Detecting Credit Card Fraud Using Selected Machine Learning Algorithms

Conference Paper · May 2019

DOI: 10.23919/MIPRO.2019.8757212

2 authors, including:

Ljiljana Brkic
University of Zagreb
19 PUBLICATIONS **160** CITATIONS

SEE PROFILE

Detecting Credit Card Fraud Using Selected Machine Learning Algorithms

Maja Puh, Ljiljana Brkić
University of Zagreb Faculty of Electrical Engineering and Computing,
Zagreb, Croatia,
maja.puh@fer.hr
ljiljana.brkic@fer.hr

***Abstract* - Due to the immense growth of e-commerce and increased online based payment possibilities, credit card fraud has become deeply relevant global issue. Recently, there has been major interest for applying machine learning algorithms as data mining technique for credit card fraud detection. However, number of challenges appear, such as lack of publicly available data sets, highly imbalanced class sizes, variant fraudulent behavior etc. In this paper we compare performance of three machine learning algorithms: Random Forest, Support Vector Machine and Logistic Regression in detecting fraud on real-life data containing credit card transactions. To mitigate imbalanced class sizes, we use SMOTE sampling method. The problem of ever-changing fraud patterns is considered with employing incremental learning of selected ML algorithms in experiments. The performance of the techniques is evaluated based on commonly accepted metric: precision and recall.**

Keywords - credit card fraud detection, machine learning, class imbalance

I. INTRODUCTION

In today's ever-growing global society, the issue of committing a fraud has become more relevant than ever. One of the areas that has experienced immense growth is e-commerce. Due to the increased number of possibilities in making payments of any kind and their easiness of use, e-commerce business establishments gained user confidence. Unfortunately, increased number of users followed by increased revenue makes them vulnerable to fraudulent behavior. In 2016, total value of card fraud using cards issued in Single Euro Payments Area (SEPA) was €1.8 billion, 0.041% of total value of card transactions made. Card fraud increased in terms of volume by 92% compared with 2012 [1]. Most common card fraud types are application fraud, lost or stolen card, account takeover and counterfeit cards [2]. Fraud caused by obtaining physical card or falsifying one by using counterfeit data is of less interest in the context of credit card fraud detection (CCFD) tools and analysis of cases of fraud. The reason being is that the majority, of all fraudulent transactions made are card-not-present (CNP) fraud (i.e. 73% in 2016). In CNP scenario, credit card information is obtained without the knowledge of card holder and is used remotely in attempt to commit fraud [3].

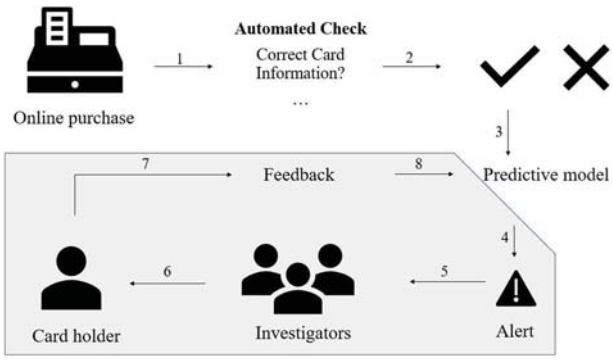

Figure 1 Typical credit card fraud detection scenario

Fraud detection assumes identifying fraud as quickly as possible once it has been committed. Methods for detecting fraud are continuously adapting to everchanging fraud strategies. Presently, fraud detection has been implemented by several methods such as statistics, rule engines, artificial intelligence and data mining. Fraud detection occurs on different levels, in terms of time expired from the occurrence of the single transaction. As shown in Figure 1, first level is automated card validation, which is performed in real time, for the benefit of user satisfaction. If a card is not rejected, transaction goes through the second level, denoted as *Predictive model*. Throughout the past decade, there has been increased interest in applying machine learning as a data mining approach for second level of credit card fraud detection [4].

The second level takes place in respective amount of time after transaction is made [5]. Automated part of the second level, predictive model, discovers fraud from anomalies in data and patterns. For the suspicious transaction an alert is raised which then requires human expert intervention. The shaded part in the Figure 1. happens only for those transactions. After examining detailed transaction data and, in some cases, contacting card holder, investigators decide whether the transaction is fraudulent or not and provide feedback in order to improve the accuracy of the used predictive model [6].

Solving fraud detection problem with data mining technique boils out to classifying transactions to one of the two categories: fraud or nonfraud. Classification is a data mining method that assigns items in a collection to target categories or classes. The goal of classification is to accurately predict the target class for each case in the data.

The simplest type of classification problem is binary classification where the target attribute has only two possible values [7]. Credit card fraud detection is, in fact, binary classification problem. This paper implements supervised machine learning algorithms for classification of a credit-card transaction as either fraudulent or not-fraudulent.

The rest of this paper is organized as follows. In Section 2, the challenges concerning credit card fraud detection as well as some of the state-of-the-art solutions for individual challenge are described. Section 3 introduces required concepts and proposed setup. Section 4 gives all the details about performed experiments. Finally, the paper ends with conclusion and future works.

II. CHALENGES IN CREDIT CARD FRAUD DETECTION

A. Data defficiency

Essentially, the biggest problem in dealing with CCFD scientifically is that real data is hardly ever available for exploration, due to the issue of confidentiality [8]. Nevertheless, researchers are not discouraged by this obstacle given the fact that they can often carry out scientific work in cooperation with respective industrial partner, provider of the data. Also, some suggest using synthetic data which simulates dataset of transactions [9].

B. Imbalanced data

As mentioned earlier, on the global level fraudulent transaction are amounted to less than 0.05% of the total transactions. This ratio is detained in the distribution of any credit card fraud dataset, resulting in highly imbalanced classes [10]. If this problem had not been taken into consideration, any machine learning algorithm that classifies correctly only genuine transactions would perform outstanding, with accuracy level above 99%, disregarding the fact that all the minority class transactions are classified falsely. There are possible solutions to this problem both at data and algorithmic level. We will focus on data level further on. Data level methods, such as oversampling and under sampling, alter the size of the dataset used for training. While the level of imbalance is reduced, problems of overfitting and ignoring useful data appear, respectively [11]. Possible improved approach is using one-sided selection methods focusing on removing some outlier or noisy majority class examples to decrease the disproportion between classes [12]. Outlier is an extreme data value that does not come from the typical population of data. In a normal distribution, outliers are typically at least 3 standard deviations from the mean [7]. Example of another complex sampling method is SMOTE, which oversamples the minority class generating synthetic examples by interpolating k minority class nearest neighbors [13]. Through this process the classifier builds larger decision regions that contain nearby examples from the minority class, which has shown improvements in application.

C. Behavioral variation

Fraudulent behavior tends to alter over time in order to avoid detection. Therefore, CCFD predictive model should not be static, i.e. constructed once and never updated [8]. In terms of machine learning, this problem is known as concept drift. It occurs during learning in nonstationary environment and manifests in changes in underlying data. Known methods used for overcoming this problem are adaptive base learners and ensembles [14]. Adaptive base learners have a drift detection mechanism that updates the current model when the drift is detected while ensembles have natural ability to simultaneously retain relevant information and acquire new knowledge [15]. In terms of CCFD, ensemble techniques are widely used [8], [15], [16].

D. Cost sensitive problem

CCFD is a cost sensitive problem, meaning that the cost produced by misclassifying genuine transaction (false positive) is different than the cost of misclassifying fraudulent one (false negative) [5]. When a CCFD system marks a genuine transaction as fraudulent, the financial institution has a measurable cost of employee engagement for examining and arbitrating the status of suspicious transactions and the immeasurable cost of customer dissatisfaction. On the other hand, failure to detect a fraudulent transaction (false negative), causes the financial loss of the amount of that transaction. The cause for the misclassification, besides concept drift in fraudulent behavior, lies in the distribution of the data. Precisely, the problem occurs because of the overlapping data - when many genuine transactions resemble fraudulent ones and vice versa, resulting in false positives and false negatives [10].

E. Evaluation metric

As mentioned in the chapter 2.2, accuracy is not a suitable metric for CCFD, due to the class imbalance. Metrics that are commonly used for fraud detection combine precision, recall (true positive rate) and fall-out (false positive rate) all shown in TABLE I. Metrics are based on a well-known machine learning concept confusion matrix with the following four categories defined:

- TP (true positive) - correctly predicted positive class
- FP (false positive) - incorrectly predicted positive class
- TN (true negative) - correctly predicted negative class
- FN (false negative) - incorrectly predicted negative class.

The main goal here is to improve the classification of false negatives – recall without hurting the classification of false positives – precision [17]. This is difficult because the two measures have an opposite relationship: decrease in one results in increase of the other [8]. The reason why we need to improve evaluation of both types (FP, FN) of the incorrectly classified examples is that each type causes different cost, making this cost sensitive evaluation [18].

Cost of false negatives is of financial kind, its significance varying depending on the amount of the transaction. On the other hand, cost of false positives is measured in terms of customer dissatisfaction. Metric that combines TPR and FPR is called The Receiver Operating Characteristic (ROC) curve and is represented by calculating area under the ROC curve (AUC). AUC is a well-accepted measure for handling class imbalance [6].

TABLE I. COMMON METRICS

Precision	$\frac{TP}{TP + FP}$
Recall (TPR)	$\frac{TP}{TP + FN}$
Fall-out (FPR)	$\frac{FP}{FP + TN}$

Technique that takes into account ranking of the examples is Precision-Recall curve, which gives Average precision, calculated as area under P-R curve [19].

The important aspect to consider regarding to detection costs is also time dimension. Fraud should be detected promptly after it occurred, as it then enables more potential frauds to be prevented [19]. Therefore, single algorithm can be evaluated as superior among others if it can detect fraudulent transactions before every other algorithm.

III. EXPERIMENTAL SETUP

The focus of this study is to examine performance of three selected machine learning algorithms. The selection of algorithms is based on our previous research [20] and complies with the most often used ML algorithms in CCFD. To overcome the unbalanced classes issue we use oversampling technique described later. This section describes the data used for training and testing the models, learning approaches and performance measures used.

A. Dataset

The dataset used in experiments is the only publicly available dataset suitable for the CCFD [21]. Dataset contains transactions made in September 2013 by European cardholders, during the period of two days. Among 284807 transactions there are 492 fraudulent ones. Therefore, the dataset is highly unbalanced, having only 0.1727% fraudulent transactions, which are positive class. Due to confidentiality issues, the original information about the data is omitted. Dataset consists of 28 numerical input variables (named V1 - V28) which are the result of transformation made by provider of the dataset and 2 additional not transformed variables - Time and Amount. Time is presented in the shape of seconds elapsed from the first recorded transaction, whereas Amount shows amount spent in a single transaction. In terms of supervised machine learning, all the above described variables are features. The dataset also contains field named Class representing the information whether the transaction is fraudulent or not. Possible Class values are 1 if transaction is fraudulent and 0 otherwise, making this a binary classification problem. Each transaction is considered as one input value, i.e. one transaction is represented with single vector.

B. Data preparation

Considering all the above stated guidelines regarding individual challenges, for handling class imbalance we have chosen SMOTE method. This method is, according the scientific literature, one of the most used [15],[16],[22]. The sampling strategy parameter, representing the desired ratio of the number of samples in majority class over the number of samples in minority class, may vary. The value used in experiments equals to 0.4 (number of minority samples is increased to 40% of majority class size). It was determined empirically by choosing the best performing value among various attempts.

In the dataset used, before it was made public, due to data privacy issue, features were already transformed using PCA (Principal Component Analysis) [23] for dimensionality reduction. Because of the unknown meaning of the original or the constructed features there was not enough information for creation of additional features. The only other preprocessing done on the dataset was scaling the feature Amount. It was standardized by removing mean and scaled to unit variance.

C. Algorithms

Based on our previous research, in experiments we used three algorithms that are among five most used in CCFD: Random Forest (RF), Support Vector Machine (SVM) and Logistic Regression (LR) [20].

Random Forest is an ensemble of decision tree models that are quite popular for various machine learning problems because they are easy to use and interpret. Each decision tree by itself is sensitive to overfitting, but combined they perform well. Random Forest is a bagging classifier and it implements two stochastic decision levels in its process of learning – for each individual decision tree in ensemble it chooses subset of samples as well as subset of features for training [24].

SVM is a classifier that maps features from the non-linear input space to feature space of higher dimension. The reason for this transformation is to convert initially more complex classification problem to linear in a higher dimensional space. This mapping is achieved by using kernel functions [25].

Finally, Logistic Regression has its history in appliance in fraud classification [3]. Logistic Regression is probabilistic model – it assigns probability to each classified sample. This gives us more possibilities than pure binary classification. In terms of CCFD, this allows us to rank the transactions by their probability to be fraudulent and choose the threshold for most probable frauds that will raise the alert by predictive model.

Parameters for the models were determined from a variety of preliminary tests conducted on training data. For Random Forest, number of trees is set to T=100. For SVM, we use Gaussian radial basis function as kernel which performs better than linear kernel, prone to overfitting in this case. The cost parameter C and the kernel parameter gamma were set to values 10 and 0.01 respectively. These values were selected after cross validation grid search which included different combinations of these values. In Logistic Regression we set C to 100 and used L2-regularization.

D. Learning approaches

There are different approaches to the learning process. The simplest, but still most widely used, is static approach which creates predictive model by processing available items all at once in a single learning batch [19]. The major drawback of such one-time trained model is its inability to adapt to any changes in the input data i.e. to support concept drift. Periodic re-training approach is more resistant to nonstandard input items, although changes that occur after the last trained data will not be included in the classification model. As opposed to static, incremental approach has been introduced [16]. Incremental learning is suitable for nonstationary environment because it processes data in chunks as they arrive. Due to the unpredictability of fraudulent behavior, it may happen that patterns once used and then altered may re-occur in the future. For that reason, incremental approach uses an ensemble which contains all previously generated classifiers that are evaluated on the newest data chunk and the classifiers with the best current performance are used in prediction. In that way, it preserves knowledge from the past while learning from new observations.

In experiments, we compared influence of static and incremental learning on chosen algorithms. In static approach, training and testing was done once using all data. For incremental learning, we divided data into two chunks, representing two days that are observed. We are aware that real life scenario includes larger number of data chunks, but this limitation is caused by the available data. Training and testing were done on each data chunk separately which generated two models. We implemented weighted ensemble of models h_i, which is defined as

$$E = \sum_{i=0}^{M} w_i h_i \quad (1)$$

where M denotes total number of models (in our case two). Weight w_i is calculated as precision (formula in TABLE I.) on observed data chunk and then normalized to 0-1 range.

E. Metrics

We decided to evaluate our results by two measures, area under the ROC curve (AUC) and average precision (AP). ROC curve is plotted as Recall (TPR) against Fall-out (FPR) at various classification thresholds. Transaction with fraud probability that equals or is above threshold value is considered fraudulent. The best classifier corresponds to the point (0,1) where there are no false positives or false negatives [24]. AUC metric measures how much is the ROC curve of single classifier close to the optimal point.

Second measure, average precision, approximates area under the precision-recall curve. A precision-recall curve is a plot of precision versus recall at different probability thresholds. Intuitively, each threshold represents rank and we want to examine how specific model performs for highest ranked transactions. Average precision is calculated as mean precision averaging over recall values [26].

Since data set, used in experiments, includes transactions made in just two days it did not make any sense to perform evaluation that includes time aspect.

IV. EXPERIMENTAL RESULTS

All the required programming code is written in the Python programming language, and standard scikit learn implementation was used. We used cross-validation for learning process and measuring model performance. Data set is divided into a training set and a test set in the typical ratio of 70:30. Results of evaluating models by AUC are shown in TABLE II. Results of evaluating models by average precision are shown in TABLE III. and Figures 2. and 3., respectively.

TABLE II. AUC SCORES

Static learning	Random Forest	0.9148
	SVM	0.8877
	Logistic Regression	0.9114
Incremental learning	Random Forest	0.9013
	SVM	0.8678
	Logistic Regression	0.9107

AUC scores reveal that all three algorithms have similar performance, with SVM having slightly lower results than the other two algorithms. All three algorithms have better AUC scores for static models than incremental ones.

AP, shown in table below, determines Random Forest as best option in terms of static learning. On the other hand, all three algorithms have similar performance in incremental version, while SVM is somewhat better when measured with Average Precision, as oppose to its performance measured with AUC. When comparing same algorithms in their static and incremental version, Random Forest is better in static (same as when measured with AUC), and SVM and Logistic Regression are better in incremental, with latter having substantially better results than its static version.

TABLE III. AP SCORES

Static learning	Random Forest	0.8483
	SVM	0.7978
	Logistic Regression	0.7337
Incremental learning	Random Forest	0.8293
	SVM	0.8036
	Logistic Regression	0.8413

Figure 2. shows comparison of all three algorithms measured with AP (plotted as Precision-Recall curve) in terms of static learning. In Figure 2. is their comparison in incremental learning.

Because the curves are overlapping, there is no conclusion on which algorithm is significantly better than the others. In Figure 2., for lower recall values, Random Forest and SVM are performing similar, while for the higher recall values Logistic Regression comes closer to Random Forest and SVM decreases in performance. In Figure 3., SVM is better for the low recall values and both Logistic Regression and Random Forest outperform it for higher recall values.

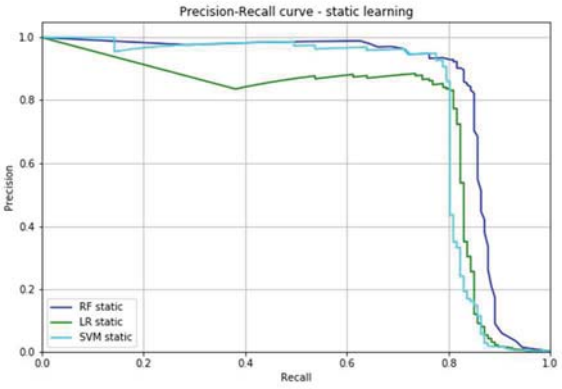

Figure 2. Precision-Recall curve – static learning

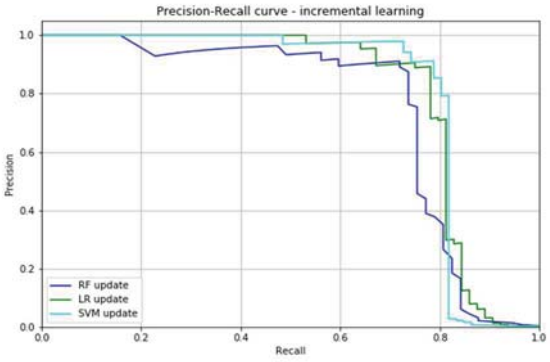

Figure 3. Precision-Recall curve – incremental learning

V. RELATED WORK

Since the data set, we used in experiments, is currently the only publicly available data set based on real-time data, there are a number of papers in which authors report on their results in similar research. In [27] authors also compared three algorithms (Naïve Bayes, k-nearest neighbors and Logistic Regression). Skewness was overcomed by a hybrid technique of undersampling and then oversampling. Their evaluation metrics includes precision and sensitivity (i.e. recall). Results show k-nearest neighbors as superior across all used evaluation metrics on different data distributions.

In [28] classification is done with Logistic Regression, nearest neighbors, SVM, Decision trees, Random Forest and Naïve Bayes. Evaluation is done by several metrics, also including precision and recall. All algorithms were applied on skewed data, without any balancing techniques. Logistic Regression outperformed other algorithms.

Importance of evaluation technique used in CCFD is pointed out in [29]. They compared ROC and Precision-Recall curves using Logistic Regression as on algorithmic level and random undersampling on data-level. Results show that Precision-Recall curve has more advantages than ROC curve in dealing with CCFD on this dataset with conclusion that the former captures the effect of improving both precision and recall better than the latter.

Authors in [30] put focus on data sampling methods used for overcoming class imbalance. They propose new technique called VOS (Variational Oversampling) and compared it with two existing oversampling techniques, one of them being SMOTE. Performance of three algorithms (Logistic Regression, Multi-layer perceptron and Random Forest) is evaluated by precision and recall, as well as accuracy. The proposed VOS method achieved the best score among all compared algorithms.

VI. CONLUSION

In this paper we have outlined main issues in the CCFD field and proposed state-of-the-art solutions. Using the only publicly available dataset suitable for the CCFD we have implemented and measured performance of three selected ML algorithms: Random Forest, Support Vector Machine and Logistic Regression. The selected algorithms belong to the most often used ML algorithms in CCFD and its selection came from scientific literature review we have performed in our previous research. Experiments were done using two approaches: (i) static and (ii) incremental learning of chosen algorithms. The following measures are used to evaluate the performance of the algorithms: area under the ROC curve (AUC) and average precision (AP). From the results presented in paper, it is evident that SVM shows the poorest performance in both static and incremental setup. The difference in performance between RF and LR is slight. LR shows marginally better results in incremental setup but this should be taken with caution because of the limitations of the dataset used - it contains transactions made in just two days. Since static training approach is not a long-term solution, we plan to investigate incremental learning on more realistic dataset. Richer datasets are not publicly available so to further our research we plan to work on producing realistic synthetic data.

REFERENCES

[1] European Central Bank, "Card Fraud Report," 2018. [Online]. Available: https://www.ecb.europa.eu/pub/cardfraud/html/ecb.cardfraudreport201809.en.html#toc3.

[2] T. P. Bhatla, V. Prabhu, and A. Dua, "Understanding Credit Card Frauds," *Tata Consult.*, vol. 3, no. 1, pp. 1–14, 2003.

[3] S. Bhattacharyya, S. Jha, K. Tharakunnel, and J. C. Westland, "Data mining for credit card fraud: A comparative study," *Decis. Support Syst.*, vol. 50, no. 3, pp. 602–613, 2011.

[4] P. H. Tran, K. P. Tran, T. T. Huong, C. Heuchenne, P. HienTran, and T. M. H. Le, "Real Time Data-Driven Approaches for Credit Card Fraud Detection," *Proc. 2018 Int. Conf. E-bus. Appl. - ICEBA 2018*, no. February, pp. 6–9, 2018.

[5] J. A. Gómez, J. Arévalo, R. Paredes, and J. Nin, "End-to-end neural network architecture for fraud scoring in card payments," *Pattern Recognit. Lett.*, vol. 105, no. May, pp. 175–181, 2018.

[6] A. Dal Pozzolo, "Adaptive real-time machine learning for credit card fraud detection," no. September, 2013.

[7] Oracle, “Data Mining Concepts.” .
[8] M. Zareapoor and P. Shamsolmoali, “Application of credit card fraud detection: Based on bagging ensemble classifier,” *Procedia Comput. Sci.*, vol. 48, no. C, pp. 679–686, 2015.
[9] A. Gupta, D. Kumar, and A. Barve, “Hidden Markov Model based Credit Card Fraud Detection System with Time Stamp and IP Address,” *Int. J. Comput. Appl.*, vol. 166, no. 5, pp. 33–37, 2017.
[10] S. Maes, K. Tuyls, B. Vanschoenwinkel, and B. Manderick, “Machine Learning Techniques for Fraud Detection,” *Proc. 1st Int. naiso Congr. neuro fuzzy Technol.*, no. January, pp. 261–270, 2002.
[11] X.-Y. Liu, J. Wu, and Z.-H. Zhou, “Exploratory Undersampling for Class-Imbalance Learning,” *IEEE Trans. Syst. MAN, Cybern. B Cybern.*, vol. VOL. 39, no. 2, pp. 539–550, 2009.
[12] G. Batista, A. Carvalho, and M. C. Monard, “Applying one-sided selection to unbalanced datasets.,” *MICAI 2000 Adv. Artif. Intell.*, pp. 315–325, 2000.
[13] N. V Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “SMOTE: Synthetic Minority Over-sampling Technique,” *J. Artif. Intell. Res. 16*, pp. 321–357, 2002.
[14] T. R. Hoens, R. Polikar, and N. V. Chawla, “Learning from streaming data with concept drift and imbalance: An overview,” *Prog. Artif. Intell.*, vol. 1, no. 1, pp. 89–101, 2012.
[15] G. Ditzler and R. Polikar, “An ensemble based incremental learning framework for concept drift and class imbalance,” *Proc. Int. Jt. Conf. Neural Networks*, no. May 2014, 2010.
[16] G. Ditzler, R. Polikar, and N. Chawla, “An incremental learning algorithm for non-stationary environments and class imbalance,” *Proc. - Int. Conf. Pattern Recognit.*, pp. 2997–3000, 2010.
[17] N. V. Chawla, “Data mining for imbalanced datasets: an overview,” *DATA Min. Knowl. Discov. Handb.*, pp. 854–867.
[18] C. Phua, D. Alahakoon, and V. Lee, “Minority report in fraud detection,” *ACM SIGKDD Explor. Newsl.*, vol. 6, no. 1, p. 50, 2004.
[19] A. Dal Pozzolo, O. Caelen, Y. A. Le Borgne, S. Waterschoot, and G. Bontempi, “Learned lessons in credit card fraud detection from a practitioner perspective,” *Expert Syst. Appl.*, vol. 41, no. 10, pp. 4915–4928, 2014.
[20] I. Mekterović, L. Brkić, and M. Baranović, “A Systematic Review of Data Mining Approaches to Credit Card Fraud Detection,” vol. 15, pp. 437–444, 2018.
[21] “Credit Card Fraud Dataset.” [Online]. Available: https://www.kaggle.com/mlg-ulb/creditcardfraud.
[22] R. Akbani, S. Kwek, and N. Japkowicz, “Applying Support Vector Machines to Imbalanced Datasets,” pp. 39–50, 2004.
[23] A. Herv´e and L. J. Williams, “Principal Component Analysis,” *Wiley Interdiscip. Rev. Comput. Stat. 2*, vol. 2, 2010.
[24] E. Alpaydin, *Introduction to Machine Learning*. 2014.
[25] C. Bishop, *Pattern Recognition and Machine Learning*. 2006.
[26] K. P. Murphy, *A probabilistic perspective*. 2012.
[27] J. O. Awoyemi and S. A. Oluwadare, “Credit card fraud detection using Machine Learning Techniques : A Comparative Analysis,” 2017.
[28] A. Kumar and G. Gupta, *Fraud Detection in Online Transactions*. Springer Singapore, 2018.
[29] R. Fayzrakhmanov, A. Kulikov, and P. Repp, “The Difference Between Precision- recall and ROC Curves for Evaluating the Performance of Credit Card Fraud Detection Models,” no. 17, pp. 17–22, 2018.
[30] V. Andrei, F. David, F. Roshanak, and H. Charu, “VOS : a Method for Variational Oversampling of Imbalanced Data.”