Data Stream Mining & Processing

Proceedings of
IEEE Third International Conference on
Data Stream Mining & Processing

August 21-25, 2020
Lviv, Ukraine

IEEE

Ukrainian Catholic University

Manhattan College

DSMP

Proceedings of the

2020 IEEE Third International Conference on Data Stream Mining & Processing (DSMP)

Organized by

IEEE Ukraine Section

IEEE Ukraine Section (Kharkiv) SP/AP/C/EMC/COM Societies Joint Chapter

IEEE Ukraine Section (West) AP/ED/MTT/CPMT/SSC Societies Joint Chapter

IEEE Ukraine Section IM/CIS Societies Joint Chapter

Ukrainian Catholic University

Manhattan College

Kharkiv National University of Radio Electronics

Lviv, Ukraine
August 21-25, 2020

Additional copies may be ordered from:

IEEE Conference Operations
445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1331 USA

DSMP'2020 Organizing Committee
E-mail: dsmp.conference@gmail.com

IEEE Catalog Number: CFP20J13-ART

ISBN: 978-1-7281-3214-3

IEEE Third International Conference on Data Stream Mining & Processing
August 21-25, 2020, Lviv, Ukraine

Hybrid Machine Learning System for Solving Fraud Detection Tasks

Olena Vynokurova
GeoGuard
Kharkiv, Ukraine
vynokurova@gmail.com

Dmytro Peleshko
GeoGuard
Lviv\Kharkiv, Ukraine
dpeleshko@gmail.com

Oleksandr Bondarenko
GeoGuard
Kharkiv, Ukraine
alexander@geoguard.com

Vadim Ilyasov
GeoGuard
Kharkiv, Ukraine
vadim@geoguard.com

Vladislav Serzhantov
GeoGuard
Kharkiv, Ukraine
vladislav@geoguard.com

Marta Peleshko
Lviv State University of Life Safety
Lviv, Ukraine
marta.peleshko@gmail.com

***Abstract* — In parallel with technological development the problem of fraud detection is becoming more and more important. Increasing number of electronic transactions in various business environments, on the one hand, and software and technology development, on the other hand, lead to an active increase in electronic crime. In the paper the hybrid system of machine learning for solving tasks of anomalies detection has been proposed. This hybrid system consists of two subsystems – anomalies detection subsystem (based on unsupervised learning) and the interpretation subsystem of anomaly type (based on supervised system). The advantage of proposed hybrid system is the high-speed data processing when the data are fed in real time. The effectiveness of the proposed approach was confirmed during the solution of the detecting anomalies problem based on real data streams.**

Keywords— fraud detection, anomaly detection, hybrid system, isolation forest, random forest, transactions, machine learning

I. INTRODUCTION

In parallel with technological development the problem of fraud detection is becoming more and more important. Increasing number of electronic transactions in various business environments, on the one hand, and software and technology development, on the other hand, lead to an active increase in electronic crime. Authentication methods are no longer the only way to protect against fraud. Early detection of fraud is one of the main ways to prevent fraud. Blocking anomalous electronic transactions in some cases is almost the main way to avoid fraud. However, the development of mathematical methods for detecting fraud stimulates the skilled development of ways for concealing fraud. This leads to the fact that the practical algorithms of fraud detection are no longer universal.

In many cases, in order to increase the accuracy of early identification of anomalous electronic transactions, it is necessary to develop specialized software solutions. The essence of specialization is to use models that are adapted to the specifics of the company's business activities. For example, most of the available scientific papers on detecting fraud-related anomalies are related to credit cards.

This means that fraud detection methods that focus on the specific nature of input data that contain information from electronic transactions related to credit cards. And the resulting classification models are tightly tied to this business domain. This specialization is not quite a disadvantage, as it allows easy enough scaling of systems or expansion of types of detection when new types of fraud appear.

II. RELATED WORK

Using set of rules is one of first approach for developing fraud detection systems. This approach has been developed in the form of knowledge base system. The most well-known mechanisms for their implementation are expert systems [1].

Using a predefined set of rules simplifies the software development of fraud detection systems, but in general such systems have a number of disadvantages:

- the development of rules depends on the quality of the examination of the business environment. This determines the direct dependence of the effectiveness of the set of rules on the qualifications of expert analysts who create these rules [2].
- expanding the system of rules is costly. New experts are needed to expand the set of rules. Therefore, the appearance of new types of fraud leads to increased costs for modification of software systems.
- with a significant increase in the set of rules, the speed of the system can significantly decrease. This problem is getting more intense when large feature vectors are used.
- in the case, when the rules use a threshold, it is very difficult to achieve the adaptability of these values to environmental conditions.

Another defining characteristic of rule-based systems is the size of the rule base [1]. Small size databases occur primarily in cases where the input data vectors have a small dimension. Therefore, software solutions based on such databases are characterized by high speed. But the accuracy of these systems will again depend on experts.

In terms of support, small databases are much easier to administer. This is another advantage of small databases.

However, modern operational processes manipulate large-scale vectors. And this fact leads to a significant increase in the database and reduce the advantages of using rules for fraud detection tasks.

Other methods for solving a fraud detection tasks are statistical methods. The group of statistical methods includes methods that are based on elements of probability theory, mathematical statistics and data collected over a period of time.

Using statistical methods is one of the modern main directions of development of fraud detection methods. On the one hand, high accuracy of anomaly detection is obtained. On the other hand, the use of various inaccurate estimation parameters greatly reduces the flexibility of these methods and adaptability to changes in input data. For example, many methods require setting thresholds. Other methods require information on the statistical distribution of input data, etc. [3].

Today, all static methods of fraud detection can be divided into two categories: supervised and unsupervised methods. Both categories of these methods are united by the use of historical data (record of observations from the past) for effective fraud detection. The depth of this story for each method of different categories may be different. One of the main problems of supervised methods is the need to have sets of labeled features at the input.

This is not always possible and therefore contributes to the development of unsupervised methods. In the [4] authors have used a combination of unsupervised and supervised methods based on a self-organizing map and a neural network. Another example of a combination is hybrid methods from [5]. In [6] the classification of various hybrid methods is presented. This classification describes a variety of combined uses of unsupervised and supervised methods. In addition, a significant number of comparative experiments were conducted to assess the effectiveness of their use.

In point of view of the development and operation of software systems for fraud detection, machine learning methods have three main advantages:

1. An increase of the electronic transactions number usually leads to an increase in the accuracy of fraud detection models.

2. The dimensions of modern data sets make it impossible to analyze them without automation. Machine learning methods significantly simplify and increase the processing speed of large data sets.

3. The use of machine learning methods makes it possible to detect hidden dependencies. This is important to improve the accuracy of the systems and to increase the resistance of the system to the emergence of new types of fraud.

As the analysis of scientific results shows that among the most popular methods for Fraud Detection are Logistic regression, Random Forests and Support Vector Machines [7, 8]. It should be noted that [7] shows the efficiency of classification of anomalies using SVM. And in [8] the effectiveness of anomaly detection using Random Forests.

III. Anomaly Detection Hybrid Architecture for Fraud Detection Problems

For solving task of fraud detection the pipline for real-time anomality detection is proposed.

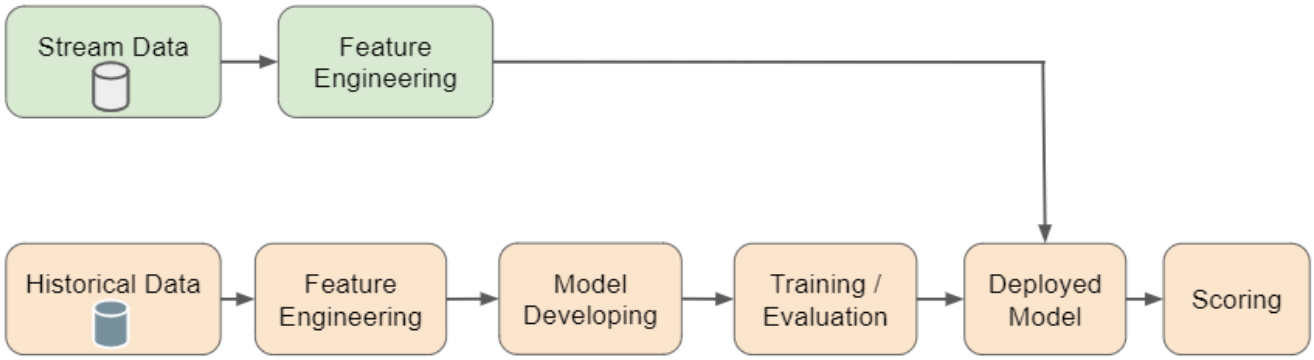

Fig. 1. Architecting an Anomaly Detection Pipeline

This pipeline consists of two flows: training/testing flow, which trains and tests the developed model and retrains it, evaluation flow that processing data stream from server in real time. The data are collected using no-sql DB Elasticsearch and each transaction is stored in json file. Depending on the solution type of transaction (Android, IOS, gdk (Windows and MAC), Solus (html), Plugin (Windows and MAC)), the different type of information about transaction can be stored in json file.

For training/testing flow we have collected the historical data for the dataset, which must be balanced for all type of transactions and their combining. After that, the data are fed to the stage of feature embedding, which is the most time-consuming, at this stage is cleaning, normalizing and embedding data. At this stage, the final training and testing dataset for the developed model is formed.

The next stage is the development of an anomaly detector model and a system for interpreting anomalies type. Then the model is trained and tested and after that we get a ready model for use on the evaluation flow.

The evaluation flow consists of capturing the data stream from the server and forwarding it to the feature engineering stage. Based on the developed embedding methods on the

training / testing flow, a dataset is obtained which is fed to the deployed model. After that the calculation of transaction scoring and making decision are performed.

Feature Engineering stage is the most time-consuming. It consists of the selecting field from database, the correlation analysis, the cleaning and preprocessing data and features embedding. Based on correlation analysis we select 64 fields for building dataset. Furthermore, we have to fill gaps in data. Since each transaction may have different filled fields in the connection with solution type. It is necessary to fill in all gaps based on machine learning methods or expert analysis of each field. It is important because quality of filling gaps affects accuracy of anomaly detection. And, also, we need to normalize and code numeric type data. The next stage of feature engineering is Categorical Data Embedding. Among the analyzed fields, 70% of fields are categorical variables. Different fields require different encoding, or the combination and encoding of several features together. We used: Label Encoding, One Hot Encoding, Embedding Vector, Binary Encoding, Hashing, Crosstab, Frequency Encoding and some modified methods.

The current version of the anomaly detector model is based on data from 64 fields of database. These fields describe the id of users and devices; the information about geolocation of the users and devices, which perform this transaction the history of transactions; the information about connection type (gsm, gps, wifi, ip); the information about running process and software on the devices that can be rooted; spoofed information about users and their location and etc.

Based on these fields and its combination 41 features is developed

$$x(k) = \{x_1(k), x_2(k), \ldots, x_n(k)\} \quad (1)$$

where $x(k)$ is transaction, $x_i(k)$, $i = 1 \ldots n$ are features (in our case n=41), k is real time.

This set of features forms the training and testing dataset.

On the case study stage, it was determined that the solution of the problem of fraud detection involves the need not only to determine anomalous transactions, but also to interpret why the transaction was failed. All existing methods of detection anomalies solve only the first problem. Thus, it is necessary to develop a hybrid model that could solve the problem of fraud detection and interpretation of the transaction anomaly type.

Proposed approach in the paper involves the use of two sequential models as entities to solve the problem of fraud detection. The first model is a binary classifier that solves the 'fraud' or 'non-fraud' problem. The next model is a multi-class classifier that defines the 'fraud' type. The general architecture is shown in fig. 2 and a more detailed architecture of the deployed model are shown in Fig. 3.

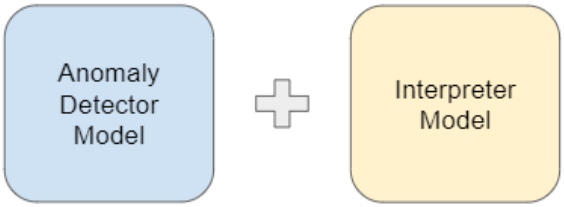

Fig. 2. General architecture of model

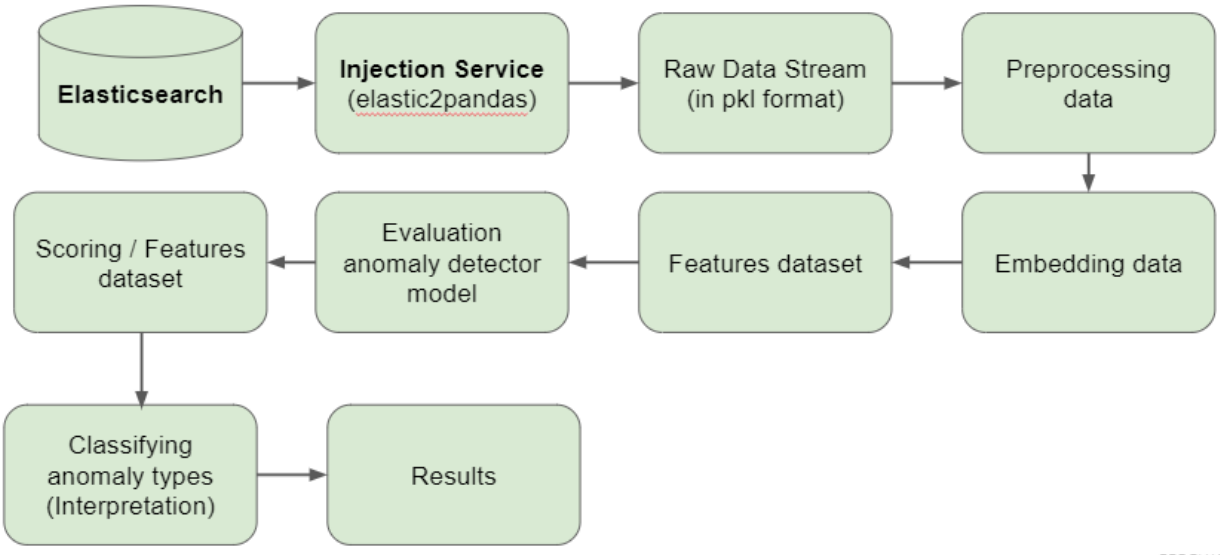

Fig. 3. Detailed architecture of deployed model

Nowadays there are a lot of methods for anomaly detection, the Robust Covariance method [9], One-class SVM method [10], Local Outlier Factor method [11], KNN method [12] and Isolation Forest [13] have been investigated, and for

developing model of anomaly detector we have selected Isolation Forest. Among the different anomaly detection algorithms, Isolation Forest is one with unique capabilities. It is a model free algorithm that is computationally efficient, can easily be adapted for use with parallel computing paradigms, and has been proven to be highly effective in detecting anomalies. In our case, it gave best accuracy among others.

Thus, the vector $x(k) = \{x_1(k), x_2(k), \ldots, x_n(k)\}$ from the constructed dataset is fed to generate an isolation tree. x is recursively separated by randomly selecting a feature and a random value of this feature between min(x_q) and max(x_q) the values of the selected feature and so on until the tree is constructed. Thus, we get an isolation tree which is a proper binary tree, where each node in the tree has exactly zero or two daughter nodes.

The task of detecting anomalies is to provide a rating of transactions that reflects the degree of their anomaly. Thus, one way to detect an anomalous transaction is to sort the transactions according to their length or anomaly scores; and anomalies are transactions that will be at the top of the list. The path length and anomaly estimate are determined by the algorithm proposed in [13].

In the case of Isolation Forest, anomaly score is defined as:

$$s(x,l) = 2^{-\frac{E(p(x))}{m(l)}} \tag{2}$$

where $p(x)$ is the path length of observation x, $m(l)$ is the average path length of unsuccessful search in a Binary Search Tree and l is the number of external nodes. More on the anomaly score and its components can be read in [13].

Each observation is given an anomaly score and the following decision can be made on its basis:

- a score close to 1 indicates anomalous transactions;
- score much smaller than 0.5 indicates normal transactions;
- if all scores are close to 0.5 then the entire sample does not seem to have clearly distinct anomalous transactions

The random forest method [14] was chosen to develop an interpreter model of the fraud detection hybrid system. The interpreter model provides an explanation to the provider-companies, why the transaction was determined as abnormal. Because the system under development may have situations where several types of anomalies can be present in a single transaction, the use of random forest-based methods is a priority.

For each decision tree, Scikit-learn calculates a nodes importance using Gini Importance, assuming only two child nodes (binary tree):

$$n_j^{im} = w_j U_j - w_j^r U_j^r - w_j^l U_j^l \tag{3}$$

where n_j^{im} is the importance of node j, w_j is weighted number of samples reaching node j, U_j is the impurity value of node j, $\bullet_j^r$ is child node from right split on node j, $\bullet_j^l$ is child node from left split on node j.

The importance for each feature on a decision tree is obtained in the form:

$$f_i^{im} = \frac{\sum_j n_j^{im}}{\sum_{k \in all\ nodes} n_k^{im}} \tag{4}$$

where f_i^{im} is the importance of feature i, n_j^{im} is the importance of node j.

After that these features importance are normalized using expression

$$\overline{f}_i^{im} = \frac{f_i^{im}}{\sum_{j \in all\ features} f_j^{im}}. \tag{5}$$

The final feature importance, at the Random Forest level, is its average over all the trees. The sum of the feature's importance value on each tree is calculated and divided by the total number of trees:

$$RF_i^{im} = \frac{\sum_{j \in all\ trees} \overline{f}_{ij}^{im}}{Tr} \tag{5}$$

where RF_i^{im} is the importance of feature i calculated from all trees in the Random Forest model, $\overline{f}_{ij}^{im}$ is the normalized feature importance for i in tree j, Tr is total number of trees.

Anomalous transactions and borderline transactions that have been detected by the anomaly detector model are fed to the interpreter model. Cross validation has used for tuning hyperparameters of interpreter model. Therefore, we have a cascade of classifiers which in general presents the proposed anomaly detector model.

IV. EXPERIMENTS

The proposed hybrid system was developed to solve the problem of detecting anomalies in the geolocation of users during transactions (GPS spoofing, Wi-Fi spoofing, location jumping, etc.). Experimental studies were conducted on the DB of GeoGuard company.

The specific feature of the system is that the decision cannot be made at the level of an anomaly, the type of anomaly must be explained to justify the decision.

The input vector consisted of 41 features based on information collected in the no-sql Elasticsearch database in json form for each transaction. Transaction information depends on the type of user's operating system (Android, IOS, MacOs, Windows) and consists of fields describing user geolocation when executing transaction from gsm, gps, wi-fi sources types and fields describing user's device. Frequency of transaction appearance in the environment is 800 transactions per 1 minute.

Training dataset consists of about 90 000 samples, number of trees was 300 trees for anomaly detection model and 500 trees for interpreter model.

Table I shows the results of detection and accuracy of interpretation of proposed hybrid machine learning system.

TABLE I. THE RESULTS OF ANOMALIES DETECTION AND THEIR INTERPRETATION

Solution	**Accuracy, %**		
	Training Detector	***Testing Detector***	***Classification & Interpreter***
all solution	91	90	90
ios	91	90	95
android	92	90	95
plugin (Windows+ MacOs)	99	96	99
gdk (Windows+ MacOs)	97	96	98

The results show that the anomaly detection on each individual solution is more accurate than when all solutions are combined into one dataset. The specificity of the system is the need to balance the dataset, in which all known anomalies on each operating system must be presented. It is also a feature of the system that multiple anomalies may be present in a transaction at the same time, which complicates the process of interpretation.

V. CONCLUSION

In the paper a hybrid system of machine learning for solving problems of anomaly detection is proposed. Such hybrid system consists of two subsystems - subsystem of anomaly detection and subsystem of anomaly type interpretation (classification), which are based on a cascade of decision trees with supervised and unsupervised learning. The advantage of the hybrid system is the speed of processing the data that are fed in real time. The effectiveness of the proposed approach has been confirmed in solving the practical problem of detecting anomalies in user geolocation during transactions execution (GPS spoofing, Wi-Fi spoofing, location jumping, etc.).

REFERENCES

[1] N.F. Ryman-Tubb, P. Krause, and W. Garn. “How Artificial Intelligence and machine learning research impacts payment card fraud detection: A survey and industry benchmark,” Engineering Applications of Artificial Intelligence, no. 76, pp. 130–157, 2018.

[2] R. P. Dazeley, “To The Knowledge Frontier and Beyond: A Hybrid System for Incremental Contextual- Learning and Prudence Analysis,” PhD thesis, University of Tasmania, 2006 http://https://eprints.utas.edu.au/8173

[3] A. Patcha, and J. M. Park, “An overview of anomaly detection techniques: Existing solutions and latest technological trends,” Computer Networks, vol. 51(12), pp. 3448–3470, 2007.

[4] J. T. Quah, and M. Sriganesh “Real-time credit card fraud detection using computational intelligence,” Expert Systems with Applications, vol. 35 (4), pp. 1721–1732, 2008.

[5] Y. Moreau, E. Lerouge, H. Verrelst, C. Stormann, P. Burge, and K. U. Leuven, “A hybrid system for fraud detection in mobile communications,” Neural Networks, pp. 447–454, April 1999.

[6] C. Phua, V. Lee, K. Smith, and R. Gayler, “A Comprehensive Survey of Data Mining-based Fraud Detection Research.” DOI: 10.1016/j.chb.2012.01.002. Arxiv:1009.6119., 2010.

[7] S. Bhattacharyya, S. Jha, K. Tharakunnel, and J. C. Westland. “Data mining for credit card fraud: A comparative study,” Decision Support Systems, vol. 50 (3), pp. 602–613, 2011.

[8] D. Meyer, F. Leisch, K. Hornik. “The support vector machine under test,” Neurocomputing, vol. 55 (12), pp. 169 –186, 2003.

[9] P. J. Rousseeuw and M. Hubert, “Anomaly Detection by Robust Statistics” arXiv:1707.09752v2 [stat.ML] 14 Oct 2017

[10] R. Zhang, Sh. Zhang, S. Muthuraman, and J. Jiang. “One Class Support Vector Machine for Anomaly Detection in the Communication Network Performance Data,” In: 5th WSEAS Int. Conference on Applied Electromagnetics, Wireless and Optical Communications, 2007.

[11] M. M. Breunig, H.-P. Kriegel, R. T. Ng, J. Sander, “LOF: Identifying Density-Based Local Outliers” in Proc. ACM SIGMOD 2000 Int. Conf. On Management of Data, Dalles, TX, 2000

[12] Y. Djenouri, A. Belhadi, J. C. Lin and A. Cano, "Adapted K-Nearest Neighbors for Detecting Anomalies on Spatio–Temporal Traffic Flow," in IEEE Access, vol. 7, pp. 10015-10027, 2019. doi: 10.1109/ACCESS.2019.2891933

[13] T.L. Fei, M. T. Kai, Z. Zhi-Hua, “Isolation Forest,” in Proceedings of the 2008 Eighth IEEE International Conference on Data Mining, pp. 413–422, December 2008, https://doi.org/10.1109/ICDM.2008.17

[14] L. Breima. “Random Forests. Machine Learning”, v. 45 (1), pp. 5–32, 2010, doi:10.1023/A:1010933404324