

Fraud Detection Technology in Payment Systems

Iulia Khlevna, Bohdan Koval

Taras Shevchenko National University of Kyiv, Volodymyrska str., 60, Kyiv, 01033, Ukraine

Abstract

The paper outlines the relevance of fraud detection from the standpoint of the integrity of the research. It proposes the solution - the development of technology to detect fraud in payment systems. The definition of such technology has been given. It determines that the approach to the architecture of fraud detection technology in payment systems is based on the discovery of non-typical behavior in the payment system and classifying anomaly transactions. It has been established that in terms of technology it is important to develop an effective and optimized model for the classification of fraud in payment systems from the standpoint of all stages of the research. It has been discovered that there is no efficient algorithm that would be the standard for all financial institutions in detecting fraud. The work proposes generalization and development of methods and algorithms. The outcome technology of fraud detection in payment systems has been compiled and researched based on a transaction database of 6.36 million records. The solution to the problem is implemented using machine learning methods, using Python programming language as a base for the software product. The research resulted in a technology based on a model with an accuracy of 99.97% and an AURPC of 99.86%. The prospects of further research - the implementation of the information system itself, which can be integrated into the software of financial institutions - has been outlined, based on the obtained data. The paper identifies the challenges and perspectives of the research.

Keywords ¹

data science, machine learning, deep learning, data visualization, binary classification

1. Introduction

Technology evolution is transforming financial markets. Autonomy of financial institutions' clients have entered everyday life due to the expansion of the set of tools for transactions on different levels. From the financial institutions' point of view, the consequence of such autonomy is high-speed and high-quality processing of operations, reduction of costs and risks associated with the storage and transportation of cash. Customers get the convenience and simplicity of real-time transaction procedures. However, there is a dynamic problem associated with maintaining the integrity and veracity of transactions, the problem of fraud. Financial institutions have financial losses, reduced customer loyalty and their loss in case of fraud within the customers' transactions. Thus, in Ukraine in 2016, losses from fraudulent transactions were about 330 million UAH, now, in 2019, - about 1 billion UAH, which indicates that the existing systems are not effective. Moreover, Ukraine is almost 50% ahead of the United States in the percentage of fraudulent transactions (on 10,000 transactions) and this gap is only growing, indicating that not only the industry as a whole needs new solutions, but they are especially needed in Ukraine. To prevent this, we need to transform approaches and means to monitor, detect and control illegal actions. No doubt, the best way to deal with fraud is to prevent it. Such warning is possible due to the development of a system based on the prediction of fraudulent actions at the level of suspicious transactions and forecasting the probability of its occurrence.


¹IT&I-2020 Information Technology and Interactions, December 02–03, 2020, KNU Taras Shevchenko, Kyiv, Ukraine

EMAIL yuliya.khlevna@gmail.com (Iulia Khlevna); bohkoval@gmail.com (Bohdan Koval)

ORCID: 0000-0002-1807-8450 ; (Iulia Khlevna); 0000-0002-3757-0221 (Bohdan Koval)

© 2020 Copyright for this paper by its authors.

Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

Peculiarities of using fraud detection models are presented in the works [1 – 5]. The main essence of Peculiarities of using fraud detection models are presented in the works [1 – 5]. The main essence of the presented works is the classification models of fraud transactions by different methods. It is appropriate to use this approach for fraud detection techniques that have been observed before. The work [6] forms the approach according to which a transaction is considered fraudulent if it differs from normal user behavior. This is based on the assumption that attackers will behave completely differently than the account owner. The approach to handling fraud risks with payment systems, which involves a combination of the presented approaches, is defined in the works [7, 8]. Taking into account the above, we must first develop and identify a model of credit card user behavior, and then detect fraud. Various methods and algorithms can be used to solve this problem [9]. However, there is no efficient and precise algorithm in the literature on credit card fraud that would be the standard for all financial institutions [10]. Each technique has its advantages and disadvantages. In addition, fraud approaches are dynamic and require constant reworking of their forecasting. Therefore, the study of fraud detection models, changes in their parameters, combination of algorithms to maintain each other's advantages and cover their weaknesses in detecting fraud with financial payment systems from the standpoint of systematics is of both scientific and practical interest.

2. Algorithm for constructing fraud classification technology and detection of anomalies in systems

Definition 1. Technology of fraud detection in payment systems (TFDPC) is a set of systematized ways to provide forecasting, detection and control of fraudulent transactions in financial systems. Models, methods and algorithms of machine learning underlie the core of such technology.

According to the literature [1 - 10] it has been established that the basis of fraud detection in payment systems is the task of classification. Therefore, it is proposed to base the development of an effective and optimized model of fraud classification in payment systems on the construction of TFDPC from the standpoint of all stages of the study. To implement the full cycle of TFDPC, we choose Python3 programming language because of its simple syntax, broad programming community support and a huge amount of available documentation.

2.1. Software environment

The technology will be based on Python programming language and its broad ecosystem. This choice is justified by the fact that Python allows you to quickly write code and test hypotheses, and is able to run on almost any device. In addition, the biggest advantage of Python is the presence of many libraries and frameworks, which significantly reduce development time.

It is expedient to choose such libraries for realization of the solution:

- pandas - data processing and analysis, the library provides data structures and operations for table and time series manipulation, it is optimized for working with large amounts of data;
- numpy - adds support for large, multidimensional arrays and matrices, along with a vast collection of high-level mathematical functions for operations with them;
- matplotlib - offers an object-oriented software interface for inserting various visualizations;
- scikit-learn - introduces various algorithms for solving problems of classification, regression analysis, clustering, including the method of support vectors, K-nearest neighbors, random forests and others;
- xgboost - framework to work with gradient boosting algorithm.

The use of these libraries will form a ready-to-use application, which will serve as a kind of foundation for the solution.

2.2. Input transactions data

The solution will be implemented based on the transaction data set of the anonymous payment system. It is worth noting that the technology is implemented in a flexible, scalable way, and therefore

this solution can be adapted to any data set that has a similar structure (similar transaction records). We will describe our data structure for a clearer understanding.

The reference structure for analysis is an object with rectangular data, therefore, in fact, it is a table. The table consists of 10 columns (attributes) and 6,362,620 records in total, so our dataset is a matrix of dimension 6362620 x 10. Also, there are no empty cells in the data set, all attributes of all records are filled. Each line is a record of an individual transaction. Columns are the transaction's attributes and include such:

- step - displays the unit of time in reality. The transactions in the dataset were completed within 743 hours. That is, "1" is the first hour of observation, "743" is the 743rd hour of observation;
- type - transaction type. Possible types: CASH-IN (cash replenishment), CASH-OUT (cash withdrawal), DEBIT (sending funds to the account), PAYMENT (payment for goods or services), TRANSFER (money transfer);
- amount - transaction amount;
- nameOrig - the client that initiated the transaction;
- oldBalanceOrig - the initial balance of the client before the transaction;
- newBalanceOrig - customer balance after the transaction;
- nameDest - ID (identifier) of the transaction recipient;
- oldBalanceDest - the initial balance of the recipient;
- newBalanceDest - recipient's balance after the transaction;
- isFraud - flag if the transaction is fraud (1) or non-fraud (0).

Thus, the analysis will be performed on a set of banking transactions that were performed by individuals on their own - for example, using mobile banking, card or terminal - so those operations that were not performed with the help of a bank or other supervisory authority. Therefore, this data set is especially useful today - because the level of money digitization is increasing, and the share of transactions made in cash or at the bank - is declining. That's why this set allows us to build modern solutions and identify patterns in banking transactions.

2.3. Exploratory data analysis and data preparation

Before we start creating a concept of the technology and build a model, let's focus on manual data research. This is an important step in building any model, because it will allow you to identify important factors, focus on them and ignore any "white noise". The data preparation is based on:

1. Determining which type of transaction is most often fraudulent (and adapting the dataframe accordingly) on Appendix 1.

As you can see, all fraudulent transactions are either transfers between accounts (TRANSFER) or cash withdrawals (CASH_OUT). In the next steps, it is effective to get rid of records of all other types of transactions - they will only interfere with the model, not allowing it to identify fraudulent transactions.

For clarity and optimization, let's put binary code on transaction type: "0" will correspond to the type TRANSFER, "1" - the type CASH_OUT (Appendix 2).

2. Substantiation of anomalies during the funds transfer.

It is clear that when making a transfer (amount! = 0) the initial recipient account (oldBalanceDest) can not be empty at the same time as the final account (newBalanceDest) - they must differ in the amount of the transaction on Appendix 3.

Part of anomaly transactions among fraudulent: 0.4955558261293072

Part of anomaly transactions among regular (non-fraudulent): 0.0006176245277308345

It was found that almost every 2nd anomalous transaction is fraudulent (which is not surprising), while among ordinary transactions such are about 0.62%. On the one hand, this is good news, because there is a clear indicator that the operation is fraudulent. However, to build a model, such a feature is rather harmful - the presence of a clear indicator aggressively classifies the data set, and therefore

fraudulent transactions that do not fall under this classification will be easily lost. Therefore, it is advisable to mark the appropriate balances in such records as "-1" instead of "0" - this will allow the model to be more smart, because in this way we will distinguish the following records from the rest on Appendix 4.

3. Analysis of anomalous transactions from the transaction initiative side.

From the transaction sender point of view (oldBalanceOrig - newBalanceOrig) such cases also occur, so we will similarly separate them, but we will mark such transactions differently so as not to confuse with abnormal transactions of the recipient (Appendix 5).

4. Quantitative detection of a certain anomaly in the transaction

Thus, it is clear that the difference between the initial balance and the final balance must differ by the amount of the transaction (which is negative for the sender and positive for the recipient). We will introduce 2 new columns and calculate the error in the balance of the recipient (errorBalanceDest) and the initiator (errorBalanceOrig) - how much the expected account balance (transaction amount) differs from the actual one:

2.4. Visualization of transaction data

The best way to confirm that a dataset contains enough information to build a model that can make the right predictions is to visualize the differences between fraudulent and ordinary transactions.

The plot (Figure 1) shows how regular and fraudulent transactions have different projections (fingerprints) of their distribution over the observation time. It is clear from the plot that fraudulent transactions are more evenly distributed over time than ordinary ones (fraudulent transactions are weakly correlated with time, can occur at any time, while ordinary transactions occur at approximately equal intervals). In addition, it has been determined that the number of cash withdrawal transactions (CASH_OUT) outweighs the number of transfers (TRANSFER) in normal transactions, while in fraudulent transactions - their number is approximately the same. This allows us to make the following assumption - the money transfer operation is more likely to be fraudulent.

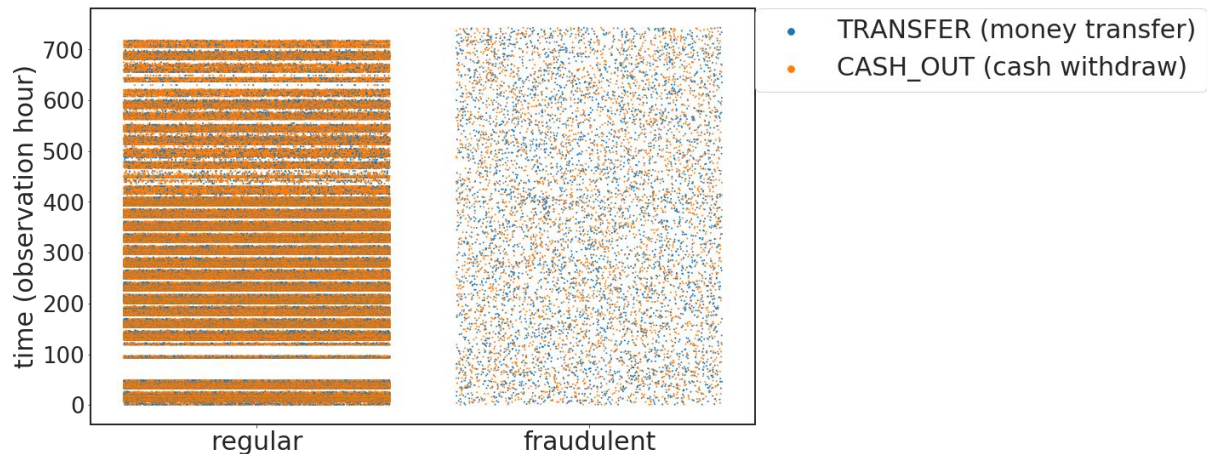


Figure 1: Stripped distribution in regular transactions vs. homogeneous distribution in fraudulent transactions (over time)

The plots (Figure 2 and Figure 3) below show that although fraud in a transaction can be detected based on the initial "amount" attribute, the newly introduced "errorBalanceDest" attribute (the difference in expected balance and actual one) is the best classifier. Thus, Figure 2 shows that regular transactions have a much larger number of transfers than fraudulent, which are also not limited in the amount of the transaction. Also, fraudulent transactions are balanced by type (money transfer or cash withdrawal) and also do not exceed the limit.

The distribution of transactions based on the "errorBalanceDest" attribute almost clearly distinguishes the difference between normal and fraudulent transactions (Figure 3) - as we can see, in the fraudulent transactions this error is usually negative. That is, in normal transactions, the amount of oldBalanceDest+amount is usually greater than the newBalanceDest, which is acceptable (for

example, part of the amount was spent on some commissions), and also, in fraudulent transactions, this amount is usually less than the new balance, which confirms the anomaly and identifies such transactions as fraudulent.

The 3D plot (Figure 4) best demonstrates the differences in the distribution of regular and fraudulent transactions. It is determined that the initial attribute "step" (transaction over time) weakly reflects the relationship of the transaction to a particular class. It is also worth noting the striped nature of the data distribution over time.

For the final visualization of patterns in the data set, we use a simple but reliable method of visualization - a heat map, which will show the correlation between attributes (Figure 5).

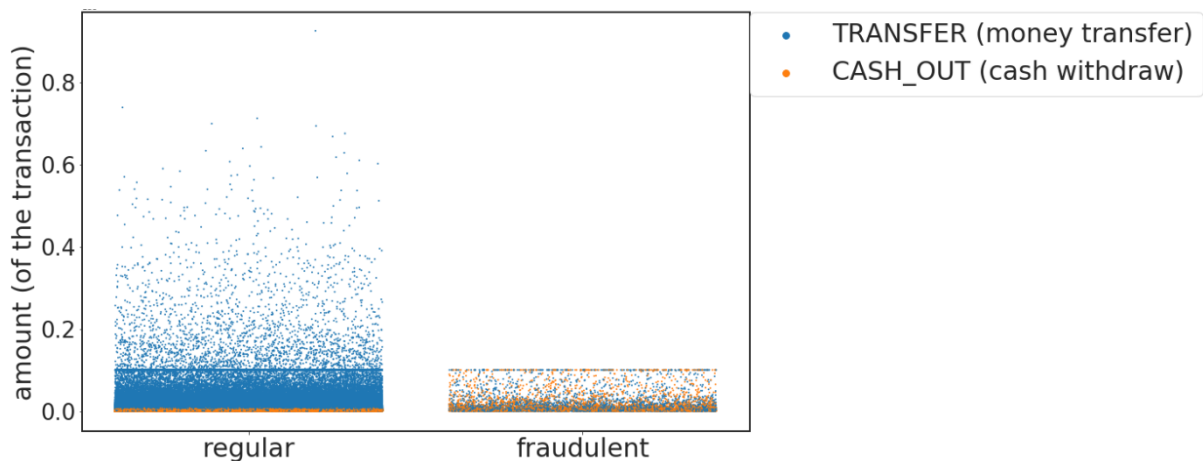


Figure 2: Transactions amount distribution among regular and fraudulent transactions

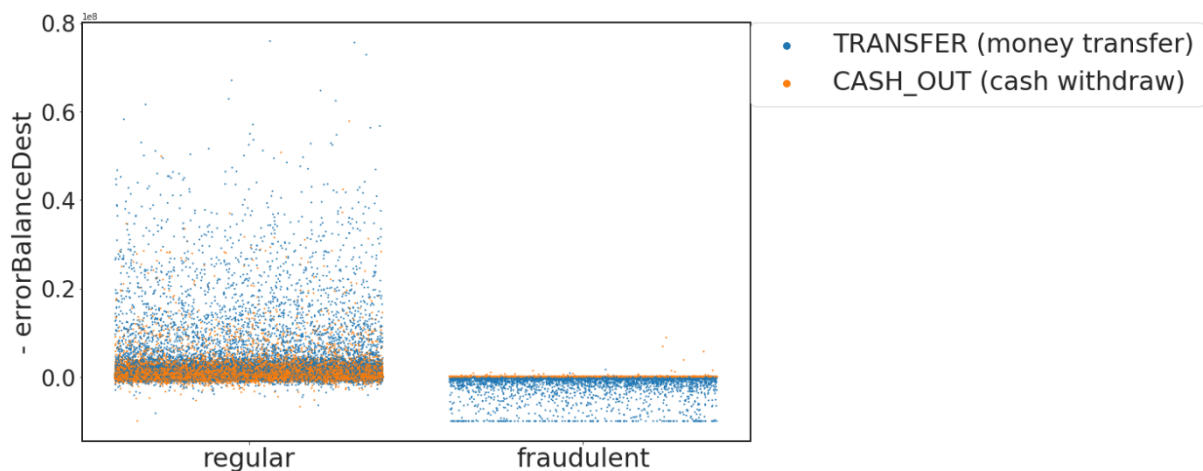


Figure 3: Opposite distribution polarity of error balances in regular vs. fraudulent transactions

Let's get rid of attributes that are irrelevant (have neutral color) and do not affect the result (Appendix 6). We get a clean, ready-to-build dataset model. Since we have previously identified the critical attributes of the model and logically separated them at the program level, we move on to the implementation stage. Here are the received attributes based on which we will build the model (Appendix 7).

3. Building the concept and implementing the model of fraud detection in unbalanced data set

Having identified and formed attributes that allow to clearly separate fraudulent transactions from regular ones, it is advisable, when creating TFDP, to take into account that data is highly

unbalanced, skewed towards regular ones, so there is much more regular transactions than fraudulent, which can result in a biased model. The share of fraudulent transactions is only 0.3%.

It is also important to note which metrics to use to evaluate the model and, in fact, with which algorithm to implement the model. That is, it is necessary not only to create the model itself, but also to be able to qualitatively analyze it to get a real result.

Metric selection: as the data is very unbalanced, we will use the area under the response accuracy curve (AUPRC) to evaluate the model, rather than the usual area under the recipient performance curve (AUROC), because the first one is more sensitive to differences between algorithms and their parameters.

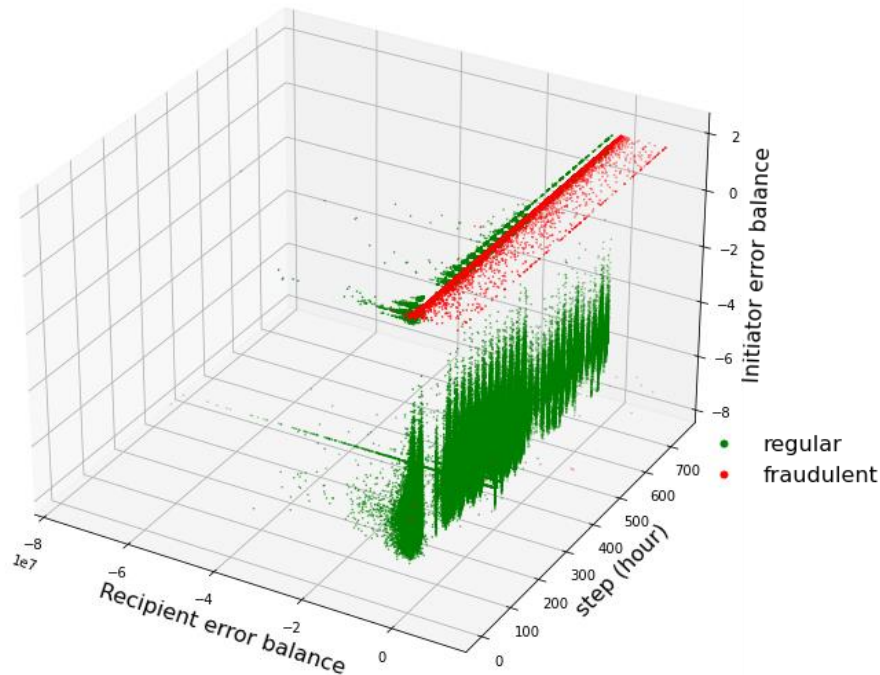


Figure 4: Difference in distribution of fraudulent and regular transactions basing on error balances

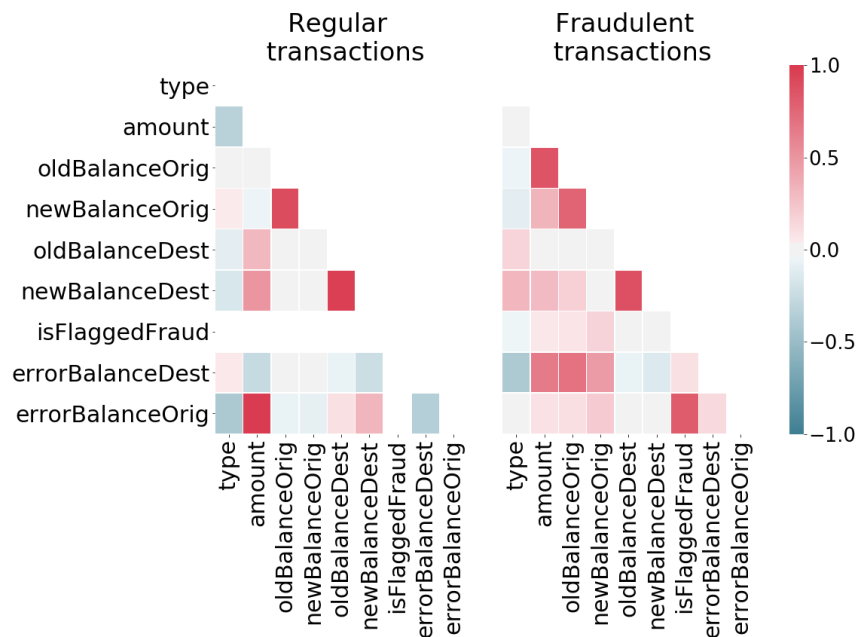


Figure 5: Heats maps of attributes correlation in regular and fraudulent transactions

Algorithm selection: the first approach to working with unbalanced data is to balance them by discarding most classes before applying the algorithm (so-called sub-sampling, or under-sampling).

The disadvantage of subsampling is that the model trained in this way will not work well with real skewed test data, as almost all the information has been discarded. The best approach may be to expand minority class records, for example, using the SMOTE technique (contained in the imblearn library). 5 different models were created, focusing on the detection of anomalies and teacher training: logistic regression, K-nearest neighbors, the method of support vectors (SVM), the Bayesian classifier [6, 11, 12].

But the best result is achieved by applying an algorithm based on ensembles of decision trees, which works effectively on unbalanced data. Such algorithms not only allow us to build a model that can handle potentially missing values in our data, but they also have the shortest data processing time, which will allow faster analysis of the result, potential information system to work faster. Among these algorithms (based on decision tree ensembles) there are 2 most effective - Random Forest and XGBoost, and the latest gradient boosting algorithm still shows the best result. In addition, XGBoost allows you to weigh the positive class (fraud) more efficiently than the negative class (no fraud) - which allows us to process unbalanced data more efficiently.

Let's divide the initial data set into training and test in the ratio of 80:20 (Appendix 8). We initialize the fraud detection classifier implemented on the basis of gradient boosting (XGBoost) and find its estimate using the previously mentioned AUPRC method (Appendix 9).

0.9986361116985445

The algorithm constructed using the AURPC evaluation method has a score of 0.9986, which indicates a very high efficiency of the classifier.

Let's calculate the actual accuracy of the model - the percentage of transactions that were correctly classified. That is, the correctly classified include those that:

- 1) are classified as fraudulent and are in fact fraudulent, or
- 2) are classified as regular and are in fact regular.

Accordingly, others:

- 1) those that are fraudulent, but classified as regular, or
- 2) regular, and have been classified as fraudulent fall into the category of incorrectly classified.

The accuracy of the model will be defined as the ratio of the sum of the correct classifications to the total number of transaction records. We will also create a confusion matrix for visual representation (Figure 6).

Thus, the accuracy of the model implemented using the technology of gradient boosting is 99.97%. What attributes signal a fraudulent operation? We demonstrate the effect of each attribute on the final forecast in the plot below.

As you can see, the artificially calculated variable "errorBalanceOrig" is the most indicative of transaction fraud compared to other attributes (Figure 7).

For a visual representation that the implemented model is not a "black box" - let's visualize the decision-making model. Due to the fact that the decision is based on decision trees, the visualization is simple and clear. From so, the received model classifies transactions being guided by the following decisions (Figure 8).

4. Conclusion and perspectives

The technology of fraud detection in payment systems has been implemented. This technology is based on a model with an accuracy of 99.97% and an AURPC of 99.86% - which is not just a high result, the technology can be recommended for use in business and banking, because among 554082 test transactions only 3 transactions that were classified as regular, turned out to be fraudulent, 166 really regular transactions were identified as fraudulent.

The advantage of the model is that almost no fraudulent transaction was missed. After all, blocking regular transactions, although it worsens the user experience of the system, is more acceptable than losing money. The TFDPC implementation has been accomplished in 4 stages:

1. exploratory data analysis;
2. data visualization with subsequent adaptation of the data set;

3. creation of the technology using existing classification algorithms;
4. visualization of the obtained model and results.

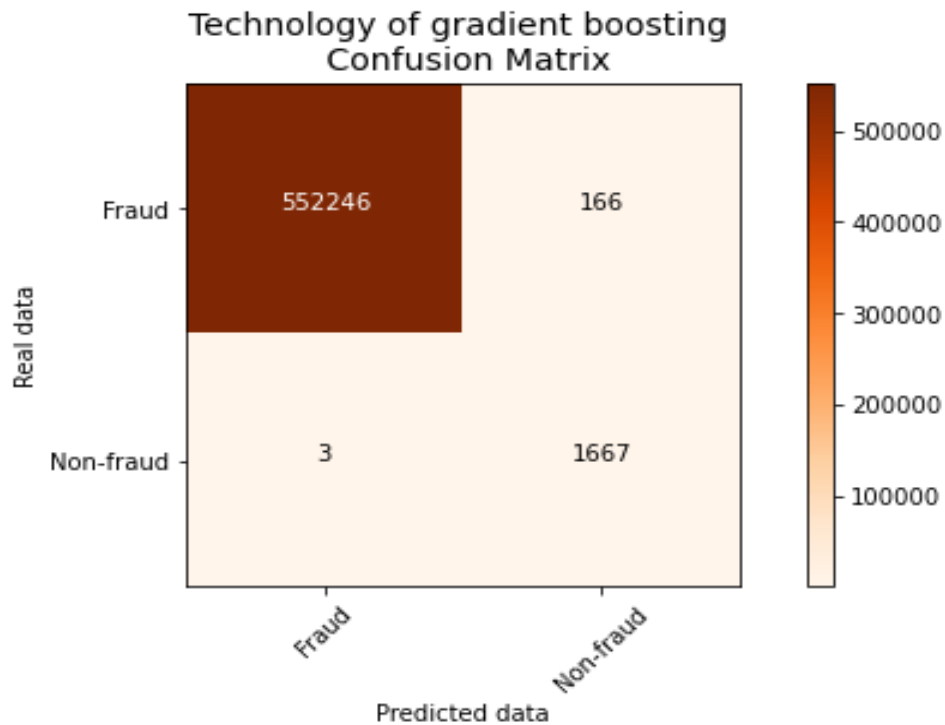


Figure 6: Confusion matrix of TFDPC

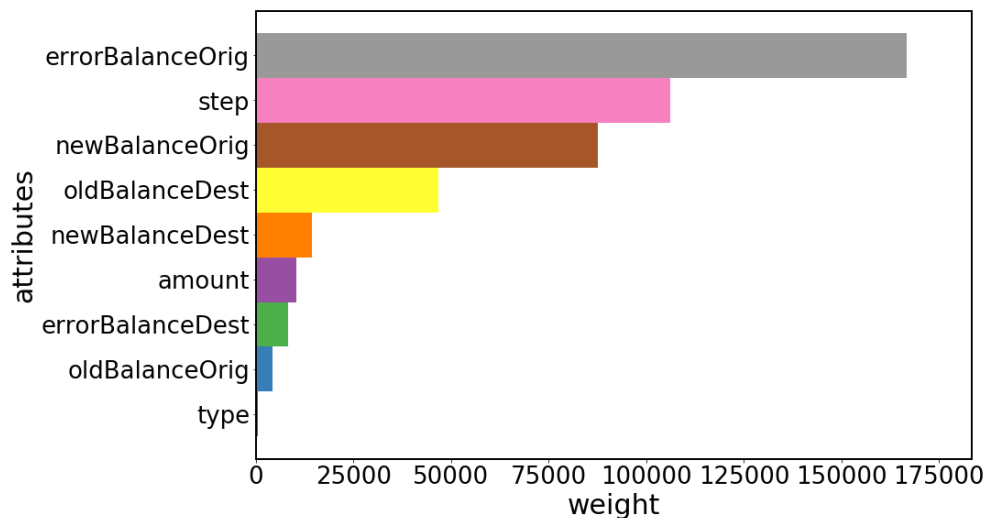


Figure 7: The order of attributes importance in the implemented model

It is worth noting that stages 1 and 2 (preparation and research of data) were no less critical than the creation of the model itself - they allowed a better understanding of the data, their applied value, and highlighted what is really important.

Further development of technology is the implementation of the information system itself, which, based on the created model, could analyze bank transactions in real time and trigger appropriate actions - to block such transactions that were classified as fraudulent. This system would have commercial value, could be integrated into the banking software, payment systems and other financial institutions. The challenge here is to configure the interaction of the system not with the ready-to-go dataset, but dynamically received data. Such a system must withstand high loads, deliver results as

quickly as possible, be secure and highly accessible, and the project architecture must be flexible to adapt to possible changes [13].

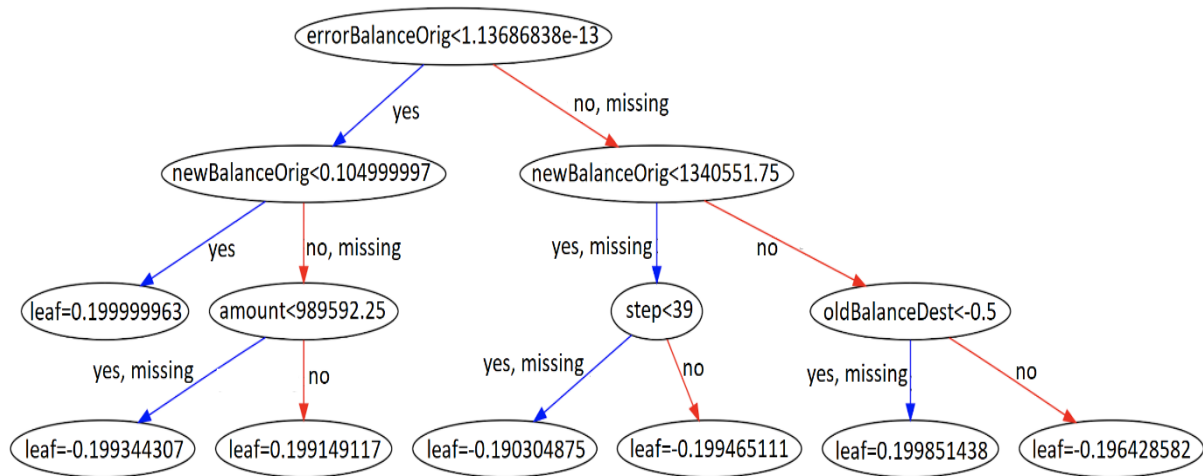


Figure 8: Decision making tree of the implemented fraud detection technology

5. References

- [1] Fraud Detection Techniques: Data and Technique Oriented Perspective / S. Sorournejad, Z. Zojaji, R.E. Atani, Amir Hassan Monadjemi / Cornell University Library, 2016. Mode of access: <https://arxiv.org/ftp/arxiv/papers/1611/1611.06439.pdf> .
- [2] Lebichot, B., Le Borgne, Y.-A.: Deep-Learning Domain Adaptation Techniques for Credit Cards Fraud Detection. In: Oneto, L., Navarin, N., Sperduti, A., Anguita, D. (eds.) Recent Advances in Big Data and Deep Learning, pp. 78–88. Springer, New York (2019)
- [3] Caelen, O., Smirnov, E.N.: Improving Card Fraud Detection Through Suspicious Pattern Discovery. In: Benferhat, S., Tabia, K., Ali, M. (eds.) Advances in Artificial Intelligence: From Theory to Practice, pp. 181–190. Springer, New York (2017)
- [4] Pozzolo, A.D., Caelen, O., Bontempi, G., Johnson, R.A.: Calibrating Probability with Undersampling for Unbalanced Classification. Paper presented at the 2015 IEEE Symposium Series on Computational Intelligence, Cape Town, South Africa, 7-10 December 2015
- [5] Lebichot B., Le Borgne YA., He-Guelton L., Oblé F., Bontempi G. (2020) Deep-Learning Domain Adaptation Techniques for Credit Cards Fraud Detection. In: Oneto L., Navarin N., Sperduti A., Anguita D. (eds) Recent Advances in Big Data and Deep Learning. INNSBDDL 2019. Proceedings of the International Neural Networks Society, vol 1. Springer, Cham. https://doi-org-443.webvpn.jnu.edu.cn/10.1007/978-3-030-16841-4_8
- [6] Sorournejad, S. A Survey of Credit Card Fraud Detection Techniques: Data and Technique Oriented Perspective / S. Sorournejad, Z. Zojaji, R.E. Atani, Amir Hassan Monadjemi / Cornell University Library, 2016. Mode of access: <https://arxiv.org/ftp/arxiv/papers/1611/1611.06439.pdf>
- [7] Kuznietsova, N.V. Analysis and forecasting the risks of credit card fraud. Informatics and Mathematical Methods in Simulation Vol. 8 (2018), No. 1, pp. 16-25
- [8] Kuznietsova, N.V. Scoring Technology for Risk Assessment of Fraud in Banking / Selected Papers of the XVI International Scientific and Practical Conference "Information Technologies and Security" (ITS 2016). — 2016. — Pp. 54-61 .
- [9] Linda Delamaire, Hussein Abdou, John Pointon, "Credit card fraud and detection techniques: a review", Banks and Bank Systems, Volume 4, Issue 2, 2009.
- [10] MasoumehZareapoor, Seeja.K.R, M.Afshar.Alam, "Analysis of Credit Card Fraud Detection Techniques: based on Certain Design Criteria", International Journal of Computer Applications (0975 – 8887) Volume 52– No.3, 2012.

- [11] Beninel, F. Transfer Learning Using Logistic Regression in Credit Scoring / F. Beninel, W. Bouaguel, G. Belmufti / Cornell University Library, 2012. Mode of access: <https://arxiv.org/pdf/1212.6167.pdf>
- [12] Trainable neural networks modelling for a forecasting of start-up product development. Morozov, V., Mezentseva, O., Proskurin, M. Proceedings of the 2020 IEEE 3rd International Conference on Data Stream Mining and Processing, DSMP 2020, 2020, pp. 55–60, 9204264
- [13] Teslia I., Yehorchenkov O., Khlevna I., Khlevnyi A. (2018). Development concept and method of formation of specific project management methodologies. Eastern European Journal of Advanced Technologies, 5/3(95), 6 – 16.

6. Appendix

Appendix 1

```
fraudTransactions = df.loc[df.isFraud == 1].type.drop_duplicates().values
print(list(fraudTransactions))
['TRANSFER', 'CASH_OUT']
```

```
dfTransactions = df.loc[(df.type == 'TRANSFER') | (df.type == 'CASH_OUT')]
dfFraud = dfTransactions['isFraud']
del dfTransactions['isFraud']
```

Appendix 2

```
dfTransactions.loc[dfTransactions.type == 'TRANSFER', 'type'] = 0
dfTransactions.loc[dfTransactions.type == 'CASH_OUT', 'type'] = 1
dfTransactions.type = dfTransactions.type.astype(int)
```

Appendix 3

```
dfTransactionsFraud = dfTransactions.loc[dfFraud == 1]
dfTransactionsNonFraud = dfTransactions.loc[dfFraud == 0]
fractionAnomalyTransactionsInFraud = len(dfTransactionsFraud.loc[
    (dfTransactionsFraud.oldBalanceDest == 0)
    & (dfTransactionsFraud.newBalanceDest == 0)
    & (dfTransactionsFraud.amount)
]) / (1.0 * len(dfTransactionsFraud))
print(
    "Part of anomaly transactions among fraudulent: ",
    fractionAnomalyTransactionsInFraud
)

fractionAnomalyTransactionsInNonFraud = len(dfTransactionsNonFraud.loc[
    (dfTransactionsNonFraud.oldBalanceDest == 0)
    & (dfTransactionsNonFraud.newBalanceDest == 0)
    & (dfTransactionsNonFraud.amount)
]) / (1.0 * len(dfTransactionsNonFraud))
print(
    "Part of anomaly transactions among regular (non-fraudulent): ",
    fractionAnomalyTransactionsInNonFraud
)
```

Appendix 4

```
dfTransactions.loc[
```

```
(dfTransactions.oldBalanceDest == 0)
& (dfTransactions.newBalanceDest == 0)
& (dfTransactions.amount != 0), \
    ['oldBalanceDest', 'newBalanceDest']] = - 1
```

Appendix 5

```
dfTransactions.loc[
    (dfTransactions.oldBalanceOrig == 0)
    & (dfTransactions.newBalanceOrig == 0)
    & (dfTransactions.amount != 0), \
    ['oldBalanceOrig', 'newBalanceOrig']] = np.nan
```

Appendix 5

```
dfTransactions['errorBalanceDest'] = \
    dfTransactions.oldBalanceDest + dfTransactions.amount \
    - dfTransactions.newBalanceDest
dfTransactions['errorBalanceOrig'] = \
    dfTransactions.newBalanceOrig + dfTransactions.amount \
    - dfTransactions.oldBalanceOrig
```

Appendix 6

```
dfTransactions = dfTransactions.drop(
    ['nameOrig', 'nameDest', 'isFlaggedFraud'],
    axis = 1
)
```

Appendix 7

```
list(dfTransactions)
['step', 'type', 'amount', 'oldBalanceOrig', 'newBalanceOrig', 'oldBalanceDest',
 'newBalanceDest', 'errorBalanceDest', 'errorBalanceOrig']
```

Appendix 8

```
randomState = 5
np.random.seed(randomState)
trainX, testX, trainY, testY = train_test_split(
    dfTransactions, dfFraud, test_size = 0.2, random_state = randomState
)
```

Appendix 9

```
weights = (dfFraud == 0).sum() / (1.0 * (dfFraud == 1).sum())
classifier = XGBClassifier(max_depth = 3, scale_pos_weight = weights, \
    n_jobs = 4)
predictions = classifier.fit(trainX, trainY).predict_proba(testX)
AURPC = average_precision_score(testY, predictions[:, 1])
AURPC
```