# A systematic Approach to Fraud Detection on automated Banking using Machine learning Technics.

Tuleun Terhemen Daniel

Moscow Institute of Physics and Technology (MIPT), National Research University Russia.

tuleun.t@phystech.edu OCID 0000-0002-6982-2710.

Md Mehedi Hassan

Moscow Institute of Physics and Technology (MIPT), National Research University Russia. Khasan.m@phystech.edu

Alexey Nikolaevich Nazarov

MIREA - Russian Technological University, Moscow, Russia. a.nazarov06@bk.ru, ORCID ID-0000-0002-0497-0296

Abdulrashid Saadat

Moscow Institute of Physics and Technology (MIPT), National Research University Russia. Saadat.a@phystech.edu



## Abstract:

Internet banking has a greater impact on every nation economy but if not properly handled can cause the downfall of the economy of a whole country or a continent at large. Hence, there is every need for researchers in the field of data science to be up to date in order to track any malicious attack or any attempt that will lead to tempering with the holistic nature of legitimacy of our financial transactions on the internet. The objective of this paper is to find the patterns of transactions performed and help algorithms to learn those patterns by identifying the fraudulent transactions and flag them. The algorithm is built to identify and prevent fraudulent activities on the banking website to ensure a safe and trustworthy online experience for the customers. Thereby developing a robust and accurate system that can identify and prevent fraudulent activities in online payment transaction. In order to achieve this we selected some strong and interesting machine learning algorithms where a python programing language is used to train the dataset and great results were obtain. All our selected seven algorithms perform excellent with a greater competitive accuracy between XGBoost and Random forest. Finally, Random forest is considered the best model with the accuracy of 100.

**Key Words:** Fraud detection, Automated banking, Machine learning, Cash transaction, thresholds value**.**

## I. INTRODUCTION

The advent of internet as the digital revolution has rising and has a greater effect to every aspects of our lives. One of the most important digital revolution happened in financial system and especially transacting money to someone from any part of the world digitally. Digital transactions have become a part of daily life like purchasing a product online, sending money to friends, depositing cash in

bank account, investment purposes etc., they had a lot of benefits so does it paved way for fraudulent activities. People started using digital money transactions medium to launder money and make the money look like it comes from a legal source. This work is divided into 5 sections, beginning with the introduction, section 2 is the review of the related work, section 3 present the methodology, section 4 Result and analysis and finally section 5 which is the conclusion and further studies. The main contribution to knowledge of this research work is the fact that, our model did a great job as it was not only limited to detecting the fraud but it was also able to trigger the alarm for any malicious activity and try to prevent the fraud from occurring.

## II. RELATED WORK

There are few published works about fraud detection within the domain of online banking applications. This is most likely due to the privacy, the secrecy and the commercial interests concerning this domain, rather the absence of research [3]. Therefore, due to the limited exchange of ideas, the development of new fraud detection methods in the banking area is difficult. Most published work is related to the domain of credit card, computer intrusion and mobile communication. Some relevant works on fraud detection are reviewed next. Credit card frauds- Most of the works on preventing and detecting credit card fraud were carried out with special emphasis on data mining and neural networks. Aleskerov, Freisleben and Rao [4] describe a neural network based database mining system in which a neural network is trained with the past data of a particular customer and the current spending patterns is processed to detect possible anomalies. However, Bolton and Hand [5] proposed a detection technique in which break point analysis is used to identify changes in spending behavior. 166 ICDS 2011 : The Fifth International Conference on Digital Society Copyright (c) IARIA, 2011. ISBN: 978-1-61208-116-8 Computer intrusion- Intrusion detection approaches in computers is broadly classified into two categories based on a model of intrusions: misuse and anomaly detection. Misuse detection attempts to recognize the attacks of previously observed intrusions in the form of a pattern or a signature and then monitors such occurrence. Anomaly detection tries to establish a historical normal profile for each user, and then uses sufficiently large deviation from the profile to indicate possible intrusions [6]. Denning [7] presents a statistical model for real-time intrusion detection based in anomaly detection. Ghosh and Schwrtzbard [8] describe an approach that employs artificial neural networks used for both anomaly and misuse detection. Mobile communication frauds - Fraud in communication networks refers to the illegal access to the network and the use of its services. Cortes and Pregibon [9] define statistical summaries, denominated signatures, of users over two time windows, namely, current and historical, respectively. The current network activity is compared with the historical activity for any deviation. Fawcett and Provost [10] present rule-based methods and neural networks for detecting fraudulent calls based on profiling subscriber behavior. In all domains above mentioned, fraudsters tends to adapt to new prevention and detection measures. In the same way, legitimate users may gradually change their behavior over a longer period of time. Therefore, fraud detection techniques need to be adaptive and to evolve over time in order to avoid false alarms. Models can be updated at fixed time points or continuously over time [9][10]. Panigrahi, Kundu, Sural, and Majumdar [11] describe a framework for fraud detection in mobile communication networks using rule-based deviation method. The main point of this paper is the detailed description of the use of Dempster-Shafer theory in order to combine the evidences of fraud given by two rules. The system proposed in this paper combines three different approaches: (1) differential analysis using statistical models in order to detect local evidence of fraud; (2) an innovative approach using a probabilistic model for evaluating the likelihood of a

transaction being a fraud based on its global behavior; and (3) Dempster-Shafer theory for combining evidences of fraud. In this work we have tried to do fraud detection on a bank payment data and we have achieved remarkable results with our classifiers. Since fraud datasets have an imbalance class problem we performed an oversampling technique called SMOTE and generated new minority class examples. We have investigated some classification results without SMOTE in order to check the accuracy performance of our model. As earlier said, fraud datasets will be imbalanced and most of the instances will be non-fraudulent. Imagine that we have the dataset here and we are always predicting non-fraudulent. Our accuracy would be almost 99 % for this dataset and mostly for others as well since fraud percentage is very low. Our accuracy will be very high but we may not be detecting any frauds so it will be a useless classifier. So the base accuracy score should be better at least than predicting always non-fraudulent for performing a detection.

## A. Dataset description

1. Step - maps a unit of time in the real world. In this case 1 step is 1 hour of time. Total steps 744 (30 days simulation).
2. Type - CASH-IN, CASH-OUT, DEBIT, PAYMENT and TRANSFER.
3. Amount - amount of the transaction in local currency.
4. Name Orig - customer who started the transaction
5. Old balance Org - initial balance before the transaction
6. new balance Orig - new balance after the transaction
7. name Dest - customer who is the recipient of the transaction
8. Old balance Dest - initial balance recipient before the transaction. Note that there is not information for customers that start with M (Merchants).
9. New balance Dest - new balance recipient after the transaction. Note that there is not information for customers that start with M (Merchants).
10. Is Fraud - This is the transactions made by the fraudulent agents inside the simulation. In this specific dataset the fraudulent behavior of the agents aims to profit by taking control or customers' accounts and try to empty the funds by transferring to another account and then cashing out of the system.
11. Is Flagged Fraud - The business model aims to control massive transfers from one account to another and flags illegal attempts. An illegal attempt in this dataset is an attempt to transfer more than 200,000 in a single transaction.

## B. Pivot table analysis

Numbers is everything in transaction monitoring. Numbers decide whether it is a fraudulent activity or not. Let us look at the overall numbers using pivot function

Feature engineering. As per the current rule based algorithm, there has been no flags during fraud transactions in case of cash out, which a serious concern to the anti-money laundering system. Also, there are only 16 transactions which are flagged as fraud whereas around 4k transactions are actually fraud. Our mission is now to build an efficient algorithm to mitigate this risk of letting fraud transactions unblocked see the screenshot of table one from our simulation below. Time to get our hands dirty with feature engineering. With the available information it is hard to train the model and get better results. Hence we move onto create new features by altering the existing features. In this we create three functions which creates a highly relevant feature for the domain.

**Table.1** summary of the data transaction.

| | sum | | | std | | |
|---|---|---|---|---|---|---|
| | amount | isFlaggedFraud | isFraud | amount | isFlaggedFraud | isFraud |
| type | | | | | | |
| CASH_IN | 236367391912.459991 | 0 | 0 | 126508.255272 | 0.000000 | 0.000000 |
| CASH_OUT | 394412995224.489990 | 0 | 4116 | 175329.744483 | 0.000000 | 0.042851 |
| DEBIT | 227199221.280000 | 0 | 0 | 13318.535518 | 0.000000 | 0.000000 |
| PAYMENT | 28093371138.369999 | 0 | 0 | 12556.450186 | 0.000000 | 0.000000 |
| TRANSFER | 485291987263.169983 | 16 | 4097 | 1879573.528908 | 0.005479 | 0.087344 |
| All | 1144392944759.770020 | 16 | 8213 | 603858.184009 | 0.001586 | 0.035905 |

**Difference in balance:** It is a universal truth that the amount debited from senders account gets credited into the receivers account without any deviation in cents. But what if there is a deviation in case of the amount debited and credited. Some could be due to the charges levied by the service providers, yet we need to flag such unusual instances.



**Surge indicator:** Also we have to trigger flag when large amount are involved in the transaction. From the distribution of amount we understood that we have a lot of outliers with high amount in transactions. Hence we consider the 75th percentile (450k) as our threshold and amount which is greater than 450k will be triggered as a flag

**Frequency indicator:** Here we flag the user and not the transaction. When there is a receiver who receives money from a lot of people, it could be a trigger as it can be for some illegal games of chance or luck. Hence it is flagged when there is a receiver who receives money for more than 20 times.

**Merchant indicator:** The customer ids in receiver starts with 'M' which means that they are merchants and they obviously will have a lot of receiving transactions. So we also flag whenever there is a merchant receiver.

**Split and Standardize:** In this module we create the independent and dependent feature, then split them into train and test data where training size is 70%. Later we collect all the numerical features and apply Standard Scaler () function which transforms the distribution so that the mean becomes 0 and standard deviation becomes 1.

**Tokenization.**
We had the customer ids and merchant ids stored in object type. It is bad to apply one hot encoding in it as it can lead to more features and curse of dimensionality can incur. Hence we

applied tokenization here as it can create an unique id number which is in 'int' type for each customer id.
Dropping unnecessary columns
We don't need the sender and receiver id as we have tokenized them, also we don't required is Flagged Fraud as it is just an outcome of current algorithms**.**

## III. **Methodology**.

In this work we consider a systematic Approach to Fraud Detection on automated Banking system using Machine learning Technics. Two datasets were collected and further divided in to training, testing and Visualization after that, many Machine learning (algorithms) such as Decision tree, Naïve Bayes, K-Nearest Neighbor, Random Forest, XGBOOST Classifier, Support vector classifier and Logistics Regression were used in training, testing and prediction of the dataset. When the data was collected, after data cleaning, pre-processing, and wrangling, the first step we did was to feed it to an outstanding model and of course, get output in probabilities. After that a tokenization was done on the data to help create a unique id for each customer and a confusion matrix was used to measure the effectiveness and the performance of the model

### Data source

The dataset generated using the simulator called PaySim as an approach to such a problem. PaySim uses aggregated data from the private dataset to generate a synthetic dataset that resembles the normal operation of transactions and injects malicious behavior to later evaluate the performance of fraud detection methods. PaySim simulates mobile money transactions based on a sample of real transactions extracted from one month of financial logs from a mobile money service implemented in an African country. The original logs were provided by a multinational company, who is the provider of the mobile financial service which is currently running in more than 14 countries all around the world. The dataset used in this research is available on kaggle and it is downloaded and saved as a csv file. Both the dataset and the notebook can be provided to anyone for any reasonable request that has to do with research purpose.



## IV. Result and Discussion.

**Machine learning** can be used for the detection of fraud transaction. Predictive models produce **good precision score** and are capable of detection of fraud transaction. There are 2 flags which stand out to me and it's interesting to look onto: is Fraud and is Flagged Fraud column. From the hypothesis, **is Fraud** is the indicator which indicates the **actual fraud transactions** whereas **is Flagged Fraud** is what the system prevents the transaction due to **some thresholds** being triggered. From the table above we can see that there are some relation between other columns and is Flagged Fraud thus there must be relation between is Fraud. The total number of fraud transaction is 8213.The total number of fraud transaction which is marked as fraud 16.Ratio of fraud transaction versus non-fraud transaction is 1:773. Thus in every 773 transaction there is 1 fraud transaction happening. Amount lost due to these fraud transaction is $12056415427. The accuracy of our model has a slight different with data without SMOTE, but precision, recall, f1 score is higher than data without SMOTE. We have successfully processed the data and served the data to the model. It is time consuming to find out which model works best for our data. Hence we have utilized pipeline to run our data through all the classification algorithm and select the best which gives out the maximum accuracy. We can see who won the prize-it is Random forest. Other algorithms have also performed in part with Random Forest especially XGBoost Classifier and Naïve Bayes.

Using XGBoost classifier, we managed to build a classifier that is robust enough to classify fraudulent transaction. By using XGBoost classifier, we can build the model without using any kind of resampling method. Adding the weight on the minority class using scale pos_ weight enabled us to build a robust model even on imbalanced dataset Moreover, by tuning the hyper parameter of the model based on recall as the performance metric, the model managed to capture as much fraudulent transaction as possible. Only misclassifying a few fraudulent transaction as legit transaction. The model can still be improved to be able to reduce the amount of false positive. This can be done by picking F1 as the metric.
The table below shows the results for the seven algorithms for used in our research.

Table: Result of the selected algorithms.

| Name of the model | Precision | F1-score | Recall | Accuracy |
|---|---|---|---|---|
| XGBOOST CLASSIFIER | 99 | 99 | 99 | 100 |
| LOGISTICS REGRESSION | 98 | 98 | 98 | 97 |
| DECISION TREE | 99 | 99 | 99 | 98 |
| SUPPORT VECTOR CLASSIFIER | 98 | 98 | 98 | 98 |
| RANDOM FOREST | 100 | 100 | 100 | 100 |
| NAÏVE BAYES | 99 | 99 | 99 | 99 |
| KNN | 99 | 99 | 99 | 98 |

## A. Data Visualization.



The best way of confirming that the data contains enough information so that a ML algorithm can make strong predictions, is to try and directly visualize the differences between fraudulent and genuine transactions. Motivated by this principle, we visualize these differences in several ways in the plots below. The plot below shows how the fraudulent and genuine transactions yield different fingerprints when their dispersion is viewed over time. It is clear that fraudulent transactions are more homogenously distributed over time compared to genuine transactions. Also apparent is that CASH-OUTs outnumber TRANSFERs in genuine transactions, in contrast to a balanced distribution between them in fraudulent transactions. Note that the width of each 'fingerprint' is set by the 'jitter' parameter in the plot Strip function above which attempts to separate out and plot transactions occurring at the same time with different abscissae.

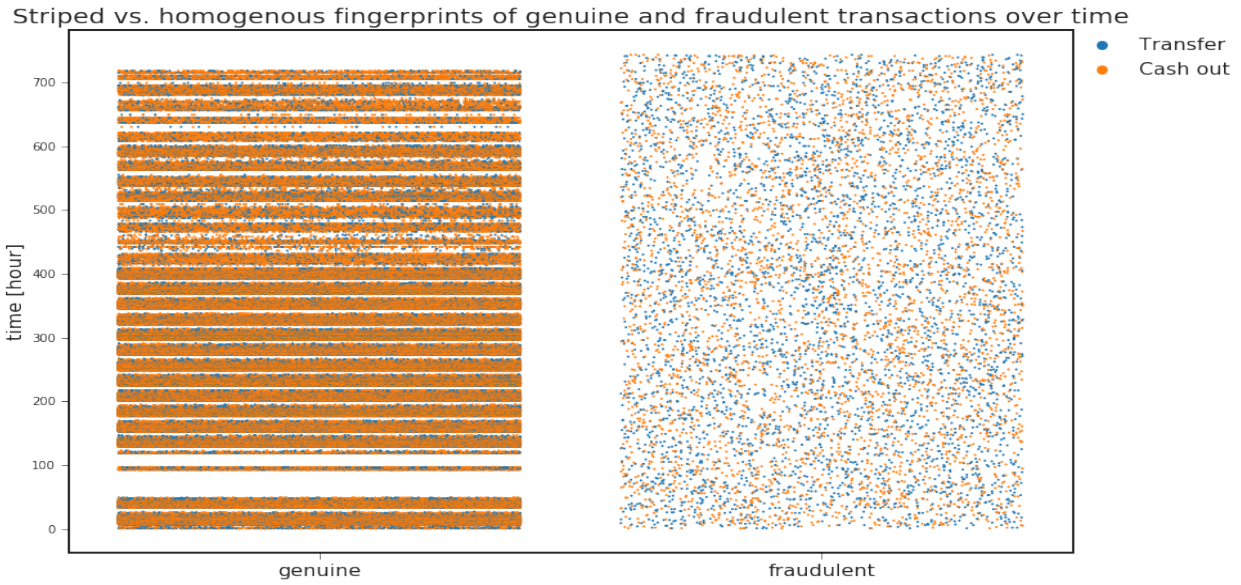


Fig.1 Finger printing comparison

The two plots below shows that although the presence of fraud in a transaction can be discerned by the orig. amount feature, the new error Balance Dest feature is more effective at making a distinction.



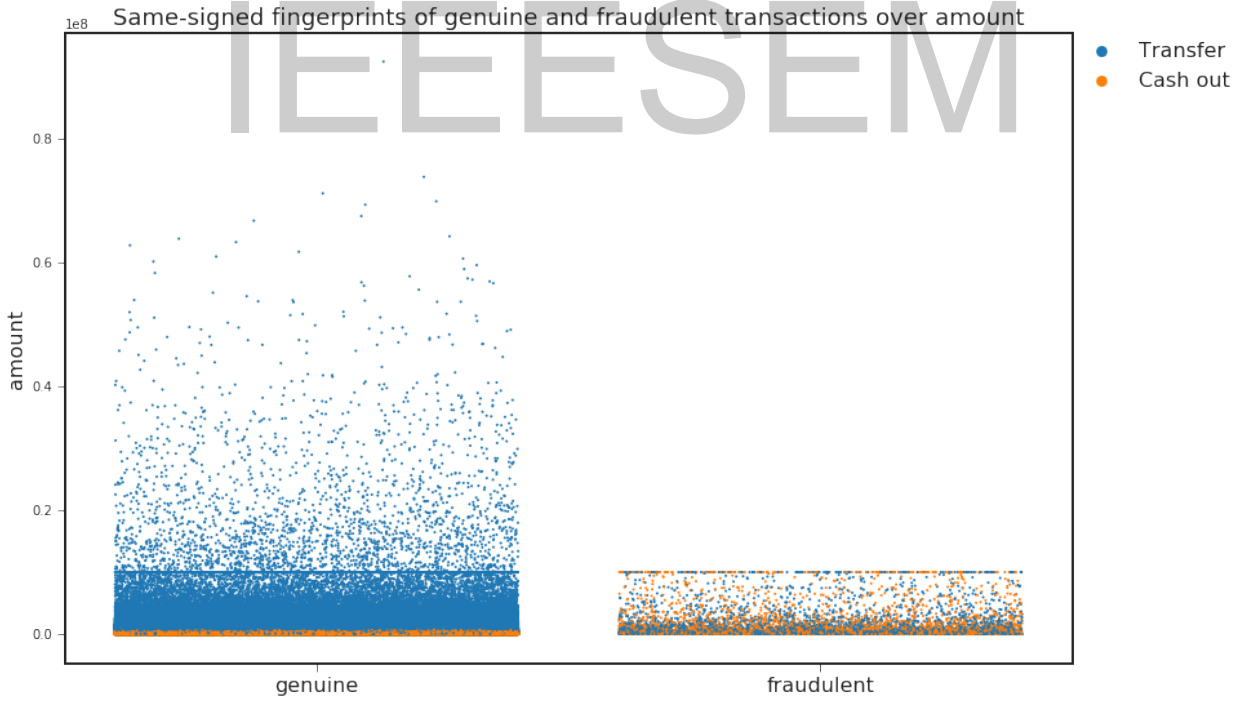


Fig2. Same-signed finger prints of genuine and fraudulent transactions over amount.

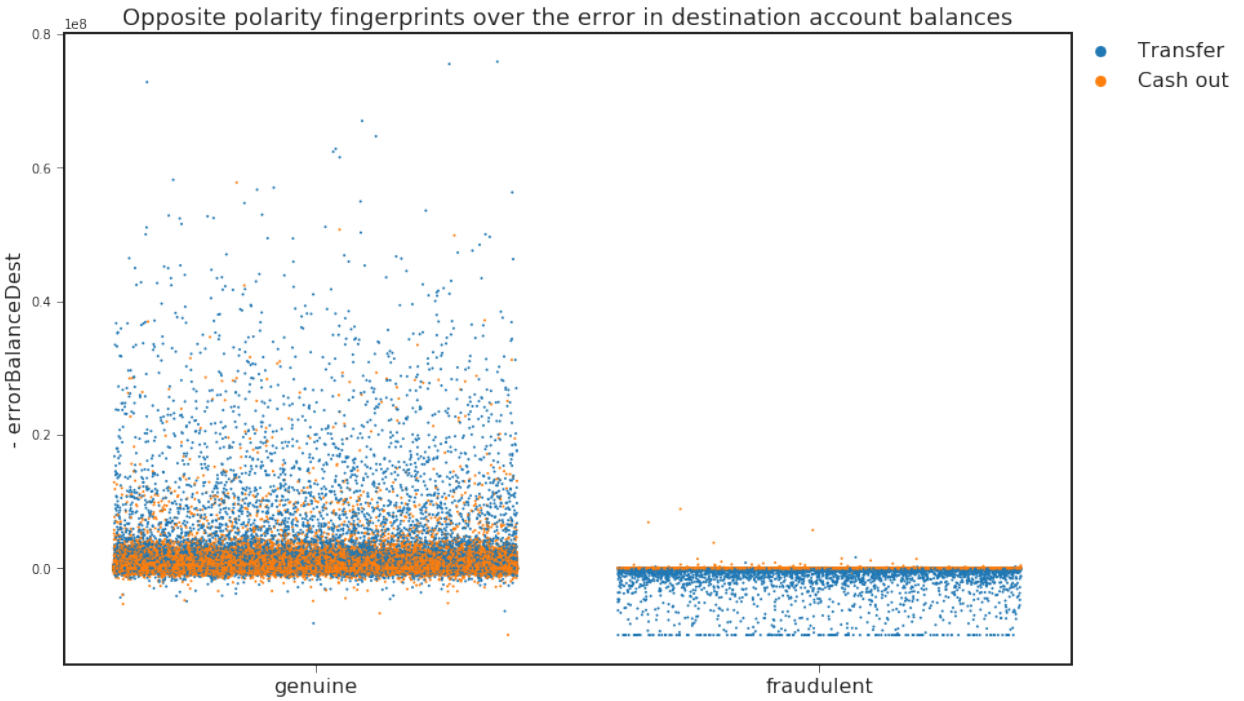


Fig2. Polarity fingerprints over error in destination account balance.

Smoking gun and comprehensive evidence embedded in the dataset of the difference between fraudulent and genuine transactions is obtained by examining their respective correlations in the heat map below.



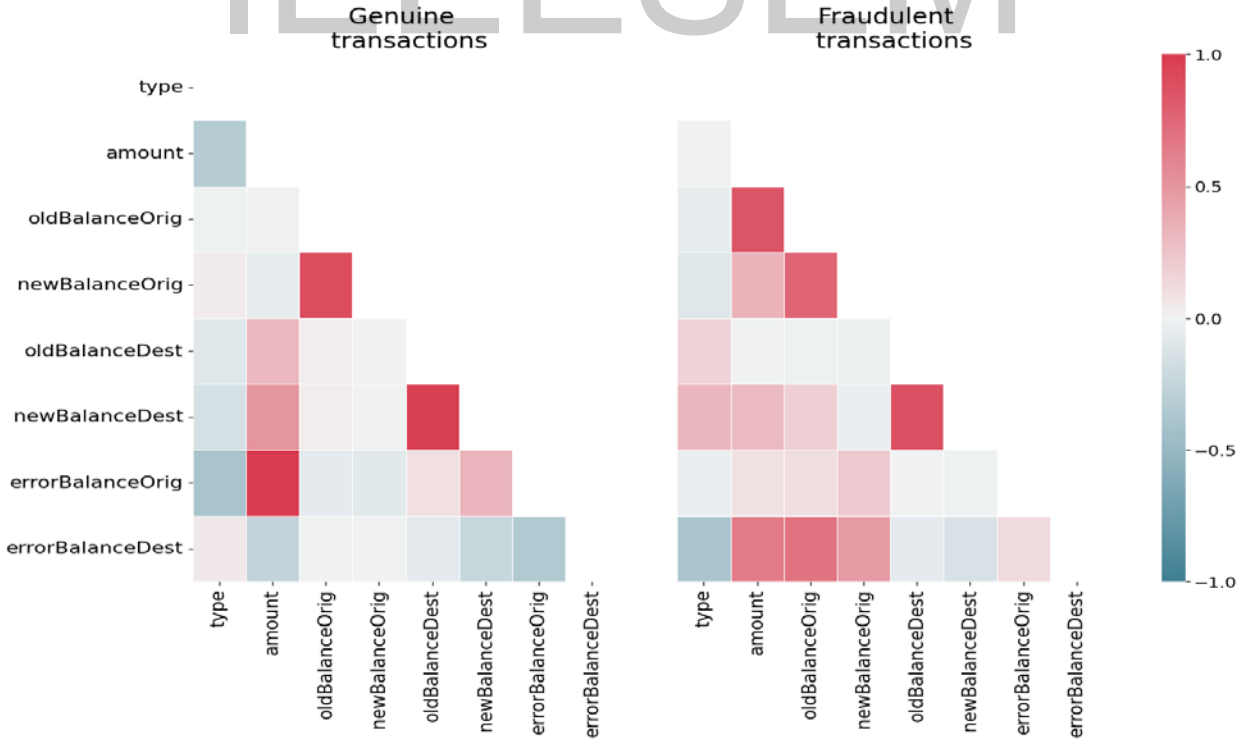


Fig5. Heat map for the genuine and fraudulent transactions.

## B.Machine Learning to Detect Fraud in Skewed Data

Having obtained evidence from the plots above that the data now contains features that make fraudulent transactions clearly detectable, the remaining obstacle for training a robust ML model is the highly imbalanced nature of the data. *Selection of metric*: Since the data is highly skewed, I use the area under the precision-recall curve (AUPRC) rather than the conventional area under the receiver operating characteristic (AUROC). This is because the AUPRC is more sensitive to differences between algorithms and their parameter settings rather than the AUROC ( Davis and Goadrich, 2006).

*Selection of ML algorithm*: A first approach to deal with imbalanced data is to balance it by discarding the majority class before applying an ML algorithm. The disadvantage of under sampling is that a model trained in this way will not perform well on real-world skewed test data since almost all the information was discarded. A better approach might be to oversample the minority class, say by the synthetic minority oversampling technique (SMOTE) contained in the 'imblearn' library. Motivated by this, I tried a variety of anomaly-detection and supervised learning approaches. I find, however, that the best result is obtained on the original dataset by using a ML algorithm based on ensembles of decision trees that intrinsically performs well on imbalanced data. Such algorithms not only allow for constructing a model that can cope with the missing values in our data, but they naturally allow for speedup via parallel-processing. Among these algorithms, the extreme gradient-boosted (XGBoost) algorithm used below slightly outperforms random-forest. Finally, XGBoost, like several other ML algorithms, allows for weighting the positive class more compared to the negative class, a setting that also allows to account for the skew in the data. Split the data into training and test sets in a 80:20 ratio.

The figure.6 below shows that the new feature *error Balance Origin* that we created is the most relevant feature for the model. The features are ordered based on the number of samples affected by splits on those features. The 3D plot in figure 7 below distinguishes best between fraud and non-fraud data by using both of the engineered error-based features. Clearly, the original *step* feature is ineffective in separating out fraud. Note the striped nature of the genuine data versus time which was anticipated from the figure.1.



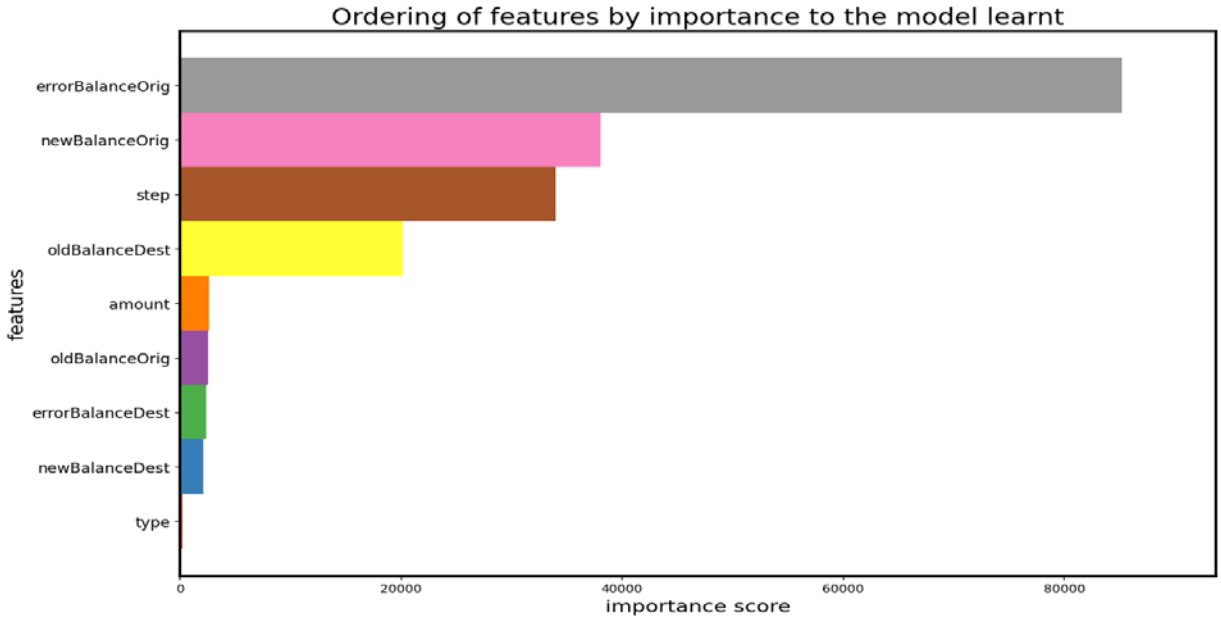


Fig 6. Feature arrangement for the model.

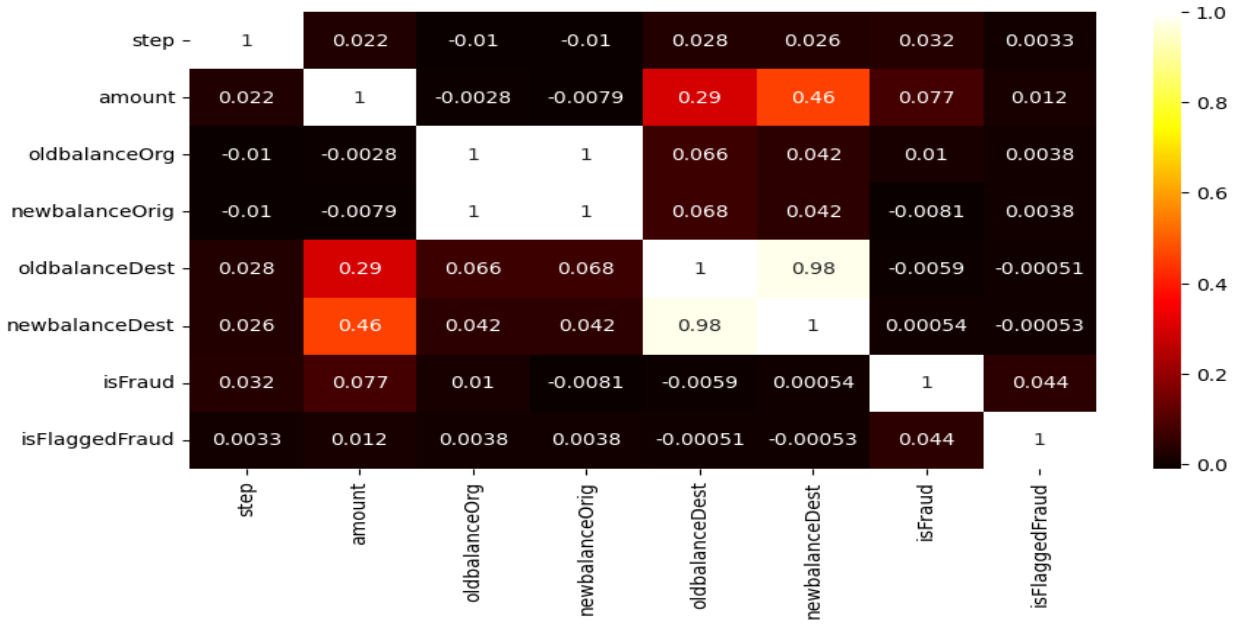


Fig 7. Heat map from the visualization of the data.



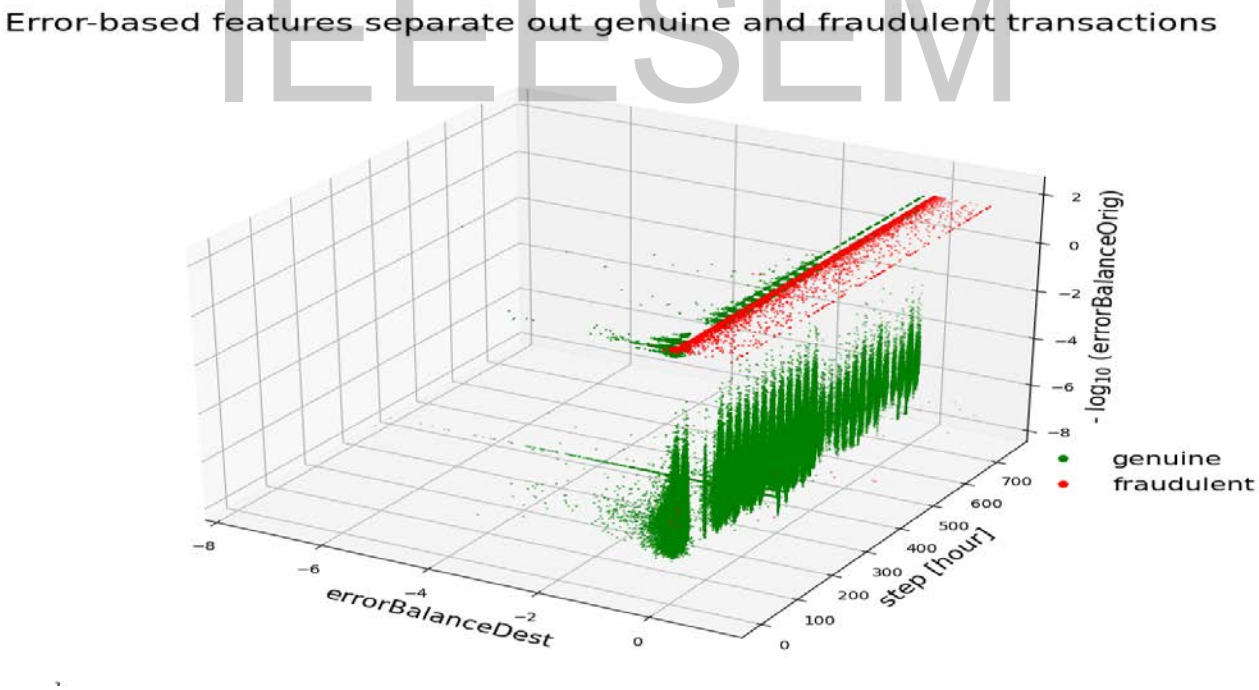


Fig 7. Error-Based features separation of genuine and fraudulent transaction.

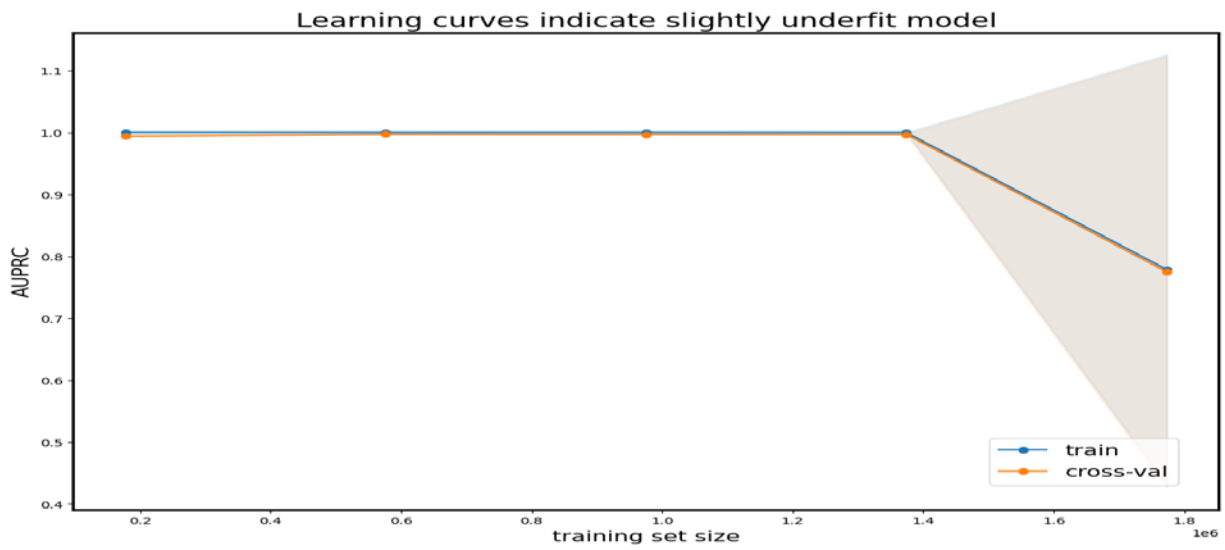


Fig 7.Learning curves indicating the behavior of the model on the data.

The root node in the decision tree visualized below is indeed the feature error Balance Orig, as would be expected from its high significance to the model. The model we have learnt has a degree of bias and is slightly underfit. This is indicated by the levelling in AUPRC as the size of the training set is increased in the cross-validation curve above. The easiest way to improve the performance of the model still further is to increase the max_depth parameter of the XGBClassifier at the expense of the longer time spent learning the model. Other parameters of the classifier that can be adjusted to correct for the effect of the modest under fitting include decreasing min_child_weight and decreasing reg_lambda.



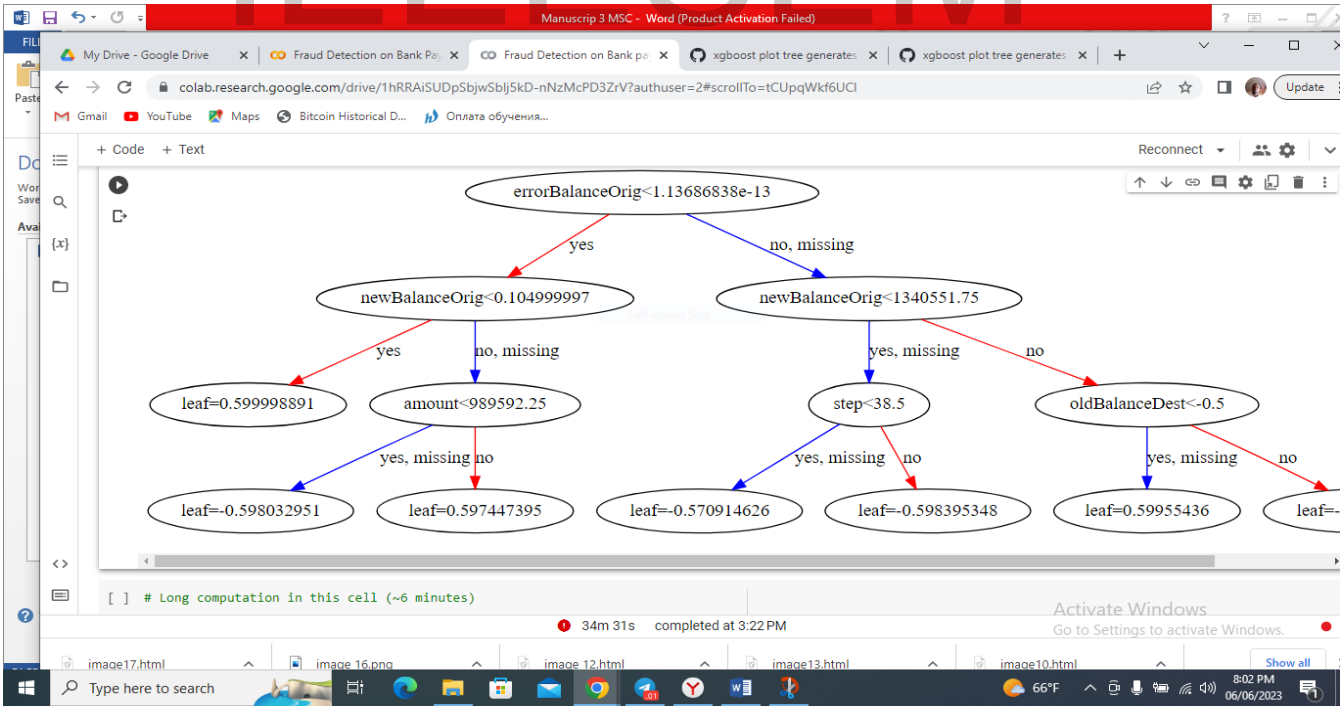


Fig. 8 The root node in the decision tree visualization.

## V. CONCLUSION.

With the advent of digital transactions, the possibility of money laundering have also soared up with the use of technology. Millions of investigators are on the field fighting against the fraudulent transactions. In the current industry we have a large inflow of false positives hits and it consumes a long time to clear the false positive hits. Customers across the world using financial technology platforms demand lightning fast services. Hence automating the hits with machine learning and reducing the false positive hits is our aim. But not at the cost of leaving out the false negatives. Hence we were more mindful about false negatives when we try to reduce the false positives. In this research a systematic approach for fraud detection on internet banking is studied using different Machine learning techniques, the provided data has the financial transaction data as well as the target variable **is Fraud**, which is the actual fraud status of the transaction and, **is Flagged Fraud** is the indicator which the simulation is used to flag the transaction as being malicious or abnormal using some **threshold value.** We thoroughly interrogated the data at the outset to gain insight into which features could be discarded and those which could be valuably engineered. The plots provided visual confirmation that the data could be indeed be discriminated with the aid of the new features. To deal with the large skew in the data, we chose an appropriate metric and used an ML algorithm based on an ensemble of decision trees which works best with strongly imbalanced classes. We also make use of the smote to improve the performance of the model. The accuracy has slight different with data without SMOTE, but precision, recall, f1 score is higher than data without SMOTE. The method used in this work should therefore be broadly applicable to a range of such problems in the future.



## Acknowledgement.

We are grateful to our teachers especially Professor Alexey Nikolaevich Nazarov, Ilia Mikhailovich Voronkov and Professor Ivaschenko for the knowledge they impacted to us during the period of our program here at MIPT and also having time to vet the manuscript before publication. I must sincerely acknowledge the support giving to me during this research by my fiancée Salome Iwanger Orban.

## REFERENCES

[1] R. J. Bolton and D. J. Hand, “Statistical fraud detection: A review”.Statistical Science. Vol. 17, No. 3, 2002, pp 235-255.

[2] U. Murad and G. Pinkas, “Unsupervised profiling for identifying superimposed fraud”, in Proceedings of the 3rd European Conference

on Principles of Data Mining and Knowledge Discovery, 1999, pp.251-266

[3] K. N. Karsen and T. G. Killingberg, “Profile based intrusion detection for Internet banking systems”, Master Thesis, Norwegian University of Science and Technology, Norway, 2008

[4] E. Aleskerov, B. Freisleben, and B. Rao, “CARDWATCH: A neural network based database mining system for credit card fraud detection”, in Computational Intelligence for Financial Engineering.

Proceedings of the IEEE/IAFE, 1997, pp 220- 226. IEEE, Piscataway,NJ.

[5] R. J. Bolton and D. J. Hand, “Unsupervised profiling methods for fraud detection”, in Conference on Credit Scoring and Credit Control 7”, Edinburgh, UK, 5-7 Sept., 2001.

[6] Y. Kou, C.T. Lu, S. Sirwonqattana, and Y.P. Huanq, “Survey of fraud detection techniques”, in Proceedings of the IEEE International Conference on Networking, Sensing and Control, vol. 1, 2004, pp.

749-754.

[7] D. E. Denning. “An intrusion detection model”. IEEE Transactions on Software Engineering, 13:222-232, February 1987.

[8] A. K. Ghosh and A. Schwartzbaxd. “A study in using neural networks for anomaly and misuse detection”, in Proceedings of the 8th USENIX Security Symposium, 1999.

[9] C. Cortes and D. Pregibon, “Signature-based methods for data streams,” Data Mining and Knowledge Discovery, vol. 5, no. 3, pp.167-182, 2001.

[10] T. Fawcett and F. Provost, “Adaptive fraud detection”, Data Mining and Knowledge Discovery Journal, Kluwer Academic Publishers,Vol. 1, No. 3, 1997, pp. 291-316.

[11] S. Panigrahi, A. Kundu, S. Sural, and A. K Majumbar, “Use of Dempster-Shafer theory and Bayesian inferencing for fraud detection in communication networks”, Lecture Notes in Computer Science,

Spring Berlin/ Heidelberg, Vol. 4586, , 2007, p.446-460.

[12] P. Burge and J. Shawe-Taylor, “Detecting cellular fraud using adaptive prototypes”, Proceedings of the AAAI-97 Workshop and AI Approaches to Fault Detection and Risk Management. Mento Park,

CA: AAAI Press, 1997, pp. 9-13.
[13] Y. Wang, H. Yang, X. Wang, and R. Zhang, “Distributed Intrusion detections Based on data fusion method.”, in Proceedings of the 5th World Congress on Intelligent Control and Automation, 2004, pp.

4331–4334.

[14] Q. Chen and U. Aickelin, “Anomaly detection using the DempsterShafer method,” in Proc. of the 2006 International Conference on Data Mining, DMIN 2006, 2006, pp. 232–240.