= http://dx.doi.org/10.1016/j.inffus.2008.04.001.

[18] Pun, J. Improving Credit Card Fraud Detection using a MetaClassification Strategy. 2014. *International Journal of Computer Applications*. 56, 10 (Oct. 2012), 41-46. DOI= http://dx.doi.org/10.5120/8930-3007.

[19] Quah, J.T.S. and Sriganesh, M. Real-time credit card fraud detection using computational intelligence. 2008. *Expert Syst. Appl.* 35, 4 (Nov. 2008), 1721-1732. DOI= http://dx.doi.org/10.1016/j.eswa.2007.08.093.

[20] Sahin, Y. and Duman, E. Detecting Credit Card Fraud by Decision Trees and Support Vector Machines. 2011. In *Proceedings of the International Multiconference of Engineers and Computer Scientists* (Hong Kong, March 16 – 18, 2011). IMECS’11. DOI= http://dx.doi.org/ 10.1.1.421.359