STRUČNI RAD UDK 004.8

Credit Card Fraud Detection Using Machine Learning Algorithms

Ivan Lorencin[1], Nikola Anđelić[2], Deni Vale[3], Marko Mavrinac[4]

[1] dr. sc.; Istarsko veleučilište, Riva 6, Pula, Hrvatska

[2] dr. sc. Sveučilište u Rijeci – Tehnički fakultet, Vukovarska , Rijeka, Hrvatska

[3] mag. phys.; Istarsko veleučilište, Riva 6, Pula, Hrvatska

[4] mag. ing. el. Elektroindustrijska i obrtnička škola Rijeka, Zvonimirova 12, Rijeka, Hrvatska

Abstract

One of the main challenges to the security of an online business is credit card fraud. For this reason, algorithms based on artificial intelligence and machine learning are being introduced to enable the most accurate and fast detection of card fraud. This paper presents an approach to the detection of card fraud based on machine learning algorithms more specifically, a multilayer perceptron (MLP) and a Decision tree. The aforementioned algorithms were trained and tested using a publicly available data set on card fraud. The data set used consists of 7 characteristics of the card transaction and information on whether there was card fraud or not. In total, the data set contains information on 1,000,000 transactions, and it is highly imbalanced. To handle the class imbalance, random under-sampling, SMOTE, and SMOTE-Tomek algorithms were proposed. From the achieved results it can be seen that the highest performances are achieved if MLP (AUC = 0.99, f1 = 0.99, MCC = 0.98, and Kappa = 0.98) and Decision tree (AUC = 0.99, f1 = 0.99, MCC = 0.99, and Kappa = 0.98) are trained by using data set re-sampled by using SMOTE-Tomek algorithm. If the performance of the mentioned algorithms is examined using fewer characteristics of the transaction, it can be seen that by reducing the number of characteristics a significant decrease in classification performances can be noticed if a Decision tree in combination with SMOTE-Tomek is used. However, if an MLP in combination with SMOTE-Tomek is used, a significantly lower decrease in performance can be observed, pointing to the higher robustness to input vector dimension reduction. Such a robust system can provide information about transaction validity even in a condition where the input data is limited to a few input variables. From the achieved results, it can be concluded that MLP in combination with the SMOTE-Tomek algorithm can be used for credit card fraud detection, even in conditions with a lower number of input variables.

1. Introduction

Online trading has taken a large share in worldwide trade in recent years (Rouf et al., 2021). More and more different products and services can be bought and sold online (Melović et al., 2021). Such a trend reached its peak in the past years due to the COVID-19 pandemic and reduced physical contact (Wynn and Olayinka, 2021). For the above reasons, safe and verified trade is imperative for the exchange of goods and services to be carried out smoothly and without major delays (Kim et al., 2022). One of the main challenges for the security of an online business is certainly card fraud, and the timely detection of fraud represents a significant saving of resources and time (Arora et al., 2022). For this reason, algorithms based on artificial intelligence (AI) and machine learning (ML) are being introduced to enable the most accurate and fast detection of card fraud (Alarfaj et al., 2022). Utilization of AI and ML has an application in the security area, ranging from protection of (Internet of Things) IoT systems and computer networks (Kuzlu et al., 2021, Mattos et al., 2020, Qiu et al., 2019), over financial systems (Melnychenko, 2020, Bredt, 2019) to video surveillance (Nguyen et al., 2020, Lorencin et al., 2019). Similar techniques can be used in credit card fraud detection. The authors in (Shanthakumara et al., 2022) have proposed the utilization of XGBoost and Random Forest Classifier to detect credit card frauds by using data about fraudulent transactions. The described approach has resulted in F1 score ranging from 0.81, for the case of XGBoost to 0.87 for the case of Random Forest Classifier. Sudha and Akila in (Sudha and Akila, 2021) have proposed a method based on a majority voting-based ensemble (MVE) classifier. In this case, the F1 score up to 0.9 was achieved. Asha and Kumar (Asha and KR, 2021) have proposed multiple machine-learning algorithms for credit card fraud detection. If Support Vector Machine (SVM) was used, precision of 0.98 and recall of 0.90 were achieved. On the other hand, if a neural network was used precision of 0.81 and recall of 0.76 were achieved. Roseline et. al. have proposed a solution based on Long Short-Term Memory-Recurrent Neural Network (LSTM-RNN). By using this approach, high classification performances were achieved. The authors in (Carcillo et al., 2021) have proposed a combination of supervised and unsupervised learning for credit card detection. The authors have concluded that such an approach enables significant performance increase. The authors in (Chen and Lai, 2021) have proposed an approach based on the utilization of convolutional neural networks. Such an approach has resulted in an accuracy of 99%. Varmedija et. al. (Varmedja et al., 2019) have proposed an approach based on multiple algorithms: Logistic Regression, Random Forest, Naive Bayes, and Multilayer Perceptron (MLP). The authors have concluded that utilization of all aforementioned algorithms has resulted in high classification performances. Taha et. al. (Taha and Malebary, 2020) have proposed a solution based on an Optimized Light Gradient Boosting Machine. By using this approach, high accuracy bud low F1 score of only 0.57 was achieved. Fukas et al. (Fukas et al., 2022) demonstrate the application of shapely additive explanations to determine the models and individual feature importance for financial fraud detection. A downstream set of methods is applied and the results enable the description of the main factors which indicate fraudulent behaviors. Chaquet-Ulldemolins et al. (Chaquet-Ulldemolins et al., 2022) demonstrate the application of Interpretable Autoencoders (IA). These methods allow for the creation of individualized transaction ranking (ITR), which in turn allows for a higher accuracy that reaches up to 93%. Hasan and Rizvi (Hasan and Rizvi, 2022) demonstrate the application of AI-driven detection on Indian e-Commerce transaction data, during its increase within the COVID-19 pandemic period. The authors conclude that AI-driven methods show promise, especially when the amount of data processed is significantly increased. Li (Li, 2022) applies the information fusion technology (IFT) which merges several ML and data mining techniques such as Logistic regression (LR) and SVM. The research focuses on the analysis of the individual making the transaction, especially in a business-to-business (B2B) environment and shows significant promise in determining attempts at fraudulent transactions and requests. AI-JasCon is an AI-based containerization system for bayesian fraud detection in complex networks proposed by Nonum et al. (Nonum et al., 2022). The goal of the research is the preparation of an AI-based high-precision system for the detection of problematic transactions which can easily be applied

within existing pipelines for continuous integration and continuous delivery (CI/CD). Chang et al. (Chang et al., 2022) test a multitude of algorithms, including LR, k-nearest neighbors (KNN), random forest (RF), and autoencoders to determine the best-performing algorithms for fraud detection. When undersampling and feature reduction are applied the best-performing algorithms are shown to be RF and LR. Fraud detection with a multitude of methods is also performed by Navaneethakrishnan and Viswanath (Navaneethakrishnan and Viswanath, 2022). Their research includes a total of seven algorithms, the best of which are shown to be feed-forward ANNs, with accuracy and precision of 99% and fallacy rate of 0.1%. The ever-changing nature of fraudulent transactions is noted by Aschi et al. (Aschi et al., 2022), who encode the pre-processing and model training inside a batch layer. This approach allows for periodic retraining with newly collected data. Grossi et al. (Grossi et al., 2022) focus on the application of a Quantum SVM (QSVM) algorithm. The main benefit of this research is the application on quantum computer architecture via the Qiskit stack, paired with the automatic best-feature selection based on the QSVM feature-map characteristic.

From the presented literature overview it can be seen that there is no information about the comparison of different data set balancing techniques used on highly-imbalanced data sets related to credit card fraud detection. Furthermore, there is a lack of information about the performances of ML techniques for credit card fraud detection in conditions where the input data is limited to a few input variables.

According to the presented literature overview, the following questions can be asked:

- Is it possible to design an intelligent system for credit card fraud detection using transaction data and ML algorithms such as MLP and Decision tree combined with data set resampling algorithms?
- Which combination of ML algorithm and data set resampling algorithm will achieve the highest classification performances?
- How will the developed models perform in conditions with a limited number of input variables?

To summarize the novelty of this paper, in this research an approach base on two basic ML algorithms (Decision tree and MLP) is presented. To handle the imbalanced data, random under-sampling, SMOTE, and SMOTE-Tomek algorithms were used. Furthermore, on both Decision tree and MLP, a robustness test for input vector dimension reduction is performed for all data sets. The dimension reduction is performed according to the correlation of each input variable to the target. The performed investigations can provide information about the proper methodology for the utilization of highly-imbalanced data sets related to credit card fraud. Furthermore, the part of the investigation related to the examination of input dimension reduction robustness can provide the information about possibility of utilizing such algorithms in a condition where the input data is limited to a few input variables.

2. Credit card fraud detection system

The system proposed in this article consists of a classification algorithm that uses purchase data to detect credit card fraud. Such an algorithm is used to determine possible credit card fraud by using data characteristics for a single purchase. The fraud detection procedure using a classification algorithm is presented in Figure 1. According to the presented data flow, it can be seen that certain data is required to detect credit card fraud with high accuracy. In this paper, different combinations of input variables were used to determine the minimal number of input characteristics that will result in satisfying fraud detection performances.

3. Used data set

In this research, a publicly available data set on credit card frauds was used (Varmedja et al., 2019). The used data set consists of 1,000,000 data points on credit card transactions. However, the original

data set is unbalanced, where only 8.57% of samples represent fraud. The characteristics of the data set are presented in Table 1.

The first input variable corresponds to the customer's distance from his residence in kilometers. The second variable represents the distance in kilometers from the last transaction, while the third variable corresponds to the ratio of the incurred cost to the median of all transactions. Other variables represent a logical expression of Boolean algebra and provide information on whether the same transaction is repeated, whether the chip card and pin were used during the transaction, and whether payment was made for an online order.

Table 1: Description of the used data set

Input-Output	Variable name	Data type	Minimal value	Maximal Value
X_1	distance from home	Float	0.104184119	965.910612
X_2	distance from last transaction	Float	0.001448486	990.0703152
X_3	ratio to median purchase price	Float	0.016932756	41.02344879
X_4	repeat retailer	Bool	0	1
X_5	used chip	Bool	0	1
X_6	used pin number	Bool	0	1
X_7	online order	Bool	0	1
Y	Class	Bool	0	1

3.1 Resampling techniques

It can be noticed that the used data set is significantly unbalanced. For these reasons, multiple data resampling techniques were used to achieve a more balanced data set.

The application of methods to increase the number of samples is absolutely necessary because such an approach reduces the influence of imbalanced classes such as those present in the data set in question. Furthermore, this approach enables a larger number of samples for traning, which increases the performance and robustness of the developed model.

In this section, a brief description of the used data set resampling techniques will be presented.

3.1.1 Random under-sampling

Random under-sampling is performed by randomly reducing the number of the dominant class. The resulting data set consists of all members of class positive and a partition of class negative. This data set consists of 16,384 points that represent credit card fraud and 16,384 data points that represent a valid transaction. The 16,384 valid transactions were selected from the original 983,616 data points uniformly randomly, in order to represent the original data as similarly as possible, according to the law of large numbers (Abou Jaoude, 2013).

3.1.2 SMOTE

One of the used resampling techniques is Synthetic Minority Over-sampling Technique (SMOTE). SMOTE is an oversampling technique, standardly used for classification problems (Chawla et al., 2002). For a data set of n samples and X features (dimensions) in the feature space. The feature space of the minority class (positive in this case) is over-sampled by defining k-nearest neighbors of each sample contained in the minority class. The new data point in feature space is then made between the random data and the randomly selected k-nearest neighbor. The SMOTE methodology used in this research is presented in Figure 2.

The procedure is repeated until minority class has the same proportion of as the majority class.

Figure 1: The data-flow diagram of the proposed credit card fraud detection system

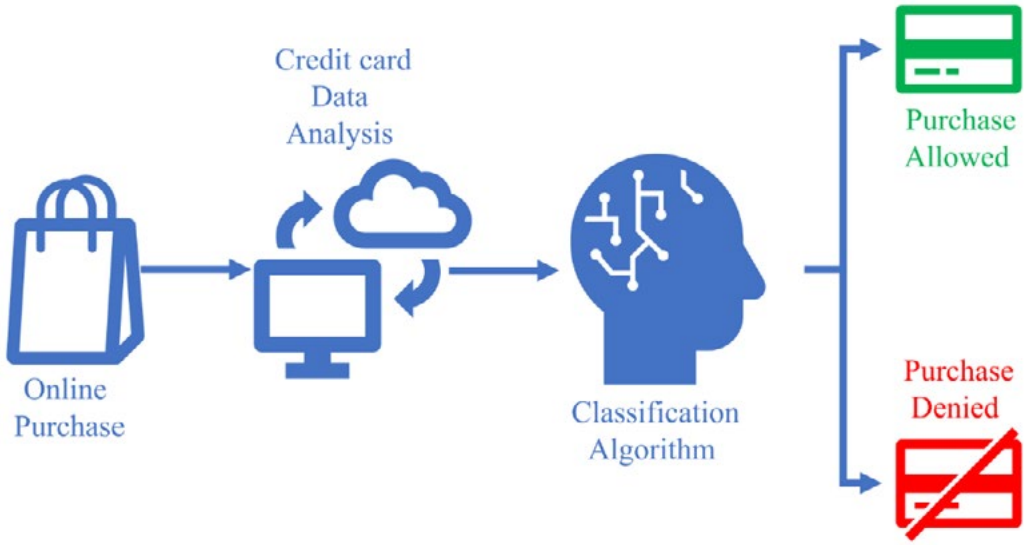
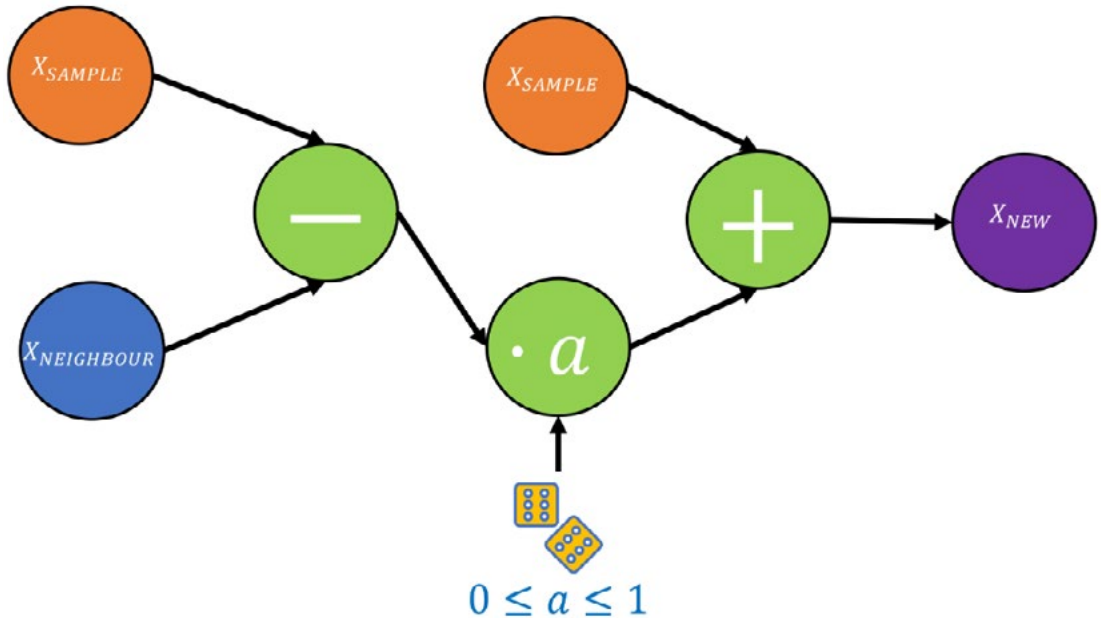


Figure 2: The graphical representation of Synthetic Minority Over-sampling Technique (SMOTE) over-sampling method



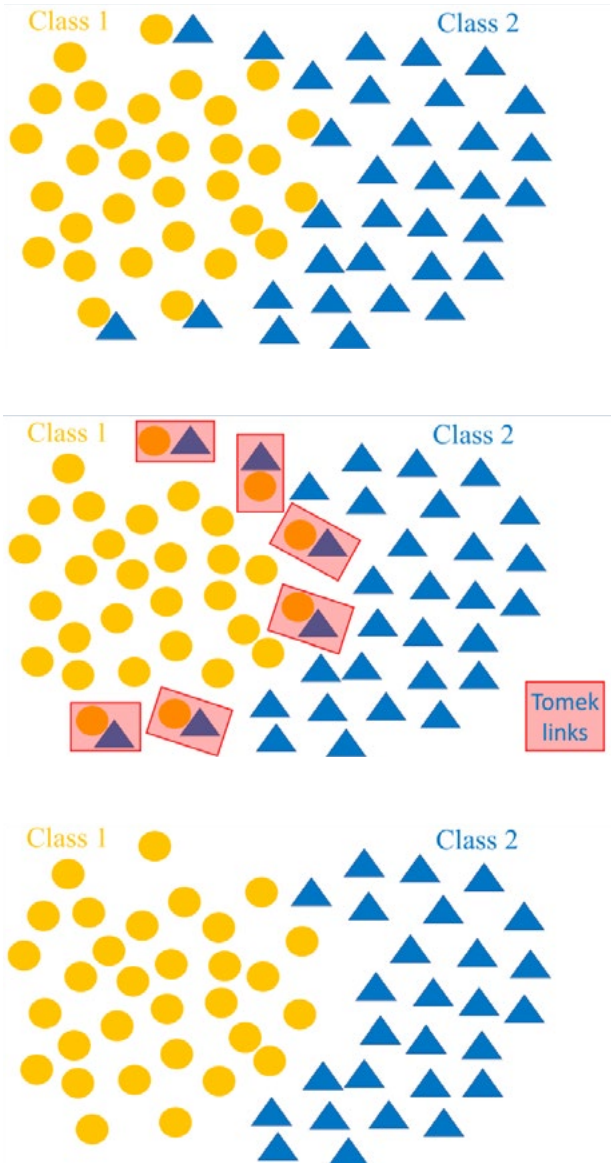
3.1.3 SMOTE-Tomek

SMOTE-Tomek algorithm represent a combination of the described SMOTE algorithm and Tomek under-sampling algorithm. Tomek algorithm is based on the elimination of Tomek links from the data set and it represents a undersampling method. Unlike the presented random under-sampling method, Tomek under-sampling method selects data pairs (X_1 and X_2) according to the following criteria:

1. X_1 is the nearest neighbour of X_2,
2. X_2 is the nearest neighbour of X_1, and
3. X_1 and X_2 are members of different classes.

The members of the majority class that are included into the Tomek links are than removed from the data set. In other words, Tomek algorithm is used to remove the majority class samples that have the lowest Euclidean distance with the minority class samples. A graphical overview of the described Tomek Algorithm is presented in Figure 3.

Figure 3: Overview of Tomek procedure (a) Initial class distribution, b) Construction of Tomek links; c) Removal of majority class samples from Tomek links)



a) Class members

c) Tomek links

c) Samples removal

The SMOTE-Tomek resampling algorithm is designed to combine SMOTE up-sampling procedure with Tomek under-sampling. By using this approach, the generated synthetic data has a higher Euclidean distance to original majority class.

3.1.4 An overview of the used data set

By using the described resampling techniques, the new data sets are designed. A brief overview of the designed data sets and their division into training, validation, and testing data sets are presented in Figure 4.

3.2 Dimension reduction

From Table 1, it can be seen that the data set consists of seven input variables (X_1, X_2, X_3, X_4, X_5, X_6, and X_7) and one output variable (Y). From Table 1, it can be seen that three input variables are float (X_1, X_2, and X_3) and four are bool variables (X_4, X_5, X_6, and X_7). The output variable (Y) is a bool variable, where 0 corresponds with a valid transaction and 1 corresponds to credit card fraud.

With the aim of reducing the number of input variables, it is necessary to determine how each of the input variables influences the output variable. In other words, it is necessary to determine which

input variables are a more important indicator for credit card fraud detection. For these reasons, correlation analysis is performed. The correlation matrix for all input variables and output variables is presented in Figure 5.

From the correlation matrix, it can be seen that X_3 has the highest correlation to the output, indicating that there is a connection between the two data variables. Variables marked with X_1, X_2, and X_7 have lower but still significant correlation factors. On the other hand, variables X_4, X_5, and X_6 have no significant influence on the output variable. For these reasons, during the procedure of input dimension reduction, these variables will be omitted. The procedure of input dimension reduction is presented in Figure 6.

Figure 4: Data set distribution

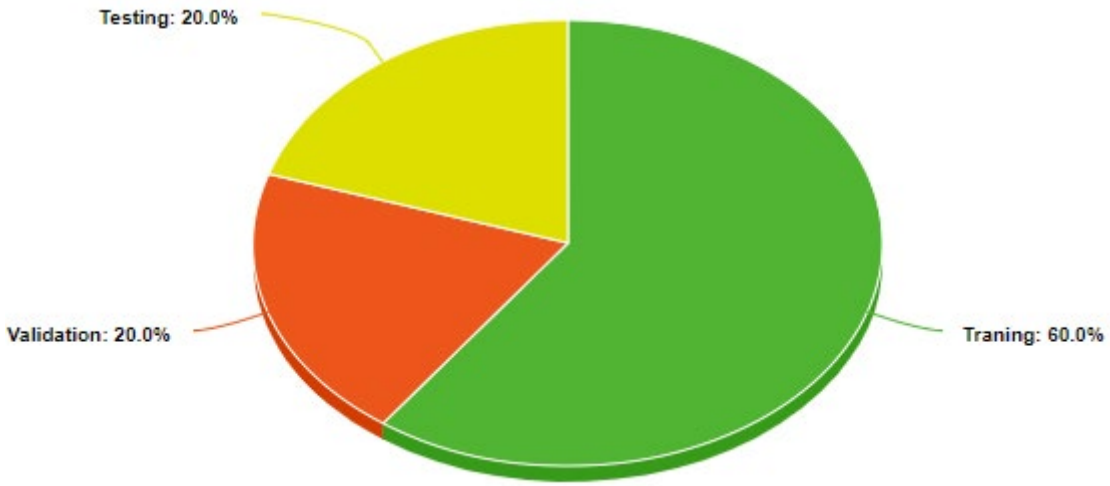


Figure 5: The Pearson correlation matrix of the used data set development of the classification algorithm.

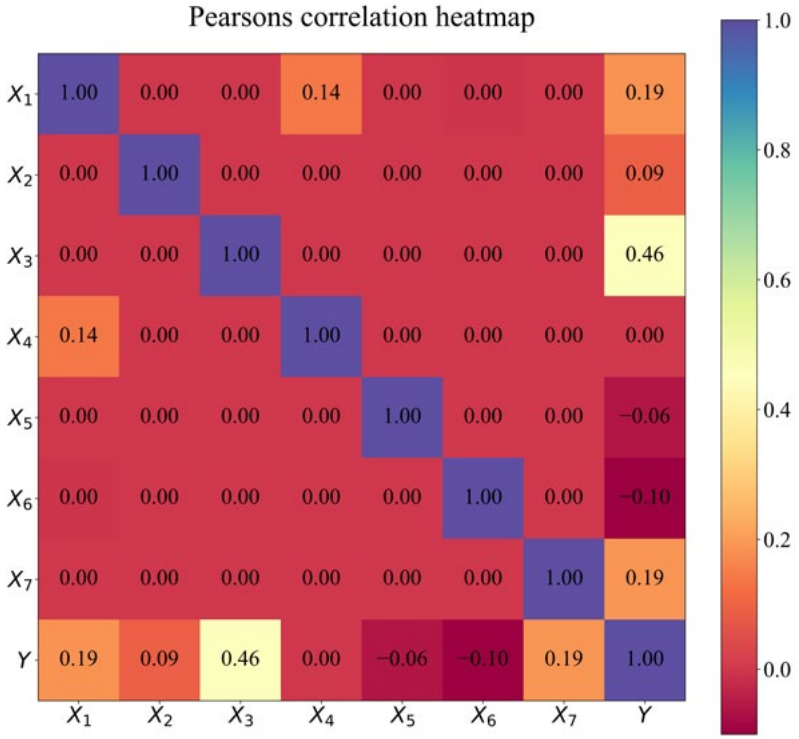
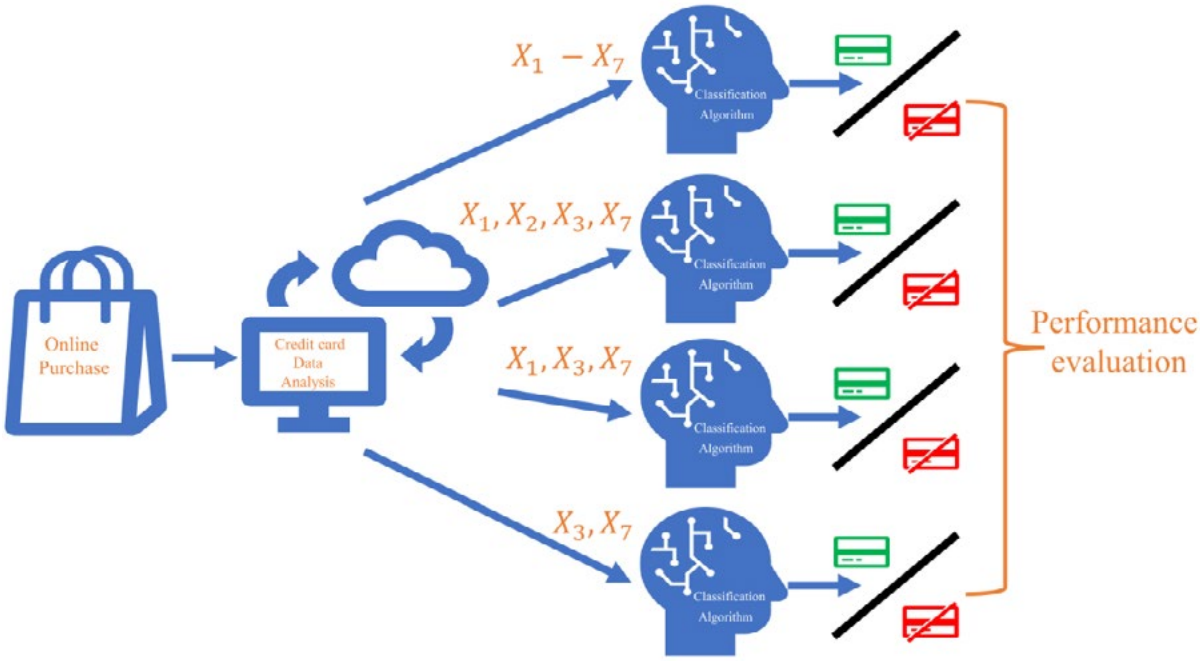


Figure 6: The process of dimension reduction for the input data



4. Used algorithms

In this section, a brief overview of the used classification algorithms is provided. In this research, two different classification algorithms were used:

- MLP and
- Decision tree.

To determine the configuration that produces the highest classification performances, a grid-search procedure is performed in the case of both algorithms. Classification performances are evaluated by using classification measures. In this case, measures based on the confusion matrix are used (Luque et al., 2019). From the confusion matrix, precision and recall are determined. By using precision and recall, F1 score is calculated. F1 score is used as a single, scalar value used to determine the classification quality (Yacouby and Axman, 2020).

4.1 Multilayer perceptron

Multilayer perceptron (MLP) represents one of the standard Artificial Neural Networks (ANN). It is successfully used in both classification and regression problems in various fields from power engineering (Khosravi and Syri, 2020, Olatunji et al., 2019, Moon et al., 2018), through medicine (Rehman et al., 2019, Lee et al., 2020, Lorencin et al., 2020, Car et al., 2020) to robotics (Segota et al., 2021, Colli-Alfaro et al., 2019). MLP is characterized by three types of layers:

- Input layer,
- Hidden layer(s), and
- Output layer.

Each of the aforementioned layers consists of at least one artificial neuron. The number of neurons in the input layer is determined by a number of dimensions in the input vector. Analogously, the number of output layers is determined as the number of output classes. The optimal number of hidden layers (and the number of neurons in them) can vary from problem to problem and it is determined by using the grid-search procedure. During the development of the MLP-based classifier used in this research, seven different configurations of hidden layers are used.

Each neuron consists of an activation function. An activation function defines the relationship between signals on the input of a neuron and the output signal of a neuron, in a similar manner as

activation potential in a physical neuron (Karlik and Olgac, 2011). In this research, three different activation functions are used.

To fit MLP layers to available training data, algorithms called solvers are used. The solver is used to minimize the training error of the network and to fit the network weights to the input and output data of the training data set. In this research, three different solvers were used.

All mentioned hyper-parameters are given in Table 2.

During the development procedure, all combinations of the described hyper-parameters are used to determine the combination with the highest classification performances.

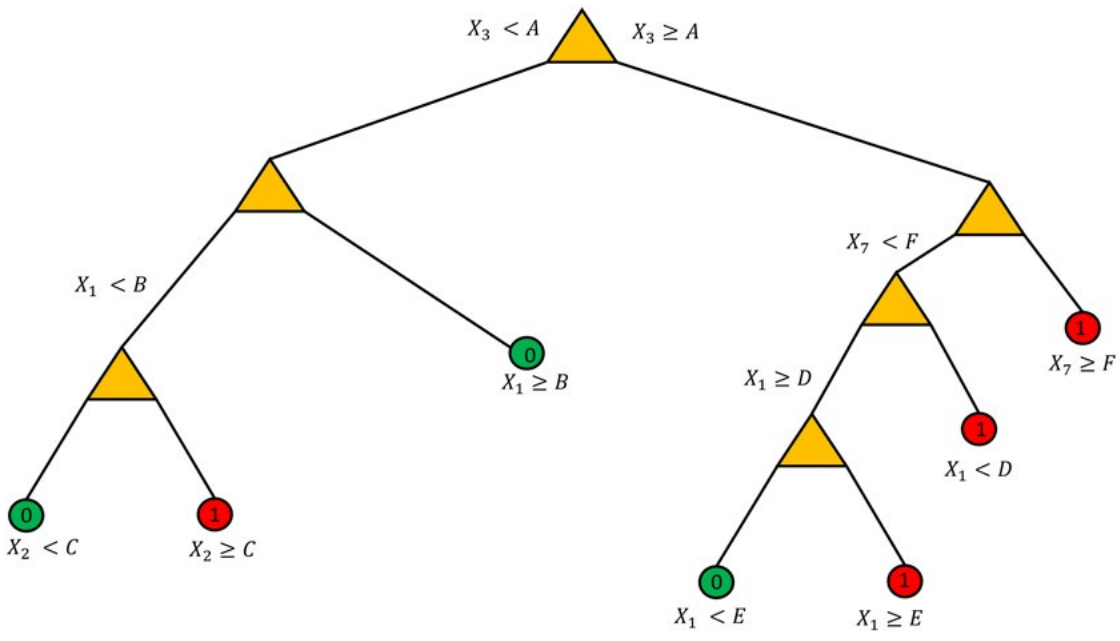
Table 2: Used hyper-parameters

Hiddel layers	Activation function	Solver
(10)	Rectified Linear Unit (ReLU),	Limited memory Broyden–Fletcher–Goldfarb–Shanno (LBFGS) (Fei et al., 2014),
(10,10)	Logistic Sigmoid	Stochastic Gradient Descent (SGD) (Sopy la and Drozda, 2015),
(100,100)	Hyperbolical Tangent (Tanh).	Adam solver (Rausch et al., 2017).
(10,10,10)		
(100,10,10)		
(100,100,100,10)		

4.2 Decision tree

Another algorithm used in this research is Decision tree. Decision tree represents a classification approach based on a set of rules formed as a tree structure (Li et al., 2019). Such a structure is used as a non-parametric algorithm that follows the paradigm of supervised learning. during the training process, multiple structures are created to determine the set of rules that will result in the highest classification performances. An example of a Decision tree is given in Figure 7.

Figure 7: An example of a Decision tree



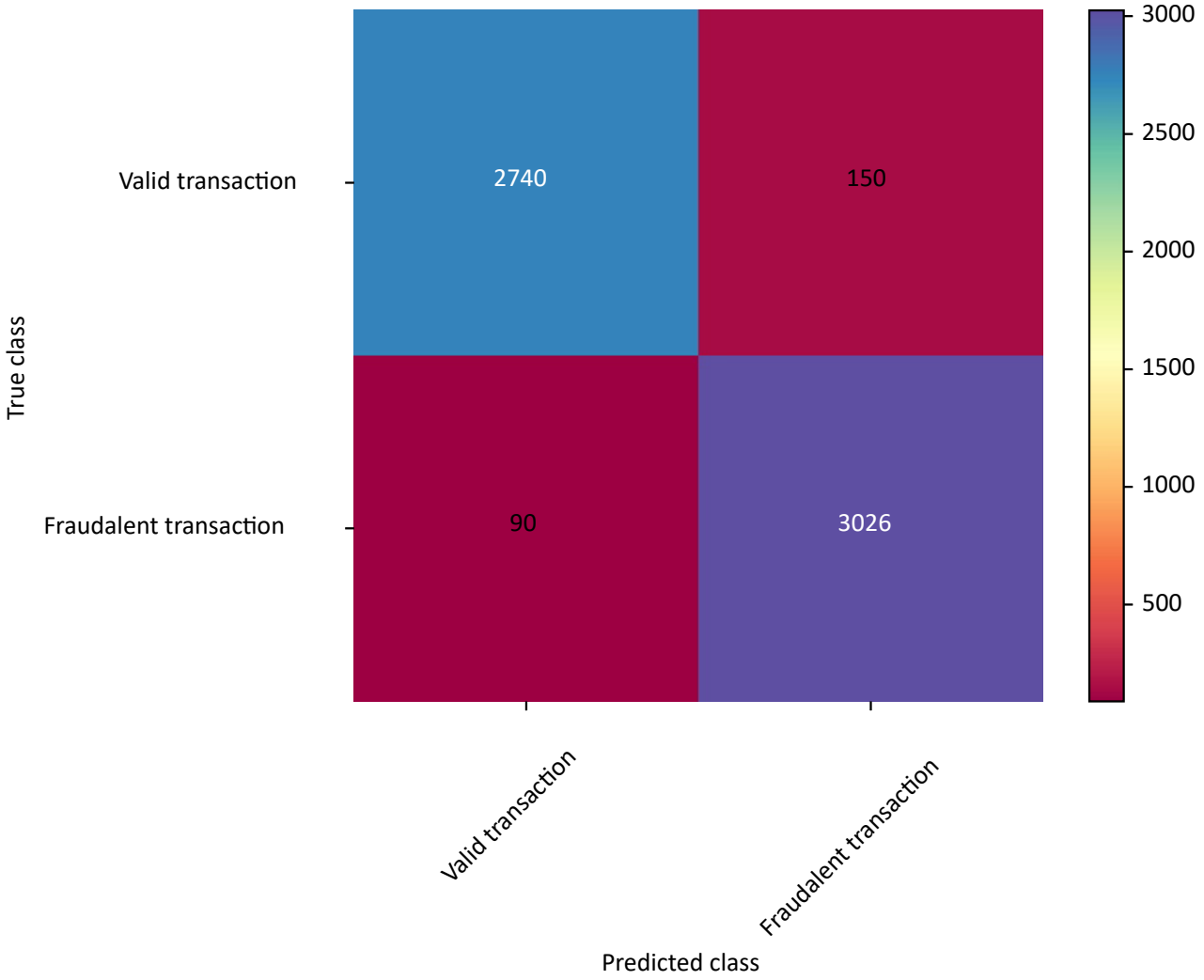
The configuration of a Decision tree, i.e. the set of rules is defined during the training procedure. During the training procedure, multiple trees are designed and evaluated to define the tree with the highest classification performance. To perform the training procedure, several parameters are defined. The performances of the designed tree can vary with the change of training parameters (Vanfretti and Arava, 2020, Ahmed et al., 2018). With the aim of achieving higher classification performances, a grid search procedure is performed to determine the training parameters that will result in a higher F1 score. During this research, three training parameters are varied:

- Maximal number of splits,
- Minimal leaf size, and
- Minimal parent size.

5. Performance evaluation methods

To determine the performance of the designed classifiers, it is necessary to define the metrics. In this research, methods based on the confusion matrix were used. Confusion matrix is a special form of a matrix that can be formed as a table to enable the visualization of classification performances (Wu, 2022, Kim and Cho, 2022). Each row in the matrix represents one actual class contained in the data set target. Each column of the confusion matrix represents predicted class (Li et al., 2022). An example of a binary confusion matrix is presented in Figure 8.

Figure 8: An example of a confusion matrix



According to the confusion matrix, multiple metrics used for classification performance evaluation can be formed. Precision or positive predicted value (*PPV*) can be defined as a ratio between the number of samples correctly classified as fraudulent and the total number of samples classified as fraudulent, or:

$$PPV = \frac{TP}{TP + TN} \tag{1}$$

where TP is a number of true positive and TN a number false positive values.

On the other hand, recall or true positive rate can be defined as a ratio between the number of samples correctly classified as fraudulent and the total number of truly fraudulent samples, or:

$$TPR = \frac{TP}{TP + FN} \tag{2}$$

where FN is the number of false negative values.

5.1 AUC-ROC

One of the measures used in this research is the area under the *ROC* curve (*AUC*). *ROC* curve is a curve constructed in *FPR*–*TPR* plane, where *FPR* is on the *x* and *TPR* on *y* axis. The *AUC* is the area under the *ROC* curve, and it represents a single, scalar value that can be used as a classification measure. *AUC* represents a float value ranging from 0.5 (coin-flip) classification to 1 (perfect classification) (Lorencin et al., 2020).

5.2 F1

Alongside *AUC*, *F1* score was used. *F1* score represent the harmonic mean of *PPV* and *TPR*. Such a harmonic mean can be defined as:

$$F1 = 2\frac{PPV \cdot TPR}{PPV + TPR} \tag{3}$$

or:

$$F1 = \frac{2\,TP}{2\,TP + TN + FN} \tag{4}$$

F1, alongside *AUC*, represents a standard classification measure, used in various classification problems. Due to a higly imbalanced data set, two new classification measures are introduced, and these are:

- Matthews Correlation Coefficient and
- Cohen's Kappa Coefficient.

In the next few paragraphs, a brief description of the aforementioned measures is provided.

5.3 Matthews Correlation Coefficient

Another method used for statistical evaluation of classification performances is Matthews Correlation Coefficient (*MCC*). *MCC* can be defined by using data

$$F1 = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{5}$$

determined from the confusion matrix as (Chicco and Jurman, 2020):

5.4 Cohen's *Kappa* Coefficient

The last classification measure used in this research is Cohen's *Kappa* Coefficient. Cohen's *Kappa* is a quantitative measure of reliability for two raters that are rating the same thing, corrected for how often that the raters may agree by chance. Similarly to already presented measures, *Kappa* can be calculated by using confusion matrix (Wang et al., 2019).

In the case of binary classification, *Kappa* can be reduced to:

$$Kappa = \frac{2(TP \cdot TN - FN \cdot FP)}{(TP + FP)(FP + TN) + (TP + FN)(FN + TN)} \tag{6}$$

6. Results and discussion

In this section, the achieved results will be presented and discussed for the case of the classification based on the original data set and on data sets created by using random under-sampling, SMOTE resampling, and SMOTE-Tomek resampling. The results will be presented for both MLP and Decision tree-based classification.

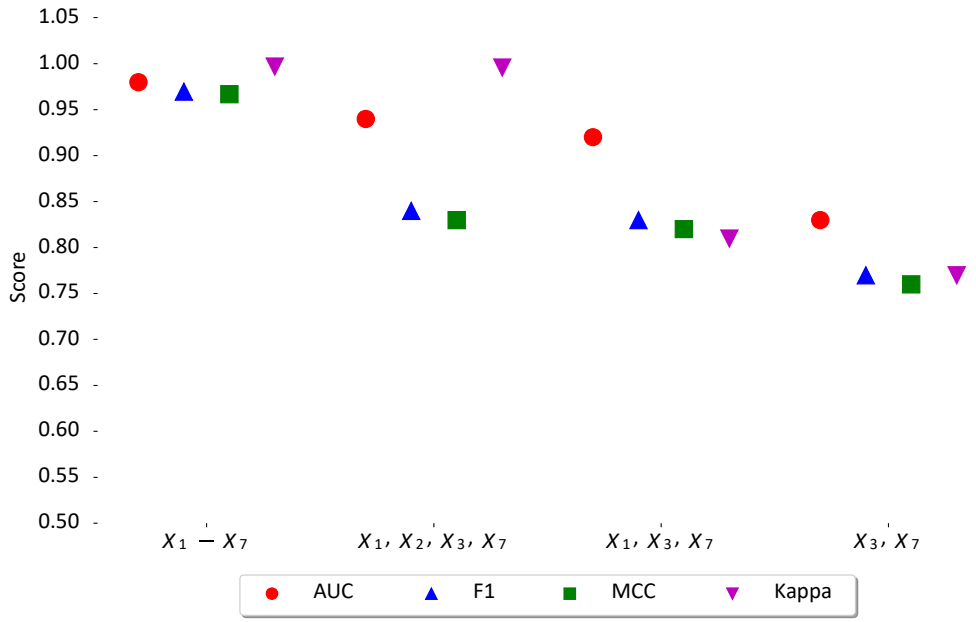
6.1 Results achieved with original data set

In this subsection, the results achieved with MLP and Decision tree trained with the original data set are presented.

6.1.1 Results achieved with Multilayer perceptron

During the development of the MLP, multiple architectures were examined. Furthermore, each MLP variation is trained and tested using different combinations of input variables. For each combination of input parameters, the MLP configuration with the highest classification performances is presented, alongside the achieved scores. It can be noticed that, in the case of the input vector that is designed using all 7 input parameters, the highest performances were achieved if an MLP is designed by using ReLU activation function. Furthermore, it can be noticed that the satisfying classification performances were achieved only in the case when all 7 input variables are used during the construction of the input vector, as presented in Figure 9.

Figure 9: Comparison of *AUC, F1, MCC*, and *Kappa* scores achieved with a Multilayer perceptron according to the combination of input parameters



It can be noticed that the highest classification performances are achieved if an MLP of intermediate model complexity is used. Such a conclusion is consistent with the theoretical knowledge of model selection (Hastie et al., 2009, Bishop and Nasrabadi, 2006, Goodfellow et al., 2016). All described MLP configurations are presented in Table 3.

Table 3: MLP hyperparameters and achieved metrics for the case of all combinations of input variables

Input variables	Solver	Hidden layers	Activation function
X_1–X_7	Adam	100,10	ReLU
X_1, X_2, X_3, X_7	LBFGS	100	Tanh
X_1, X_3, X_7	Adam	10	ReLU
X_3, X_7	LBFGS	10,10	Logistic

6.1.2 Results achieved with Decision tree

When the results achieved with a Decision tree are compared, it can be seen that the highest classification performances are achieved only if all 7 input variables are used for input vector construction. It is interesting to notice that, in all other cases, the underperformance can be noticed only if *MCC* and Kappa measures are observed. In the case of *AUC* and *F1*, the performance drop is not obvious, as presented in Figure 10.

Such a performance drop is a consequence of the highly imbalanced training data set.

Hyper-parameters used for the design of the analyzed Decision trees are presented in Table 4.

Figure 10: Comparison of *AUC, F1, MCC*, and *Kappa* scores achieved with a Decision tree according to the combination of input parameters

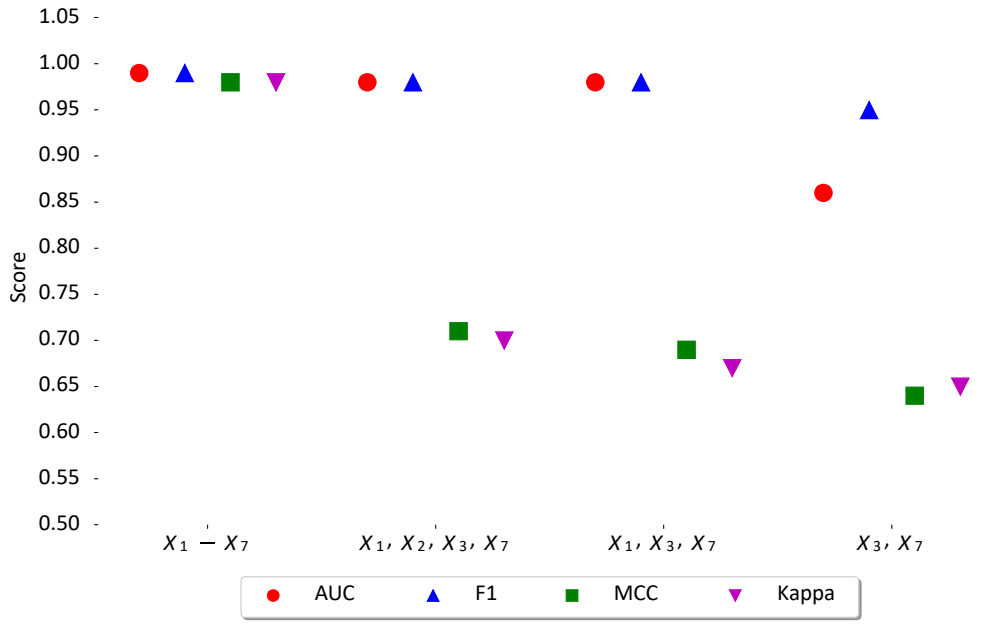


Table 4: Decision tree hyperparameters and achieved metrics for the case of all combinations of input variables

Input variables	Maximal number of splits	Minimal leaf size	Minimal parent size
X_1–X_7	20	1	10
X_1, X_2, X_3, X_7	18	1	1
X_1, X_3, X_7	9	1	1
X_3, X_7	8	1	1

6.2 Results achieved with random under-sampling

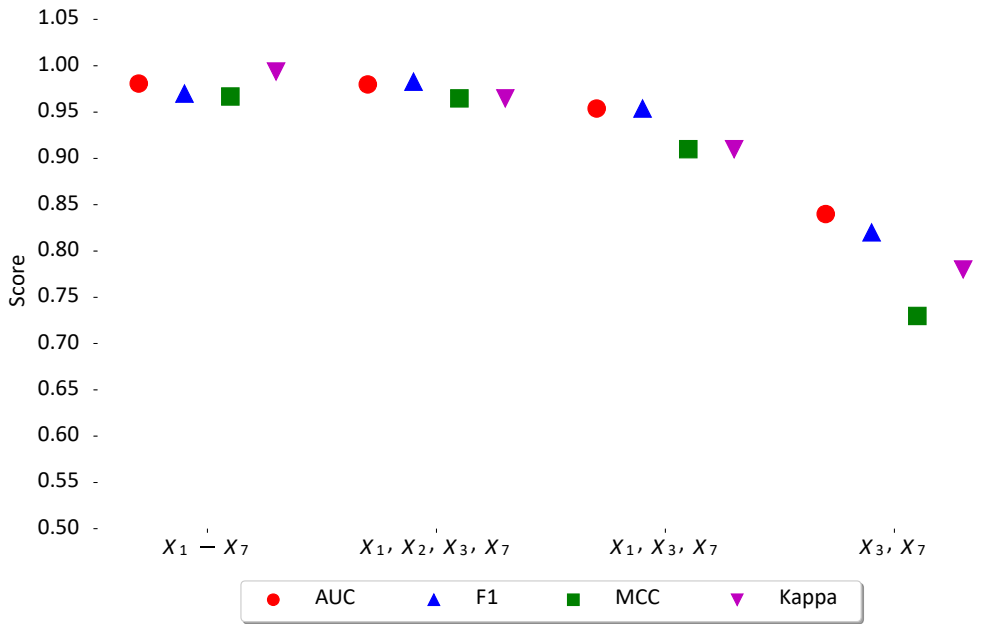
In this subsection, the results achieved with MLP and Decision tree trained with the randomly under-sampled data set are presented.

6.2.1 Results achieved with Multilayer perceptron

If the results achieved with MLP trained and tested by using random under-sampling are observed, it can be noticed that the highest classification

performances are achieved when all 7 variables are used during input vector construction. In the cases when a lower number of input variables is used for input vector construction, a slight drop in classification performances can be noticed. It is interesting to notice that the performances higher than 0.9 are achieved in all cases except for the case when X_3 and X_7 are used for input vector construction, as presented in Figure 11.

Figure 11: Comparison of *F1* scores achieved with a Multilayer perceptron and random under-sampling according to the combination of input parameters



The presented results are pointing towards the conclusion that MLPs trained with a lower number of input parameters can be used for credit card fraud detection to in more extent. Such a characteristic can be noticed regardless of the metric used. Such a characteristic can be attributed to the fact that the data set is highly balanced.

The hyper-parameters used for the design of the analyzed MLP models are presented in Table 5.

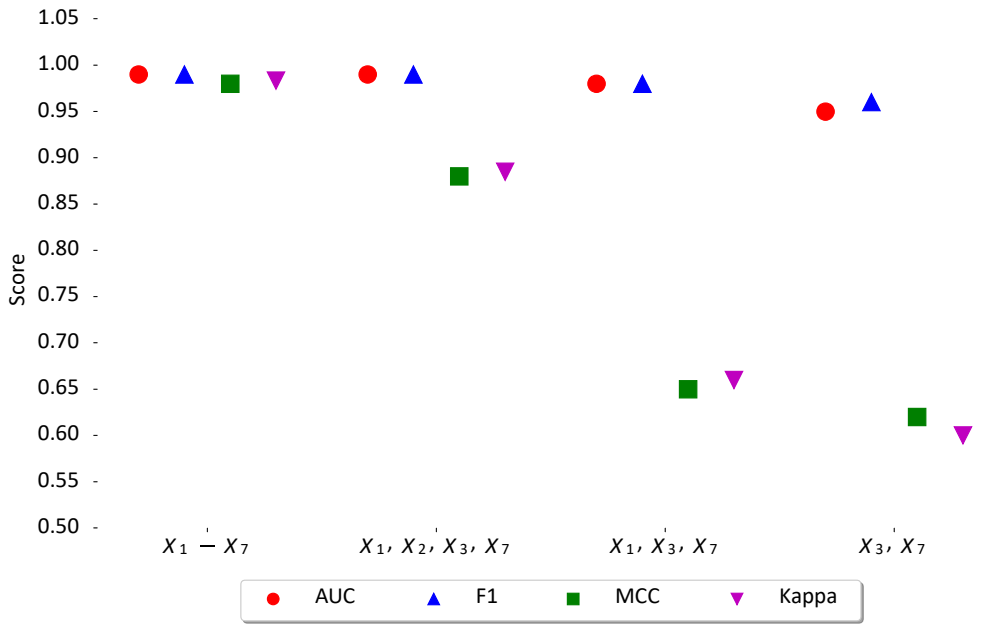
Table 5: MLP hyperparameters and achieved metrics for the case of all combinations of input variables

Input variables	Solver	Hidden layers	Activation function
X_1–X_7	LBFGS	10,10,10	Tanh
X_1, X_2, X_3, X_7	LBFGS	100,100,100,10	ReLU
X_1, X_3, X_7	LBFGS	10,10,10	ReLU
X_3, X_7	LBFGS	10	ReLU

6.2.2 Results achieved with Decision tree

If a Decision tree is used on a randomly under-sampled data set, it can be noticed that the highest classification performances are achieved only when all 7 input variables are used for input vector construction. In all other cases, a significant drop in *MCC* and *Kappa* score can be noticed, as can be seen in Figure 12. Due to the low *MCC* and *Kappa*, all other combinations of input parameters must be omitted.

Figure 12: Comparison of *AUC, F1, MCC*, and *Kappa* scores achieved with a Decision tree according to the combination of input parameters and random under-sampling



Hyperparameters used for the design of the analyzed trees are presented in Table 6.

Table 6: Decision tree hyperparameters and achieved metrics for the case of all combinations of input variables and random under-sampling

Input variables	Maximal number of splits	Minimal leaf size	Minimal parent size
X_1–X_7	13	1	19
X_1, X_2, X_3, X_7	8	1	1
X_1, X_3, X_7	14	7	19
X_3, X_7	3	1	1

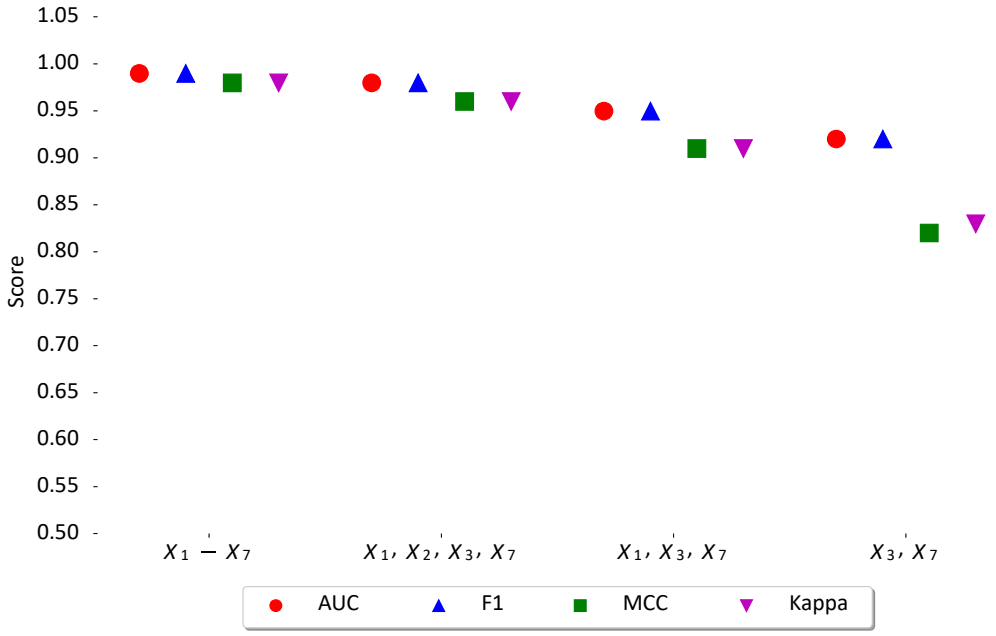
6.3 Results achieved with SMOTE

In this subsection, the results achieved with MLP and Decision tree trained with the SMOTE data set are presented.

6.3.1 Results achieved with Multilayer perceptron

When the results achieved with MLP trained by using SMOTE-generated date set, it can be noticed that the highest classification performances are achieved if all input variables are used. However, slightly lower performances are achieved if MLP is trained by using four or three input variables. In this case, the classification score is higher than 0.9, regardless of the metric used. If two variables are used for input vector construction, classification performances are significantly lower. This property is particularly emphasized in the case of *MCC* and *Kappa*, as presented in Figure 13.

Figure 13: Comparison of *AUC*, *F1*, *MCC*, and *Kappa* scores achieved with a Multilayer perceptron according to the combination of input parameters and SMOTE resampling



The hyper-parameters used for the design of the analyzed MLP models are presented in Table 7.

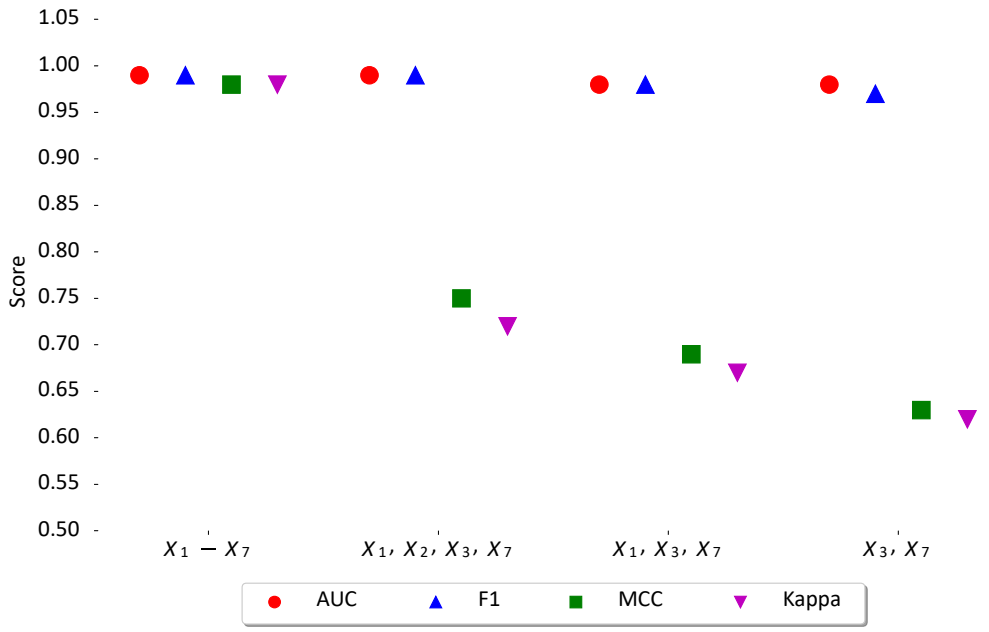
Table 7: MLP hyperparameters and achieved metrics for the case of all combinations of input variables

Input variables	Solver	Hidden layers	Activation function
X_1–X_7	LBFGS	10,10	Tanh
X_1, X_2, X_3, X_7	LBFGS	10,10,10	ReLU
X_1, X_3, X_7	LBFGS	100,10	Tanh
X_3, X_7	LBFGS	10,10	ReLU

6.3.2 Results achieved with Decision tree

If the results achieved with Decision tree are observed, it can be noticed that the high classification performances are achieved only if all 7 input variables are used. In all other cases, the performances are significantly lower. Such a property can be seen from low MCC and Kappa scores, as presented in Figure 14.

Figure 14: Comparison of *AUC*, *F1*, *MCC*, and *Kappa* scores achieved with a Decision tree according to the combination of input parameters and SMOTE resampling



Hyperparameters used for the design of the analyzed trees are presented in Table 8.

Table 8: Decision tree hyperparameters and achieved metrics for the case of all combinations of input variables and random under-sampling

Input variables	Maximal number of splits	Minimal leaf size	Minimal parent size
X_1–X_7	18	1	15
X_1, X_2, X_3, X_7	16	6	1
X_1, X_3, X_7	8	6	1
X_3, X_7	2	1	1

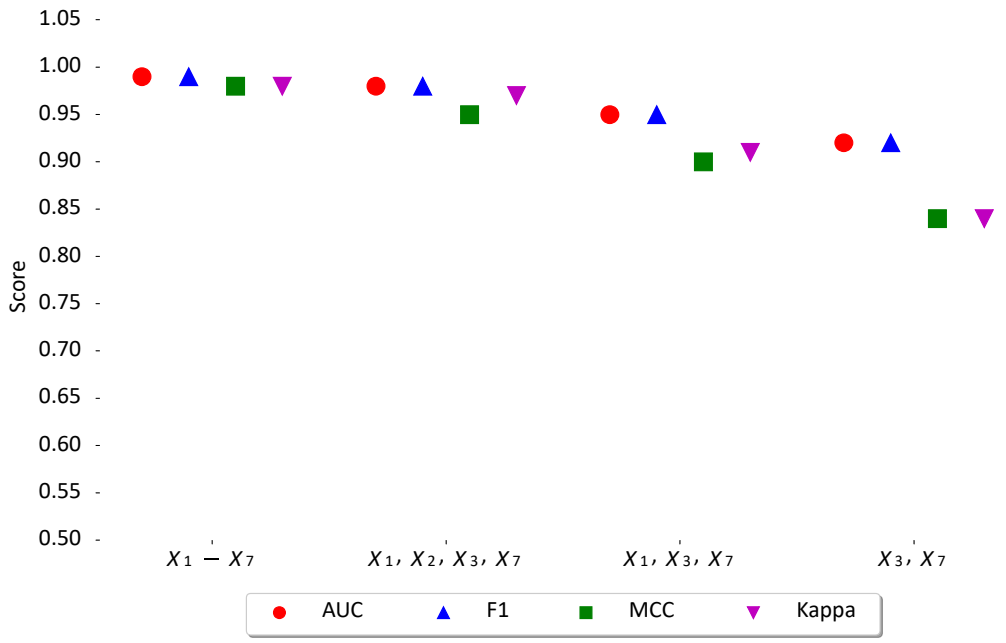
6.4 Results achieved with SMOTE Tomek

In this subsection, the results achieved with MLP and Decision tree trained with the SMOTE-Tomek data set are presented.

6.4.1 Results achieved with Multilayer perceptron

If the results achieved with MLP trained with SMOTE-Tomek generated data set are compared, it can be noticed that in all cases, a classification score higher than 0.8 is achieved. Such a property can be noticed regardless of the measure used. However, it can be noticed that the highest performances are achieved if all 7 input variables are used. On the other hand, MLP trained with four and three variables can achieve a score over 0.9, regardless of the measure utilized, as presented in Figure 15.

Figure 15: Comparison of *AUC*, *F1*, *MCC*, and *Kappa* scores achieved with a Multilayer perceptron according to the combination of input parameters and SMOTE-Tomek resampling



From the presented result, it can be seen that MLP trained with a data set resampled by using SMOTE-Tomek algorithm can achieve higher robustness on input dimension reduction. The hyper-parameters used for the design of the analyzed MLP models are presented in Table 9.

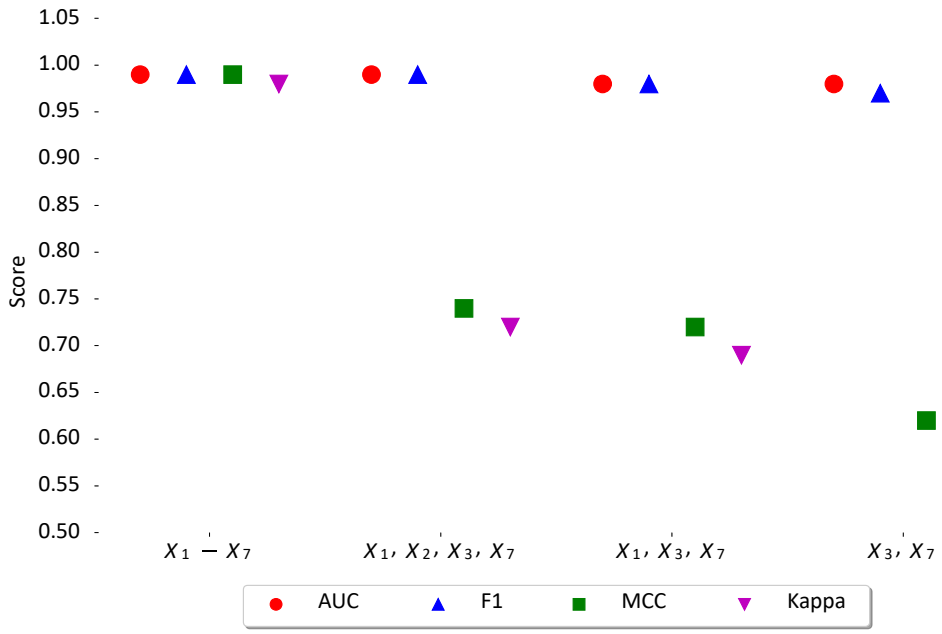
Table 9: MLP hyperparameters and achieved metrics for the case of all combinations of input variables

Input variables	Solver	Hidden layers	Activation function
X_1–X_7	LBFGS	100	Tanh
X_1, X_2, X_3, X_7	LBFGS	100	Tanh
X_1, X_3, X_7	LBFGS	100	Logistic
X_3, X_7	Adam	100,10	ReLU

6.4.2 Results achieved with Decision tree

When the results achieved with Decision tree are compared, it can be noticed that high classification performances are achieved only if all 7 input variables are used. In all other cases, the classification performances are significantly lower. Such a conclusion can be derived from low MCC and Kappa scores, as presented in Figure 16.

Figure 16: Comparison of *AUC*, *F1*, *MCC*, and *Kappa* scores achieved with a Decision tree according to the combination of input parameters and SMOTETomek resampling



Hyperparameters used for the design of the analyzed trees are presented in Table 10.

Table 10: Decision tree hyper-parameters and achieved metrics for the case of all combinations of input variables and random under-sampling

Input variables	Maximal number of splits	Minimal leaf size	Minimal parent size
X_1–X_7	20	8	1
X_1, X_2, X_3, X_7	15	6	1
X_1, X_3, X_7	8	6	1
X_3, X_7	2	1	1

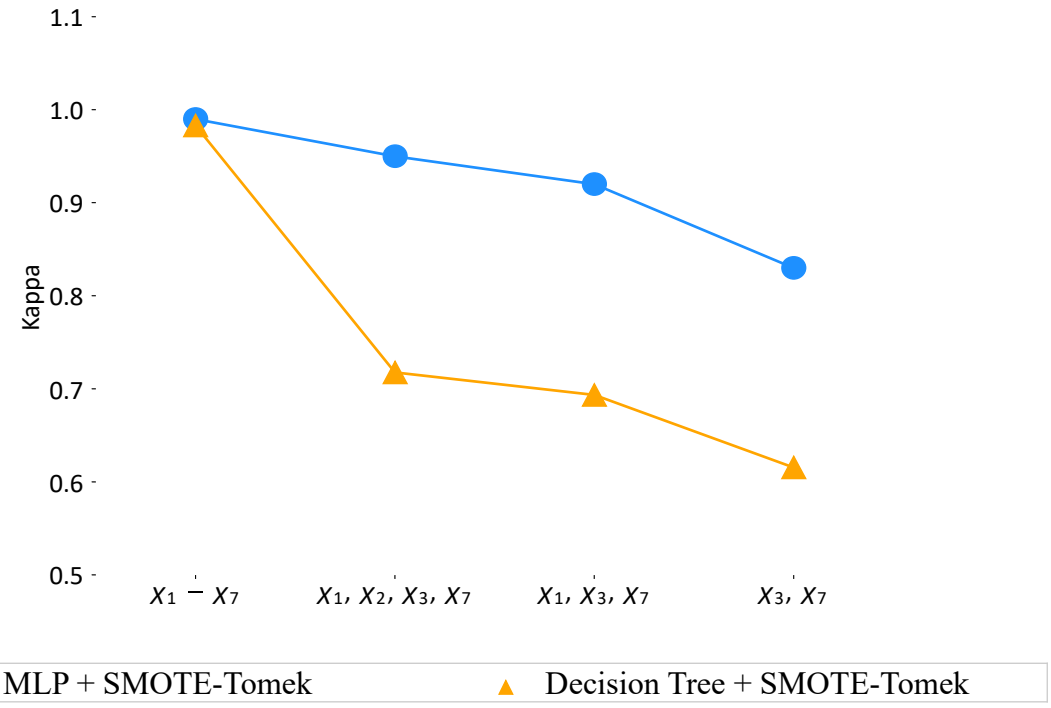
6.5 Method comparison

When all achieved results are compared, it can be noticed that the highest classification performances are achieved if SMOTE-Tomek resampling algorithm was used. SMOTE-Tomek has shown the dominant performances regardless of the algorithm used. Both algorithms, if trained with the data set re-sampled by using SMOTE-Tomek and by using all 7 input variables, have achieved the highest classification scores, regardless of the metric used. If the performances of MLP trained with SMOTE-Tomek data set are closely observed, it can be noticed that 1 in 100 valid transactions will be falsely classified as fraudulent. On the other hand, it can be noticed that 1 in 1000 fraudulent transaction will be classified as valid. Such a relationship can be described as positive, due to the fact that the detection rate of a fraudulent transaction is high. If the Decision tree is used, it can be seen that only 1 in 1000 transactions will be falsely detected as fraudulent. On the other hand, it can be noticed that 4 in 1000 transactions will be falsely detected as valid.

By comparing the achieved results with the results achieved with MLP, it can be seen that by using Decision tree, a lower number of transactions are falsely detected as fraudulent. On the other hand, the number of falsely valid transactions is four times higher, when compared with MLP.

When the performances of MLP and Decision tree according to input vector dimension reduction are compared, it can be noticed that a significant performance drop occurs if Decision tree is used. In this case, *Kappa* score is higher than 0.9 only if all 7 input variables are used, as presented in Figure 17.

Figure 17: Comparison Kappa values according to the input variables for MLP and Decision tree trained with data set re-sampled by using SMOTE-Tomek



On the other hand, if MLP is used, *Kappa* higher than 0.8 is achieved even if only two input variables are used. Such a property is pointing towards the conclusion that MLP has higher robustness to input vector dimension reduction, and it can be used in conditions whit a limited number of input data.

7. Conclusions

In this paper, a machine learning-based method for credit card fraud was proposed. The proposed method was based on the utilization of MLP and Decision tree. Due to the highly imbalanced data set, random undersampling, SMOTE, and SMOTE-Tomek algorithms were used. From the achieved results, it can be concluded that:

- It is possible to design an intelligent system for credit card fraud detection based on 382 Decision tree and MLP.
- The highest classification performances are achieved if MLP and Decision tree are trained by using data set re-sampled with SMOTETomek algorithm. Such a conclusion can be derived from high classification scores, determined with different classification performance measures.
- By using Decision tree, a lower number of transactions are falsely detected as fraudulent. By using MLP, a lower number of transactions are falsely detected as valid.

- MLP has shown higher robustness to input vector dimension reduction. Such a robust algorithm can be applied for credit card fraud detection even in conditions where data about transactions is limited.

The main disadvantage of this research is the limitations of SMOTE and SMOTE-Tomek algorithms and their aim for generalization. For these reasons, future work will be based on the investigation of the utilization of more complex data set resampling algorithms. Furthermore, future work will be based on different classification methods such as classifiers based on genetic programming. Furthermore, the possibility of utilizing more complex ensemble methods will be examined.

References

(Abou Jaoude, 2013) Abou Jaoude, A. (2013). The complex statistics paradigm and the law of large numbers. Journal of Mathematics and Statistics, 9(4):289.

(Ahmed et al., 2018) Ahmed, A. M., Rizaner, A., and Ulusoy, A. H. (2018). A novel decision tree classification based on post-pruning with bayes minimum risk. PLoS One, 13(4):e0194168.

(Alarfaj et al., 2022) Alarfaj, F. K., Malik, I., Khan, H. U., Almusallam, N., Ramzan, M., and Ahmed, M. (2022). Credit card fraud detection using state-of-the-art machine learning and deep learning algorithms. IEEE Access, 10:39700–39715.

(Arora et al., 2022) Arora, B. et al. (2022). A review of credit card fraud detection techniques. Recent Innovations in Computing, pages 485–496.

(Aschi et al., 2022) Aschi, M., Bonura, S., Masi, N., Messina, D., and Profeta, D. (2022). Cybersecurity and fraud detection in financial transactions. In Big Data and Artificial Intelligence in Digital Finance, pages 269–278. Springer.

(Asha and KR, 2021) Asha, R. and KR, S. K. (2021). Credit card fraud detection using artificial neural network. Global Transitions Proceedings, 2(1):35–41.

(Bishop and Nasrabadi, 2006) Bishop, C. M. and Nasrabadi, N. M. (2006). Pattern recognition and machine learning, volume 4. Springer. (Bredt, 2019) Bredt, S. (2019). Artificial intelligence (ai) in the financial sector—potential and public strategies. Frontiers in Artificial Intelligence, 2:16.

(Car et al., 2020) Car, Z., Baressi Šegota, S., Anđelić, N., Lorencin, I., and Mrzljak, V. (2020). Modeling the spread of covid-19 infection using a multilayer perceptron. Computational and mathematical methods in medicine, 2020.

(Carcillo et al., 2021) Carcillo, F., Le Borgne, Y.-A., Caelen, O., Kessaci, Y., Obl´e, F., and Bontempi, G. (2021). Combining unsupervised and supervised learning in credit card fraud detection. Information sciences, 557:317–331.

(Chang et al., 2022) Chang, V., Di Stefano, A., Sun, Z., Fortino, G., et al. (2022). Digital payment fraud detection methods in digital ages and industry 4.0. Computers and Electrical Engineering, 100:107734.

(Chaquet-Ulldemolins et al., 2022) Chaquet-Ulldemolins, J., GimenoBlanes, F.-J., Moral-Rubio, S., Mun˜oz-Romero, S., and Rojo-Alvarez, ´ J.-L. (2022). On the black-box challenge for fraud detection using machine learning (ii): Nonlinear analysis through interpretable autoencoders. Applied Sciences, 12(8):3856.

(Chawla et al., 2002) Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). Smote: synthetic minority over-sampling technique. Journal of artificial intelligence research, 16:321–357.

(Chen and Lai, 2021) Chen, J. I.-Z. and Lai, K.-L. (2021). Deep convolution neural network model for credit-card fraud detection and alert. Journal of Artificial Intelligence, 3(02):101–112.

(Chicco and Jurman, 2020) Chicco, D. and Jurman, G. (2020). The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. BMC genomics, 21(1):1– 13.

(Colli-Alfaro et al., 2019) Colli-Alfaro, J. G., Ibrahim, A., and Trejos, A. L. (2019). Design of user-independent hand gesture recognition using multilayer perceptron networks and sensor fusion techniques. In 2019 IEEE 16th International Conference on Rehabilitation Robotics (ICORR), pages 1103–1108. IEEE.

(Fei et al., 2014) Fei, Y., Rong, G., Wang, B., and Wang, W. (2014). Parallel l-bfgs-b algorithm on gpu. Computers & graphics, 40:1–9.

(Fukas et al., 2022) Fukas, P., Rebstadt, J., Menzel, L., and Thomas, O. (2022). Towards explainable artificial intelligence in financial fraud detection: Using shapley additive explanations to explore feature importance. In International Conference on Advanced Information Systems Engineering, pages 109–126. Springer.

(Goodfellow et al., 2016) Goodfellow, I., Bengio, Y., and Courville, A. (2016). Deep learning. MIT press.

(Grossi et al., 2022) Grossi, M., Ibrahim, N., Radescu, V., Loredo, R., Voigt, K., Von Altrock, C., and Rudnik, A. (2022). Mixed quantumclassical method for fraud detection with quantum feature selection. IEEE Transactions on Quantum Engineering.

(Hasan and Rizvi, 2022) Hasan, I. and Rizvi, S. (2022). Ai-driven fraud detection and mitigation in e-commerce transactions. In Proceedings of Data Analytics and Management, pages 403–414. Springer.

(Hastie et al., 2009) Hastie, T., Tibshirani, R., Friedman, J. H., and Friedman, J. H. (2009). The elements of statistical learning: data mining, inference, and prediction, volume 2. Springer.

(Karlik and Olgac, 2011) Karlik, B. and Olgac, A. V. (2011). Performance analysis of various activation functions in generalized mlp architectures of neural networks. International Journal of Artificial Intelligence and Expert Systems, 1(4):111–122.

(Khosravi and Syri, 2020) Khosravi, A. and Syri, S. (2020). Modeling of geothermal power system equipped with absorption refrigeration and solar energy using multilayer perceptron neural network optimized with imperialist competitive algorithm. Journal of Cleaner Production, 276:124216.

(Kim et al., 2022) Kim, J., Jung, H., and Kim, W. (2022). Sequential pattern mining approach for personalized fraudulent transaction detection in online banking. Sustainability, 14(15):9791.

(Kim and Cho, 2022) Kim, S.-C. and Cho, Y.-S. (2022). Predictive system implementation to improve the accuracy of urine self-diagnosis with smartphones: Application of a confusion matrix-based learning model through rgb semiquantitative analysis. Sensors, 22(14):5445.

(Kuzlu et al., 2021) Kuzlu, M., Fair, C., and Guler, O. (2021). Role of artificial intelligence in the internet of things (iot) cybersecurity. Discover Internet of things, 1(1):1–14.

(Lee et al., 2020) Lee, S.-J., Tseng, C.-H., Lin, G.-R., Yang, Y., Yang, P., Muhammad, K., and Pandey, H. M. (2020). A dimension-reduction based multilayer perception method for supporting the medical decision making. Pattern Recognition Letters, 131:15–22.

(Li, 2022) Li, J. (2022). E-commerce fraud detection model by computer artificial intelligence data mining. Computational Intelligence and Neuroscience, 2022.

(Li et al., 2022) Li, J., Sun, H., and Li, J. (2022). Beyond confusion matrix: learning from multiple annotators with awareness of instance features. Machine Learning, pages 1–23.

(Li et al., 2019) Li, M., Xu, H., and Deng, Y. (2019). Evidential decision tree based on belief entropy. Entropy, 21(9):897.

(Lorencin et al., 2019) Lorencin, I., Anđelić, N., Mrzljak, V., and Car, Z. (2019). Marine objects recognition using convolutional neural networks. NASE MORE: znanstveni časopis za more i pomorstv, 66(3):112–119.

(Lorencin et al., 2020) Lorencin, I., Anđelić, N., Španjol, J., and Car, Z. (2020). Using multi-layer perceptron with laplacian edge detector for bladder cancer diagnosis. Artificial Intelligence in Medicine, 102:101746.

(Luque et al., 2019) Luque, A., Carrasco, A., Mart´ın, A., and de Las Heras, A. (2019). The impact of class imbalance in classification performance metrics based on the binary confusion matrix. Pattern Recognition, 91:216–231.

(Mattos et al., 2020) Mattos, D. M. F., Krief, F., and Rueda, S. J. (2020). Blockchain and artificial intelligence for network security.

(Melnychenko, 2020) Melnychenko, O. (2020). Is artificial intelligence ready to assess an enterprise's financial security? Journal of Risk and Financial Management, 13(9):191.

(Melović et al., 2021) Melović, B., Sehović, D., Karadžić, V., Dabić, M.,and Čirović, D. (2021). Determinants of millennials' behavior in on-´ line shopping–implications on consumers' satisfaction and e-business development. Technology in society, 65:101561.

(Moon et al., 2018) Moon, J., Kim, Y., Son, M., and Hwang, E. (2018). Hybrid short-term load forecasting scheme using random forest and multilayer perceptron. Energies, 11(12):3283.

(Navaneethakrishnan and Viswanath, 2022) Navaneethakrishnan, P. and Viswanath, R. (2022). Fraud detection on credit cards using artificial intelligence methods. Elementary Education Online, 19(2):2086–2086.

(Nguyen et al., 2020) Nguyen, M. T., Truong, L. H., Tran, T. T., and Chien, C.-F. (2020). Artificial intelligence based data processing algorithm for video surveillance to empower industry 3.5. Computers & Industrial Engineering, 148:106671.

(Nonum et al., 2022) Nonum, E., Okafor, K., Nosike, I., and Misra, S. (2022). Ai-jascon: An artificial intelligent containerization system for bayesian fraud determination in complex networks. In Artificial Intelligence for Cloud and Edge Computing, pages 299–319. Springer.

(Olatunji et al., 2019) Olatunji, O. O., Akinlabi, S., Madushele, N., Adedeji, P. A., and Felix, I. (2019). Multilayer perceptron artificial neural network for the prediction of heating value of municipal solid waste. AIMS Energy, 7(6):944–956.

(Qiu et al., 2019) Qiu, X., Du, Z., and Sun, X. (2019). Artificial intelligence-based security authentication: applications in wireless multimedia networks. IEEE Access, 7:172004–172011.

(Rausch et al., 2017) Rausch, V., Hansen, A., Solowjow, E., Liu, C., Kreuzer, E., and Hedrick, J. K. (2017). Learning a deep neural net policy for end-to-end control of autonomous vehicles. In 2017 American Control Conference (ACC), pages 4914–4919. IEEE.

(Rehman et al., 2019) Rehman, I. U., Nasralla, M. M., and Philip, N. Y. (2019). Multilayer perceptron neural network-based qos-aware, contentaware and device-aware qoe prediction model: A proposed prediction model for medical ultrasound streaming over small cell networks. Electronics, 8(2):194.

(Rouf et al., 2021) Rouf, N., Malik, M. B., Arif, T., Sharma, S., Singh, S., Aich, S., and Kim, H.-C. (2021). Stock market prediction using machine learning techniques: a decade survey on methodologies, recent developments, and future directions. Electronics, 10(21):2717.

(Segota et al., 2021) Šegota, S. B., Anđelić, N., Mrzljak, V., Lorencin, I., Kurić, I., and Car, Z. (2021). Utilization of multilayer perceptron for determining the inverse kinematics of an industrial robotic manipulator. International Journal of Advanced Robotic Systems, 18(4):1729881420925283.

(Shanthakumara et al., 2022) Shanthakumara, A. et al. (2022). A comparative analysis of supervised classifiers for detecting credit card frauds. In 2022 International Conference on Computer Communication and Informatics (ICCCI), pages 1–6. IEEE.

(Sopy la and Drozda, 2015) Sopy la, K. and Drozda, P. (2015). Stochastic gradient descent with barzilai–borwein update step for svm. Information Sciences, 316:218–233.

(Sudha and Akila, 2021) Sudha, C. and Akila, D. (2021). Majority vote ensemble classifier for accurate detection of credit card frauds. Materials Today: Proceedings.

(Taha and Malebary, 2020) Taha, A. A. and Malebary, S. J. (2020). An intelligent approach to credit card fraud detection using an optimized light gradient boosting machine. IEEE Access, 8:25579–25587.

(Vanfretti and Arava, 2020) Vanfretti, L. and Arava, V. N. (2020). Decision tree-based classification of multiple operating conditions for power system voltage stability assessment. International Journal of Electrical Power & Energy Systems, 123:106251.

(Varmedja et al., 2019) Varmedja, D., Karanovic, M., Sladojevic, S., Arsenovic, M., and Anderla, A. (2019). Credit card fraud detectionmachine learning methods. In 2019 18th International Symposium INFOTEH-JAHORINA (INFOTEH), pages 1–5. IEEE.

(Wang et al., 2019) Wang, J., Yang, Y., and Xia, B. (2019). A simplified cohen's kappa for use in binary classification data annotation tasks. IEEE Access, 7:164386–164397.

(Wu, 2022) Wu, M.-T. (2022). Confusion matrix and minimum crossentropy metrics based motion recognition system in the classroom. Scientific Reports, 12(1):1–10.

(Wynn and Olayinka, 2021) Wynn, M. and Olayinka, O. (2021). Ebusiness strategy in developing countries: A framework and checklist for the small business sector. Sustainability, 13(13):7356.

(Yacouby and Axman, 2020) Yacouby, R. and Axman, D. (2020). Probabilistic extension of precision, recall, and f1 score for more thorough evaluation of classification models. In Proceedings of the first workshop on evaluation and comparison of NLP systems, pages 79–91.

Sažetak

Jedan od glavnih izazova za sigurnost internetskog poslovanja su kartične prijevare. Iz tog razloga uvode se algoritmi temeljeni na umjetnoj inteligenciji i strojnom učenju kako bi se omogućilo što točnije i brže otkrivanje kartičnih prijevara. Ovaj rad predstavlja pristup otkrivanju kartičnih prijevara koji se temelji na algoritmima strojnog učenja, točnije višeslojnom perceptronu (MLP) i stablu odlučivanja. Navedeni algoritmi trenirani su i testirani korištenjem javno dostupnog skupa podataka o kartičnim prijevarama. Skup podataka koji se koristi sastoji se od 7 karakteristika kartične transakcije i informacija o tome je li bilo kartične prijevare ili ne. Skup podataka ukupno sadrži informacije o 1.000.000 transakcija i vrlo je neuravnotežen. Kako bi se riješila neravnoteža klasa, predložene su metode temeljene na nasumičnom pod-uzorkovanju, SMOTE i SMOTE-Tomek algoritmi. Iz postignutih rezultata vidljivo je da se najveće performanse postižu ako se MLP (AUC = 0,99, f1 = 0,99, MCC = 0,98 i Kappa = 0,98) i stablo odlučivanja (AUC = 0,99, f1 = 0,99, MCC = 0,99, i Kappa = 0,98) treniraju korištenjem skupa podataka koji je balansiran primjenom SMOTE-Tomek algoritma. Ako se performanse spomenutih algoritama ispitaju korištenjem manjeg broja karakteristika transakcije, može se vidjeti da se smanjenjem broja karakteristika može primijetiti značajno smanjenje klasifikacijskih performansi ako se koristi Stablo odlučivanja u kombinaciji sa SMOTE-Tomek balansiranjem. Međutim, ako se koristi MLP u kombinaciji sa SMOTE-Tomek balansiranjem, može se primijetiti značajno niži pad performansi, što ukazuje na veću robusnost na smanjenje dimenzije ulaznog vektora. Takav robustan sustav može pružiti informacije o valjanosti transakcije čak i u uvjetima kada su ulazni podaci ograničeni na nekoliko ulaznih varijabli. Iz postignutih rezultata može se zaključiti da se MLP u kombinaciji sa SMOTE-Tomek algoritmom može koristiti za detekciju prijevara s kreditnim karticama, čak i u uvjetima s manjim brojem ulaznih karakteristika.

Istarsko veleučilište
Università Istriana di scienze applicate

Istarsko veleučilište
Università Istriana di scienze applicate

Riva 6, 52100 Pula
Hrvatska

www.iv.hr