




Leveraging augmentation techniques for tasks with unbalancedness within the financial domain: a two-level ensemble approach

Golshid Ranjbaran¹, Diego Reforgiato Recupero², Gianfranco Lombardo³ and Sergio Consoli^{4*} 

*Correspondence:

sergio.consoli@ec.europa.eu

⁴Joint Research Centre (DG JRC),
European Commission, Via E. Fermi
2749, 21027 Ispra (VA), Italy
Full list of author information is
available at the end of the article

Abstract

Modern financial markets produce massive datasets that need to be analysed using new modelling techniques like those from (deep) Machine Learning and Artificial Intelligence. The common goal of these techniques is to forecast the behaviour of the market, which can be translated into various classification tasks, such as, for instance, predicting the likelihood of companies' bankruptcy or in fraud detection systems. However, it is often the case that real-world financial data are unbalanced, meaning that the classes' distribution is not equally represented in such datasets. This gives the main issue since any Machine Learning model is trained according to the majority class mainly, leading to inaccurate predictions. In this paper, we explore different data augmentation techniques to deal with very unbalanced financial data. We consider a number of publicly available datasets, then apply state-of-the-art augmentation strategies to them, and finally evaluate the results for several Machine Learning models trained on the sampled data. The performance of the various approaches is evaluated according to their accuracy, micro, and macro F1 score, and finally by analyzing the precision and recall over the minority class. We show that a consistent and accurate improvement is achieved when data augmentation is employed. The obtained classification results look promising and indicate the efficiency of augmentation strategies on financial tasks. On the basis of these results, we present an approach focused on classification tasks within the financial domain that takes a dataset as input, identifies what kind of augmentation technique to use, and then applies an ensemble of all the augmentation techniques of the identified type to the input dataset along with an ensemble of different methods to tackle the underlying classification.

Keywords: Augmentation techniques; Ensemble method; Financial sector; Machine learning; Unbalanced data

1 Introduction

Financial Technology (Fintech) aims to introduce new approaches to improve and automate the delivery and usage of financial services [11–15]. When Fintech emerged for the

© The Author(s) 2023. **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

first time in the 21st century,¹ the term was initially referred to the back-end systems of established financial institutions. Then, there was a shift to more consumer-oriented services. Today, Fintech includes different sectors and industries such as education, retail banking, fundraising and nonprofit, and investment management.

The advent of Big Data and all the recent advances in Machine Learning (ML) and Deep Learning (DL) have the potential to revolutionize the banking industry through practical applications. Indeed, the fact that ML and DL can process a vast amount of data at high speed plays a key role that makes them suitable and applicable to real-world scenarios.

Several financial problems hide different computer science tasks related to classification [21]. For a certain classification task, given an input set (fixed) of categories and a set of objects, the goal is to assign one or more categories to each object. In practical datasets within the financial domain, depending on the underlying task, data are often distributed unevenly among classes. This makes the dataset unbalanced and leads to a decrease in the predictive performances of ML and DL approaches [70]. In particular, this phenomenon leads to classification issues for datasets where only a very small number of samples belong to a certain class (minority class) whereas the remaining classes (majority classes) have a very large number of data instances. When imbalanced datasets need to be handled, basic ML and DL approaches mainly focus on the majority classes due to their occurrences. Identifying the instances in the minority classes becomes more difficult as they are often mislabeled as noise.

To overcome the unbalancedness issue, many techniques and algorithms have been implemented to reduce the gap between the classes to classify [21]. They apply a resampling process at the data level which aims at balancing minority and majority data samples before the training of ML and DL approaches on the data. These resampling techniques can be mainly categorised into under-sampling and over-sampling approaches [2, 6]. Under-sampling is commonly performed by randomly removing data samples from the majority class of the training set [25]. However, this random process is very likely to remove critical or important data samples from the training set, resulting in a critical loss in performance of the classification algorithm [24]. In addition, the under-sampling is not even applicable to all those datasets whose unbalancedness is too extreme (like in the finance domain), since the removal process would drop too many data samples yielding a highly poor training set, which would make the training of any ML and DL approaches on such data practically impossible.

In this context, to handle highly unbalanced data, different over-sampling, or data augmentation, strategies can be adopted [80]. Their goal is to increase the size of data used for training a model by artificially replicating the data instances of the minority class adopting some intelligent look-ahead mechanisms. Within the financial domain where datasets are often highly unbalanced, augmentation techniques are beneficial and help ML and DL approaches in increasing their prediction accuracies. This is the object of the work reported in this paper. In particular, we consider several problems within the financial domain most of which have the characteristic of being associated with very unbalanced datasets and focus on various data augmentation techniques in order to properly handle such extremely unbalanced data. Specifically, we tackle the following tasks:

¹<https://thepaymentsassociation.org/article/fintech-the-history-and-future-of-financial-technology/>.

- Classifying Polish and American companies depending on whether they went in bankrupt or not;
- Classifying success or failure cases of bank marketing where an agent calls a client to sell a long-term deposit via telemarketing calls;
- Classifying a credit card transaction as regular or fraudulent.
- Classifying a credit approval as granted or not from the bank for a certain customer asking for a loan;

The datasets we have used for each task, except for the credit approval, are characterized by unbalancedness. Although the credit approval has a balanced dataset, it consists of too few instances to train well a classifier. We aim to experiment with our proposed augmentation strategies in this particular kind of context too. Thus, we apply state-of-the-art augmentation strategies to all the datasets and evaluate the results for several ML and DL approaches. The considered augmentation strategies are of two kinds: i) those that augment the instances associated with the minority class and ii) those that augment evenly a balanced dataset represented by too few instances. We then compare the classification results against two levels of baselines: i) when no augmentation strategies are applied or ii) when basic augmentation strategies are used to increase the training data. On the basis of the results, which prove the benefit brought by the augmentation techniques, we present a two-level ensemble approach to perform classification within the financial domain which: i) takes a dataset as input, ii) identifies from its features whether it contains (and what kind) any unbalance issue, iii) applies all the augmentation techniques that work best with the discovered kind, iv) performs an ensemble of different ML methods using the different augmented datasets. The reader notices that a first-level ensemble is applied when generating the augmented datasets and a second-level ensemble is used when employing several ML methods for the classification task.

Therefore, the contributions of this paper are the following:

- We tackle various tasks within the financial domain characterized by different kinds of unbalancedness in the data that we had to deal with using augmentation techniques;
- We leverage state-of-the-art augmentation techniques to balance the training data and to bring benefits to the subsequent classification task;
- We show that in every case, augmentation techniques are beneficial for classification tasks within the financial domain and suggest best practices to adopt such strategies in other similar problems in the same domain;
- We adopt augmentation techniques on the bankruptcy dataset in the USA, which has been recently presented in [48] and has only been used for classification tasks by just leveraging undersampling techniques because of the strong unbalance condition of the bankruptcy class;
- We define a two-level ensemble approach focused on the financial domain for classification tasks that can be used to select the best combination (augmentation, ML) approach and also to evaluate any method trying to automatically infer either the augmentation approach to use or the ML approach to run;
- We share the code in a public repository² and keep it general so that it is possible to replicate our work and easily adapt it to tackle other classification tasks suffering from similar unbalance problems.

²<https://github.com/golshidr/Augmentation-methods-datasets>.

The remainder of this paper is organized as follows. Section 2 contains the related work on augmentation techniques for tasks within the financial domain. Section 3 describes the financial tasks that we have considered and tackled in this paper. The augmentation techniques that we have analysed and used within the mentioned tasks are detailed in Sect. 4. In Sect. 5 we present an overview of the ML algorithms we used to address the considered financial problems. The experiments that we have carried out containing the obtained results for the aforementioned tasks and the proposed augmentation techniques are described in Sect. 6. Furthermore, Sect. 7 includes the details of a two-level ensemble approach, previously mentioned, that we propose in this paper. Finally, Sect. 8 ends the paper with conclusions and future directions where we are headed.

2 Related work

A number of problems in finance can be formulated from a ML perspective as classification tasks, where is often the case that the class to categorize is much smaller than the number of samples in the other classes [70]. As an example, suppose you want to classify in a financial market the companies which will bankrupt. Here, the number of bankrupt companies is much smaller than the others, and the performance of any classifier trained on this data is generally poor, since it is complex to estimate an effective decision boundary to distinguish such companies from the healthy ones with few observations. As a result, the minority class of bankrupt companies is typically categorized as data outliers or even noise of the healthy companies distribution. On the other hand, the design choice of undersampling the majority class of healthy companies leads to an improvement of the recall over the bankruptcy events but a very poor precision over the healthy ones. In both cases, performance is low, especially in dynamic and temporal contexts where the data distribution is not stationary over time [48]. The issue of imbalanced data in the financial domain is therefore very critical, and a number of research works have appeared in the literature to handle various financial problems by resampling the data to balance minority and majority classes before training on a classifier [65]. Augmentation techniques, in particular, obtain a balanced dataset by increasing the individuals of the minority class with new synthetic samples [80], and are usually employed when highly unbalanced data need to be handled [24].

In this context, to manage a different over-sampling or data augmentation, different strategies can be adopted [80]. Among these, maybe the most used and effective one is the Synthetic Minority Oversampling Technique (SMOTE) by Chawla et al. [17], which generates synthetic data for the minority class by using the similarities computed with k -Nearest Neighbors (kNN) for each of the minority samples. Veganzones and Séverin [74] used SMOTE in the context of bankruptcy prediction from the financial ratios of a large set of companies in France. Dal Pozzolo et al. [26] adopted SMOTE and an ensemble of incremental learning classifiers for the detection of fraudulent credit card transactions. A main disadvantage of SMOTE is that the over-sampled synthetic data may overlap in some cases with samples of the majority class, creating redundancy in the training phase of the ML algorithm. To deal with this issue, variations of this method have also arisen in the literature. Le et al. [43] considered various resampling techniques based on SMOTE to improve the performance of basic classifiers on the Korean companies' bankruptcy data, ranging from 2016 to 2017. Similarly, in [31] the authors tested a hybrid approach combining SMOTE with an ensemble of classification algorithms for bankruptcy prediction

on a real dataset from Spain. Pranavi et al. [60] combined SMOTE with a Random Forest for detecting fraudulent transactions, increasing the overall classification accuracy of the algorithm to a remarkable 90%. Garcia [33] proposed a combination of SMOTE with cluster-based under-sampling, leading to promising classification results for bankruptcy prediction. In [45] the authors proposed a fast and accurate ML model called XGBS, using the extreme gradient boosting model and the squared logistics loss (SqLL) for handling the bankruptcy forecasting problem, and validating the approach on imbalanced datasets for firms in Korea, US, and Japan. In [44], the authors developed an ensemble approach to handle the problem of data imbalance in bankruptcy forecasting, combining three algorithms, namely the CBoost algorithm [46], the technique with a cost-sensitive (HAOC) framework, and the XGBS algorithm [45]. The CBoost uses the k-means clustering algorithm to calculate the initial weight vector for the training set. Next, CBoost performs several iterations to determine a set of weak classifiers, and finally, XGBS combines this set to create the final classifier.

Other relevant augmentation techniques for financial unbalanced problems have been also proposed in the literature. Authors in [36] proposed a novel boosting regression data resampling method based on a conditional variational autoencoder that can be used in different tasks for regression including unbalanced datasets. Others designed deep learning approaches for the prediction of hourly movement directions of different banking stocks leveraging stock prices and technical features [34]. These last were reduced through a recursive feature elimination selection [81].

Alarfaj et al. [3] compared different decision tree splitting criteria for credit card fraud detection, deriving a new measure for separating class samples which obtains decision tree solutions with higher performance. Alfaiz and Fati [4] considered over-sampling in various ML techniques, including random forests, decision trees, logistic regression, support vector machines, and artificial neural networks, to detect fraudulent credit card transactions, achieving top performance when the data augmentation was included relative to the models alone. Last, but not least, Chugh and Malik [19] employed random forest and the Adaboost algorithm with data augmentation to detect fraudulent transactions in various countries.

Beside the issue of imbalanced data, it is worth mentioning another important problem when handling economic and financial data, which is data imputation. It is common to find missing values in financial time series, like e.g. stock market data. The reasons might be diverse, like the closing periods of markets during holidays, or the inability to capture financial data in a specified period of time, recording errors and noise, and so on [68]. Missing data makes it daunting to predict future financial time series points using the most up-to-date market information. Thus, when the problem of missing data arises, hence there is an urgent need to handle it [72]. A number of data imputation methods have been proposed in the literature. For a comprehensive overview the reader is referred to [18, 67], and in particular to [39] for specifically handling financial time series.

Another common issue that is often encountered in financial time series is conditional heteroskedasticity [42]. In these cases, the level of volatility cannot be predicted over time, and weighted regressions represent a frequent approach to produce estimations [20, 47]. With heteroscedasticity, the least squares assumption of constant variance in the residuals is violated. Weighted regression minimizes, with the correct weight set, the sum of weighted squared residuals to produce residuals with a constant variance [20].

As it will be shown in the following, in our paper we have used data augmentation on various classical, very unbalanced, financial tasks. We have sampled the datasets using different data augmentation techniques to overcome class imbalances. The balanced datasets were used to train a set of popular classifiers, which have been validated according to common classification metrics. In line with the findings of the state-of-the-art works that applied specific augmentation techniques and specific machine learning or deep learning approaches, we have proved the benefit of augmentation techniques on different financial tasks involving unbalanced datasets and on different machine learning methods. Also, ours is the first work to propose a classification task using augmentation techniques on a new dataset related to the bankruptcy of publicly traded companies in the American stock market.

3 Proposed tasks

In this section, we describe the tasks that we have tackled in this paper.

3.1 Bankruptcy prediction

Corporate bankruptcy prediction is one of the main tasks in credit risk assessment due to its economic damage and social consequences. After the 2007/2008 financial crisis, it has become a priority for most financial institutions, regulatory agencies, and academics [66]. Bankruptcy prediction has been widely researched as a binary classification problem with several ML techniques [27, 54, 78, 84]. The basic goal is to assess the likelihood of companies' default by looking for relationships among different types of financial data, and the financial status of a firm in the future [30]. Barboza et al. [7] show that, on average, ML models exhibit 10% higher accuracy than scoring-based ones [55, 77]. Specifically, Support Vector Machines (SVM), Random Forests (RF), as well as bagging and boosting techniques were tested for predicting bankruptcy events and compared with results from the discriminant analysis, Logistic Regression, and Neural Networks. However, the main open problem related to this task consists in dealing with a very large imbalance among the classes due to the rarity of bankruptcy events in the real economy. This issue becomes even worse when considering DL approaches that usually require a vast amount of data for training [40, 51]. Private and public companies have in general different dynamics that may significantly impact the probability of dealing with financial troubles like bankruptcy. In light of this, to further analyze our results, we applied the same methodology to two different publicly available datasets:

- *Bankruptcy prediction for private companies*: This dataset is related to the bankruptcy prediction of private companies in Poland between 2000 and 2012. The dataset has been proposed in [84] and is publicly available on the UCI ML Repository.³ It provides financial statements for both healthy and bankrupted companies in their last 5 years of activity. Table 1 shows the total number of companies under investigation, within each time frame. Each company-year observation is composed of 64 accounting variables from the financial statements. A detailed description of these features is provided in the UCI repository.
- *Bankruptcy prediction for public companies in the stock market*: This dataset is related to the bankruptcy prediction of publicly traded companies in the American stock

³<https://archive.ics.uci.edu/ml/datasets/Polish+companies+bankruptcy+data>.

Table 1 Statistical data about the bankruptcy dataset for Polish companies

Time frame	Financial statements (features)	Total companies	Bankrupt companies	Active companies
1st Year	64	7027	271	6756
2nd Year	64	10,173	400	9773
3rd Year	64	10,503	495	10,008
4th Year	64	9792	515	9277
5th Year	64	5910	410	5500

Table 2 Firms distribution by year for the American stock market dataset

Year	Total firms	Bankrupt firms	Year	Total firms	Bankrupt firms
2000	5308	3	2010	3743	23
2001	5226	7	2011	3625	35
2002	4897	10	2012	3513	25
2003	4651	17	2013	3485	26
2004	4417	29	2014	3484	28
2005	4348	46	2015	3504	33
2006	4205	40	2016	3354	33
2007	4128	51	2017	3191	29
2008	4009	59	2018	3014	21
2009	3857	58	2019	2723	36

market (New York Stock Exchange and NASDAQ) for the period between 1999 and 2018. It has been proposed in [48] and is publicly available on GitHub.⁴ It provides data from 8262 different companies and, in particular, 18 accounting variables for each fiscal year. Companies are labelled each year depending on their state in the next year and according to the Security Exchange Commission (SEC) rules. Table 2 shows the dataset distribution.

3.2 Bank marketing

Marketing managers try to improve the effects of their campaigns by carefully choosing the target audiences and the best communication channels. ML techniques can be used to improve these direct marketing initiatives [76]. One of the freely available datasets collected for this purpose is from the Portuguese marketing campaign companies.⁵ Deposit subscriptions as well as actual statistics were gathered from a marketing campaign of a Portuguese banking institution. Finding a model that can explain a contact's success, or whether a client subscribes to a deposit, is the business's goal. Better use of the available resources (such as human effort, phone calls, and time) and the selection of a high-quality and affordable group of potential buyers are all advantages of using such a strategy to boost campaign efficiency [53]. In this task, telemarketing calls are used to sell long-term deposits to target clients. Depending on who initiated the contact (the client or the contact center), contacts can be categorized as inbound or outbound. Both categories are present in the dataset. Human agents call a list of clients during a campaign to sell the deposit (outbound); otherwise, if the customer calls the contact center for another reason, he/she is requested to subscribe to the deposit (inbound) [53]. The result is an interaction that can either be successful or a failure, which translates to a binary classification task to solve.

⁴https://github.com/sowide/bankruptcy_dataset.

⁵<https://archive.ics.uci.edu/ml/datasets/Bank+Marketing>.

Table 3 Examples of some Bank Marketing features

Name of feature	Description
Age	age of contact
Marital status	Married, Single, Divorced, Windowed
Annual balance	in euro currency
Debit card	Yes or No
Loans in delay	Yes or No
Agent	Human that answered the call
Date and Time	Referring to when the contact was made
Duration	Of the contact (in seconds)

This study takes into account actual data that was gathered from a Portuguese retail bank between May 2008 and June 2013, for a total of 52,944 phone contacts. Only 6557 entries of the dataset are related to successes, rendering the samples highly imbalanced. Each record included the output target, the contact outcome (“failure”, “success”), and candidate input features like age, education, housing, marital status, etc. Some of these features are text, so they must be encoded into numbers. Some bank marketing features considered in the research studies are reported in Table 3. In the mentioned table, for example, the age, marital status, and annual balance of each consumer are specified. It has also reported whether the customer owns a debit card, if he/she experienced a loan payment delay, etc. The customer’s agent who made the call, the date, and the duration of the conversation are additional details further included in the dataset.

3.3 Credit card frauds

Fraud in credit cards is a growing problem as more and more transactions are conducted online. Researchers have been using different machine learning methods to detect and analyze frauds in online transactions [10]. Credit card fraud is a significant issue that affects millions of people worldwide. According to the Federal Trade Commission (FTC), consumers reported losing more than \$5.8 billion to fraud in 2021, an increase of more than 70 % over the previous year. The FTC received fraud reports from more than 2.8 million consumers last year [32]. In addition to the direct financial losses suffered by consumers, credit card fraud also has broader economic impacts. According to the Nilson Report, U.S. losses from card fraud are forecasted to total \$165.1 billion over the next 10 years [62]. Datasets on credit card frauds are extremely unbalanced, and with a large difference in the number of samples between legal and fraud transactions [83]: indeed, on average, 98% of real-world transactions are legal, and only 2% are frauds. There are several datasets available for credit card fraud detection. One such dataset is available on Kaggle [73] and contains transactions made by credit cards in September 2013 by European cardholders. This dataset presents transactions that occurred in two days, where there are 492 frauds out of 284,807 transactions. The dataset is highly unbalanced, with the positive class (frauds) accounting for only 0.172% of all transactions [73]. The numerical input variables contained in the dataset are derived from a Principle Component Analysis (PCA) transformation [25] since it is not possible to obtain the original features and additional context due to confidentiality concerns. Only the original features “Time” and “Amount” have been retained and not changed by the PCA approach. The other features are then the principal components derived by PCA. The target variable, labeled “Class”, takes value 1 in the event of fraud, and 0 in all other circumstances [26]. The dataset is public and can be freely obtained from the Kaggle website [73].

Table 4 Representation of the features in the credit card approval dataset

Feature	Type	Values
A1	Nominal	a, b
A2	Continues	13.75 - 80.25
A3	Continues	0 - 28
A4	Nominal	u, y, l, t
A5	Nominal	g, p, gg
A6	Nominal	c, d, cc, i, j, k, m, r, q, w, x, e, aa, ff
A7	Nominal	v, h, hbb, j, n, z, dd, ff, o
A8	Continues	0 - 28.5
A9	Nominal	t, f
A10	Nominal	t, f
A11	Continues	0 - 67
A12	Nominal	t, f
A13	Nominal	g, p, s
A14	Continues	0 - 2000
A15	Continues	0 - 100,000
Class	Nominal	+ , -

3.4 Credit approval

The procedure a company or person must go through in order to be approved for a loan or to be able to pay for products and services over a long period of time is known as credit approval. The decision to approve a credit application is based on the lender's assessment of the borrower's ability and willingness to repay the loan, or pay for the purchased products plus interest in a reasonable time frame, as well as the creditor's desire to lend the money. Typically, companies get permission to withdraw money from client accounts and also permit clients to bank loans from them. Between the borrower and the lender, credit implies a condition of trust for the repayment of the borrowed money (or products) [75]. We focus on the real-world Australian credit approval dataset, which can be publicly obtained from the UCI ML Repository.⁶ There are 690 cases in this dataset [63], of which 307 involve creditworthy applicants and 383 involve uncreditworthy ones. Each instance has class labels (accepted or rejected), eight numeric attributes, and six nominal attributes. The attribute types in the dataset are well-balanced and include continuous, nominal with few classes, nominal with many classes, as well as a few missing values. To preserve the privacy of the data, all feature names have been altered to anonymous values. Nonetheless, there are 67 records of missing values in 7 feature attributes. In Table 4 the features of the dataset can be seen.

Although this dataset is balanced, the contained data samples are too few with respect to the included features, making the training of a classifier upon it extremely difficult and low performing. Therefore, we included this dataset in our study to experiment with our proposed augmentation strategies also in this particular context.

4 Augmentation techniques

As there are insufficient samples of the minority class, imbalanced classification has the limitation that a model cannot efficiently learn the decision boundary. Given that under-sampling approaches are not well-suited to properly handle extremely unbalanced data [25], as it is in our considered financial datasets, in our work we focused on three categories of data augmentation methods. The first category, which represents our baseline,

⁶<https://archive.ics.uci.edu/ml/datasets/Credit+Approval>.

uses statistical information from each data column, the second category is based on the SMOTE approach [17], and the last category is based on deep neural networks [80].

As far as the first category is concerned, its purpose is to define a baseline and empirically prove that it is outperformed by the approaches of the other categories. For the baseline methods, we exploited the information about each feature of the input datasets. Statistical functions such as minimum, maximum, mean, and standard deviation have been used to create synthetic data. We defined two different baselines using such information.

For the second category, SMOTE is used to generate synthetic data points from raw data. This algorithm generates new instances of the minority class by creating convex combinations of neighbouring instances [17]. It takes samples of the feature space for each target class and its neighbours and then generates new instances that combine the features of the target cases with the features of its neighbours. It is worth noting that the data generation process employed by SMOTE is intrinsically linear, as most data augmentation techniques. Therefore, if the minority class does not have enough representative samples, there is no guarantee that the given data distribution reflects the (true) underlying data distribution (in other words it may not constitute a representative sample in the statistical sense). The newly generated data points would not be able to introduce much variance to the data, being only slightly different than the original points, which could potentially lead to a bias in the estimation [58]. Thus, for these cases, oversampling the whole data, without extra assumptions about the underlying distribution, is an unbiased approach in the statistical sense. To overcome this issue, it is crucial to perform the splitting of the data between training and test sets, and only afterwards balance the data uniquely in the training set. In this way, the test is left as unbiased as possible in order to get an objective evaluation of the model's performance.

The third category is represented by a state-of-art data augmentation algorithm that is based on deep neural networks, namely the Variational Autoencoders (VAEs) approach [21]. As it has been shown in the survey by Wen et al. [80], VAEs perform very well in problems with an extremely shrunk number of data samples. This kind of augmentation methodology can be employed also in problems where, although balanced, there are a few data samples for training learning models, as is the case, for instance, in the classification of credit approvals that we further consider in our study. VAEs estimate the Probability Density Function (PDF) of the training data and next the model samples the learned PDF to generate new data records that are similar to the original dataset distribution [41]. Relative to classical augmentation techniques like SMOTE, this generation process has the advantage to maintain the original data statistical properties minimising the introduction of bias during the process.

In the following sections, we describe in detail the two used baselines, the SMOTE-based methods, and the Variational Autoencoders approach.

4.1 Baseline: min-max approach

The first baseline is pretty straightforward. Here, the minimum and maximum of each column of the various datasets have been calculated, and a random number has been generated from this interval for each column and assigned as a new feature to a new synthetic data sample, which is then additionally included in the original dataset. Basically, for a given feature i , the algorithm generates random values x_i , such that $minimum_i \leq x_i \leq maximum_i$, to create the new additional data record.

Table 5 An example of the Min-Max approach. The reported samples are labelled with the minority class, and it is shown the generation of two additional data samples

Minority class				
#	Feature 1	Feature 2	Feature 3	Class
1	58	2143	12	1
2	33	231	19	1
3	47	446	14	1
4	33	512	87	1
Min	33	231	12	
Max	58	2143	87	
$Min \leq x_i \leq Max$	$33 \leq x_i \leq 58$	$231 \leq x_i \leq 2143$	$12 \leq x_i \leq 87$	
Synthetic Data	45	467	54	1
Synthetic Data	52	1232	18	1

Table 6 An example of the Min-Max approach. The reported samples are labelled with the majority class

Majority class				
#	Feature 1	Feature 2	Feature 3	Class
1	35	1787	18	0
2	61	675	26	0
3	42	245	78	0
4	48	545	39	0
5	51	891	10	0
6	31	880	2	0

Let us consider a dataset consisting of two classes and 3 features. Class 1 is the minority class whereas class 0 is the majority, and the data augmentation method is supposed to increase the number of samples of the minority class.

An example of the data augmentation method with the Min-Max approach can be seen in Tables 5–6. In Table 6 six samples of the majority class are illustrated, whereas in Table 5 the four samples of the minority class can be seen. The minimum and maximum of each column in the minority class are calculated separately, and a random number is chosen in this interval for each of the three features in order to generate synthetic data. In the mentioned interval, as many synthetic data as needed can be generated randomly. In particular, in the example reported in Table 5, two synthetic data are required to have the equivalent number of records between the minority and majority classes.

4.2 Baseline: mean-std approach

In this approach, for each column, we compute the mean and the standard deviation. Then, for a certain feature i , each new value x_i is generated as $x_i = Std_i - Mean_i$.

As mentioned before, let us consider a dataset with three features and two classes. Class 1 is the minority class, class 0 is the majority class, and the Mean-Std approach is intended to increase the amount of minority class samples.

Tables 7–8 provide the example of the data augmentation procedure by means of the Mean-Std Approach. Table 8 shows six elements for the majority class. In Table 7 four elements for the minority class are displayed as well as the mean and the standard deviation feature. Then, two synthetic data are created by subtracting the standard deviation from the mean. The disadvantage of this method is that it produces constant numbers for the values of the same features of the new synthetic data, creating redundancy.

Table 7 An example of the Mean-Std approach. The reported samples are labelled with the minority class, and it is shown the generation of two additional data samples

Minority class				
#	Feature 1	Feature 2	Feature 3	Class
1	58	2143	12	1
2	33	231	19	1
3	47	446	14	1
4	33	512	87	1
Mean	42.75	833	33	
Std	10.49	763.43	31.28	
$x_i = \text{Mean} - \text{Std}$	42.75 - 10.49	833 - 763.43	33 - 31.28	
Synthetic Data	32.26	69.57	1	1.72
Synthetic Data	32.26	69.57	1.5	1.72

Table 8 An example of the Mean-Std approach. The reported samples are labelled with the majority class

Majority class				
#	Feature 1	Feature 2	Feature 3	Class
1	35	1787	18	0
2	61	675	26	0
3	42	245	78	0
4	48	545	39	0
5	51	891	10	0
6	31	880	2	0

4.3 SMOTE

One way to solve the imbalance problem is to duplicate minority samples. However, this does not provide any new information to the machine learning algorithm training on the data. SMOTE creates new synthetic samples by interpolating between existing minority class samples [50]. More precisely, SMOTE [17, 52] generates as many entries as the minority class until there is the same number of entries in both classes. SMOTE has been widely used and has been shown to be effective in many applications. It has also been extended and modified in various ways to improve its performance in specific scenarios [71].

SMOTE generates synthetic data for the minority class using the nearest neighbour of the data samples. New instances that combine the features of the target class with the features of its neighbours can be generated. The fundamental concept is that k -nearest neighbours of the samples are used to generate a synthetic instance for the minority class. The k -nearest elements are chosen from the samples in the minority class. Afterwards, the SMOTE algorithm randomly selects n samples and saves them. Suppose the new data samples are named X_i . The new samples X' are generated based on the following equation:

$$X' = X + \text{rand} \cdot (X_i - X), \quad i = 1 \dots n,$$

where *rand* follows a uniform distribution in the range (0, 1).

In Fig. 1 a graphical representation of the generation process of the SMOTE algorithm is shown. The main idea is that to generate a synthetic instance for the minority class, k -nearest neighbours of the samples are used. The samples of the minority class are shown in green color and the majority class is shown in blue. To create a new sample, the distances among the samples of the minority class are calculated, and new data can be created within

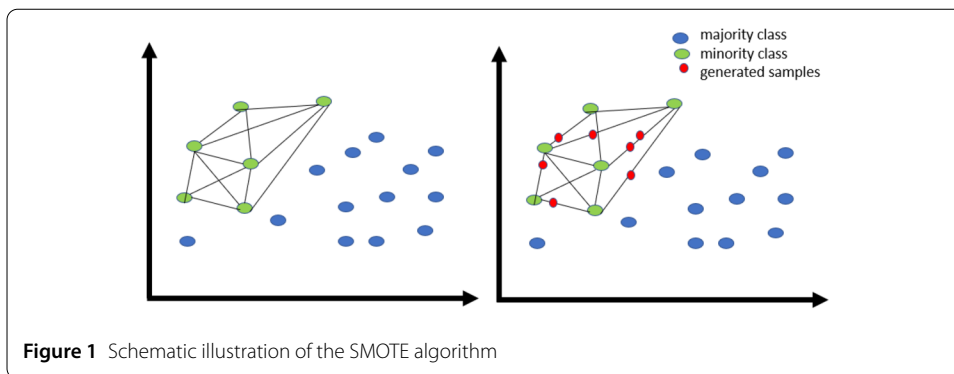


Table 9 A numerical example of SMOTE augmentation

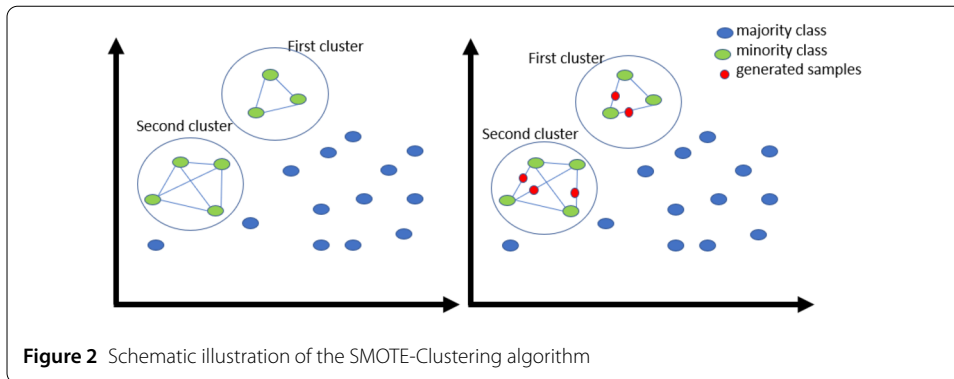
Consider a sample (10, 14) and let (8, 11) be its nearest neighbour.
 (10, 14) is the sample for which k-nearest neighbours are being identified.
 (8, 11) is one of its k-nearest neighbours.
 Let us consider:
 $x_{1,1} = 10, x_{1,2} = 14, x_{1,2} - x_{1,1} = 4$
 $x_{2,1} = 8, x_{2,2} = 11, x_{2,2} - x_{2,1} = 3$
 The new samples are generated as
 $(x'_1, x'_2) = (10, 14) + \text{rand}(0, 1) * (4, 3)$
 $\text{rand}(0, 1)$ generates a random number between 0 and 1.

the distances among samples (in the Euclidean space case, they would be the lines created connecting each pair of samples) until both classes have the same size. The generated data is shown in red in the right illustration reported in Fig. 1. Table 9 shows a numerical example of the overall generation process of SMOTE.

4.4 SCUT: SMOTE-clustering

SMOTE and Clustered Undersampling Technique (SCUT) [1] is an over-sampling method derived from SMOTE using the Expectation Maximization (EM) algorithm. The EM algorithm down-samples the hard clusters generated iteratively by the classical SMOTE with a Gaussian probability distribution. Each member has then a probability of belonging to a certain Gaussian instead of just being assigned to a specific cluster. An advantage of using EM is that the number of clusters does not have to be specified beforehand. EM assigns a probability distribution to each instance relative to each particular cluster.

Figure 2 shows a graphic view of how the SCUT algorithm works. Instead of just being assigned to a certain cluster as SMOTE, each synthetic data has a chance of falling into a particular Gaussian distribution. This algorithm works similarly to SMOTE, with the difference that it is slightly improved by using the Gaussian probability distribution, such that it generates more accurate data. The data of the class that is going to be increased is divided into several clusters, and the new data is assigned with a probability in each of these clusters. In the example given in Fig. 2, the data of the minority class are clustered into two clusters, and the new data in the red color are placed in these clusters (on the right side of the figure).



4.5 VAEs: variational autoencoders

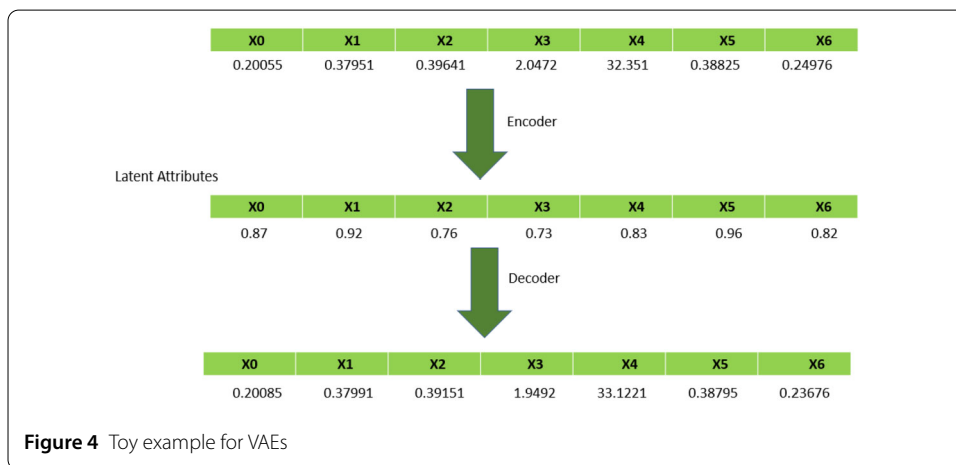
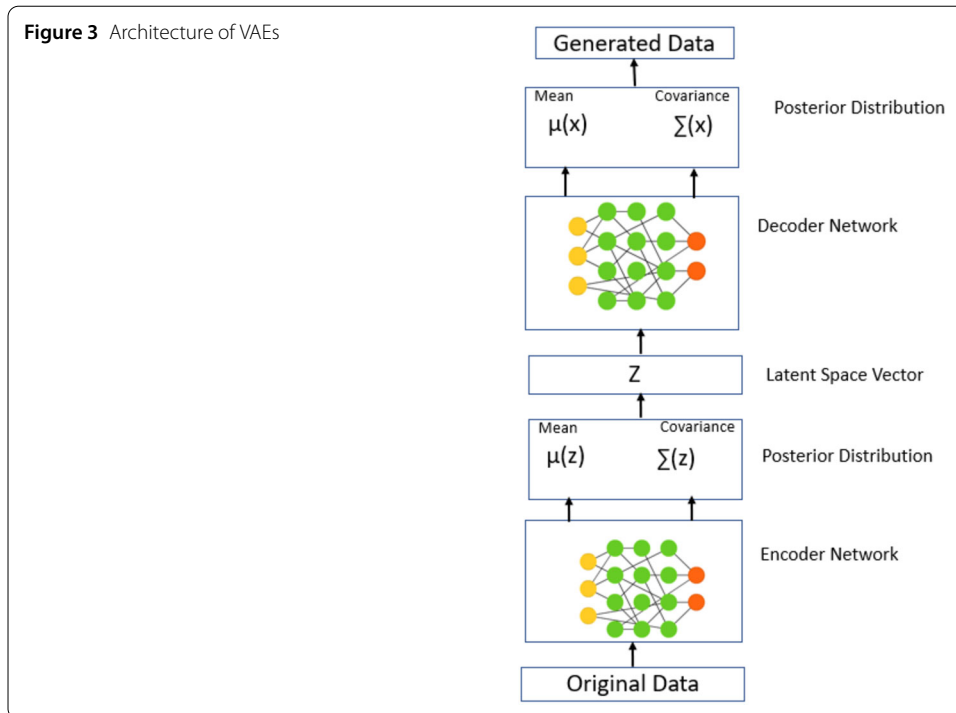
A Variational Autoencoder (VAE) is a generative model that can be used for data augmentation. It consists of an encoder network that maps input data to a latent space and a decoder network that maps points in the latent space back to the data space. By sampling points in the latent space and decoding them, new synthetic samples can be generated [37, 79].

A VAE is composed of a (deep) neural network’s encoder, a decoder, and a loss function. VAE is a technique used in probability models to describe approximate inference in a latent Gaussian model where the approximate posterior and model likelihood is parametrized by neural networks. Generally, an autoencoder network is made of a pair of two connected networks: an encoder and a decoder. The procedure starts with the encoder compressing the original data into a short code ignoring the noise. Then, the decoder uncompresses that code to generate data as close as possible to the original input [28].

VAEs are auto-encoders that encode inputs as distributions rather than points, and their hidden space is tuned by constraining the distributions returned by the encoder to be close to the standard Gaussian [64]. VAEs are a probabilistic version of autoencoders that address the problem of latent space irregularity. They allow the generation of synthetic data with different attributes. VAEs can be seen as the decoder part of an autoencoder which learns the set of parameters θ to approximate the conditional $p_{\theta}(x|z)$ to generate data, based on a sample from a true prior, $z \sim p_{\theta}(z)$. The true prior $p_{\theta}(z)$ is generally distributed as a standard Gaussian [41], that is:

$$P_{\theta}(x) = \int P_{\theta}(x|z)P_{\theta}(z) dz.$$

In Fig. 3 the architecture of VAEs is shown. The idea is to map the original data to a latent space (by the encoder) and reconstruct back values from the latent space into their original dimension (by the decoder). Although this encoded-decoded double transformation initially seems computationally onerous, it is only required to formulate a quantified reconstruction error. The VAEs training goal is to reduce this error, which converts it into the appropriate transformation function while another regularization regulates the latent distribution’s shape. Figure 4 provides a brief example of how VAEs behave. Seven features from the original data are coded into the latent space. For each feature, the latent attribute is extracted and the latent vector is generated. This latent vector is converted into fresh synthetic data with the aid of the decoder.



5 ML algorithms

In this section, we provide an overview of the ML algorithms that we have used for our financial classification tasks. It is worth noting that, given that the goal of our work is to prove whether or not the inclusion of data augmentation helps in improving the classification performance for highly imbalanced financial problems, we have used classical implementations for the considered ML algorithms with standard parameters settings.⁷

5.1 Naive Bayes classifier (NB)

The Naive Bayes Classifier [82] is a supervised ML technique using a training dataset with given target classes to predict the class of upcoming instances. In its most basic form, a

⁷<https://scikit-learn.org/>.

Naive Bayes technique assumes that the “presence or absence” of one attribute in a set is independent of the presence or absence of any other attributes in the same set. The NB method performs comparably well relative to other popular ML algorithms for classification, according to a variety of experiments carried out on real-world datasets (see for example the work by Osisanwo et al. [57]). This method takes the assumption that the class-given attributes are independent. Following that, the classification is carried out using the “Bayes” method to determine the likelihood of the correct class of new instances [38]. In Bayesian statistics, the prior probability is the likelihood of an event occurring before fresh data are gathered. This is the most logical estimate of the probability of an outcome based on the available data before an experiment. It is defined as follows.

$$\text{PRIOR PROBABILITY OF } X : \left(\frac{\text{Number of } X \text{ instances}}{\text{Total number of instances}} \right),$$

$$\text{LIKELIHOOD OF } Y \text{ GIVEN } X : \left(\frac{\text{Number of } X \text{ in Presence of } Y}{\text{Total number of } X} \right).$$

According to the Bayesian theory, the final classification is then created by integrating the two sets of data (likelihood, prior) to create a posterior probability, representing the classification outcome provided by the algorithm:

$$\text{POSTERIOR} = \left(\frac{\text{Prior} * \text{Likelihood}}{\text{Evidence}} \right),$$

where the Evidence is a scaling factor depending only on the observations X .

We have developed the Naive Bayes Classifier using the sklearn library in Python⁸ with the default algorithmic parameters.

5.2 Support vector machines (SVMs)

SVMs were first developed by Cortes and Vapnik [22] with the specific aim of addressing binary classification problems. Given the input parameters $x \in X$ and their corresponding output parameters $y \in Y = \{-1, 1\}$, the separation between classes is achieved by fitting the hyperplane $f(x)$ that has the optimal distance to the nearest data point used for training of any class, that is:

$$f(x) = \sum_{i=1}^n \alpha_i y_i \langle x_i, x \rangle + b, \quad (1)$$

where n is the total number of parameters. The goal is to find the hyperplane which maximizes the minimum distances of the samples on each side of the plane. However, the solution to the above problem is not always possible, since fitting a plane could result in samples being on the “wrong” side of the plane. To account for this, a penalty is associated with the instances which are misclassified and added to the minimization function. This is done via the parameter C in the minimization formula:

$$f(x) = \omega^T x + b \frac{1}{2} \|\omega\|^2 + C \sum_i^n c(f, x_i, y_i). \quad (2)$$

⁸https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.GaussianNB.html.

By varying C , a trade-off between the accuracy and the stability of the function is defined. Larger values of C result in a smaller margin, leading to potentially more accurate classifications, however, over-fitting can occur. The above approach only allows for the separation of linear data. In most real-world problems, this is not the case. To overcome this issue, a mapping of the data into a richer feature space, including non-linear features is applied prior to the hyperplane fitting. For the purpose of this mapping, kernel functions $k(x, x')$ are used.

Different kernel functions can be used, including exponential, polynomial, sigmoid kernels, or Gaussian radial-basis (RBF) [5]. In our paper, we focus on the latter, which represents a widely used kernel performing well in many works in the literature:

$$K(x_i, x') = \exp(-\gamma \|x_i - x'\|^2), \quad \gamma > 0, \quad (3)$$

where γ defines the variance of the RBF, practically defining the shape of the kernel function peaks: lower γ values set the bias to low and corresponding high γ to high bias. For our implementation, we have used the SVM model available in the scikit-learn library in Python⁹ and used the default parameters setting.

5.3 Multilayer perceptron (MLP)

This technique belongs to the category of feed-forward artificial neural networks. At least three layers of nodes make up an MLP: the input layer, the hidden layer, and the output layer. Each node is a neuron that utilises a non-linear activation function, except for the input nodes. In an MLP, the data flow from the input to the output layer in the forward direction, much like a feed-forward network. The backpropagation learning approach is used to train the MLP's neurons. In contrast with a linear perceptron, MLP has numerous layers and uses nonlinear activation functions. It is capable of separating data that cannot be separated linearly. Each input data has n features and is fed to the perceptron neural network. One of the input data enters the neural network at each step. The perceptron neural network generates an output proportional to the input data and weights by using a set of weights (W) and a bias value [29]. The perceptron can be used to define a linear decision boundary with this discrete output, which is controlled by the activation function. The separation hyperplane between misclassified data and the decision boundary is determined to be as short as possible. The equation which modifies the network weights with the help of the activation functions is represented by:

$$y = \text{function} \left(\sum_{i=1}^n w_i I_i + \text{Bias} \right) = \text{function}(W^T I + \text{Bias}),$$

where y is expected to map an input vector I to an output class, *function* is the activation function, W is the set of parameters, or weights, in the layer, and b is the bias vector. Note that in the case of a deep MLP neural network, which is composed by a set of consecutive MLP layers, the input vector of a layer, I , is the output of the previous layer. We have developed the Multilayer Perceptron Classifier with the popular Rectified Linear Unit

⁹<https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html>.

(ReLU) activation function and Adam optimization solver, and using the sklearn library in Python¹⁰ with the default parameters' values.

5.4 K-nearest neighbours (KNN)

The KNN algorithm is a supervised ML algorithm that can be used to solve classification problems [35]. One of its advantages consists of the simplicity of its implementation. The KNN approach is indeed a simple approach that classifies any new instances based on a similarity metric. The KNN algorithm is an example of an instance-based learner which works as follows. Each new instance is compared to the previous ones using a distance metric, and the class is assigned to the current instance based on the closest previous instance. The majority class of the closest k neighbours is allocated to the new instance when more than one nearest neighbour is used. A distance or similarity metric between two data instances must be defined for the K-Nearest Neighbor, such as, e.g., the Euclidean distance, Manhattan Distance, and Cosine similarity, among others [35]. We have developed the KNN Classifier adopting experimentally the Euclidean distance and using the scikit-learn library in Python¹¹ with the default parameters' values.

5.5 Random forests

Random Forests are a class of supervised ML algorithms that are widely used to handle classification problems [9, 23]. On various data samples, it builds decision trees and uses its majority vote to classify data. Decision tree predictions are used by the Random Forest algorithm to determine the classification outcome. It predicts by taking the average or mean of the output from the various trees. The accuracy of the result usually improves as the number of trees grows, although the algorithm complexity increases yielding higher computational running times. Finding a good balance is indeed very important for the performance of the algorithm. A slight change in the data can result in a big change in the decision tree's structure, which can lead to instability. On the other hand, overfitting of the data is a problem in decision trees, but not in random forests where the numerous trees decrease the occurrence of that event. Random Forests are therefore able to boost the precision of the classification task [59]. We have developed the Random Forests Classifier using the scikit-learn library in Python¹² and using the default parameters' values.

5.6 Stochastic gradient descent (SGD)

Stochastic Gradient Descent (SGD) is a popular machine-learning approach for classification that uses an optimization routine to determine the parameters of an adopted kernel that minimizes the cost function of the classification task [49]. SGD has been used to solve large-scale, sparse ML problems that are frequently experienced in classification. The approach basically implements a standard stochastic gradient descent learning procedure that supports various classification loss functions and penalties [69].

The model is built in a stage-wise fashion, generalizing to any classification routine through the optimization routine, which can be deployed to any arbitrary differentiable loss function. A common and widely adopted choice consists in deploying an SGD-based

¹⁰https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html.

¹¹<https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html>.

¹²<https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>.

estimator for classification tasks by using the SGD routine within regularized linear approaches. We have developed the Stochastic Gradient Descent Classifier experimentally adopting Hinge loss and L2 regularization, and using the scikit-learn library in Python¹³ with the default parameters' values.

6 Experimental evaluation

In this section, the effectiveness of the various data augmentation techniques is tested on the mentioned tasks and datasets. We show in particular the adopted evaluation metrics and the results we obtained with our augmented classification for the different tasks.

6.1 Evaluating metrics

We used the F1 score, Macro F1, Micro F1, and the precision and recall of the minority class to measure the performances of our models. We do not take into account the accuracy as in our single-label binary classification it corresponds to the Micro F1.

The F1 score is utilized to evaluate classification techniques and is recommended in the case of unbalanced data as it focuses on the recognition of the minority class. The harmonic mean of a classifier's precision and recall is used to calculate the F1 score, which integrates both metrics into a single value [6]. In particular:

$$precision = \frac{TP}{TP + FP},$$

$$recall = \frac{TP}{TP + FN},$$

and

$$F1 = 2 * \frac{precision * recall}{precision + recall},$$

where TP, TN, FN, and FP correspond, respectively, to the True Positives, True Negatives, False Negatives, and False Positives obtained by the model [6].

We have also employed the Macro-F1 and Micro-F1 scores [56]. A macro-average computes the metric independently for each class and then takes the average (hence treating all classes equally), whereas a micro-average aggregates the contributions of all classes to compute the average metric. Their formulas are defined as follows:

$$\text{Macro-F1} = \frac{1}{N} \cdot \sum_{i=1}^N F1_i,$$

where i is the class/label index and N the number of classes/labels, and

$$\text{Micro-F1} = 2 \cdot \frac{\text{Microprecision} \cdot \text{Microrecall}}{\text{Microprecision} + \text{Microrecall}}.$$

6.2 Computational results

In this section, we show the results that we obtained for the financial tasks presented in Sect. 3. To provide fair and reliable results, the augmented data affects the training set

¹³https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDClassifier.html.

only, whereas the test set is always composed of real data. Furthermore, in our settings, we made sure that the test set was composed of samples with the same number of minority and majority classes.

6.2.1 Bankruptcy in Poland

In this dataset, a total of 41,514 active companies and 2091 bankrupt companies were analyzed. As mentioned in Sect. 3, the dataset we have used is divided into 5 groups. All tests were run independently in each group and the discussed results are the average of all of them. We have used different kinds of augmentation methods as mentioned above, like the min-max approach, mean-std, SMOTE, and SMOTE-Clustering approach. In each approach we balanced the dataset, that is the size of the minority class became the same as the size of the majority class. After applying data augmentation techniques, the augmented dataset included 41,514 instances of active companies and 41,514 bankrupt companies. The candidates for the test set were extracted from the real data (before the augmentation) and included all the samples from the minority class and the same number of samples from the majority class. The dimension of the test set for each run was fixed to 100. On the other hand, the training set consisted of the augmented data including the candidates for the test set except the 100 samples used for the current iteration. We used a k -fold cross-validation setting in each of the five groups. Table 10 shows the number of samples in the training and the test set and the value of k for the validation. As an example, the first row indicates that for the first year of the dataset we have 13,412 elements in the training set (corresponding to $6756 \cdot 2 - 100$ as can be seen in Sect. 3.1), 542 candidate elements for the test set (evenly balanced), and 100 of these are chosen as actual test set (again evenly balanced) for the current iteration (the other 442 were already included in the count of the training set). A value of k equal to 5 means that at least 500 real elements (250 of the minority class and 250 of the majority class) must have been chosen before the augmentation so that the k -fold cross-validation with $k = 5$ could be carried out.

The obtained results are shown in Table 11 and represent the precision and recall and F1 scores of the minority class obtained by the various classifiers for the five classes. In our experiments, we always show the comparison against the classifiers adopting the various augmentation methods, along with their original non-augmented versions. As it can be observed in Table 11, the SMOTE-Clustering method has outperformed the other methods and has shown in particular superior F1 scores relative to almost all the other algorithms except SGD. Overall, with an F1 micro score of 0.94%, the Random Forest method has produced the best-performing scores. It is worth noting that the analyzed datasets contain a number of non-numerical features that have been encoded as numerical features and utilized in this way to train appropriately the ML algorithms.¹⁴ Moreover, it

Table 10 Details about the bankruptcy in Poland dataset

# Groups	# Training samples	#Candidates for test set	# Test set in each run	K value
First	13,412	542	100	5
Second	19,446	800	100	8
Third	19,916	990	100	9
Forth	18,454	1030	100	10
Fifth	10,900	820	100	8

¹⁴<https://scikit-learn.org/stable/modules/classes.html#module-sklearn.preprocessing>.

Table 11 Classification results for the bankruptcy in Poland dataset

Evaluation metric	Naive Bayes	SGD	KNN	MLP	Random forest	SVM
<i>Without augmentation</i>						
F1 score macro	0.3584	0.4019	0.3583	0.5317	0.3639	0.40
F1 score micro	0.5006	0.5175	0.5692	0.6249	0.5691	0.5161
Precision minority class	0.4997	0	0	0.6824	0	0.6118
Recall minority class	0.9732	0.0868	0	0.2335	0.0062	0.0773
<i>Baseline: min-max approach</i>						
F1 score macro	0.3689	0.364	0.5429	0.5243	0.7141	0.3583
F1 score micro	0.4509	0.511	0.6125	0.6023	0.7359	0.5692
Precision minority class	0.4351	0.3145	0.9937	0	0.9966	0
Recall minority class	0.9254	0.0494	0.2247	0.212	0.4725	0
<i>Baseline: mean-std approach</i>						
F1 score macro	0.3691	0.3708	0.3583	0.4868	0.7241	0.3583
F1 score micro	0.4513	0.5468	0.5692	0.6052	0.7449	0.5692
Precision minority class	0.4349	0.204	0	0.6491	0.9976	0
Recall minority class	0.9236	0.0284	0	0.1744	0.4815	0
<i>SMOTE approach</i>						
F1 score macro	0.3157	0.5932	0.9013	0.8829	0.9381	0.6826
F1 score micro	0.4335	0.6028	0.9137	0.8948	0.9474	0.7029
Precision minority class	0.4305	0.5551	0.8292	0.8541	0.9074	0.6166
Recall minority class	0.9738	0.6581	0.9694	0.8782	0.844	0.6975
<i>SMOTE-clustering approach</i>						
F1 score macro	0.4168	0.4494	0.9176	0.6447	0.9082	0.6772
F1 score micro	0.5801	0.4603	0.9293	0.6834	0.9169	0.6949
Precision minority class	0.4537	0.4345	0.8394	0.6814	0.881	0.612
Recall minority class	0.5098	0.6437	0.9836	0.4935	0.9144	0.7091

should be highlighted that the increase in performance is mainly due to a better capability of the models to classify the minority class when the augmentation methods are employed. Among them, SMOTE and SMOTE-Clustering appear to be the algorithms achieving the top performance.

6.2.2 Bankruptcy in USA

We used another dataset related to companies located in the USA to tackle the same task of correctly classifying bankruptcy. In the survey of bankrupt and active companies in the USA, 8261 instances were examined. Of these, 561 companies had declared bankruptcy, and 7700 were active. As it is obvious, this dataset is very unbalanced so data augmentation methods have been used in this dataset.

After augmenting this dataset, we have 7700 data for companies that are still active and 7700 samples for companies that have gone bankrupt. The candidates for the test set have been chosen before augmenting the data and have been fixed to 1122. The test set size was set to 100 for each iteration. We applied a k -fold cross-validation with k equal to 11. Table 12 illustrates the split of training and test set we have used and the value of k . 15,300 is the size of the training set ($7700 \times 2 - 100$), 1122 is the number of candidates for the test set, and 100 of these are chosen in the test set for the current iteration while the others are added to the training set.

Results are shown in Table 13, which indicates that the data augmentation techniques employed in most of the considered classification methods were successful in improving the F1 scores. Random Forest, in particular, achieved very high performance, outperforming all the other classification methods for all the metrics. The other methods, although with lower F1 scores values, obtained improved results when adopting the augmentation

Table 12 Details about the bankruptcy in USA dataset

# Training samples	#Candidates for test set	# Test set in each run	K value
15,300	1122	100	11

Table 13 Classification results for the bankruptcy dataset in USA

Evaluation metric	Naive Bayes	SGD	KNN	MLP	Random forest	SVM
<i>Without augmentation</i>						
F1 score macro	0.3416	0.5779	0.3917	0.3855	0.4059	0.3478
F1 score micro	0.5202	0.6278	0.5398	0.5381	0.5482	0.521
Precision minority class	0	0.7791	0.8651	0	0.95	0
Recall minority class	0	0.3195	0.0493	0.0443	0.0632	0.0067
<i>Baseline: min-max approach</i>						
F1 score macro	0.347	0.4792	0.4797	0.4819	0.359	0.343
F1 score micro	0.5202	0.5772	0.5816	0.5312	0.5277	0.5202
Precision minority class	0	0.7465	0.8588	0.5157	0	0
Recall minority class	0.0061	0.1637	0.1582	0.2434	0.0157	0.0017
<i>Baseline: mean-std approach</i>						
F1 score macro	0.3416	0.5315	0.3429	0.3781	0.372	0.35
F1 score micro	0.5202	0.5953	0.5052	0.5344	0.5331	0.5227
Precision minority class	0	0.7194	0	0	0.9762	0
Recall minority class	0	0.2693	0.01	0.0367	0.0288	0.0083
<i>SMOTE approach</i>						
F1 score macro	0.7136	0.7051	0.9246	0.7693	0.9823	0.5228
F1 score micro	0.7157	0.7093	0.9253	0.7715	0.9825	0.5272
Precision minority class	0.7113	0.7231	0.8639	0.77	0.9804	0.5051
Recall minority class	0.6863	0.6507	1	0.7436	0.9823	0.5016
<i>SMOTE-clustering approach</i>						
F1 score macro	0.7111	0.7442	0.9179	0.7428	0.9875	0.5261
F1 score micro	0.7132	0.7458	0.9186	0.7452	0.9875	0.5305
Precision minority class	0.7097	0.7244	0.8537	0.7534	0.9852	0.5094
Recall minority class	0.6813	0.7715	1	0.699	0.99	0.495

methods too. The reason is likely because the features in this dataset are statistically dependent on each other and, therefore, using statistical attributes like mean, standard deviation, minimum, and maximum to augment data, has shown to be successful in creating synthetic data with the same characteristics as the original ones. Moreover, by leveraging SMOTE-based approaches, we achieved the current best results on this dataset over the minority class with respect to the state-of-the-art to date presented by Lombardo et al. [48].

6.2.3 Bank marketing

For this other task, the bank marketing dataset contains 45,211 observations, which includes 5289 for positive instances and 39,922 for negative instances. The number of successful and unsuccessful samples has been balanced by using data augmentation techniques. In particular, the amount of failed items has increased. Similar to the previous tasks, the test set has been separated before the augmentation. The test set is divided into subsets with size 1000, each of which contains 500 samples that were successful and 500 samples that failed. A k -fold cross-validation was performed using an appropriate k which satisfies the previous constraint. Table 14 shows the split of training and test sets and the value of k . In particular, 5289 real elements from the minority class and 5289 real elements from the majority class have been extracted before the augmentation and represent the

Table 14 Details about the bank marketing dataset

# Training samples	#Candidates for test set	# Test set in each run	K value
78,844	10,578	1000	10

Table 15 Classification results for the bank marketing dataset

Evaluation metric	Naive Bayes	SGD	KNN	MLP	Random forest	SVM
<i>Without augmentation</i>						
F1 score macro	0.3734	0.405	0.4312	0.3517	0.3791	0.4338
F1 score micro	0.557	0.5405	0.5753	0.5464	0.5518	0.5524
Precision minority class	0	0.4347	0.8308	0	0.5362	0.5464
Recall minority class	0.0222	0.1129	0.0862	0.0011	0.0308	0.1106
<i>Baseline: min-max approach</i>						
F1 score macro	0.3738	0.4475	0.5326	0.311	0.477	0.3083
F1 score micro	0.3824	0.5413	0.5714	0.419	0.5836	0.454
Precision minority class	0.3726	0.5074	0.5274	0.4331	0.5569	0.454
Recall minority class	0.5384	0.1496	0.6065	0.8976	0.257	1
<i>Baseline: mean-std approach</i>						
F1 score macro	0.3116	0.3504	0.5324	0.5002	0.4594	0.3083
F1 score micro	0.3226	0.4935	0.5713	0.5214	0.5259	0.454
Precision minority class	0.2815	0.1737	0.5273	0.4882	0.5107	0.454
Recall minority class	0.3576	0.0282	0.6062	0.7362	0.5497	1
<i>SMOTE approach</i>						
F1 score macro	0.8055	0.4632	0.9105	0.7237	0.9957	0.5509
F1 score micro	0.82	0.567	0.9149	0.7534	0.9957	0.5628
Precision minority class	0.8749	0.4522	0.8525	0.8819	0.9997	0.5187
Recall minority class	0.7024	0.2902	0.9679	0.5326	0.9917	0.5115
<i>SMOTE-clustering approach</i>						
F1 score macro	0.8058	0.6152	0.8916	0.7269	0.9972	0.548
F1 score micro	0.82	0.6742	0.8968	0.7546	0.9972	0.5574
Precision minority class	0.8723	0.7603	0.828	0.8784	0.9992	0.5107
Recall minority class	0.707	0.4435	0.9575	0.5346	0.9952	0.5444

candidates for the test set. During each fold, the test set is formed taking 1000 elements from these candidates in a balanced way. The others are part of the training set.

The obtained F1 scores are illustrated in Table 15, which demonstrates how data augmentation techniques have improved the classification performances across all ML algorithms. KNN and Random Forest, in particular, have outperformed all the others and have generally provided superior results when using SMOTE-based techniques with respect to the other augmentation approaches.

6.2.4 Credit card frauds

For the credit card fraud task, there are 284,315 samples total in the used dataset, and only 492 among them are labelled as fraud transactions. Once again, before performing the augmentation, we created our list of candidates for the test by taking real data: 492 fraud transactions and 492 non-fraud transactions. Then we performed the augmentation and generated our complete and balanced training set. We applied a k -fold cross-validation using test sets with sizes equal to 100. Table 16 shows the split of training and test sets and the value for k we have chosen.

The test results shown in Table 17 demonstrate that, after data augmentation, KNN, MLP, and Random Forest algorithms have provided high F1 scores. The performance of the algorithms with SMOTE and SMOTE-Clustering approaches appears to be very similar although slightly better performances are achieved with SMOTE-Clustering. As in the

Table 16 Details about the credit card frauds dataset

# Training samples	#Candidates for test set	# Test set in each run	K value
567,546	984	100	9

Table 17 Classification results for the credit card frauds dataset

Evaluation metric	Naive Bayes	SGD	KNN	MLP	Random forest	SVM
<i>Without augmentation</i>						
F1 score macro	0.8218	0.3373	0.3351	0.3395	0.8781	0.3351
F1 score micro	0.8277	0.5052	0.5042	0.5062	0.8802	0.5042
Precision minority class	0.9942	0	0	0	1	0
Recall minority class	0.6566	0.002	0	0.004	0.7587	0
<i>Baseline: min-max approach</i>						
F1 score macro	0.7282	0.3351	0.3351	0.3351	0.8615	0.3351
F1 score micro	0.7471	0.5042	0.5042	0.5042	0.8647	0.5042
Precision minority class	1	0	0	0	1	0
Recall minority class	0.4902	0	0	0	0.7266	0
<i>Baseline: mean-std approach</i>						
F1 score macro	0.5465	0.3351	0.3351	0.3351	0.3788	0.3351
F1 score micro	0.615	0.5042	0.5042	0.5042	0.5244	0.5042
Precision minority class	0.9726	0	0	0	1	0
Recall minority class	0.2297	0	0	0	0.041	0
<i>SMOTE approach</i>						
F1 score macro	0.8847	0.7158	0.9849	0.9485	0.976	0.5409
F1 score micro	0.8862	0.7605	0.9849	0.9486	0.976	0.5528
Precision minority class	0.9923	0.8797	0.9819	0.9867	0.9936	0.5087
Recall minority class	0.7769	0.7316	0.9878	0.909	0.958	0.5015
<i>SMOTE-clustering approach</i>						
F1 score macro	0.8798	0.8812	0.9879	0.9395	0.9898	0.549
F1 score micro	0.8814	0.8833	0.9879	0.9397	0.9898	0.5584
Precision minority class	0.9822	0.9887	0.9901	0.9871	0.9963	0.5127
Recall minority class	0.7759	0.775	0.9857	0.8908	0.9832	0.5454

previous tasks, although the performance growth is smaller by leveraging augmentation techniques, it is clear that the models benefit once again from a better capability of learning the minority class with a higher recall over the minority class obtained by the classifiers.

6.2.5 Credit approval

For the credit approval, there are 698 samples, 306 of which are assigned to the positive class and 383 to the negative class. In contrast to the cases mentioned above, this dataset is not unbalanced but it consists of too few samples to be fed to ML approaches. As it is widely acknowledged, in this case, there are not enough instances available to train ML algorithms [61]. The VAEs auto-encoder has been utilized to solve this problem by generating the required data artificially. A copy of the original data is obtained at the start of the procedure for testing. Then, by applying a VAEs auto-encoder, we obtained 10,000 samples for each class. They have been used to train ML algorithms. The test set was taken before applying the augmentation technique and contained 50 negative samples and 50 positive samples out of a candidate set of 698 elements. A k -fold cross-validation was applied using the split of training and test shown in Table 18. Again, the candidates for the test set have been extracted before the augmentation and consisted of 698 samples. During each cross-validation iterations, 100 balanced elements from the 698 were chosen as a test set and the others were left for training.

Table 18 Details about the credit approval dataset

# Training samples	#Candidates for test set	# Test set in each run	K value
19,900	698	100	6

Table 19 Classification results for the credit approval dataset

Evaluation metric	Naive Bayes	SGD	KNN	MLP	Random forest	SVM
<i>Without augmentation</i>						
F1 score macro	0.7703	0.657	0.6283	0.7498	0.7824	0.5326
F1 score micro	0.8087	0.6811	0.6541	0.778	0.8088	0.5658
Precision minority class	0.786	0.7574	0.6765	0.7762	0.8164	0.5909
Recall minority class	0.8486	0.6943	0.7114	0.8343	0.8171	0.62
<i>DL approach</i>						
F1 score macro	0.7138	0.629	0.7797	0.8884	0.9771	0.3674
F1 score micro	0.773	0.687	0.8084	0.8932	0.9771	0.4712
Precision minority class	0.8183	0.8017	0.8068	0.8759	0.9853	0.639
Recall minority class	0.7057	0.5629	0.8229	0.9343	0.9686	0.3629

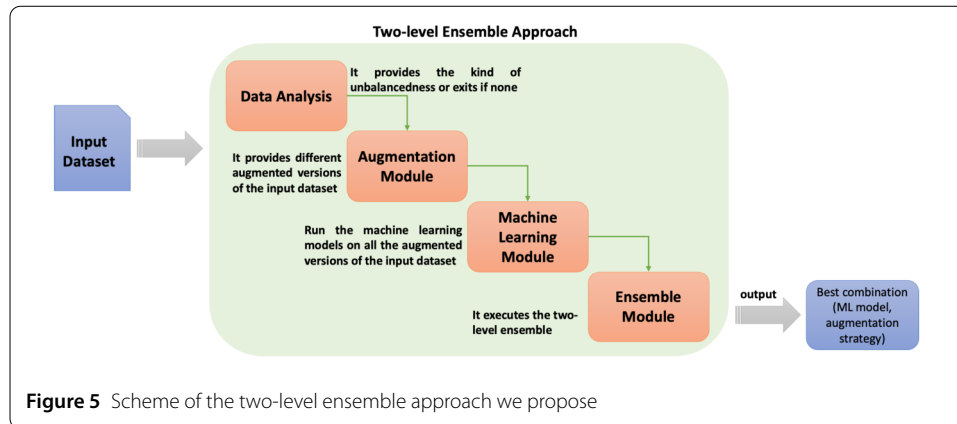
Table 19 illustrates the results obtained for the credit approval dataset. In particular, the reader can notice how the data augmentation approach has improved again the F1 scores in KNN, MLP, and Random Forest, the last of which provides the highest values for F1 macro, F1 micro, precision, and recall of the minority class. These results further confirm the superiority in performance of the over-sampling classifiers also in this particular case of a balanced dataset with very low samples, suggesting an overall validity of the data augmentation approach for financial tasks.

7 A two-level ensemble approach for financial classification tasks

The results we showed in Sect. 6 confirmed that the tested augmentation methods for the mentioned classification tasks within the financial domain bring benefit to the overall accuracy. Moreover, we proved that these methods enable to achieve better performance in heterogeneous conditions where the samples are composed of different types of data such as:

- Numerical financial variables from financial reports for the bankruptcy prediction;
- Personal information with several nominal attributes and textual data for the bank marketing task;
- Abstract features such as the ones generated with the PCA for the frauds detection task;
- Mix of numerical/nominal data that suffer from data missing for the Credit approval task and in scenarios where privacy preserving plays a fundamental role.

However, as previously discussed, there is not any absolute or standard solution that can be prior adopted depending on the use case because although the nature of data plays a crucial role, the design should also take into account the type of unbalancedness to deal with and the different importance that a wrong prediction can have depending on the context. In addition, the financial domain is dynamic and strongly dependent on time factors and events that can emerge over time and require novel investigations and model re-training. For this reason, we propose a two-level ensemble approach to drive the experimental setup in each scenario by automatically identifying the augmentation and ML methods to apply. This is an architectural scheme we have designed that can be leveraged



to successfully employ augmentation techniques and machine/DL methods for classification tasks and provide an evaluation method for approaches which automatically identifies the augmentation or ML method to apply.

In Fig. 5 we show the approach we have outlined. For sake of understanding, we will focus on binary tasks, on three different kinds of unbalancedness, and on the machine/DL approaches we covered in Sect. 5. In the future work section, we will address its extension to other domains, other augmentation methods, and further machine and DL techniques. First, a labeled dataset has to be fed to the proposed approach. As mentioned, we assume that the label represents a binary value and the dataset may have already been pre-processed and may present numerical elements only. Each column of the dataset represents a different feature.

The Data Analysis module analyses the samples of the dataset and their related labels to identify whether the dataset presents unbalancedness or not. Several state-of-the-art techniques exist to identify whether a dataset is unbalanced or not [16] and is not the purpose of this paper to explore them. If it does, it has to recognize its kind. The three types of unbalancedness that we have initially taken into account are defined as follows.

- The dataset is strongly unbalanced. It means that the number of samples of the minority class is much smaller than the number of those corresponding to the majority class. Moreover, the samples of the minority class are too few to perform the down-sampling of the samples of the majority class as the resulting balanced dataset would be too small for ML models to be trained and executed.
- The dataset is balanced but the overall number of samples is small for ML models to work efficiently.
- The dataset is unbalanced but it is possible to perform a down-sampling of the samples belonging to the majority class to match up those in the minority class as the resulting balanced dataset would be fine for ML to work well.

We let the reader notices that in our paper we have discussed methods to tackle the first kind of unbalancedness (i.e., VAEs) and others to tackle the second kind (SMOTE and SCUT). The reader is referred to [8] for a survey paper about an exhaustive list of augmentation techniques for text classification. Once the input dataset has been analysed, assuming one of the three kinds of unbalancedness has been found, the Augmentation module is triggered. All the augmentation techniques available in this module and corresponding to the underlying kind of unbalancedness are employed to create different augmented versions of the original dataset. Then, the Machine Learning module is called to run all the

ML models present here and on all the augmented versions of the input dataset. Finally, the Ensemble module is run. It includes the metrics that need to be taken into account; for the sake of simplicity, we will consider the accuracy. Then, it is employed to ensemble all the results using a majority voting strategy (clearly, other ensemble strategies may be adopted as well). The results are ensembled according to two levels. In the first level, a set of (ML algorithm, and augmentation technique) pairs are selected for each different ML algorithm (which corresponds to the key of the pair). Then, another ensemble is performed among the resulting winner pairs. As an example, let d_1, d_2, d_3 be three augmented versions of the input dataset by using the augmentation technique identified within the dataset. Different ML algorithms M_1, M_2, \dots, M_n will be applied to each of them producing $M_1(d_1), M_1(d_2), M_1(d_3), M_2(d_1), M_2(d_2), M_2(d_3), \dots, M_n(d_1), M_n(d_2), M_n(d_3)$. A first ensemble strategy is applied to each different pair (ML algorithm, augmentation technique) to select one winning pair for each key, that is the combination with the highest accuracy on the underlying classification task. That is, out of $M_1(d_1), M_1(d_2), M_1(d_3)$ we will have a winner $M_1(d_{w_1})$, out of $M_2(d_1), M_2(d_2), M_2(d_3)$ we will have a winner $M_2(d_{w_2})$, out of $M_n(d_1), M_n(d_2), M_n(d_3)$ we will have a winner $M_n(d_{w_n})$. Then, a second ensemble is applied to all the models that passed the first stage, that is $M_1(d_{w_1}), M_2(d_{w_2}), \dots, M_n(d_{w_n})$ to finally obtain the winner $M_i(d_x)$ which corresponds to the pair with the highest accuracy on the given classification task.

This scheme can be used to also evaluate any approach that automatically tries to infer one of the two elements of the pair (ML algorithm, augmentation technique) or even both of them. By ranking the results of the two-level ensemble approach in decreasing order of the adopted metric (the accuracy in our example), we can simply define a *rank* which would return the rank position of the tested algorithm with respect to all the possible combinations and assess its performance.

8 Conclusions and future directions

In this work, we analyzed how the state-of-the-art augmentation strategies deal with the financial domain, which commonly presents imbalanced conditions that prevent current ML models from correctly learning the less represented class. This is one of the main issues in intelligent Fintech, since the minority class is usually related to the rarest event that one might be interested in predicting. We evaluated several financial tasks with publicly available datasets and different imbalance conditions to finally prove that especially exploiting the SMOTE technique with clustering (SCUT) and VAEs leads to consistent and accurate improvements for all the tasks in terms of precision, recall, and F1 scores. Moreover, by analyzing the precision and recall metrics over the minority class in each dataset, we proved that the performance achieved is effectively due to a generally better capability of the models to learn the minority class. We will also explore the benefits of the same techniques for multi-class classification tasks in other domains for future comparisons.

In light of these results, as the second contribution of this paper, we also proposed a two-level ensemble approach to target classification tasks, that compares augmenting techniques given a dataset as input, and which returns the best combination of ML model and augmentation strategy as output. We are currently employing this approach to evaluate one methodology we designed which automatically identifies the augmentation technique to apply and leverage a fine-tuned transformer architecture for the targeted task.

Future developments are related firstly to further analysis of other augmentation techniques and their support in our two-level ensemble approach, such as Generative Adversarial Models (GANs) and conditional generative models. Moreover, since synthetic data generation has a positive effect on financial tasks, the same methodologies could also be exploited for deeper analysis toward the explainability and interpretability of more complex DL models available for these tasks.

Acknowledgements

The authors would like to thank the colleagues of the Competence Centre on Composite Indicators and Scoreboards (COIN) at the Joint Research Centre of the European Commission for helpful guidance and support during the development of this research work. The views expressed are purely those of the authors and may not in any circumstance be regarded as stating an official position of the European Commission.

Funding

Not applicable.

Abbreviations

Fintech, Financial Technology; ML, Machine Learning; DL, Deep Learning; SMOTE, Synthetic Minority Oversampling Technique; KNN, K-Nearest Neighbour; SqLL, Squared Logistics Loss; SVM, Support Vector Machine; RF, Random Forest; SEC, Security Exchange Commission; PCA, Principle Component Analysis; VAE, Variational Autoencoder; PDF, Probability Density Function; SCUT, SMOTE and Clustered Undersampling Technique; EM, Expectation Maximization; NB, Naive Bayes Classifier; RBF, Gaussian radial-basis function; MLP, Multilayer Perceptron; ReLU, Rectified Linear Unit; SGD, Stochastic Gradient Descent; GAN, Generative Adversarial Model; TP, True Positive; TN, True Negative; FN, False Negative; FP, False Positive.

Availability of data and materials

Data for bankruptcy prediction for private companies in Poland are available at <https://archive.ics.uci.edu/ml/datasets/Polish+companies+bankruptcy+data>. Data for bankruptcy prediction for public companies in the American stock market are available at https://github.com/sowide/bankruptcy_dataset. Data for bank marketing from Portuguese companies are available at <https://archive.ics.uci.edu/ml/datasets/Bank+Marketing>. Data for credit card frauds by European cardholders are available at <https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud>. Data for the Australian credit approval applications are available at <https://archive.ics.uci.edu/ml/datasets/Credit+Approval>.

Declarations

Competing interests

The authors declare that they have no competing interests.

Author contributions

DRR, GL and SC designed the study; GR carried out data collection; GR, GL and DRR implemented the software; GL, DRR and SC carried out the analysis and interpretation of the data; GR, DRR, GL, and SC helped to write the manuscript. All authors read and approved the final manuscript.

Author details

¹Department of Electrical and Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran. ²Department of Mathematics and Computer Science, University of Cagliari, via Ospedale 72, 09121 Cagliari, Italy. ³Department of Engineering and Architecture, University of Parma, Parco Area delle Scienze, 43125 Parma, Italy. ⁴Joint Research Centre (DG JRC), European Commission, Via E. Fermi 2749, 21027 Ispra (VA), Italy.

Received: 14 November 2022 Accepted: 26 June 2023 Published online: 10 July 2023

References

1. Agrawal A, Viktor HL, Paquet E (2015) Scut: multi-class imbalanced data classification using smote and cluster-based undersampling. In: The international joint conference on knowledge discovery, knowledge engineering and knowledge management (IC3k), vol 1. IEEE, New York, pp 226–234
2. Ahmad H, Kasasbeh B, Aldabaybah B, Rawashdeh E (2023) Class balancing framework for credit card fraud detection based on clustering and similarity-based selection (SBS). *Int J Inf Technol* 15(1):325–333
3. Alarfaj FK, Malik I, Khan HU, Almusallam N, Ramzan M, Ahmed M (2022) Credit card fraud detection using state-of-the-art machine learning and deep learning algorithms. *IEEE Access* 10:39700–39715
4. Alfaiz NS, Fati SM (2022) Enhanced credit card fraud detection model using machine learning. *Electronics* 11(4):662
5. Awad M, Khanna R (2015) Support vector machines for classification. In: *Efficient learning machines*. Springer, Berlin, pp 39–66
6. Barbaglia L, Consoli S, Manzan S, Reforgiato Recupero D, Saisana M, Tiozzo Pezzoli L (2021) Data science technologies in economics and finance: a gentle walk-in. In: *Data science for economics and finance: methodologies and applications*. Springer, Cham, pp 1–17

7. Barboza F, Kimura H, Altman E (2017) Machine learning models and bankruptcy prediction. *Expert Syst Appl* 83:405–417
8. Bayer M, Kaufhold MA, Reuter C (2022) A survey on data augmentation for text classification. *ACM Comput Surv* 55(7):146. <https://doi.org/10.1145/3544558>
9. Biau G, Scornet E (2016) A random forest guided tour. *Test* 25(2):197–227
10. Bin Sulaiman R, Schetinin V, Sant P (2022) Review of machine learning approach on credit card fraud detection. *Hum-Cent Intell Syst* 2(1–2):55–68
11. Carta S, Consoli S, Podda AS, Reforgiato Recupero D, Stanciu MM (2022) An eXplainable Artificial Intelligence tool for statistical arbitrage. *Softw Impacts* 14:100354. <https://doi.org/10.1016/j.simpa.2022.100354>
12. Carta S, Corrigan A, Ferreira A, Podda AS, Reforgiato Recupero D (2021) A multi-layer and multi-ensemble stock trader using deep learning and deep reinforcement learning. *Appl Intell* 51(2):889–905. <https://doi.org/10.1007/s10489-020-01839-5>
13. Carta S, Fenu G, Reforgiato Recupero D, Saia R (2019) Fraud detection for e-commerce transactions by employing a prudential multiple consensus model. *J Inf Secur Appl* 46:13–22. <https://doi.org/10.1016/j.jisa.2019.02.007>
14. Carta S, Ferreira A, Podda AS, Reforgiato Recupero D, Sanna A (2021) Multi-DQN: an ensemble of deep Q-learning agents for stock market forecasting. *Expert Syst Appl* 164:113820. <https://doi.org/10.1016/j.eswa.2020.113820>
15. Carta SM, Consoli S, Piras L, Podda AS, Reforgiato Recupero D (2021) Explainable machine learning exploiting news and domain-specific lexicon for stock market forecasting. *IEEE Access* 9:30193–30205. <https://doi.org/10.1109/ACCESS.2021.3059960>
16. Chawla NV (2005) Data mining for imbalanced datasets: an overview. In: Maimon O, Rokach L (eds) *Data mining and knowledge discovery handbook*. Springer, Boston, pp 853–867. https://doi.org/10.1007/0-387-25465-X_40
17. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) SMOTE: synthetic minority over-sampling technique. *J Artif Intell Res* 16:321–357
18. Chhabra G, Vashisht V, Ranjan J (2019) A review on missing data value estimation using imputation algorithm. *J Adv Res Dyn Control Syst* 11(7):312–318
19. Chugh B, Malik N (2023) Machine learning classifiers for detecting credit card fraudulent transactions. In: *Information and communication technology for competitive strategies. Lecture notes in networks and systems*, vol 400. Springer, Singapore, pp 223–231
20. Cleveland WS, Devlin SJ (1988) Locally weighted regression: an approach to regression analysis by local fitting. *J Am Stat Assoc* 83(403):596–610
21. Consoli S, Reforgiato Recupero D, Saisana M (2021) *Data science for economics and finance: methodologies and applications*. Springer, Cham. <https://doi.org/10.1007/978-3-030-66891-4>
22. Cortes C, Vapnik VN (1995) Support-vector networks. *Mach Learn* 20(3):273–297
23. Cutler A, Cutler DR, Stevens JR (2012) Random forests. In: *Ensemble machine learning: methods and applications*. Springer, New York, pp 157–175. https://doi.org/10.1007/978-1-4419-9326-7_5
24. Dal Pozzolo A, Caelen O, Bontempi G (2015) When is undersampling effective in unbalanced classification tasks? In: *Machine learning and knowledge discovery in databases. ECML PKDD 2015. Lecture notes in computer science*, vol 9284. Springer, Cham. https://doi.org/10.1007/978-3-319-23528-8_13
25. Dal Pozzolo A, Caelen O, Johnson RA, Bontempi G (2015) Calibrating probability with undersampling for unbalanced classification. In: *2015 IEEE symposium series on computational intelligence, IEEE*, New York, pp 159–166
26. Dal Pozzolo A, Caelen O, Le Borgne YA, Waterschoot S, Bontempi G (2014) Learned lessons in credit card fraud detection from a practitioner perspective. *Expert Syst Appl* 41(10):4915–4928
27. Danenas P, Garsva G (2015) Selection of support vector machines based classifiers for credit risk domain. *Expert Syst Appl* 42(6):3194–3204
28. Dantuluri A (2022) Learned data augmentation using VQ-Vae. <https://medium.com/mllearning-ai/learned-data-augmentation-using-vq-vae-339a8e12b779>
29. Delashmit WH, Manry MT et al (2005) Recent developments in multilayer perceptron neural networks. In: *Proceedings of the seventh annual Memphis area engineering and science conference, MAESC*
30. du Jardin P (2016) A two-stage classification technique for bankruptcy prediction. *Eur J Oper Res* 254(1):236–252
31. Faris H, Abukhurma R, Almanaseer W, Saadeh M, Mora AM, Castillo PA, Aljarah I (2020) Improving financial bankruptcy prediction in a highly imbalanced class distribution using oversampling and ensemble learning: a case from the Spanish market. *Prog Artif Intell* 9(1):31–53
32. Federal Trade Commission et al (2022) New data shows FTC received 2.8 million fraud reports from consumers in 2021
33. Garcia J (2022) Bankruptcy prediction using synthetic sampling. *Mach Learn Appl* 9:100343
34. Gunduz H (2021) An efficient stock market prediction model using hybrid feature reduction method based on variational autoencoders and recursive feature elimination. *Financ Innov* 7(1):28. <https://doi.org/10.1186/s40854-021-00243-3>
35. Guo G, Wang H, Bell D, Bi Y, Greer K (2003) KNN model-based approach in classification. In: *OTM confederated international conferences: on the move to meaningful Internet systems*. Springer, Berlin, pp 986–996
36. Huang Y, Liu DR, Lee SJ, Hsu CH, Liu YG (2022) A boosting resampling method for regression based on a conditional variational autoencoder. *Inf Sci* 590:90–105. <https://doi.org/10.1016/j.ins.2021.12.100>. <https://www.sciencedirect.com/science/article/pii/S0020025521013207>
37. Islam Z, Abdel-Aty M, Cai Q, Yuan J (2021) Crash data augmentation using variational autoencoder. *Accid Anal Prev* 151:105950
38. Jiang L, Wang D, Cai Z, Yan X (2007) Survey of improving naive Bayes for classification. In: *International conference on advanced data mining and applications*. Springer, Berlin, pp 134–145
39. John C, Ekpenyong EJ, Nworu CC (2019) Imputation of missing values in economic and financial time series data using five principal component analysis approaches. *CBN J Appl Stat* 10:51–73
40. Kim MJ, Kang DK (2010) Ensemble with neural networks for bankruptcy prediction. *Expert Syst Appl* 37(4):3373–3379
41. Kingma DP, Welling M (2013) Auto-encoding variational Bayes. *arXiv preprint*. [arXiv:1312.6114](https://arxiv.org/abs/1312.6114)
42. Lamoureux CG, Lastrapes WD (1990) Heteroskedasticity in stock return data: volume versus GARCH effects. *J Finance* 45(1):221–229

43. Le T, Lee MY, Park JR, Baik SW (2018) Oversampling techniques for bankruptcy prediction: novel features from a transaction dataset. *Symmetry* 10(4):79
44. Le T, Son LH, Vo MT, Lee MY, Baik SW (2018) A cluster-based boosting algorithm for bankruptcy prediction in a highly imbalanced dataset. *Symmetry* 10(7):250
45. Le T, Vo B, Fujita H, Nguyen NT, Baik SW (2019) A fast and accurate approach for bankruptcy forecasting using squared logistics loss with gpu-based extreme gradient boosting. *Inf Sci* 494:294–310
46. Le T, Vo MT, Vo B, Lee MY, Baik SW (2019) A hybrid approach using oversampling technique and cost-sensitive learning for bankruptcy prediction. *Complexity* 2019:8460934
47. Lee WS, Liu B (2003) Learning with positive and unlabeled examples using weighted logistic regression. In: *Proceedings, twentieth international conference on machine learning*, vol 1, pp 448–455
48. Lombardo G, Pellegrino M, Adosoglou G, Cagnoni S, Pardalos PM, Poggi A (2022) Machine learning for bankruptcy prediction in the American stock market: dataset and benchmarks. *Future Internet* 14(8):244
49. Loshchilov I, Hutter F (2017) SGDR: stochastic gradient descent with warm restarts. In: *Proceedings of the 5th international conference on learning representations (ICLR 2017)*, p 149804
50. Machado P, Fernandes B, Novais P (2022) Benchmarking data augmentation techniques for tabular data. In: *Intelligent data engineering and automated learning—IDEAL 2022: 23rd international conference, IDEAL 2022, Manchester, UK, November 24–26, 2022*. Springer, Berlin, pp 104–112
51. Mai F, Tian S, Lee C, Ma L (2019) Deep learning models for bankruptcy prediction using textual disclosures. *Eur J Oper Res* 274(2):743–758
52. Maulidevi NU, Surendro K et al (2022) Smote-lof for noise identification in imbalanced data classification. *J King Saud Univ, Comput Inf Sci* 34(6):3413–3423
53. Moro S, Laureano R, Cortez P (2011) Using data mining for bank direct marketing: an application of the CRISP-DM methodology. In: *European simulation and modelling conference, EUROSIS-ETI*, pp 117–121
54. Moscatelli M, Parlapano F, Narizzano S, Viggiano G (2020) Corporate default forecasting with machine learning. *Expert Syst Appl* 161:113567
55. Nanni L, Lumini A (2009) An experimental comparison of ensemble of classifiers for bankruptcy prediction and credit scoring. *Expert Syst Appl* 36(2):3028–3033
56. Opitz J, Burst S (2019) Macro F1 and macro F1. *CoRR abs/1911.03347*. <http://arxiv.org/abs/1911.03347>
57. Osisanwo F, Akinsola J, Awodele O, Hinmikaiye J, Olakanmi O, Akinjobi J (2017) Supervised machine learning algorithms: classification and comparison. *Int J Comput Trends Technol* 48(3):128–138
58. Pandya DD, Gupta NS, Jadeja A, Patel RD, Degadwala S, Vyas D (2022) Bias protected attributes data balancing using map reduce. In: *6th international conference on electronics, communication and aerospace technology, ICECA 2022 - proceedings*, pp 1540–1544. <https://doi.org/10.1109/ICECA55336.2022.10009363>
59. Paul A, Mukherjee DP, Das P, Gangopadhyay A, Chintha AR, Kundu S (2018) Improved random forest for classification. *IEEE Trans Image Process* 27(8):4012–4024
60. Pranavi NSS, Sruthi T, Naga Sirisha BJ, Nayak M, Gupta Thadikemalla VS (2022) Credit card fraud detection using minority oversampling and random forest technique. In: *2022 3rd international conference for emerging technology, INCET 2022*, pp 1–6
61. Ray S (2019) A quick review of machine learning algorithms. In: *2019 international conference on machine learning, big data, cloud and parallel computing (COMITCon)*, IEEE, pp 35–39
62. Saheed YK, Baba UA, Raji MA (2022) Big data analytics for credit card fraud detection using supervised machine learning models. In: *Big data analytics in the insurance market*. Emerald Publishing Limited, pp 31–56
63. Sakprasat S, Sinclair MC (2007) Classification rule mining for automatic credit approval using genetic programming. In: *2007 IEEE congress on evolutionary computation*, IEEE, pp 548–555
64. Saldanha J, Chakraborty S, Patil S, Kotecha K, Kumar S, Nayyar A (2022) Data augmentation using variational autoencoders for improvement of respiratory disease classification. *PLoS ONE* 17(8):e0266467
65. Santoso N, Wibowo W, Himawati H (2019) Integration of synthetic minority oversampling technique for imbalanced class. *Indones J Electr Eng Comput Sci* 13(1):102–108
66. Schönfeld J, Kuděj M, Smrčka L (2018) Financial health of enterprises introducing safeguard procedure based on bankruptcy models. *J Bus Econ Manag* 19(5):692–705
67. Silva LO, Zárate LE (2014) A brief review of the main approaches for treatment of missing data. *Intell Data Anal* 18(6):1177–1198. <https://doi.org/10.3233/IDA-140690>
68. Sohae O (2015) Multiple imputation in missing values in time series data. Master's thesis, Duke University, North California
69. St Angel L (2020) Using stochastic gradient descent to train linear classifiers. *Towards Data Science*. <https://towardsdatascience.com/using-stochastic-gradient-descent-to-train-linear-classifiers-c80f6aeaff76>
70. Sun Y, Wong AKC, Kamel MS (2009) Classification of imbalanced data: a review. *Int J Pattern Recognit Artif Intell* 23(04):687–719. <https://doi.org/10.1142/S0218001409007326>
71. Tarawneh AS, Hassanat AB, Almohammadi K, Chetverikov D, Bellinger C (2020) Smotefuna: synthetic minority over-sampling technique based on furthest neighbour algorithm. *IEEE Access* 8:59069–59082
72. Tusell-Palmer FJ (2005) Multiple imputation of time series: an application to the construction of historical price indexes. *BILTOKI* 1134-8984, Universidad del País Vasco - Departamento de Economía Aplicada III (Econometría y Estadística). <https://ideas.repec.org/p/ehu/biltok/5663.html>
73. Université Libre de Bruxelles, Machine Learning Group (2021) Credit card fraud detection. <https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud>. Accessed 16 Apr 2023
74. Veganzones D, Séverin E (2018) An investigation of bankruptcy prediction in imbalanced datasets. *Decis Support Syst* 112:111–124
75. Wang CM, Huang YF (2009) Evolutionary-based feature selection approaches with new criteria for data mining: a case study of credit approval data. *Expert Syst Appl* 36(3):5900–5908
76. Wang D (2020) Research on bank marketing behavior based on machine learning. In: *Proceedings of the 2nd international conference on artificial intelligence and advanced manufacture*, pp 150–154
77. Wang G, Hao J, Ma J, Jiang H (2011) A comparative assessment of ensemble learning for credit scoring. *Expert Syst Appl* 38(1):223–230

78. Wang G, Ma J, Yang S (2014) An improved boosting based on feature selection for corporate bankruptcy prediction. *Expert Syst Appl* 41(5):2353–2361
79. Wei S, Chen Z, Arumugasamy SK, Chew IML (2022) Data augmentation and machine learning techniques for control strategy development in bio-polymerization process. *Environ Sci Ecotechnol* 11:100172
80. Wen Q, Sun L, Yang F, Song X, Gao J, Wang X, Xu H (2021) Time series data augmentation for deep learning: a survey. In: Zhou ZH (ed) *Proceedings of the thirtieth international joint conference on artificial intelligence, IJCAI-21, international joint conferences on artificial intelligence organization*, pp 4653–4660. <https://doi.org/10.24963/ijcai.2021/631>
81. Yan K, Zhang D (2015) Feature selection and analysis on correlated gas sensor data with recursive feature elimination. *Sens Actuators B, Chem* 212:353–363. <https://www.sciencedirect.com/science/article/pii/S0925400515001872>
82. Yang FJ (2018) An implementation of naive Bayes classifier. In: 2018 international conference on computational science and computational intelligence (CSCI), pp 301–306
83. Zareapoor M, Shamsolmoali P et al (2015) Application of credit card fraud detection: based on bagging ensemble classifier. *Proc Comput Sci* 48:679–685
84. Zięba M, Tomczak SK, Tomczak JM (2016) Ensemble boosted trees with synthetic features generation in application to bankruptcy prediction. *Expert Syst Appl* 58:93–101

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- ▶ Convenient online submission
- ▶ Rigorous peer review
- ▶ Open access: articles freely available online
- ▶ High visibility within the field
- ▶ Retaining the copyright to your article

Submit your next manuscript at ▶ [springeropen.com](https://www.springeropen.com)
