European Research Studies Journal
Volume XXVII, Issue 2, 2024

pp. 571-585

Credit Card Fraud Detection Using Machine Learning Techniques

Submitted 08/09/24, 1st revision 20/10/24, 2nd revision 06/11/24, accepted 10/12/24

Michał Gostkowski[1], Andrzej Krasnodębski[2], Arkadiusz Niedziółka[3]

Abstract:

Purpose: *The rapid growth of credit fraud data and credit card fraud detection is now a challenge for machine learning algorithms. Financial fraud is increasing significantly, causing losses of billions of dollars worldwide every year. In the paper the selected techniques (artificial neural networks, decision trees and random forests) were adopted and used for credit card fraud detection.*
Design/Methodology/Approach: *Due to the large class imbalance with fraud detection datasets, the class imbalance problem and methods for preprocessing class-imbalanced datasets are presented. ML models were applied for the SMOTE dataset and compared using the F1-Score measure.*
Findings: *In data preparation step four approaches were considered (SMOTE, Oversampling, Undersampling, Original dataset). The F1-Score showed that SMOTE approach gives the highest value in comparison to other approaches.*
Practical Implications: *The approach presented in the paper can be used by financial institutions to develop the system to minimize their losses and minimize the credit card risk.*
Originality/Value: *The findings presented in the paper showed that SMOTE approach can be interesting alternative to under sampling and oversampling in data preparation step. Moreover, the comparison of the selected statistical methods showed that the random forests algorithm gives the highest accuracy.*

Keywords: *Credit fraud card, machine learning, decision trees, random forests, artificial neural networks, SMOTE.*

JEL codes: *G20, C50.*

Paper type: *Research article.*

[1]*Department of Econometrics and Statistics, Warsaw University of Life Sciences – SGGW, Poland. ORCID ID: 0000-0003-3606-1182, michal_gostkowski@sggw.edu.pl;*
[2]*Faculty of Agriculture and Economics, University of Agriculture in Krakow, Poland. ORCID ID: 0000-0001-8970-0916, andrzej.krasnodebski@urk.edu.pl;*
[3]*Faculty of Agriculture and Economics, University of Agriculture in Krakow, Poland. ORCID ID: 0000-0003-2546-4154, arkadiusz.niedziolka@urk.edu.pl;*

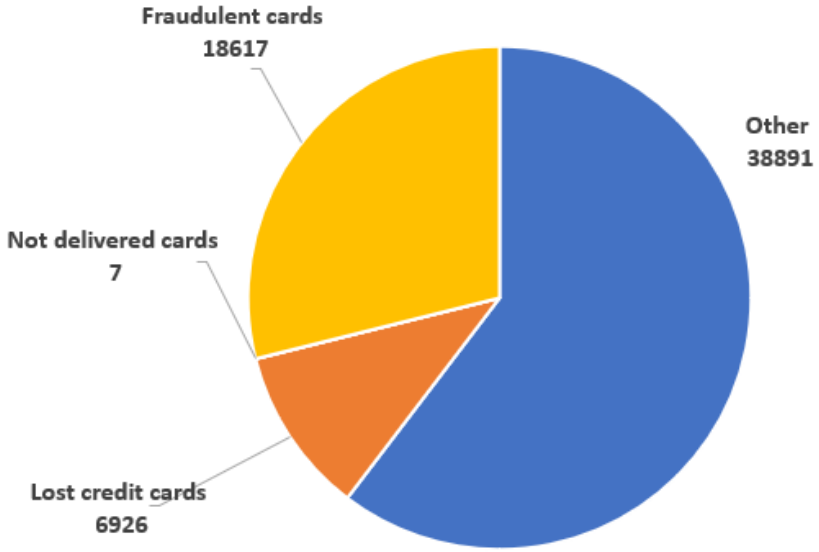
1. Introduction

A fraud transaction can be described as a deliberate fraud that is committed for some kind of profit, usually monetary. This is a dishonest and illegal practice that has been on the rise in recent times. There is a sharp increase in the use of electronic payment methods such as credit and debit cards, which in turn leads to an increase in credit card fraud. These cards can be used both online and offline to make payments.

In the case of online payments, the card does not have to be physically presented. In such cases, the card details are vulnerable to hackers or cybercriminals. Millions of dollars are lost each year as a result of this type of fraud. To overcome this obstacle, many Machine Learning (ML) algorithms have been developed and are being developed. Various detection methods are being developed to tackle fraud and theft as effectively as possible (Khatri *et al.*, 2020).

In the report published by the National Bank of Poland "Information on fraudulent transactions made using non-cash payment instruments in the third quarter of 2020", among others data on fraudulent operations made with payment cards provided by banks were presented. This data relates to fraud committed within the country and abroad with cards issued by the included banks. According to the report presented by NBP, in the third quarter of 2020, 64,441 fraudulent operations were carried out in the amount of 3,073,647 Euro.

Figure 1. *Number of fraudulent operations with payment cards by number in the third quarter of 2020.*



Source: *Own elaboration based on NBP data*.

The type of fraud that dominated among all transactions, according to the data provided by banks in the NBP report, was classified as Other. This category mainly included information about transactions made with the use of card data without its physical presence. Such transactions accounted for 60.4% of all frauds and were recorded in 38,891 cases for the amount of over 2 million Euro.

The average transaction value in this category was close to 50 Euro. The second place is occupied by frauds with counterfeit cards, which accounted for 28.9% of all frauds with an average value of 40 Euro. In the third quarter of 2020, 18,617 transactions were recorded, and the total amount of such transactions exceeded 710,000 Euro.

Another type of fraudulent transaction reported by banks were frauds with lost/stolen cards, which accounted for 10.7% of all frauds. Frauds in this category had an average amount of 58 Euro. They were recorded in 6,926 cases, while their total amount was almost 500k Euro. On average, a single fraudulent transaction using non-cash payment instruments in Q3 2020 ranged between 40-60 Euro, depending on the transaction category.

Machine learning methods for fraud detection fall into two categories: supervised and unsupervised (Gostkowski *et al.,* 2021). Supervised methods rely on estimation based on samples of fraudulent and legitimate transactions to classify new transactions as fraudulent or legitimate. In the unsupervised method, i.e. the learning method without the true category, outliers or unusual transactions are identified as potential cases of fraudulent transactions. Both of these fraud detection methods predict the likelihood of a transaction being fraudulent (Bhattacharyya *et al.,* 2011).

There are two main challenges when using supervised learning methods to detect card fraud. The first is the unbalanced size of the classes of legitimate and fraudulent transactions. In order to develop the model, some form of sampling between the two classes is usually used to obtain training data with the correct class distributions.

Various sampling approaches have been proposed in the literature. The most commonly used approaches are random oversampling of minority class cases and random undersampling of majority class cases. A second problem in developing supervised fraud models can arise from potentially undetected fraudulent transactions, leading to mislabeling of cases in the data to be used to build the model (Bhattacharyya *et al.,* 2011).

2. Materials and Methods

The paper uses data collected and analyzed as part of the Worldline and Machine Learning Group research collaboration with ULB (Université Libre de Bruxelles - Free University of Brussels) in the field of big data mining and fraud detection. The data set consists of 284,807 transactions. Among them, 492 transactions are frauds

(fraud transactions). All transactions were made using credit cards by European cardholders.

The variables in the dataset used in this paper are as follows:

- Class – target variable, takes the value 1 in case of fraud (card fraud) or value 0, when fraud (card fraud) did not take place.
- Amount – transaction amount.
- V1, V2, …, V28 – variables that have already been developed by PCA (principal component analysis)

Unbalanced datasets

Many machine learning-based classification algorithms assume that target classes have similar misclassification probabilities and costs. However, very often real datasets do not exhibit this property. A classification problem when one of the classes has a much lower probability in the training set is called an unbalanced data set problem. One popular approach to solving an unbalanced dataset is to reprocess the training data.

Data sets with unbalanced classes are typically used to detect certain anomalies, such as cancer detection, oil spill detection, network intrusion detection, fraud detection. The magnitude of the class imbalance varies from problem to problem. In the case of intrusion and fraud detection, it is not uncommon for less than ten percent of the records to represent actual intrusions and fraud. In the detection of cancer cells, typically less than one percent of the cells are actually cancerous (Liu, 2004).

Traditionally, the methods used to deal with class imbalances are based on duplicating or eliminating samples until an equilibrium is reached, for example Random Over-Sampling (ROS) and Random Under-Sampling (RUS). One of the another commonly used methods is the SMOTE (Synthetic Minority Over-sampling Technique) method, which generates new synthetic cases (Viloria *et al.,* 2020).

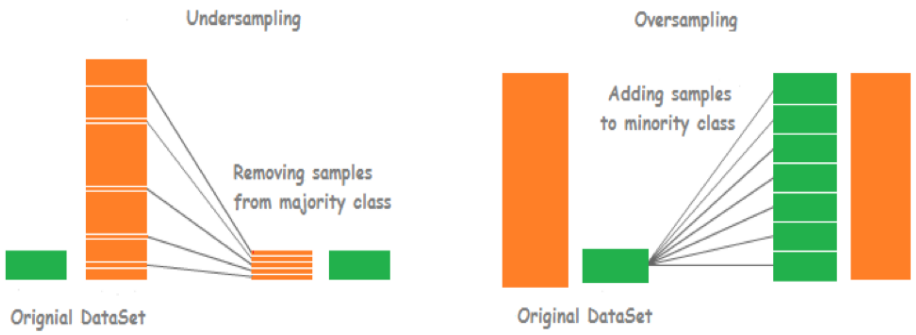
Random Oversampling

Oversampling can be done by increasing the number of minority class instances or samples to create new instances or repeat some instances (Mohammed *et al.,* 2020). Random oversampling increases the number of minority class data in the training set by randomly replicating existing minority class cases. Although a simplistic method, random oversampling performs well in empirical research, even when compared to other more complicated oversampling methods. Unfortunately, since random oversampling only replicates existing data instances, it is argued that random oversampling does not add any useful data to the training set (Liu *et al.,* 2007).

Random Undersampling

Random undersampling is a opposite approach to resampling. The majority class in the training set is randomly eliminated until the ratio between the minority and the majority class reaches the desired level until equilibrium is reached in the data set. Theoretically, one of the problems with random sampling is that you can't control what majority class information is discarded. In particular, very important information about the decision boundary between the minority class and the majority class can be eliminated (Liu, 2004).

Figure 2. *Comparison of random under sampling and random oversampling methods.*



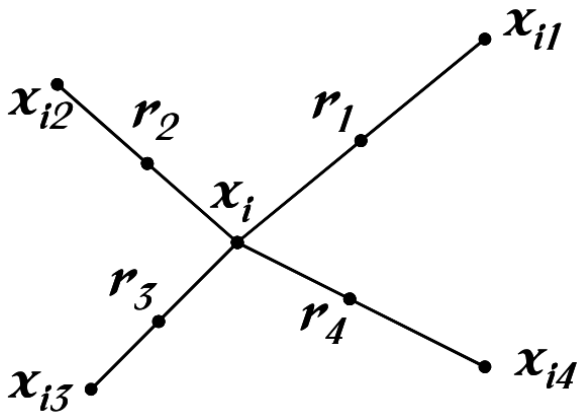
Source: *Mohammed et al., 2020.*

In an unbalanced dataset, it is often realistic to assume that many of the majority-class observations are redundant and that removing some of them at random will not significantly alter the distribution of the data. However, the risk of deleting significant observations from the dataset still exists as the deletion is unsupervised (Dal Pozzolo *et al.*, 2015).

SMOTE

The SMOTE (Synthetic Minority Oversampling Technique) algorithm is another method used for learning from unbalanced data sets. In 2002, Chawla, Bowyer, Hall and Kegelmeyer (2002) proposed a new approach as an alternative to standard random oversampling. The basis of this pre-processing technique was the creation of new minority instances (Fernandez *et al.*, 2018). The SMOTE algorithm implements an oversampling approach to balance the original training set. Instead of using simple replication of minority class cases, the key idea of SMOTE is to introduce synthetic examples (Figure 3).

Figure 3. *A simple diagram of the SMOTE method.*



Source: *Fernandez et al., 2018.*

The SMOTE algorithm can be represented as follows. First, the total amount of oversampling N (integer value) is determined, which can be set to approximate a 1:1 class distribution. Then iteration is carried out, consisting of several steps. First, a minority class instance is randomly selected from the training set and *K* of its nearest neighbors (default 5) (Fernandez *et al.,* 2018).

Artificially created instances are based on a computed neighborhood from which one neighbor is randomly selected for a new object. Each new instance is created by adding to the original object the calculated difference between a randomly selected neighbor and the source instance, which is additionally multiplied by a randomly selected value from the range (0,1). This allows you to control the final location of the artificial instance.

This increases the diversity of the artificial instances set, allowing for better use of the given decision space (Skryomski and Krawczyk, 2017). It is worth mentioning that SMOTE is characterized by significant computational complexity and memory requirements, which become visible when operating on large scales of unbalanced data (Krawczyk, 2016).

Error matrix

For the case of binary classification, there are four possible types of results. If the case is positive and is classified as positive, it is counted as true positive (TP); if it is classified as negative, it is counted as false negative (FN - false negative). If the case is negative and is classified as negative, it is recognized as true negative (TN); if it is classified as positive, it is recognized as false positive (FN - false negative).

For a given classifier and test set, an error matrix can be constructed representing the distributions of the test set. This matrix is the basis for many commonly used metrics (Powers, 2020).

Table 1. *Error matrix.*

	True Class	
Predicted Class	**T**rue **P**ositives	**F**alse **P**ositives
	False **N**egatives	**T**rue **N**egatives

Source: *Own elaboration.*

Sensitivity is the ratio of true positives (TP) to total true positives (TP) and false negatives (FN):

$$\text{Sensitivity} = \text{True Positive Rate (TPR)} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{1}$$

Specificity is the ratio of true negative (TN) results to total true negative (TN) and false positive (FP) cases:

$$\text{Specificity} = \text{True Negative Rate (TNR)} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{2}$$

Precision is the ratio of true positives (TP) to the sum of true positives (TP) and false positives (FP):

$$\text{Precision} = \text{Positive Predictive Value (PPV)} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{3}$$

The F1-Score measure represents the harmonic mean of precision and sensitivity. Its maximum score of 1 represents excellent precision and sensitivity, and a score of 0 represents the worst precision and sensitivity (Al-Antari et al., 2018):

$$F1 - Score = \frac{2}{{}^{1}/_{Precision} + {}^{1}/_{Recall}} \tag{4}$$

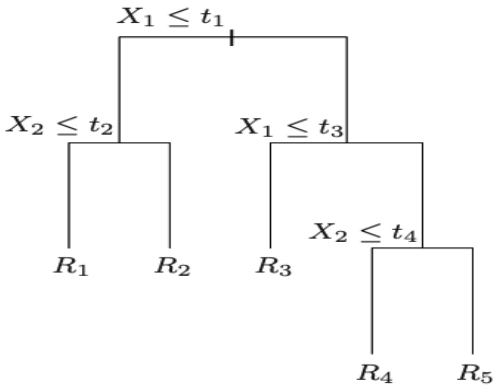
Decision trees

Decision trees were used for the binary classification problem. Tree-based methods divide the feature space into a set of rectangles and then fit a simple model to each of them (Hastie *et al.*, 2017). The model can be represented with a binary tree (Figure 5).

The complete dataset is at the top of the tree. Observations that meet the condition at each node are assigned to the left branch, and the others to the right branch. The terminal nodes or leaves of the tree correspond to the areas R1, R2, R3, R4, R5.

The key advantage of a recursive binary tree is its ease of interpretation. Decision trees can be applied to both regression and classification problems. The application of the decision tree method to these two problems differs only in the method of calculating the measure of inhomogeneity in the nodes of the tree. For the regression problem, the sum of squares (SS) method is used. For the classification problem, the following methods can be used: Gini index, cross-entropy, misclassification error.

Figure 5. *Binary tree model.*



Source: *Hastie, T., Tibshirani, R., & Friedman, J. (2017) The elements of statistical learning: data mining, inference, and prediction, Springer.*

Artificial neural networks

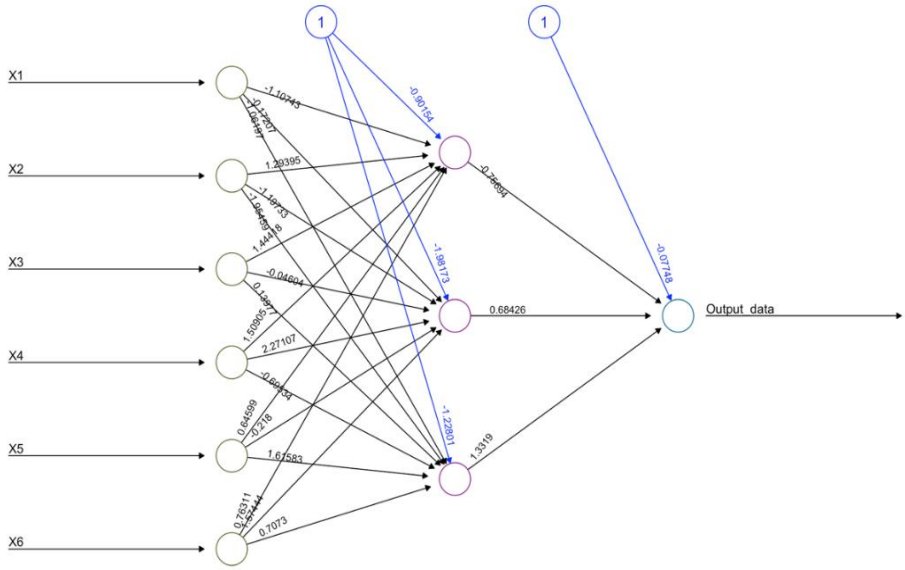
The name of artificial neural networks comes from the fact that they were built on the model of the human brain (Gajowniczek *et al.*, 2020). Each unit in the model is called a neuron. The connections between these units in the model correspond to synapses – connections in the human nervous system (Walczak, 2018). The main idea of neural networks is to extract linear combinations of inputs as derived features and then model the objective variable as a non-linear function of these features.

It is important to understand the terminology used when discussing the ANN architecture. An example ANN architecture for a supervised multi-layer learning perceptron is shown in Figure 6.

Each neuron is connected to all subsequent neurons in the next layer, with the input neurons connected to the neurons of the hidden layer, and so on until the neurons in the last hidden layer are connected to the neurons of the output layer. All these combinations have a value, called weights, which is adjusted to best match the training data (learning) (Walczak, 2018).

With too few hidden units, the model may not be flexible enough to capture non-linearities in the data. With too many hidden units, the extra weights can be reduced to zero if appropriate regularization is applied. Typically, the number of hidden units ranges from 5 to 100, with the number increasing with the number of inputs and the number of training cases.

Figure 6. *An example diagram of an artificial neural network with one hidden layer.*



Source: *Own elaboration.*

Random forests

Random forest (RF) is an ensemble method used to predict and improve the accuracy of results for regression and classification problems (Gostkowski and Gajowniczek, 2020). Random forest is a method that consists of multiple decision trees. Compared to other traditional classification algorithms, random forests have low classification error (Farnaaz and Jabbar, 2016).

The random forest algorithm for regression and classification is as follows (Hastie *et al.*, 2017):

Training set: $X = = x_1, \ldots x_n$, with a target variable $Y = y_1, \ldots y_n$

1. For $b = 1, \ldots, B$:
    a. Select a random sample of N cases retrieved with replacement (bootstrap sample) from the training set.
    b. T_b tree is trained by recursively repeating the following steps for each node in the tree until the minimum node size n_{min} is reached.

i. *m* variables should be chosen at random.
ii. The best split point is found by randomly selecting *m* out of *p* attributes ($m \leq p$). *m* parameter is usually determined as follows: $\mathrm{m} = \sqrt{p}$ for classification or $\mathrm{m} = \lfloor p/3 \rfloor$ for regression.
iii. The node is divided into 2 nodes.

2. The result of the algorithm is a set of trees$\{\mathrm{Tb}\}_1^B$.

3. Prediction at a new point *x* is calculated on the basis of the formula:

   a. For regression:

$$f^{\wedge} = \frac{1}{B}\sum_{b=1}^{B} T_b(x) \tag{5}$$

   b. For classification:

$$C^{\wedge B}_{rf}(x) = majority\ vote\ \{C^{\wedge}_b(\mathrm{x})\}_1^B \tag{6}$$

The great advantage of random forests is that they are not overfitted to the training data. Compared to decision trees, especially very deep trees, which tend to overfitting the training data (Hastie *et al.*, 2017). As such, random forests are an extremely popular method of machine learning and are implemented in many packages.

3. Results

In the first step, the data set was randomly divided into a training set (70%) and a test set (30%). Then, in order to balance the data, the following methods were used: SMOTE, random oversampling and random undersampling.

Initially, there were 284,807 observations in the source set, of which only 492 observations were fraud transactions (Class variable equal to 1), which accounted for 0.173% of all observations. The training set consisted of 320 fraud transactions, and the test set of 172.

The SMOTE method was used on the training set, where the *k* parameter was 5. Then, a new set was created based on the source data, where the random oversampling method was used. To create the last set, the method of random undersampling of the majority class was used.

For each of the sets, ML algorithms were used: decision tree, random forest and artificial neural network. As a result, 12 models were obtained, the results of which, based on the F1-Score measure, are presented in Table 2.

Table 2. *Comparison of the F1-Score measure*.

ML algorithm	Method	F1-Score
Decision tree	SMOTE	0.9894
	Random Oversampling	0.9758
	Random Undersampling	0.9690
	Source data	0.9996
Random forest	SMOTE	0.9997
	Random Oversampling	0.9997
	Random Undersampling	0.9855
	Source data	NA
Artificial neural network	SMOTE	0.9968
	Random Oversampling	0.9626
	Random Undersampling	0.9905
	Source data	NA

Source: *Own elaboration.*

The models were compared based on the F1-Score measure. Although for the decision tree method for the source set, the F1-Score measure showed the best results, for the random forest and artificial neural network, the models did not build due to too much unbalance between the classes. For the decision tree the set obtained with the SMOTE method received a very high F1-Score of almost 99%.

Because for the random forest and artificial neural network, the models for SMOTE also showed the most accurate results, this set was selected for further analysis. Further models were created based on the training set processed with SMOTE algorithm. The accuracy of prediction was tested on the test set using the F1-Score measure.

The first method that was undertaken in the work is the method of decision trees, where the optimized parameter was the CP (complexity parameter). The final value of this parameter was set at 0.001. Out of 91,874 test cases, the classifier correctly classified 90,456 negative cases and 148 positive cases (credit card fraud) (Table 3).

Table 3. *The error matrix for a decision tree with a CP parameter = 0,001*

Predicted class		True class	
		0	1
	0	90456	24
	1	1246	148

Source: *Own elaboration.*

The second method was the random forest method. In order to find the best model, the *nTree* value was optimized, which indicated the number of decision trees forming a given random forest. The final value of the parameter is 100. The results for the test set are presented below (Table 4).

Table 2. *The error matrix for a random forest with an nTree parameter = 100.*

Predicted class		True class	
		0	1
	0	91689	32
	1	13	140

Source: *Own elaboration.*

The last method was artificial neural networks, where the optimized parameter was the number of neurons in the hidden layer. On the basis of the training set, a set of models was built, where the number of neurons in the hidden layer increased by one. Finally, a model was selected that contained 30 neurons in the hidden layer (Table 5).

Table 5. *Error matrix for an artificial neural network with a h parameter = 30.*

Predicted class		True class	
		0	1
	0	91651	37
	1	51	135

Source: *Own elaboration.*

Preliminary analysis of the dataset enabled the study of the impact of methods used to overcome the problem of unbalanced classes in the dataset. For the problem of detecting card frauds and the available data set, the SMOTE method turned out to be a satisfactory data processing technique.

Among the built decision tree models, the model with a complexity level of 0.001 and an F1-Score of about 99.3% turned out to be the best model. The best models for random forests turned out to be models with the number of decision trees equal to 100. The F1-Score was 99.97%.

In turn, the analysis of the F1-Score results of the ANN models allowed to conclude that the best result was obtained by the model with the largest number of neurons in the hidden layer, i.e. the model with thirty hidden neurons. The F1-Score measure was 99.95%, which proves very good classification results.

Looking at the three selected models, based on the error matrix, it can be seen that e.g. they differ in the number of incorrectly predicted results. Based on decision trees, the model incorrectly classified 24 fraud transactions as legitimate transactions. The random forest model did not detect fraud in 32 cases and the ANN model in 37 cases. If you look at the number of cases that the models classified as

fraudulent transactions that were in fact legitimate transactions, the distribution goes the other way. For decision trees it was a much larger number of cases than for the other two models. For DT it was 1246 cases, while for artificial neural networks it was 51 cases. This means that for the ANN there were about 24 times less falsely classified negative cases than for DT.

On the other hand, for random forests it was only 13 cases, which in turn gives almost 4 times less than in the case of ANN. Although, taking into account the FN values alone, the random forests perform the worst, the FP value significantly determines that the random forests perform best among the presented ML models.

Based on the analysis of the models and their results, it can be indicated that the best ML method that detected card frauds were random forests.

4. Conclusion

With the development of modern technologies, the number of financial frauds is significantly increasing, which leads to financial loss all over the world every year. Fraudulent transactions are scattered across all transactions, and simple pattern-matching techniques are often not enough to accurately detect these frauds. Implementing effective fraud detection systems has therefore become a must for all credit card issuing banks to minimize their losses (Tripathi and Pavaskar, 2012).

In this paper, the techniques of decision trees, artificial neural networks and random forests were used to detect credit card fraud. The models built on the basis of these techniques, after selecting the appropriate parameters, showed a high level of accuracy. This proves that machine learning methods can detect fraud transactions with high accuracy. This can be significant for many companies and institutions that are vulnerable to fraud and abuse. ML-based systems can be a tool to prevent fraud.

The research included in the paper shows that the suggested method for fraud detection would be a classifier based on random forests, because for the dataset described in the paper, it gave the best results. Researchers from Southwest Jiaotong University (Liu *et al.*, 2015) in their work showed the superiority of random forests over other methods, e.g., logistic regression, the k-nearest neighbors method, decision trees, or a support vector machine for financial fraud problems.

Lakshmi and Kavilla (2018) in their work on the credit card fraud detection problem compared random forests with logistic regression and decision trees. Based on the measure of sensitivity, specificity, quality and error rate, they indicated a random forest as classifiers with the best predictive results.

Additionally, the SMOTE method was used in the paper. However, it should be emphasized that SMOTE will not be an appropriate data processing method for all algorithms and may lower the prediction results, e.g., for logistic regression or the

SVM algorithm (Ishaq *et al.,* 2021). Empirical results show that training different types of classifiers using SMOTE oversampled data leads to better classification results than training with unmodified, unbalanced data (Douzas and Bacao 2018).

References:

Al-Antari, M.A., Al-Masni, M.A., Choi, M.T., Han, S.M., Kim, T.S. 2018. A fully integrated computer-aided diagnosis system for digital X-ray mammograms via deep learning detection, segmentation, and classification. International journal of medical informatics, 117, 44-54. https://doi.org/10.1016/j.ijmedinf.2018.06.003.

Bhattacharyya, S., Jha, S., Tharakunnel, K., Westland, J.C. 2011. Data mining for credit card fraud: A comparative study. Decision support systems, 50(3), 602-613. https://doi.org/10.1016/j.dss.2010.08.008.

Dal Pozzolo, A., Caelen, O., Johnson, R.A., Bontempi, G. 2015. Calibrating probability with undersampling for unbalanced classification. In: 2015 IEEE Symposium Series on Computational Intelligence (pp. 159-166). DOI: 10.1109/SSCI.2015.33.

Douzas, G., Bacao, F., Last, F. 2018. Improving imbalanced learning through a heuristic oversampling method based on k-means and SMOTE. Information Sciences, 465, 1-20. DOI 10.1016/j.ins.2018.06.056.

Farnaaz, N., Jabbar, M.A. 2016. Random forest modeling for network intrusion detection system. Procedia Computer Science, 89, 213-217. https://doi.org/10.1016/j.procs.2016.06.047.

Fernández, A., Garcia, S., Herrera, F., Chawla, N.V. 2018. SMOTE for learning from imbalanced data: progress and challenges, marking the 15-year anniversary. Journal of artificial intelligence research, 61, 863-905. https://doi.org/10.1613/jair.1.11192.

Gajowniczek, K., Grzegorczyk, I., Gostkowski, M., Ząbkowski, T. 2020. Blind Source Separation for the Aggregation of Machine Learning Algorithms: An Arrhythmia Classification Case. Electronics, 9(3), 1-14. http://doi.org/10.3390/electronics9030425.

Gostkowski, M., Rokicki, T., Ochnio, L., Koszela, G., Wojtczuk, K., Ratajczak, M., Bełdycka-Bórawska, A. 2021. Clustering Analysis of Energy Consumption in the Countries of the Visegrad Group. Energies, 14(18), 1-24. http://doi.org/10.3390/en14185612.

Gostkowski, M., Gajowniczek, K. 2020. Weighted Quantile Regression Forests for Bimodal Distribution Modeling: A Loss Given Default Case. Entropy, 22(5), 545-555. http://doi.org/10.3390/e22050545.

Ishaq, A., Sadiq, S., Umer, M., Ullah, S., Mirjalili, S., Rupapara, V., Nappi, M. 2021. Improving the prediction of heart failure patients' survival using SMOTE and effective data mining techniques. IEEE access, 9, 39707-39716. DOI 10.1109/ACCESS.2021.3064084.

Khatri, S., Arora, A., Agrawal, A.P. 2020. Supervised machine learning algorithms for credit card fraud detection: a comparison. In: 2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence) (pp. 680-683). DOI: 10.1109/Confluence47617.2020.9057851.

Krawczyk, B. 2016. Learning from imbalanced data: open challenges and future directions. Progress in Artificial Intelligence, 5(4), 221-232. DOI 10.1007/s13748-016-0094-0.

Lakshmi, S.V.S.S., Kavilla, S.D. 2018. Machine learning for credit card fraud detection system. International Journal of Applied Engineering Research, 13(24), 16819-16824.

Liu, A.Y.C. 2004. The effect of oversampling and undersampling on classifying imbalanced text datasets. Doctoral dissertation, University of Texas at Austin.

Liu, A., Ghosh, J., Martin, C.E. 2007. Generative Oversampling for Mining Imbalanced Datasets. DMIN, pp. 66-72.

Liu, C., Chan, Y., Alam Kazmi, S.H., Fu, H. 2015. Financial fraud detection model: Based on random forest. International journal of economics and finance, 7(7). doi:10.5539/ijef.v7n7p178.

Mohammed, R., Rawashdeh, J., Abdullah, M. 2020. Machine learning with oversampling and undersampling techniques: overview study and experimental results. In: 2020 11th international conference on information and communication systems (ICICS) (pp. 243-248). IEEE. DOI: 10.1109/ICICS49469.2020.239556.

Powers, D.M. 2020. Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. arXiv preprint arXiv:2010.16061.

Skryjomski, P., Krawczyk, B. 2017. Influence of minority class instance types on SMOTE imbalanced data oversampling. Proceedings of Machine Learning Research, 74, 7–21.

Tripathi, K.K., Pavaskar, M.A. 2012. Survey on credit card fraud detection methods. International Journal of Emerging Technology and Advanced Engineering, 2(11), 721-726.

Walczak, S. 2018. Artificial neural networks. Encyclopedia of Information Science and Technology, Fourth Edition, pp. 120-131. IGI Global. DOI: 10.4018/978-1-5225-2255-3.ch011.

Viloria, A., Lezama, O.B.P., Mercado-Caruzo, N. 2020. Unbalanced data processing using oversampling: Machine Learning. Procedia Computer Science, 175, 108-113. https://doi.org/10.1016/j.procs.2020.07.018.