Universidade do Minho
Escola de Engenharia

Pedro Filipe Costa Machado

Conception and Evaluation of Data Augmentation techniques for Tabular Data

October, 2022

Universidade do Minho
Escola de Engenharia

Pedro Filipe Costa Machado

Conception and Evaluation of Data Augmentation techniques for Tabular Data

Master's Dissertation Report
Master in Informatics Engineering

Work developed under the supervision of:
Paulo Jorge Freitas de Oliveira Novais
Bruno Filipe Martins Fernandes

October, 2022

Acknowledgements

During this dissertation, I had the support of people for whom I am immensely grateful.

I would like to acknowledge and give my warmest thanks to my supervisors, Bruno Fernandes and Paulo Novais, who made this work possible. Their guidance and advice carried me through all the stages of developing and writing my dissertation.

I would also like to give my special thanks to my family as a whole, especially to my parents, my sister, my grandparents, uncles, and cousins, for all the support throughout this journey.

Finally, to my friends Alexandre Ferreira, Alexandre Miranda, Carolina Marques, Cristina Mendes, Diogo Silva, João Azevedo, Jorge Costa, Paulo Araújo, Paulo Lima, and Nuno Barbosa, I thank you for your unconditional friendship.

STATEMENT OF INTEGRITY

I hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration.

I further declare that I have fully acknowledged the Code of Ethical Conduct of the Universidade do Minho.

________________________, _______________________________________

(Place) (Date)

____

(Pedro Filipe Costa Machado)

Resumo

Concepção e Avaliação de técnicas de Data Augmentation para Dados Tabulares

O desbalanceamento dos dados, juntamente com *datasets* de tamanho reduzido, estão presentes em muitos problemas de *Machine Learning*, apesar do aumento de recolha de dados atuais por consequência do desenvolvimento tecnológico. O desbalanceamento de dados é definido por uma diferença significativa na distribuição das suas classes dentro de um conjunto de dados. Desta forma, a performance de um modelo pode diminuir drasticamente para certas classes com uma quantidade inferior de instâncias. Isto deve-se ao modelo não aprender a distribuição dos atributos dos dados e apresenta uma performance demasiado focada na classe em maioria. Este fenómeno compromete a performance dos modelos em problemas como por exemplo deteção de cancro em pacientes, uma vez que o modelo identifica poucos pacientes não saudáveis. Assim, as técnicas de *Data Augmentation* podem colmatar este problema ao gerarem dados sintéticos similares aos reais, podendo simular um ambiente de aprendizagem sem escassez de dados para os modelos. Com a aplicação destas técnicas, o número de dados disponíveis aumenta pelo que se consegue obter distribuições de classes mais equilibradas. Contudo, não existe uma técnica comum de *Data Augmentation* que possa ser aplicada em qualquer domínio com bons resultados. Desta forma, com esta dissertação pretende-se identificar quais características de um certo tipo de *dataset* beneficiam as diferentes técnicas para uma melhor performance na criação de dados sintéticos e, consequentemente, uma melhor performance dos modelos de *Machine Learning*. Os resultados obtidos nesta dissertação demonstram que a adição de dados sintéticos a *datasets*, cujos atributos sejam na sua maioria categóricos, está associada a uma acrescida dificuldade em melhorar a performance dos classificadores. No entanto, a técnica que melhor se adaptava a estas características foi o SMOTE, uma das técnicas mais clássicas de *Data Augmentation*. Por outro lado, as variações do *Variational Autoencoder*, nomeadamente a que conjuga um decaimento na *loss* e o uso de K-means, e a *GAN* geraram dados sintéticos capazes de melhorar a performance dos classificadores. Para além disto, esta dissertação comprovou que a adição de mais 25% de dados sintéticos a um *dataset* maioritariamente categórico permitiria melhores resultados, enquanto num *dataset* com maior presença de atributos contínuos era beneficiada pela adição de apenas instâncias minoritárias.

Palavras-chave: Dados Desbalanceados, Data Augmentation, Machine Learning

Abstract

Conception and Evaluation of Data Augmentation techniques for Tabular Data

Imbalanced learning and small-sized datasets are present in Machine Learning problems, even with the increased data availability provided by recent developments. The performance of learning algorithms in the presence of unbalanced data and significant class distribution skews is known as the imbalanced learning problem. The models' performance on such problems can drastically decrease for certain classes with an uneven distribution, because the models do not learn the distributive features of the data and present accuracy too favorable for a specific set of classes of data. This can have negative consequences when talking about cancer detection, for example, since the model may identify poorly unhealthy patients. Hence, Data Augmentation techniques are usually conceived to evaluate how models would behave in non-data-scarce environments, generating synthetic data similar to real data. By applying those techniques, the amount of available data can be increased, balancing the class distributions. However, there is no standardized Data Augmentation process that can be applied to every domain of tabular data. Therefore, this dissertation aims to identify which characteristics of a dataset provide a better performance when synthesizing samples by a data augmentation technique in a tabular data environment. Moreover, if the data augmentation algorithm synthesizes more real samples, it is expected to increase the classifier's performance as well. Our results demonstrate that datasets whose features are mainly categorical have an associated difficulty in increasing the classifier results by adding new samples. Furthermore, the technique that adapted best to those kinds of datasets was the more classical one, SMOTE. As for the datasets with more continuous features, the variations of Variational Autoencoder, principally the VAE with K-means and decay, as well as GAN, demonstrated an increased capability when augmenting those kinds of datasets. This dissertation demonstrated that more categorical datasets could achieve better performance by including 25% synthetic samples, whereas continuous datasets could only do so by including minority samples.

Keywords: Data Augmentation, Imbalanced Data, Machine Learning

Contents

Appendices

Annexes

List of Figures

List of Tables

Acronyms

ADASYN	Adaptive Synthetic Sampling Approach for Imbalanced Learning 1, 3, 9, 10, 11, 18, 19, 21, 24, 29, 33, 37, 40, 41
AI	Artificial Intelligence 5
CRISP-DM	Cross Industry Standard Process for Data Mining 22, 28
CURE	Clustering Using Representatives 13
DA	Data Augmentation 1, 2, 3, 4, 8, 9, 11, 13, 18, 19, 20, 21, 22, 23, 24, 25, 26, 28, 29, 30, 31, 33, 34, 35, 37, 38, 40, 41, 42
DBSCAN	Density-based Spatial Clustering of Applications with Noise 13
DCGAN	Deep Convolutional Generative Adversarial Network 18
DL	Deep Learning 24, 37
EM	Expectation-Maximization 13
GAN	Generative Adversarial Network 1, 3, 4, 8, 9, 14, 16, 17, 18, 19, 21, 24, 25, 28, 31, 33, 34, 35, 38, 40, 41, 42
GMM	Gaussian Mixture Model 9, 13, 14, 18, 19, 21, 24, 34, 40, 41
GPU	Graphics Processing Unit 2
GSVM-RU	Granular Support Vector Machines - Repetitive Undersampling 21
ML	Machine Learning 1, 2, 3, 4, 5, 7, 8, 9, 18, 19, 21, 22, 23, 24, 41, 42
MLP	Multilayer Perceptron 22, 24, 28, 33, 34, 38
NaN	Not a Number 25
PCA	Principal Component Analysis 6

SMOTE	Synthetic Minority Over-sampling Technique 1, 3, 9, 10, 18, 19, 21, 24, 29, 31, 33, 34, 35, 37, 38, 40, 41
STING	Statistical Information Grid 13
SVM	Support Vector Machine 6, 18
VAE	Variational Autoencoder 1, 3, 4, 9, 14, 15, 16, 17, 18, 21, 24, 25, 28, 29, 30, 31, 33, 34, 35, 38, 40, 41, 42

Chapter 1

Introduction

In this chapter, the context and motivation of this dissertation will be discussed, as well as the main objectives and expected results. Finally, the structure of this document will be presented.

1.1 Context and Motivation

Nowadays, developments in technology have increased the availability of data for various problems. Data engineering produces new information from raw data for a variety of purposes, including governmental decision-making support systems, medical research, and many more, ranging from microscale data analysis to macroscale knowledge discovery [1]. However, even with a huge amount of data collection, small-sized datasets still exist, and so do imbalanced ones. Moreover, imbalanced learning problems represent a recurring problem of exceptional relevance, with far-reaching repercussions that require more investigation because it may pose a problem for ML models due to the fact that most standard algorithms expect the training data to have balanced class distributions. Hence, DA techniques are usually conceived to evaluate how models would behave in non-data-scarce environments [2, 3]. The application of these techniques is helpful in problems such as fraudulent transactions where the data is clearly imbalanced [4]. DA is the practice of synthesizing new data from the data at hand. This could be applied to almost any form of data, from images to numbers, but the main focus of this research is on tabular data. Usually, synthetic data is very similar to real data. However, there is no standardized DA process that can be applied to every domain. Instead, DA refers to a process that is highly dependent on the domain where it is to be implemented. Multiple DA techniques and models have been described in the literature for computer vision, time series, and even tabular data [5–9]. Some DA methods are frequently used, such as SMOTE [10], ADASYN [11], while others, such as clustering, have few applicational examples [12]. Furthermore, GANs are popular for images and can be used for tabular data too [13], as well as VAEs [9].

1.2 The Problem and its Challenges

The main problem in this research is which DA techniques benefit the model's performance on different kinds of datasets, particularly tabular data. Therefore, it is required to collect multiple imbalanced datasets, as well as small datasets, and develop not only DA techniques but also multiple classifiers in order to analyze their performance on each problem. Moreover, knowing what kind of DA algorithm provides better performance in each dataset offers an increased capture of minority cases on an imbalanced dataset or a more capable model than one trained on a small dataset without DA. The better identification of minority cases with DA techniques is important for domains such as disease detection and fraud detection, where identifying such cases is extremely crucial [14]. DA has been shown to outperform in data-scarce environments, increasing model performance by a small margin [8]. Although is not the only way to the small and imbalanced data problems, is one that is gaining a lot of interest by the research community throughout the recent years.

The challenges can differ from collecting imbalanced datasets to the development of DA techniques. First, it is necessary to possess sets of data with different characteristics and from multiple domains. Small datasets and imbalanced ones are fundamental. Therefore, a good dataset collection is the first step to obtaining results that provide enough information for a final conclusion. This requirement provides the analyses of the DA techniques with a unique situation that could help come to a conclusion in regards to which technique is better for each dataset. Second, the data treatment can be viewed as a challenge because the research is going to use multiple datasets, each with their own kind of preprocessing due to different features. Although the ML models will be the same for each problem/dataset, they still have to be trained and fitted to acquire the best performance possible. Third, the computational cost and resources need to be taken into account when developing the algorithms. Even if the size of the data is not huge, the benchmarking of models associated with DA algorithms will be time-consuming when run locally. To overcome that, platforms such as Kaggle and Google Colab might be useful. Even though these platforms help to surpass the computational cost, they have some constraints related to the use of GPU by day or week[1]. This constraint will hinder the development of models inserted into the deep learning branch since they are more complex models that would benefit from the computational offer that GPUs provide.

[1] Google Colab has a limit of GPU use by day, whereas Kaggle limits its use by week.

1.3 Main Objectives

The main objective of this study is to research and evaluate different kinds of DA techniques for tabular data. Therefore, we expect to:

1. Collect datasets from multiple fields that contain imbalanced class distributions or datasets that are small-sized;
2. Develop standard techniques such as VAE, SMOTE, ADASYN, GAN, and clustering;
3. Benchmark the performance of ML models on multiple sets of data;
4. Determine which dataset properties indicate improved performance from one or more DA techniques.

The study and development of the multiple DA techniques is going to represent a major part of the work that is complemented with benchmarking on multiple sets of data. The performance of the DA methods could be measured by a comparison between synthetic and real data, in addition to the results acquired by the ML models.

1.4 Research Hypothesis

First, the research should verify that the use of augmented datasets increases the models' performance when predicting, as seen in [15]. This increasing performance goes from all target classes when the problem is from a small-sized dataset to the minority class from an imbalanced dataset. In order to do so, multiple DA techniques will be used with ML models on different datasets. The synthetic data should be very similar to the real data, having the same distribution and properties as the real data. Furthermore, the data should provide a similar performance when the model is trained only with synthetic data compared to when it is trained with real data.

Second, it is expected to identify which kinds of techniques provide better results when synthesizing data in a certain domain's dataset. Therefore, it is anticipated that this study can improve the choice of the DA technique for certain domains or certain characteristics of a dataset. Furthermore, the research could reveal which type of algorithm can synthesize samples that are more similar to real data, such as whether a deep learning algorithm, like GAN, can synthesize samples that are more similar to real data than a traditional ML algorithm, such as SMOTE. Moreover, algorithms proposed in new studies can evaluate and compare their new techniques' performance with the supposed best technique (stated in this document) for that problem's context.

Third, by performing statistical tests, we should be able to compare real and synthetic data distributions. Better quality synthetic data ought to have very similar properties to the real data and, therefore, statistical tests can analyze how the DA methods perform.

Finally, the number of samples to be generated is also important information that could be gained from this study. Since there is no standard number of synthetic samples to be generated, as it depends on the data and context of a certain dataset, it is also fundamental to have an idea of how the different numbers of synthetic samples generated affect the ML classifiers' performance.

In essence, the dissertation aims to answer four specific research questions (RQ) in regard to the use of DA in tabular data, mainly:

RQ1) Does DA improves the performance of ML classifiers?

RQ2) Does the dataset properties influence the quality of generated synthetic samples?

RQ3) How many samples should a DA technique generate for a certain problem?

RQ4) Which DA technique provides better quality in terms of synthetic tabular data?

1.5 Document Structure

The structure of this document is defined by the following main sections: Introduction, State of the Art, Experimental Setup, Results and Discussion, and Conclusion.

First, the Introduction will provide enough information about the context and motivation of the study, as well as its main goals and achievements. Also, it will be explained the research hypothesis that this study purports to achieve. Furthermore, the problems and challenges will be explained so it can be seen some of the main issues that might arise in future work.

Second, the State of Art describes in detail the concepts that the reader must have for a comprehensive reading of this study. Understanding concepts like ML, Imbalanced Data, and DA are fundamental because they are the main focus of this work. ML is the subject of this paper, whereas Imbalanced Data is the main problem that DA tries to solve. Also, some DA techniques are described to provide an understanding of how they work with a brief comparison of their differences and how their performance is going to be compared when applied to datasets.

Third, the Experimental Setup of this study pretends to detail the approaches of the study and its experiments. It is described which methodology is applied to each problem and how the synthetic data generated is analyzed to evaluate which method produces better quality data. Moreover, the experiments section details how the DA techniques were implemented and notes some important decisions during their implementation, mainly on methods such as VAE and GAN. Furthermore, it also describes the datasets used in this study.

Next, the Results and Discussion focus on describing in great detail all the analyses done during this study and the following conclusions that came from them.

Finally, the Conclusion chapter summarizes the main results obtained during the experiments, describes the future work, and offers some final thoughts.

Chapter 2

State of the Art

2.1 Machine Learning

In the past decades, there has been an explosion of data. All of this data could be wasted if there was no easy way to analyze it and find patterns within it. ML techniques are used to automatically find the valuable underlying patterns within complex data that we would struggle to discover [16, 17]. As seen in Figure 1, ML is a branch of AI, and its algorithms can process large quantities of data that are way too large and too complex to humans [18]. The increased interest in ML follows the falling cost of large data storage devices, the increasing ease of collecting data, and the development of robust and efficient ML algorithms to process this data, as well as the falling cost of computational power [19]. Nowadays, ML is used in various fields such as bioinformatics, information retrieval, game playing, marketing, malware detection, object detection and so on [20–23].

ML algorithms are usually divided in three paradigms: Supervised, Unsupervised and Reinforcement learning [24].

2.1.1 Supervised Learning

Supervised ML algorithms are trained with labeled datasets. This allows the models to learn and become more accurate over time [24, 26]. These kinds of algorithms need external assistance with the input dataset divided into train and test. For example, a supervised ML algorithm would be trained with pictures of dogs and cats, all labeled by humans, and the model would learn how to classify each picture.

Another kind of supervised learning problem is the regression problem, which happens when a discrete variable, for example, the price of a car, has to be predicted by the algorithm. To train the model, it would require many examples of cars, including their predictors and their labels (i.e., their prices) [27]. The

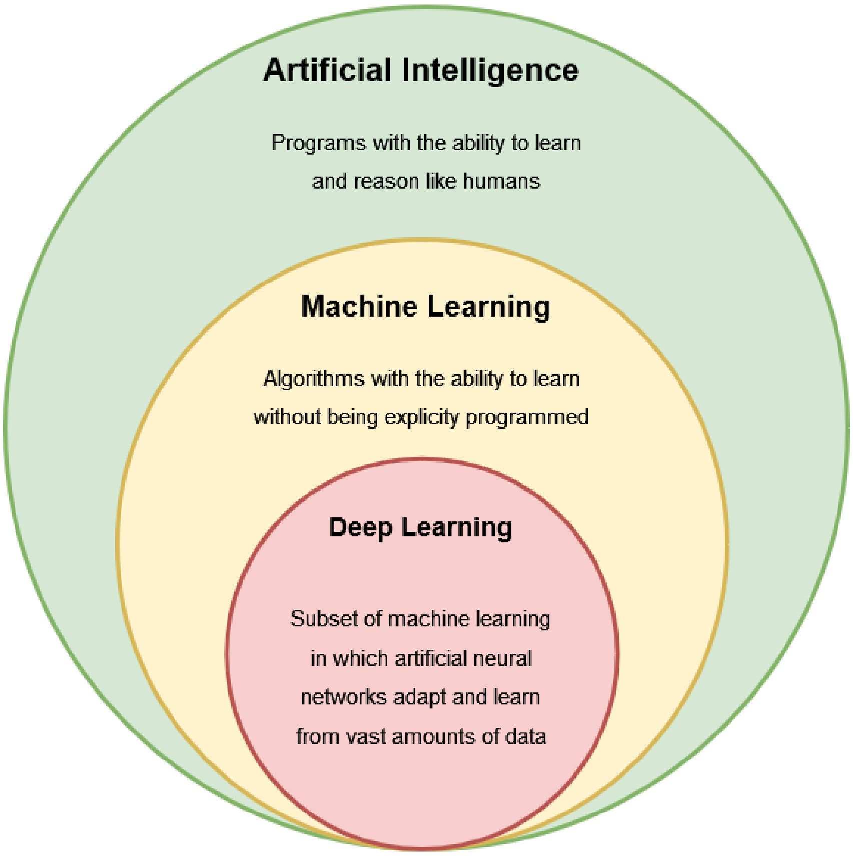

Artificial Intelligence
Programs with the ability to learn and reason like humans
Machine Learning
Algorithms with the ability to learn without being explicity programmed
Deep Learning
Subset of machine learning in which artificial neural networks adapt and learn from vast amounts of data

Figure 1: Difference between Machine Learning, Deep Learning and Artificial Intelligence [Adapted from [25]]

features taken into account, in this example, could be the age of the car, model, brand, etc. Some of the more famous supervised learning models are the Decision Tree, Linear Regression, and SVM.

2.1.2 Unsupervised Learning

On the other hand, in unsupervised learning, the algorithm searches for patterns in unlabeled data [26]. This paradigm has no teacher telling the correct answers, so the algorithms are left to their own methods to find the interesting structure in the data [24]. When new data is introduced to the algorithm, it uses the previously learned patterns/features to predict its class. Furthermore, the algorithms inserted into this paradigm can be divided into clustering, visualization, dimensionality reduction, and association rule learning. Clustering will be mentioned in detail in Section 2.4.3, as will some of its algorithms. Through visualization and dimensionality reduction, it can be analyzed how the data is organized (visualization) and which inputs better reflect the general dataset, decreasing the complexity of the data without losing information. This type of algorithm could be referred to as PCA. Finally, association rule learning involves discovering interesting relations between variables in a dataset. Apriori and Eclat, two well-known association rules algorithms (as seen in [27]), are known to provide these rules between variables. Unsupervised learning is demonstrated by an algorithm that analyzes online sales data to identify distinct types of clients purchasing based on their characteristics [26].

There is a combination of these two paradigms called semi-supervised learning. This paradigm uses a small amount of labeled data with a large amount of unlabeled data. It can be useful in areas of ML where there is unlabeled data and the process of getting labeled data is complicated [24, 27]. Generative models are an example of semi-supervised learning techniques.

2.1.3 Reinforcement Learning

Finally, Reinforcement ML trains machines through trial and error to take the best set of actions through a system of rewards, trying to maximize it [24, 26]. Furthermore, it must learn on its own what the best strategy, known as a "policy," is for reaping the greatest rewards over time [27]. As a result, these algorithms are appropriate for playing games or training autonomous vehicles to drive by informing the machine when it makes the correct decisions, rewarding it, and penalizing it when mistakes are made. Some of the most popular reinforcement learning algorithms are Q-Learning and SARSA. The SARSA algorithm is a variant of the Q-Learning algorithm. The Q-learning technique is an *off-policy* technique that learns the Q-value, which shows the potential reward for a specific action in a particular state, using a greedy approach. With the off-policy, the algorithm learns the value function according to the action of another policy. On the other hand, SARSA is a *on-policy* technique that uses the action performed by the current policy to learn the Q-value.

One famous example of using this paradigm is AlphaGo, which used reinforcement learning to learn how to play the game of Go by playing games against itself, which allowed the program to defeat the world champion [28]. Note that the algorithm disabled the learning during the games against the champion, applying only the policy it had learned.

2.2 Imbalanced Data

In recent years, the imbalanced learning problem has become a highly frequent topic among academia, industry, and government funding agencies. The fundamental issue with the imbalanced learning problem is the ability of imbalanced data to significantly decrease the performance of the ML algorithms [1]. These algorithms, when faced with imbalanced data, do not learn the distributive features of the data and present accuracies too favorable to a specific set of classes of data, in this case, the majority classes, compromising the performance of the other classes (the minority classes) because of that bias. In fairness, a dataset is considered imbalanced when it exhibits an unequal distribution between its classes. Nevertheless, the community usually considers that imbalanced data corresponds to a large unequal distribution and, in some cases, extremes. Some cases of imbalanced data are described with multiple class imbalance orders, such as 100:1, 1000:1, and 10000:1 [29, 30]. In [10], they also mention that an imbalance on the order of 100 to 1 is prevalent in fraud detection.

In some specific cases, like the medical industry, the ramifications of a ML being biased towards the majority class, for example on a mammography dataset, can be overwhelmingly costly, more so than labeling a noncancerous patient as cancerous [31]. In the presence of skewed data, additional meaningful assessment measures, such as precision-recall curves and cost curves, are required for decisive performance evaluations [1].

When the imbalance is a direct result of the nature of dataspace, it is referred to as an *intrinsic* imbalance. Biomedical applications, fraud detection, network intrusion, and oil-spill detection problems are all inserted into this kind of imbalanced data [4, 14, 32]. However, some data is unrelated to the intrinsic imbalance because the imbalance is caused by factors other than dataspace, such as time and storage. This kind of imbalanced data is called as *extrinsic* imbalanced data. Extrinsic imbalances are equally as interesting as their intrinsic counterparts, as stated in [1], because the dataspace may not be imbalanced at all. For example, when a dataset is generated from a continuous data stream of balanced data over a defined interval of time, and the transmission has occasional interruptions where data is lost (the data is not transmitted), the acquired dataset can become imbalanced. This means that the dataset is extrinsic imbalanced with a balanced dataspace.

Besides the intrinsic and extrinsic imbalances, there is one important difference between *relative imbalance* and *imbalance due to rare instances*. The first one occurs frequently in real-world problems and is often the focus of many knowledge discovery and data engineering research studies. When the minority class is not necessarily rare, but rather relative to the majority class, a relative imbalance occurs. On the other hand, imbalances due to rare instances take place when the minority class examples are very limited, i.e., they are rare. Hence, the learning will be more difficult due to the lack of representative data.

2.3 Data Augmentation

At a ML problem, the predicted results can be improved by adjusting the data treatment, tuning the algorithm's hyper-parameters, using cross-validation, changing or stacking models, and so on. However, in an imbalanced data problem (e.g., fraud detection), the real problem can't be solved with data treatment and/or model changes since the limitation is in the data itself. The same goes for a small-sized dataset, since the model cannot learn enough features to classify the problem in a real-time situation.

Therefore, DA appears as a way to surpass that limitation. DA refers to methods for constructing iterative optimization or sampling algorithms via the introduction of unobserved data or latent variables [33]. With these techniques, we can increase the amount of data, thus balancing the target variable in an imbalanced dataset. DA can be applied to images or tabular data, the focus of this thesis. If the DA is applied to images, techniques tend to apply transformations to samples of datasets like geometric transformations, flipping, color modification, cropping, etc. One other way is to introduce new synthetic images created by ML models, for example, GANs. Instead, if we are dealing with tabular data, we cannot apply simple transformations to samples, but instead synthesize samples (new or duplicated) based on the class distributions and features. Throughout the years, many DA techniques have been developed or even adapted for this purpose on images or tabular data. Since the focus of this thesis is the application of DA techniques on tabular data, those techniques are going to be studied in more detail in section 2.4.

In [30], DA is utilized to surpass the data limitations of the minority class, in this case, fraudulent transactions. The classification performance improved considerably and overfitting was alleviated, demonstrating the benefits of using a DA technique. DA can also be applied to automated skin lesion analysis by applying traditional color and geometric transformations, and more unusual augmentations such as elastic transformations, random erasing, and a novel augmentation that mixes different lesions, as stated in [5]. They prove the importance of DA techniques in both training and testing, leading to more performance gains than simply obtaining new images. Face recognition datasets can be aided by introducing images from DA techniques, enlarging the training dataset, which alleviates pose variance, illumination changes, and partial occlusions, as well as the overfitting during training [6]. Image segmentation of magnetic resonance images of brain tumors, in [7], also used DA to increase the dataset and the robustness of a ML model. Moreover, emotion classification is another example of a problem that DA algorithms can help to improve the quality of data, as demonstrated by the authors of [8]. Emotion classification is usually a problem with imbalanced data because classes (i.e. emotions) like “disgusted” are relatively more scarce than other classes like “happy” or “sad”. The authors used GANs as a DA technique that increased the classification model performance from 5% to 10%. At [9] DA expanded the size of crash events regarding a crash dataset where there existed only 625 crash events to 6.5 million non-crash events.

2.4 Data Augmentation techniques

In this section, some of the most popular DA techniques are going to be described in detail. The algorithms discussed are: SMOTE, ADASYN, GMM, VAE, and GAN. These techniques are present in multiple DA researches and are going to be developed and evaluated in order to augment data.

2.4.1 SMOTE

SMOTE is a DA technique that oversamples the minority class by creating synthetic samples [10]. One way to solve the imbalance problem is to duplicate minority samples. However, this does not provide any new information to the ML algorithm training on the data. Therefore, instead of duplicating minority samples, SMOTE synthesizes new examples from that class.

SMOTE synthesizes the minority class by operating in the feature space. It selects examples that are close in the feature space and introduces synthetic samples along the line drawn from these examples.

Specifically, SMOTE synthesizes in the following steps, as seen in Figure 2:

1. Selects a random example from the minority class;

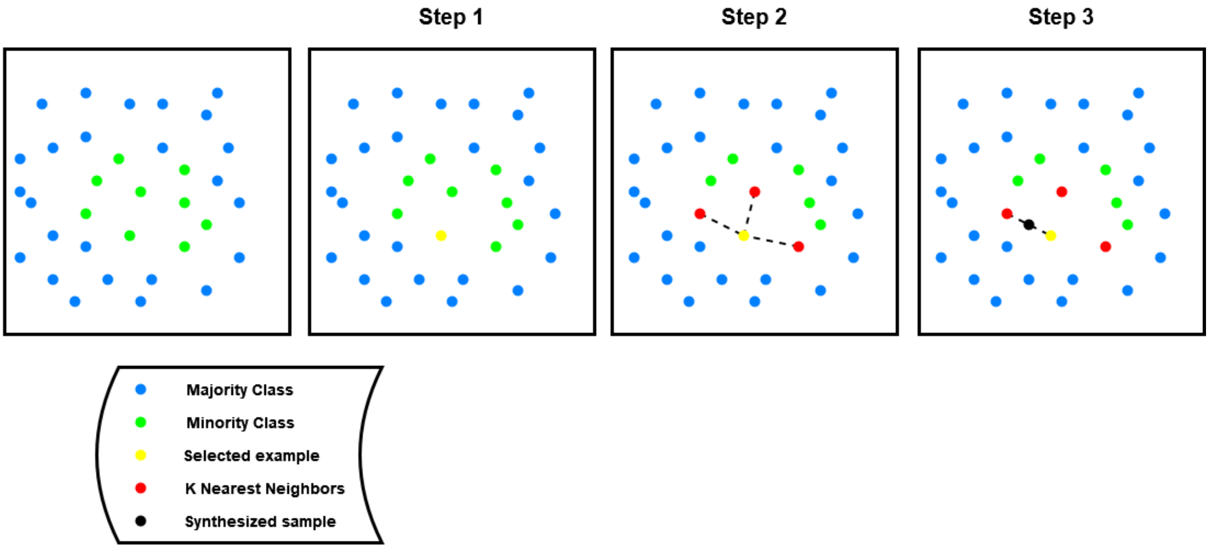


Figure 2: SMOTE algorithm

2. Then, k of the nearest neighbors (in the feature space) for that example are found[1];

3. A random neighbor is chosen, and it creates a synthetic sample at a random point between the two examples.

The technique is effective because the new synthetic samples from the minority class are somewhat close in feature space to real samples from that same class. This makes the created samples plausible.

In [10], the authors suggested undersampling the majority class and then applying SMOTE to the minority class to balance the class distributions. Undersampling could be done by erasing random samples from the majority class.

2.4.2 ADASYN

Unlike SMOTE, ADASYN approach does not focus on balancing the classes distributions, but rather on synthesizing minority samples that force the learning algorithm to focus on those difficult to learn samples, according to [11]. The objective of this technique is similar to those in SMOTEBoost [34] and DataBoost-IM [35] algorithms, compensating for the uneven distributions by giving different weights for different examples. However, the ADASYN's approach differs from the two previous algorithms because there is no hypothesis evaluation to synthesize data samples.

The algorithm only works if the dataset class distribution ratio is below the preset threshold. If so, it calculates the number of synthetic data samples that need to be created for the minority class with the following formula, with m_s and m_l being the number of minority and majority, respectively:

$$G = (m_l - m_s) \cdot \beta \tag{2.1}$$

[1] In the [10] paper, the authors use five nearest neighbors.

β is a parameter used to specify the desired balance ratio after the synthetization. $\beta \in [0,1]$, so if $\beta = 1$ it means that the dataset is fully balanced. Then, based on the Euclidean distance, find the k nearest neighbors and calculate the ratio r_i:

$$r_i = \triangle_i / k, i = 1, ..., m_s \tag{2.2}$$

where $\triangle_i$ denotes the number of samples in the k nearest neighbors of x_i that belong to the majority class, so $r_i \in [0,1]$. In order to have a density distribution[2], $\hat{r}_i$, we normalize r_i

$$\hat{r}_i = r_i / \sum_{i=1}^{m_s} r_i \tag{2.3}$$

The number of synthetic data that need to be synthesized for each x_i in the minority class is given by

$$g_i = \hat{r}_i \cdot G \tag{2.4}$$

For each minority class data sample x_i, generate g_i synthetic data by:

1. Choose randomly one minority data sample, x_j from the k nearest neighbors for data x_i

2. Generate the synthetic data sample:

$$s_i = x_i + (x_j - x_i) \cdot \lambda \tag{2.5}$$

   where $(x_j - x_i)$ is the difference vector in n dimensional spaces, and λ is a random number between zero and one.

With this procedure the ADASYN algorithm uses a density distribution, $\hat{r}_i$ as a criterion to decide the number of samples synthesized for each minority data automatically. $\hat{r}_i$ is a metric of the distribution of weights for different minority class samples in accordance with their level of difficulty in learning.

The resulting dataset after the application of the technique ADASYN is not a balanced representation of the classes distributions[3], because the technique forces the learning algorithm to focus on those difficult samples to learn.

2.4.3 Clustering

Contrarily to the other mentioned DA techniques, clustering is a type of unsupervised learning. Clustering algorithms divides the data into a number of clusters (groups or categories) [36].

The division of data into clusters is based on similarity and dissimilarity between them. Therefore, the definition of these two terms is extremely important. Once a proximity measure is determined, clustering

[2] A density distribution implies $\sum_{i=1}^{m_s} r_i = 1$.

[3] Even with the desired balance level stated in β.

can be constructed as an optimization problem. Therefore, the samples of data inside a cluster have a resemblance between them and a dissimilarity to the samples of other clusters. Typically, some of the most frequently used metrics to define the similarity between clusters are:

- Euclidean distance, and Manhattan distance for continuous variables;
- The Jaccard index is used to represent discrete or binary variables.

Similarity metrics

The Euclidean distance is the most common distance measure, and it can be explained as the length of a segment connecting two points. It is calculated from the coordinates of the two points using the Pythagorean theorem.

$$D(x,y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{2.6}$$

With Euclidean distance, it is common to normalize the data. Furthermore, this metric does not perform well with data that has a high dimensionality because higher-dimensional space does not behave as we would expect from two or three dimensional space. The most common use case for this measure is when you have low-dimensional data and the magnitude of the vectors is important to be measured.

On the other hand, Manhattan distance calculates the distance between real-valued vectors. The diagonal movement is not taken into account when calculating the distance.

$$D(x,y) = \sum_{i=1}^{k} |x_i - y_i| \tag{2.7}$$

Although Manhattan distance seems to perform better for high-dimensional data, it is a metric that is less intuitive than the previous one. Manhattan seems to perform quite well with discrete and/or binary attributes since it takes into account the paths that realistically could be taken within the values of those attributes.

Finally, the Jaccard index (or Intersection over Union) is a metric to calculate the similarity and diversity of samples. Its formula is given by dividing the intersection and the union of sample sets[4].

$$D(x,y) = 1 - \frac{|x \cap y|}{|y \cup x|} \tag{2.8}$$

Its major disadvantage is that it is highly influenced by the size of the data, since large datasets increase the union significantly whilst keeping the intersection identical.

The process used by clustering algorithms to partition data into separate groups can be understood using the similarity metrics presented. However, there are a variety of algorithms that can be used for this purpose, as we will discuss.

[4] If we have one sample in common in two sets, and there are five different entities in total, the Jaccard index is $1/5$.

Clustering models

There are four methods for clustering data:

- **Partitioning Methods**: It does partitions on the data, forming k clusters, making k different clusters. This method optimizes an objective criterion-similarity function (e.g., K-means);
- **Hierarchical Based Methods**: The clusters in this method form a tree structure based on hierarchy (e.g., CURE). New clusters are formed based on the previously formed ones, dividing them into two new categories. This method can be agglomerative (from the bottom-up approach) or divisive (from the top-down approach).
- **Density-Based Methods**: The density-based methods consider the clusters as dense regions that have some kind of similarity and differences from the lower dense region of the data space (e.g., DBSCAN).
- **Grid-based Methods**: In the grid-based methods, the data space is defined as a finite number of cells that form a grid-like structure (e.g., STING).

Of the multiple clustering algorithms from the different methods, K-means is one of the most widely used in the community. The K-means is compared to the GMM in the clustering data in [12]. Moreover, the GMM's generative nature provides an opportunity to explore its performance as a DA technique contrarily to K-means. K-means uses a pre-defined number of clusters (k) within an unlabeled multidimensional dataset. The cluster center is the arithmetic mean of all points in the cluster, making each point closer to the center of its cluster than to other cluster centers. The k cluster centers start at random positions and then iterate in phases, where, in each phase, it assigns certain points to each cluster center. The center is re-computed, forming an arithmetic mean of those points, where the closest center to each point is calculated with the chosen metric (Euclidean distance, Manhattan distance, etc.). Since the center of the cluster is not a data point from the dataset, but a mean from all points, the k-medoids algorithm was suggested. It behaves exactly as k-means with the expectation of the center that is a real data point (the closest to the calculated mean).

The use of a simple radial distance metric by k-means to assign cluster membership results in poor performance and a typical circular form for the clusters. This algorithm has no built-in way of accounting for non-circular clusters (oblong or elliptical), which do not represent the true shape of the data points sometimes. Moreover, this algorithm does not have a probabilistic nature when forming clusters.

Therefore, GMMs are an extension of the ideas behind k-means. This algorithm aims to model the data as a combination of multiple multi-dimensional Gaussian probability distributions. It works on the basis of the EM algorithm. The EM algorithm has two main steps: an estimation of the missing variables in the dataset (E-step) and the maximization of the parameters of the model in the presence of data (M-step) [37]. Because of this, the EM algorithm finds the maximum likelihood, i.e., finds a set of parameters that results in the best fit for the joint probability of the data sample [38].

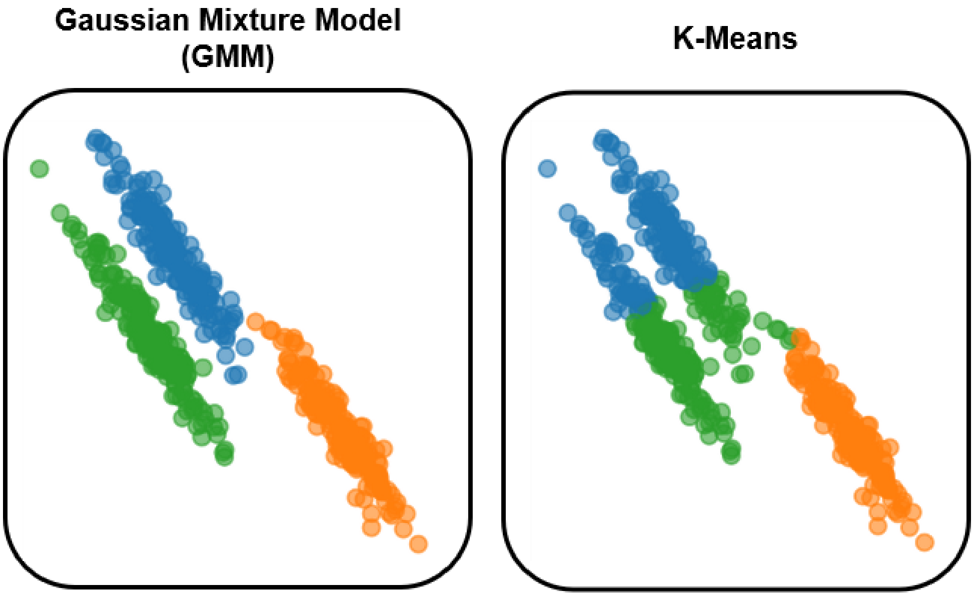


Figure 3: Differences applying GMM and K-means algorithms

GMM finds clusters in the same manner as k-means. However, it performs flawlessly for non-circular data forms. It can fit the Gaussian distribution parameters, such as the mean and standard deviation, to shape the data. The difference of clusters formations can be seen in Figure 3. Due to the generative nature of GMM, it can generate synthetic data close to the distribution of the fitted data [12]. After the algorithm fits the data and learns its distribution, it can generate an arbitrary number of samples from the learned distribution.

2.4.4 Variational Autoencoders

Nowadays, deep learning has gained a lot of interest and has made some amazing improvements regarding its performance. From the deep learning models, the family of generative models have also increased in popularity, showing a magnificent ability to produce highly realistic samples of various kinds, such as images, text, and sounds. These families of models, like all deep learning models, rely on huge amounts of data, well-structured architectures, and smart training techniques. Some popular deep learning generating models are VAEs and GANs.

In short, a VAE is an autoencoder whose encoding distribution is regularized during the training in order to ensure that its latent space[5] has good properties, allowing us to generate some new data [39].

Autoencoders are neural network architectures that have three parts in a stream: an encoder, latent space or bottleneck, and a decoder [9]. The encoder compresses the data into a lower-dimensional latent space. The decoder then tries to recreate the original/input data from the latent space. As a result, the autoencoder's goal is to optimize the iterative process in order to learn the ideal encoding-decoding scheme.

[5] Latent space is a representation of compressed data in which similar data points are closer together in space. It is useful to learn the features.

Moreover, by learning to encode and decode the input data, the autoencoder acquires knowledge of the data features and reduces noise. This is due to the algorithm's tendency to keep the most information possible while encoding and to minimize the reconstruction error when decoding. Figure 4 shows an example of the autoencoder algorithm when working with an image.

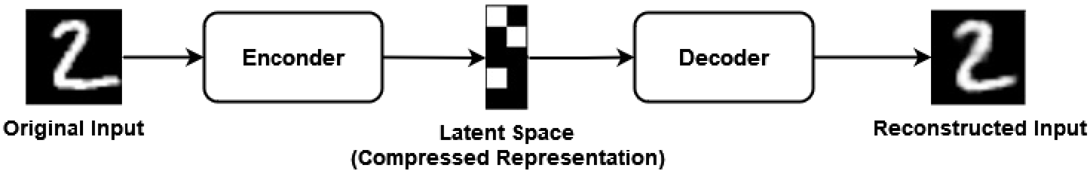


Figure 4: Autoencoder algorithm [Adapted from [40]]

A VAE consists of an encoder and a decoder, just like an autoencoder, but the loss term and the encoded layers of the autoencoder are altered in order for the model to be used as a generative model [9]. Its training is adjusted to avoid overfitting making sure that the latent space has good properties that enable the generative process. On the other hand, an autoencoder is trained to encode and decode with as few losses as possible, making no difference how the latent space is organized. The main distinction between the two encoding layers algorithms is that they encode an input as a distribution throughout the latent space rather than a single point [39]. With this in mind, the VAE avoids having some points in the latent space that would provide meaningless information once decoded.

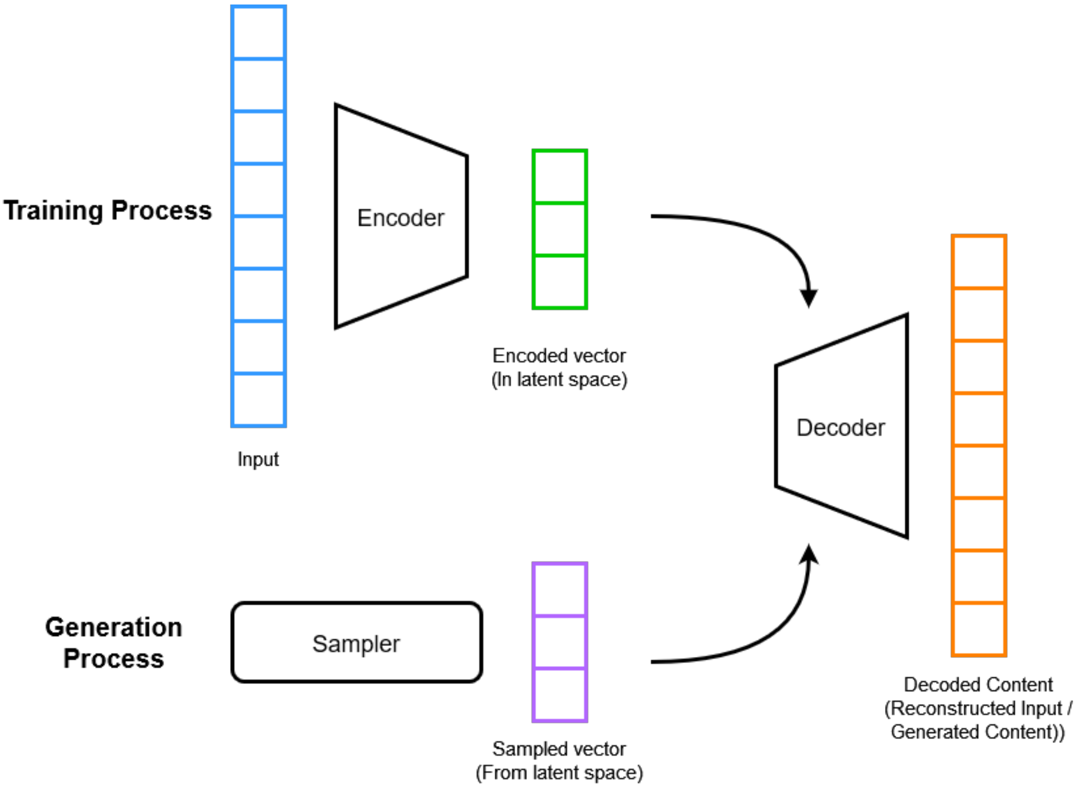


Figure 5: Variational Autoencoder algorithm [Adapted from [39]]

VAE's training process first encodes the input as a distribution over the latent space, as described in Figure 5. Next, a point from the latent space is sampled from that distribution, it is decoded, and the reconstruction error is calculated. Lastly, the weights of the network are changed based by the backpropagation of that error.

Finally, to make possible the generative process of the VAE, it is necessary for the latent space to have two properties. First, the latent space should have *continuity*, i.e., two close points in the latent space should not give completely different contents when decoded. Second, if a sampled point provides meaningful information once decoded, then the chosen distribution of the latent space has *completeness*. With these two properties, the latent space obtains regularization, i.e., the latent space distribution converges to the standard normal distribution. This regularization term prevents the model from encoding data that is far apart in the latent space and encourages returned distributions to "overlap" as much as possible[6]. Moreover, the two properties tend to create a gradient over the information encoded in the latent space [39].

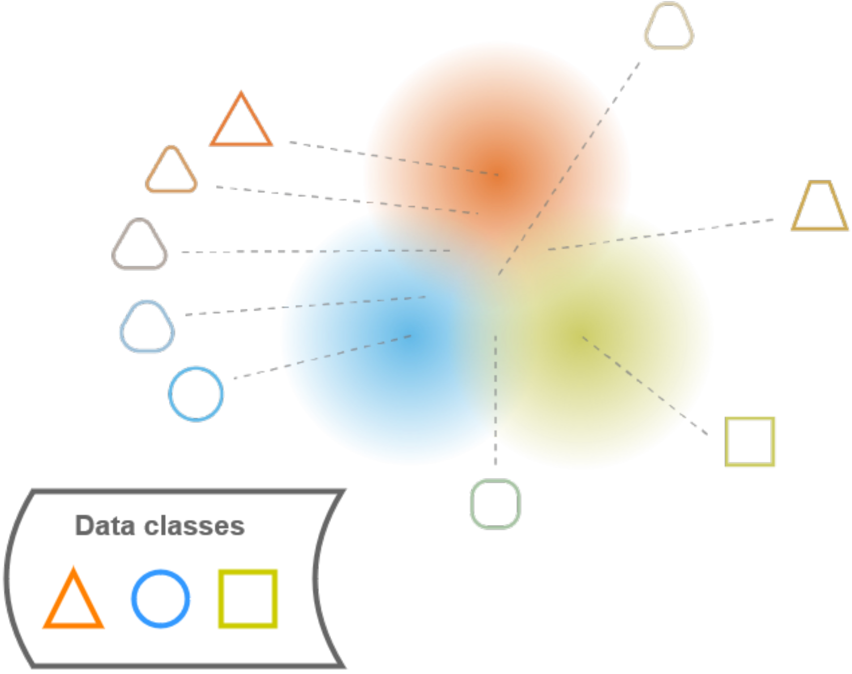


Figure 6: Overlaping classes in the latent space created by regularization [Adapted from [39]]

2.4.5 Generative Adversarial Networks

GANs, along side VAEs, are a famous deep learning generative model and were proposed in [41].

The GAN model architecture involves two neural networks: a generator and a discriminator. The generator is a model that generates new plausible samples for the problem, while the discriminator is a model that classifies examples as real (from the domain) or fake (generated) [42]. Therefore, GANs are

[6] There is an overlap between classes of data, Figure 6.

based on a game-theoretic scenario in which the generator network must compete against an adversary, the discriminator [41].

The Generator model takes a random vector, drawn from a Gaussian distribution, as input and generates a sample in the domain. This vector serves as a seed for the generative process. After its training, points in the latent space will correspond to points in the problem domain. This implies that the latent space is a compressed representation of the data distribution, much like the encoder from VAE, presented at 2.4.4. In the case of GANs, the Generator model applies sense to points in the specified latent space in such a way that new points drawn from the latent space can be fed into the Generator model as input and utilized to produce new and distinct output examples (see [42]). When the training is complete, the generator model is kept and used to generate new samples.

The Discriminator model receives as input an example that can be real or generated and predicts whether it is real or fake (generated). Therefore, the real inputs are received from the dataset, whereas the generated examples are output by the Generator model. Moreover, the Discriminator is a normal classification model and it is discarded as we are interested in the final Generator.

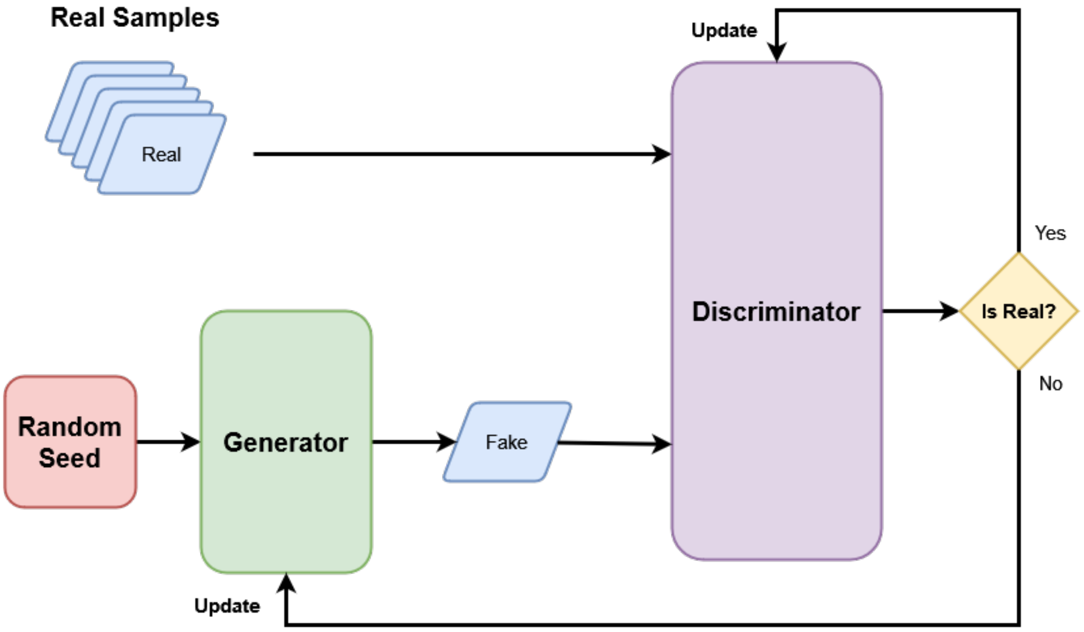


Figure 7: GAN algorithm

The Discriminator is updated to get better at discriminating between real and fake samples, and more importantly, the Generator is updated based on how well, or not, the generated samples fooled the Discriminator, demonstrated in Figure 7. Therefore, since the two models are competing against each other, they are opponents in their game. When the Discriminator identifies correctly, it is rewarded by not having to change its parameters, whereas the Generator is penalized with large updates to its parameters, and vice versa. Ideally, the Generator generates perfect samples so that the Discriminator cannot tell the difference in every case.

2.5 Comparison of Techniques

The previously mentioned ML algorithms are all specified to oversample and increase the size of data, e.g., by synthesizing the minority class of a dataset and balancing the class distributions. However, they are, in general, somewhat different DA techniques, as shown in Table 1. While techniques like GAN and VAE are inserted into the Deep Learning algorithms, others are based on more traditional ML algorithms. GMM stands out as the only unsupervised learning algorithm, not needing any kind of labeling on the data to train the model and find the structure of the clusters that best define the data. On the contrary, the other DA techniques need the labeled data to train the models, i.e., they are supervised learning models. Moreover, GMM, GAN, and VAE are part of the generative models family, i.e., they are models capable of producing new samples. Its goal is to reduce the dimensionality of data to a latent space and, through that latent space, generate samples that follow the same distribution of real data. Differences between the DA techniques are summarized in the Table 1.

Table 1: A comparison between Data Augmentation algorithms

Algorithms	**SMOTE**	**ADASYN**	**GMM**	**VAE**	**GAN**
Deep Learning	×	×	×	✓	✓
Unsupervised Learning	×	×	✓	×	×
Supervised Learning	✓	✓	×	✓	✓
Oversampling Technique	✓	✓	✓	✓	✓
Generative Model	×	×	✓	✓	✓

These algorithms have been used as DA techniques throughout multiple studies, having their performance evaluated. In [9], DA techniques were developed and their performance compared to combat the imbalanced dataset of traffic crashes, where only a minority of events are labeled as “crash”. Therefore, techniques such as VAE improved *specificity*[7] (obtained by $TN/(TN + FP)$, where TN and FP are defined in Table 2) for the Logistic Regression model when classifying data, about 2%, than both ADASYN and SMOTE while *sensitivity* (given by the metric recall, described in the next section) was lower by 4%. As for the SVM model, VAE generated better data with increased sensitivity and specificity that is comparable to the other two DA techniques. Finally, the authors concluded that DCGAN, a version of GAN, had worse results than VAE for the multiple classifiers. Overall, for this specific problem, the suggested algorithm (VAE) provided better results than the other techniques.

On the datasets used in [11], the authors proposed a new algorithm, ADASYN, motivated by the success of SMOTE. Therefore, the authors compared the algorithms’ performance on five different datasets. The chosen datasets were: vehicle (classifying the instance as one of four types of vehicles), Pima Indian Diabetes (predicting positive diabetes cases), vowel recognition (classifying different vowels), Ionosphere (classifying good or bad radar returns), and the Abalone dataset (predicting the age of abalone from physical measurements). The proposed algorithm performed competitively with SMOTE.

[7] Specificity is defined as the proportion of actual negatives that were correctly predicted as negatives.

An adopted version of GAN had better performance than oversampling with duplicated values, undersampling, SMOTE, and ADASYN in all datasets analyzed in [30]. The paper used three binary datasets for evaluation, whereas the main dataset utilized was the European credit card dataset.

Lastly, in [15], GMM synthesized data very similar to the real one, improving the ML model results in a dataset of the density of woods.

2.6 Assessment Metrics for Imbalanced Data

In order to compare the performances of all the DA techniques, it is required to define how their performances can be compared. Therefore, we need to define which ML metrics fit better into an imbalanced data problem. Traditionally, the most often used metrics are *accuracy* and *error rate*. Considering a basic two-class classification problem with positive and negative classes labels, it can be formulated a confusion matrix, as illustrated in Table 2.

Table 2: Confusion Matrix

	Real Positive	**Real Negative**
Predicted Positive	True Positive (TP)	False Positive (FP)
Predicted Negative	False Negative (FN)	True Negative (TN)

This metrics can be obtained by the following formulas:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{2.9}$$

$$ErrorRate = 1 - Accuracy \tag{2.10}$$

Although accuracy provides an easy way to describe the model's performance, it can mislead in certain situations. Imbalanced data problems are examples of that kind of deceiving, because if a minority class has 5 percent of examples and the majority has the rest of the data classes, a model that classifies all instances as being in the majority class has 95 percent accuracy. At first glance, this value appears to be an excellent classifier for the problem at hand, but it fails to identify any of the minority examples. Therefore, accuracy and error rate do not provide enough information about a classifier's functionality in terms of the sort of classification required.

In order to provide comprehensive assessments of imbalanced learning problems, the research community adopted other evaluation metrics, such as *precision*, *recall*, *F-measure*, and *G-mean*, defined as:

$$Precision = \frac{TP}{TP + FP} \tag{2.11}$$

$$Recall = \frac{TP}{TP + FN} \tag{2.12}$$

$$F - Measure = \frac{(1+\beta)^2 \cdot Recall \cdot Precision}{\beta^2 \cdot Recall + Precision} \tag{2.13}$$

where β is a coefficient that adjusts the relevance of precision versus recall[8].

$$G - Mean = \sqrt{\frac{TP}{TP+FN} \cdot \frac{TN}{TN+FP}} \tag{2.14}$$

First, precision is a metric that measures how many correct positive predictions the model makes (a measure of exactness)[9]. Therefore, precision calculates the accuracy of the positive class and is sensitive to data distribution. Second, recall is a metric that measures how many correct positive predictions were produced out of all possible positive predictions. Unlike precision, which only gives information on the correct positive predictions of all positive predictions, recall indicates the missed positive predictions and it is not sensitive to data distributions. Moreover, recall is also known as sensitivity.

When used correctly, recall and precision can evaluate an imbalanced learning problem adequately. Nevertheless, the F-measure metric combines the two previous metrics as a weighted focus on either recall or precision (given by the coefficient β). Finally, the G-mean (Geometric mean) metric evaluates the balance of classification between the majority and minority classes. Even if the negative cases are accurately identified, a low G-Mean suggests poor performance in the classification of positive cases.

To conclude, with these metrics, it is possible to evaluate the performance of the classifiers on an imbalanced data problem. Therefore, we can compare the model's results with different kinds of DA techniques. These metrics provide a specific evaluation on the prediction of the positive class (the minority class), providing more insight into a classifier's functionality than the accuracy metric.

2.7 Other Ways to Combat Imbalanced Data

While DA attempts to balance the classes distributions in a dataset, there are other ways to surpass the challenges of imbalanced data problems, e.g., Cost-Sensitive, Kernel-Based methods, and One-class learning [1].

Cost-sensitive learning methods consider the costs associated with misclassifying examples [43]. Rather than using several sampling procedures to achieve balanced data distributions, Cost-sensitive learning uses multiple cost matrices to represent the costs of misclassifying every single data example to target the unbalanced learning problem. The cost matrix can be viewed as a numerical representation of the penalty associated with classifying examples into different classes. In most cases, there is no cost for correctly identifying either class, and the cost of misclassifying minority examples is higher than the cost of correctly classifying majority examples. Cost-sensitive learning's goal is to minimize the overall cost of the

[8] Usually, the community uses $\beta = 1$, which weights precision and recall equally, and the metric is called *F1-measure* or *F1-score*.

[9] In this case, the positive class is considered the minority.

training data set. Some studies have shown that cost-sensitive learning is superior to sampling methods (DA) in some imbalanced data domains.

Although DA by sampling methods and cost-sensitive learning methods appear to dominate current research efforts in imbalanced learning, the community has also investigated a variety of additional approaches. Kernel-Based methods, such as GSVM-RU in [44], were created to deal with imbalanced data problems because they provide state-of-the-art methodologies for many of today's data engineering applications.

Finally, rather than distinguishing between instances of both positive and negative classes as in traditional learning methodologies (i.e., discrimination-based inductive methodology), one-class learning aims to recognize instances of a concept by using mostly, or only, a single class of examples (i.e., recognition-based methodology) [1].

2.8 Summary

As a result of the increased amount of data available, ML has been the target of a great deal of interest recently, being used in various and diverse fields of work. It is divided into three categories: supervised learning, unsupervised learning, and reinforcement learning. Supervised learning algorithms receive labeled data in the training process, whereas unsupervised algorithms search for patterns in unlabeled data. Furthermore, reinforcement learning trains machines through trial and error, providing them with the means to learn for themselves the best strategy for the problem at hand.

Even though the amount of data collected with recent technology is increasing, imbalanced datasets are still present. These datasets have an unequal distribution between their classes, which decreases the performance of the ML models. Therefore, DA appears as a way to surpass that limitation, increasing the amount of data. It can balance the classes distributions or increase the size of the dataset through oversampling techniques. Some of the most popular DA techniques are: SMOTE, ADASYN, GMM, VAE, and GAN. All of them are supervised models, except the GMM, which is unsupervised. Furthermore, VAE and GAN are from the generative models family and deep learning models, whereas the rest are not based on generative nor neural network models.

In order to analyze and compare which DA technique synthesizes better data, it is required to define how their performances can be compared. Since traditional metrics, such as accuracy and error rate, mislead the ML models' performance on an imbalanced learning problem, it is important to research better fitted metrics. Therefore, precision, recall, and g-mean were taken into account. These metrics provide a specific evaluation on the prediction of the positive class (the minority class), providing more insight into a classifier's functionality than the accuracy metric.

Finally, DA is not the only way to combat imbalanced learning problems. Other ways, such as cost-sensitive, kernel-based methods or one-class learning, are also alternatives without oversampling samples of the dataset.

Chapter 3

Experimental Setup

This chapter will go over the key stages of the project, detailing the methodologies and the setup of the experiments realized, which will be described in further detail in the next chapter.

3.1 Methodology

This study can be divided into two phases. The first one is data collection, treatment, and initial classifiers development. This will be done with public data from platforms such as Kaggle, Google Dataset Search, and the UCI Machine Learning Repository. Therefore, it was possible to solve imbalanced learning problems with ML models. These datasets underwent a data treatment that is typical to extract information and be ready to be learned by a classifier. After the data treatment, different ML models were developed, from the traditional ML models to deep learning, in order to classify and evaluate their performance on imbalanced data. Each problem (i.e., each prediction from the multiple datasets) throughout the research followed the CRISP-DM methodology when processing data and developing the ML models, which is a standard process model that describes the most prevalent data mining approaches [45]. Its major phases are Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation, and Deployment (as demonstrated in Figure 8). In order to evaluate their performance, the metrics stated in Section 2.6 were also used.

The second phase had the objective of developing DA techniques. These methods, previously discussed in Section 2.4, generated synthetic data, balancing classes distributions or incrementing the dataset size.

The benchmark performed in this study involved multiple analyses of the implemented DA methods throughout various datasets with multiple classifiers, such as Decision Trees, Random Forests, and MLPs.

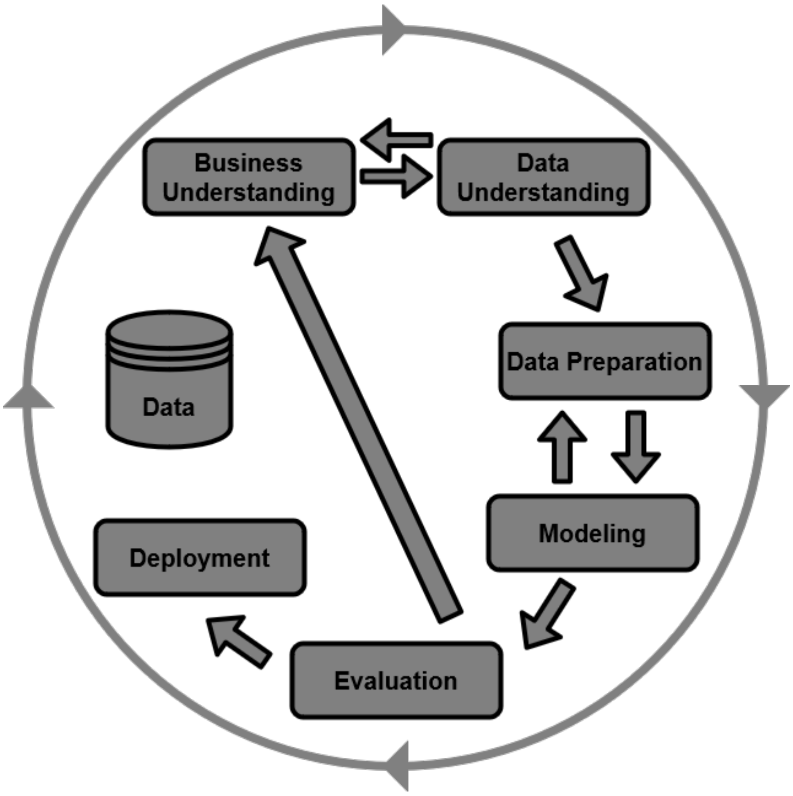


Figure 8: CRISP-DM major phases

These machine learning algorithms were chosen based on their current scientific popularity and the typically good performance associated with them.

Regarding the synthetic data generated by all DA techniques, the experiments will focus on many analyses in order to answer the previously proposed research questions. These analyses consist, mainly, of:

1. Comparing each feature's distribution throughout statistical methods;
2. Training the classifiers only with synthetic data;
3. Contrasting the number of generated samples added to the real data;
4. Training the classifiers with real and synthetic data.

Finally, the data treatment, the ML models, and DA methods were developed with the help of the Python programming language and its available libraries. For data treatment, libraries like Pandas [46] and Numpy [47] offered multiple tools to process the data. As for the ML and deep learning algorithms (i.e., classifiers and DA methods), Scikit-Learn [48], Keras [49], and Tensorflow [50] provided easy ways to develop and implement the algorithms.

3.2 Experiments

With regard to the classifiers benchmark, the ML algorithms were implemented and their results evaluated with the application of random seeds[1] in order to be able to replicate the results. In an effort to avoid overfitting the classifiers to the training data, we focused mainly on the MLP, since it was the model most susceptible to it. Therefore, to handle that, it was used as a callback to stop the training process when the performance on the validation data[2] would not increase for multiple epochs. However, the use of *dropout* layers was also intended to help avoid overfit.

Despite the fact that some DA methods were already implemented by Python packages, the VAE and GAN needed to be fully developed, particularly their architecture. The package Scikit-Learn already had implementations of SMOTE, ADASYN, and GMM, but more complex techniques (VAE and GAN) were developed in Tensorflow and Keras. Therefore, for this dissertation, both of these techniques followed an architecture of multiple dense layers with some intermediary dropouts and batch normalizations. While it is usual to use convolution layers in an image-based problem for DA techniques that are based on DL algorithms, in tabular data it is not (but could have been done by the 1D convolution layer, named Conv1D in Tensorflow). Note that for each problem, the DA methods' architecture did not change, although fine-tuning (hyperparameters optimization) could benefit the performance of the algorithms when generating synthetic samples for very specific domains. The reason for that decision was based on time and computational cost issues, since the fine-tuning of the hyperparameters of each DA technique implied complex and time-demanding solutions, such as the use of a genetic algorithm for the optimization of those hyperparameters for each domain [51].

Furthermore, during the experiments and implementation of the DA techniques, there were some obstacles with regard to the performance of some techniques, mainly the VAE and the GAN. The implemented VAE suffered from the phenomenon called *posterior collapse* [52], as well as the GAN suffered from *mode collapse* [53]. The posterior collapse happens when the information contained in the learned latent space is rendered useless. As for GAN, its generator only learns to generate a small set of outputs, making the generator over-optimizing for a particular discriminator. As a result, the minority class was unable to be synthesized. Therefore, it was crucial to find a way to surpass this limitation. The solution passed through adding noise to the discriminator's inputs and incrementing the latent space dimension.

Moreover, we explored a bit more of the possible solutions for the VAE. The first solution was to add a weighted decay to its loss. The VAE loss is composed of two factors. The first forces the decoded samples to resemble the input by penalizing the latent representation with a reconstruction loss (RC_{loss}) [54, 55]. Consequently, the RC_{loss} can be explained as:

$$RC_{loss} = -\sum_{bs=1}^{BS}\sum_{i=1}^{N} x_{bs,i} \cdot log(x'_{bs,i}) \tag{3.1}$$

[1] The same value, 42, was used as the random seed across the entire code.

[2] The available data for each problem was divided into training, validation, and test datasets.

where x is the input data, N the dimensions of the input data, BS the batch size, and x' the reconstructed input. Keep in mind that this loss is predicated on the probability of the Binary Cross-Entropy.

The second loss is the Kullback-Leibler divergence term (KL_{loss}), which serves as a regularization term to aid the model's learning of well-formed latent spaces:

$$KL_{loss} = \sum_{bs=1}^{BS} \sum_{i=1}^{N} \frac{\sigma_{bs,i}^2 + \mu_{bs,i}^2 - 1 - 2log(\sigma_{bs,i})}{2} \tag{3.2}$$

where μ, σ are the mean and standard deviation of a Gaussian distribution, respectively, BS is the batch size, N the dimensions of the input data [54, 55]. The network weights are controlled by the weight decay (W_{decay}), which penalizes the KL_{loss} more as the number of training epochs increases[3] and causes the model to become more regular. The VAE network is thus prevented from overfitting the training data, which is typically towards the majority class, by this weight decay. In essence, the VAE loss function is:

$$VAE_{loss} = RC_{loss} + KL_{loss} \cdot W_{decay} \tag{3.3}$$

As for the other solution, we explored the latent space properties, adding a k-means algorithm to be applied to the latent space. This cluster algorithm would be fitted after the VAE training in order to identify the minority and majority clusters. Therefore, before the VAE generated samples, the cluster algorithm would readjust the points of the latent space[4], making them closer to the centroid of the minority class cluster. An example of the application of this cluster algorithm to the process of synthesizing by the VAE can be seen in Section 4 by the Figure 9. Finally, the last version of VAE implemented in regards to surpassing the posterior collapse was the combination of the weight decay and the k-means.

Furthermore, the use of more computationally demanding DA methods, such as VAE and GAN, demonstrated a greater importance for each dataset preprocessing, particularly for continuous features. These continuous features were an issue during the training of these techniques because of the calculated loss. Another occurring phenomenon was that the loss had NaN values during the training due to high computations when faced by continuous features. As a result, a simple solution to this problem was to perform a normalization on each continuous feature.

[3] The weight decay has a proportional inverse behavior in regards to the number of training epochs.

[4] In this case, the latent space is the input vector that the decoder uses to generate samples.

3.3 Data

In this experiment, multiple datasets were chosen to perform a good comparison of these DA methods in generating new data. Additionally, these datasets are unbalanced[5] and inserted into many domains, including fraud detection and health. The chosen datasets were the following:

- **Adult**. The adult dataset was extracted from the census bureau and has information about multiple adults [56]. This dataset serves as a binary classification, predicting if a certain adult has an income superior to fifty thousand in a year. Regarding its features, it has seven categorical features and one continuous. Furthermore, the target class is clearly imbalanced, as the majority class (income superior to fifty thousand) is three times more frequent than the minority class. The dataset has over 30K instances.

- **Breast Cancer**. Another health domain analyzed in this experiment was breast cancer prediction [57]. This dataset was obtained from the University of Wisconsin Hospitals, Madison by Dr. William H. Wolberg and contains samples of clinical cases gathered periodically. The dataset contains a target class imbalanced with 66% of the instances belonging to benign cases and the rest being malignant. This dataset, beyond being imbalanced, is also small in size, since it only has 569 instances. Its features are all continuous.

- **Credit Card Fraud**. Fraud detection is also a recurrent domain where imbalanced data is present [58]. Therefore, in this dataset, transactions made by credit cards in September 2013 by European cardholders are analyzed. This dataset originally had more than 280K instances but was reduced (while maintaining the target class ratio) to 85K due to computational reasons. Moreover, this dataset contains only continuous attributes.

- **Cerebral Stroke**. This dataset contains features regarding individuals that may or may not suffer a cerebral stroke [59]. A cerebral stroke is when part of the brain loses its blood supply and the part of the body that the blood-deprived brain cells control stops working. Therefore, it is very useful to predict if a person may or may not suffer a stroke. This dataset is highly imbalanced, having only 2% of strokes and more categorical features than continuous. The dataset has more than 40K instances.

The characteristics of the datasets gathered for this dissertation are summarized in Table 3. Take into account that the column “Target Class Ratio” is obtained through a ratio of the number of samples in the minority class against those on the majority class. Since all chosen datasets have a binary target class, this ratio is between 0 and 100. Therefore, when the value is closer to zero, it means that the dataset is extremely imbalanced. On the other hand, with a value of 100%, the dataset is completely balanced.

[5] Note that the chosen datasets have a binary target.

Table 3: Datasets properties

Dataset	Number of Instances	Type of data	Target Class Ratio [%][1]
Adult	32561	Mostly Categorical	31.7
Breast Cancer	569	Mostly Continuous	59.4
Credit Card Fraud	85442	Mostly Continuous	0.2
Cerebral Stroke	43400	Mostly Categorical	1.8

[1] Target Class Ratio is obtained through a ratio of the number of samples in the minority class against those on the majority class.

In essence, the datasets examined in this study consist of two datasets with predominantly continuous characteristics and other two with mainly categorical features. Moreover, one of the datasets has a small size while the rest have a lot more instances. Therefore, we can analyze how the synthetic data generated performs on datasets with these characteristics.

3.4 Summary

The study of this dissertation is composed of the collection of datasets, their treatment, and the training of the classifiers. When classifying the data for each problem, the CRISP-DM methodology will be used, and the DA techniques that generated synthetic data quality will be evaluated in the second phase. This will be done by either increasing the minority class representation, balancing both class distributions, or increasing the size of a dataset. Moreover, the classifiers (Decision Tree, Random Forest, and MLP) will be implemented by packages such as Scikit-Learn, Keras, and TensorFlow.

The experiments will evaluate the synthetic data in multiple ways. First, by comparing each feature's distribution throughout statistical tests and training the classifiers only with synthetic data. Second, by contrasting the number of generated samples to the real data. Finally, the classifiers will be trained with real and synthetic data in order to evaluate if the addition of synthetic data increases the classifiers' performance.

The implementation of VAE and GAN suffered from the phenomenon called posterior collapse and mode collapse, respectively. The posterior collapse happens when the information contained in the latent space is rendered useless, while the mode collapse makes the GAN's generator over-optimizing for a particular discriminator, synthesizing only the majority class. To surpass this limitation, we implemented a decay on the VAE loss and applied the K-means to its latent space in order to approximate the feature space points for the minority class. Aiming to evaluate these two posterior collapse solutions, we performed multiple combinations of the VAE during the experiments. As for the GAN, we added noise to the discriminator's inputs and incremented the latent space dimension.

Furthermore, the more complex DA techniques (VAE and GAN) also suffered loss of computation errors due to the existence of continuous features that were not normalized. As a result, it emphasized the importance of preprocessing for each dataset, particularly for continuous features.

Finally, we chose four different datasets for the experiments, inserted in many domains, including fraud detection and health. Of the four datasets, two of them had mainly continuous features, while the rest had more categorical features. Moreover, one of the datasets was considered a small-sized dataset.

Chapter 4

Results and Discussion

In this chapter, we will go through all the results obtained by the experiments mentioned in the previous chapter. Therefore, we will answer all the research questions stated at the beginning of this dissertation and provide some insights about our findings.

4.1 Statistical Tests

First, in order to compare the quality of the synthetic data generated by each augmentation technique, we performed statistical analyses on both real and synthetic data, i.e., to determine if they came from the same distribution. Ideally, and when talking about the same number of samples, the synthetic data should have properties very similar to the real one. Therefore, we implemented some statistical methods to compare each feature distribution on the real and synthetic datasets. However, due to the different behavior of continuous and categorical features, it was necessary to apply different statistical tests. On that account, the categorical features distributions were analyzed by the *chi-square test* and the continuous features by the *Kolmogorov-Smirnov test*.

These statistical tests showed that the DA techniques generally had difficulties representing the original continuous features distributions in the generated data. SMOTE was the technique with better representation, followed by ADASYN and the variations of VAE, namely VAE with K-means and VAE with K-means and decay on its loss. However, representing the categorical feature distributions was something easier for the DA methods. Although SMOTE had a good capability to represent continuous features, it was the worst technique in regards to representing categorical features. As for the other techniques, ADASYN and the VAE with K-means and weighted decay on its loss were the ones more capable of producing similar distributions. The detailed results of these statistical tests are present in Appendix A.

4.2 Results of the application of K-means on VAE

Before analyzing the classifiers' performance with synthetic data, it was observed how using a k-means to alter the latent space on a VAE could affect the synthesized samples, and if, as expected, the number of minority samples generated would be higher. Figure 9 demonstrates that the VAE with K-means, as expected, can produce way more minority samples through a 2D visualization of the target variable on the Breast Cancer dataset. Note that this dataset was chosen to analyze the effects of the utilization of K-means on the latent space of a VAE due to the lower number of samples on this dataset, which facilitates the visualization. One important fact to take in mind is that since the K-means process is unsupervised, the minority class is defined by the cluster with less feature space points. However, if the VAE enconder does not do its job properly, for example has a weak architecture, it can chose wrongly the minority class and, therefore, benefit even more the majority class.

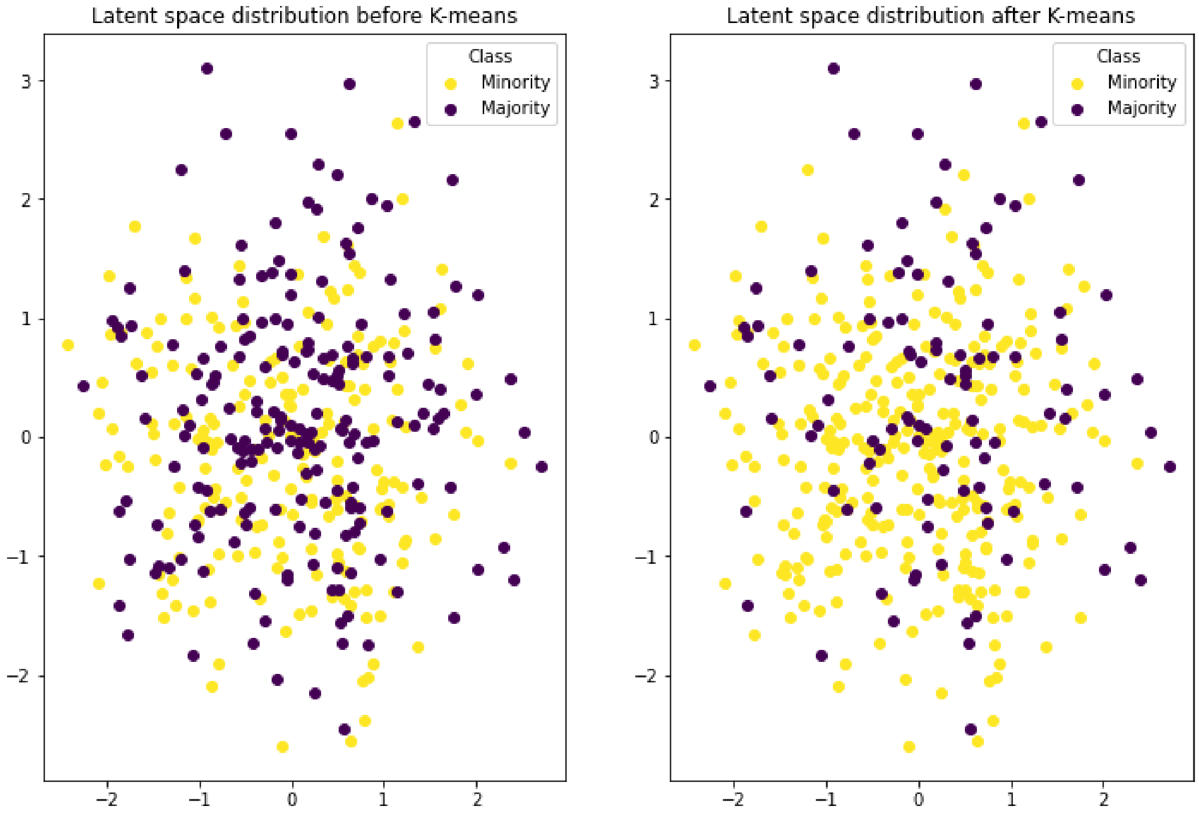


Figure 9: Application of the K-means on the process of generating samples of VAE

4.3 Synthetic Data Ratio Analysis

Although the DA choice is extremely important, the number of samples to generate is a crucial factor too. Therefore, we analyzed which ratio of synthesized samples provided the best results for the classifier performance. As a basis of comparison, we compared the various ratios to the classifiers' performance with no addition of synthetic data. The DA ratios are percentages of the size of the dataset for each problem, so when comparing a ratio of zero to one, we are comparing the classifier's mean performance when training with the original dataset versus a dataset with the same original data plus the same number of synthetic samples. Another fact to take into account is that the metric values on the following plots are averages of the various classifiers used.

As seen in Figure 10, the adult dataset showed the best results with the SMOTE technique. The classifiers' performance, with the addition of synthetic data, tended to maintain or decrease a little throughout all DA techniques. However, SMOTE has the technique with better performance, making the minority class performance increase more with the addition of 25% of synthetic samples. As for the rest of the techniques, most of them showed difficulties in improving the classifiers' performance, except for the GAN.

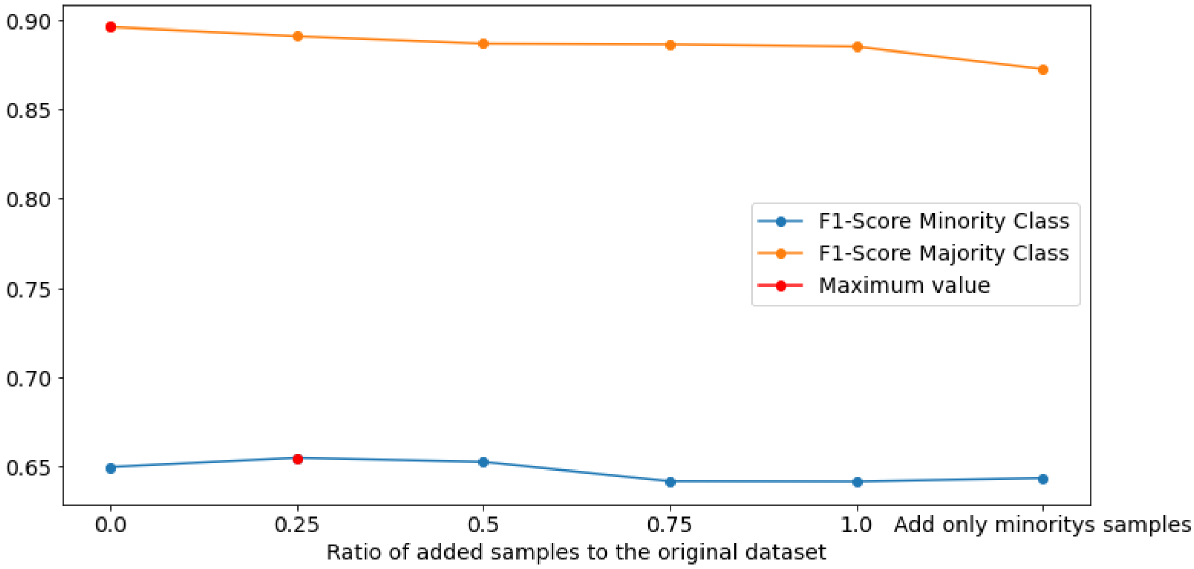


Figure 10: Ratio of data added at the Adult dataset by SMOTE

On the contrary to the adult dataset, the addition of synthetic data to the Breast cancer dataset improved both classes classification performances as described at the Figure 11. Although the addition of 25% of synthetic samples increased by a lot the performance, the experiments showed that adding only minority samples was the best choice. The techniques that demonstrated better performance were the GAN and VAE with K-means.

As for the Credit Card Fraud dataset, the ratio performance, seen in Figure 12, increased more when adding only minority samples. The VAE with K-means was the technique that improved the training data for the classifiers, followed by the other VAE variations and the GAN. The overall results were a lot similar to the Breast dataset, since the ratio and techniques that achieved better performance were, in essence, the same ones. This could be explained by the properties of the two datasets that are very similar since they are only composed of continuous features.

Finally, for the Cerebral Stroke dataset, the results were also almost identical to the dataset with similar properties (Adult dataset), as the best ratio of synthetic samples to add was 25% too, and the technique

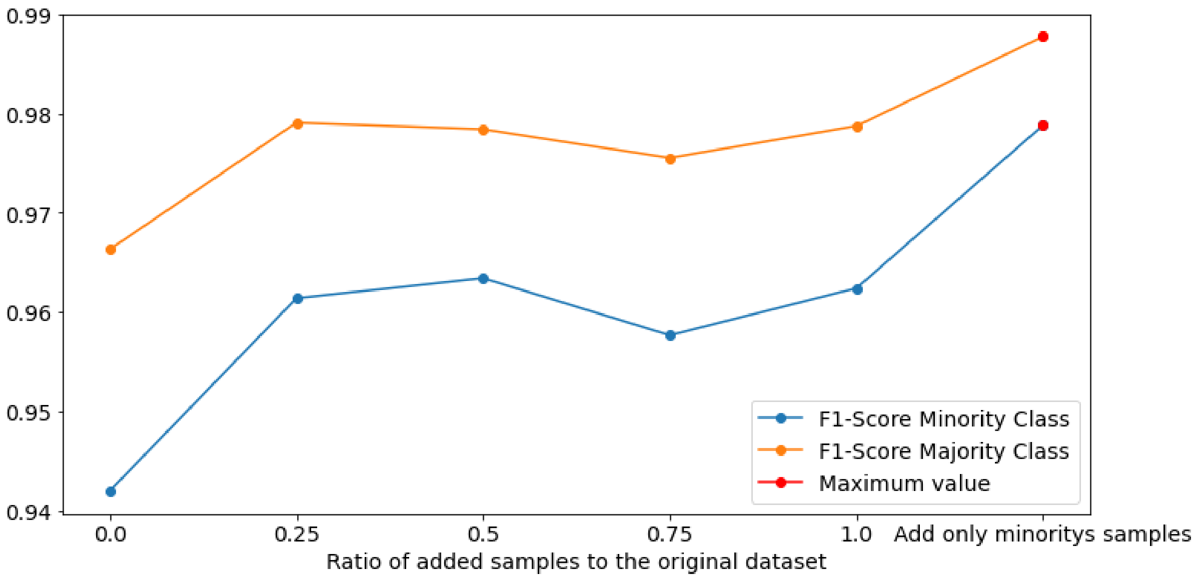


Figure 11: Ratio of data added at the Breast dataset by GAN

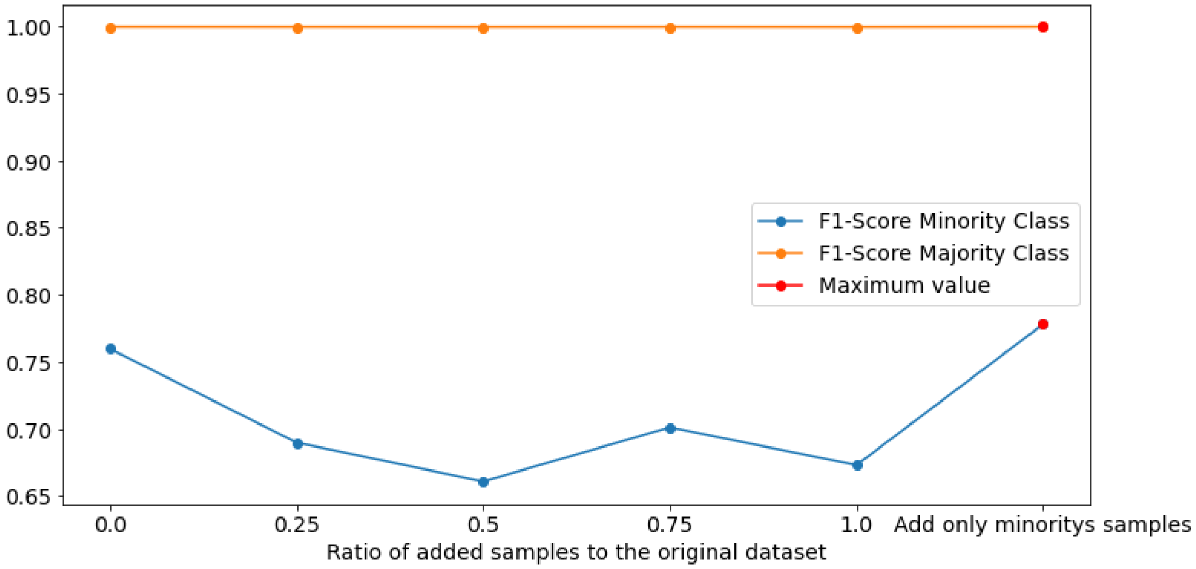


Figure 12: Ratio of data added at the Credit Card Fraud dataset by VAE with K-means

that resulted in a better performance by the classifiers was SMOTE. The results may be observed at Figure 13.

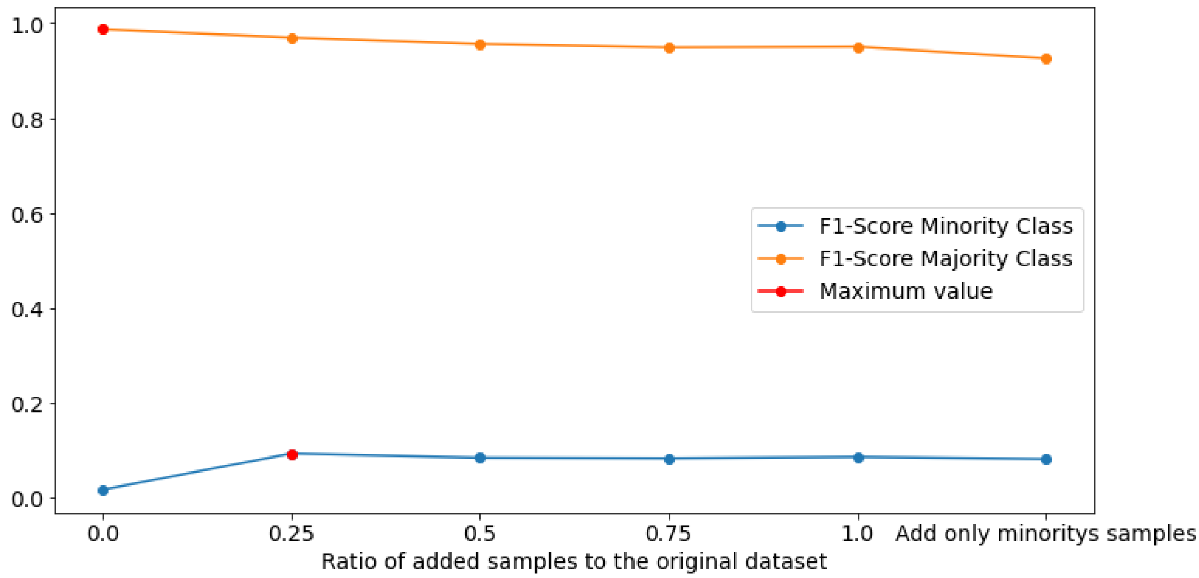


Figure 13: Ratio of data added at the Cerebral Stroke dataset by SMOTE

In essence, this experiment showed that more categorical datasets could achieve greater performance by adding 25% of samples. As for the continuous datasets, the best ratio was to add only minority samples. The chosen techniques seemed to indicate a certain pattern. The detailed results of all DA methods for all datasets are seen at Appendix B.

4.4 Synthetic Data Performance Analysis

We can still perform two additional crucial analyses to determine how reliable the generated data is after the analysis of the synthetic data properties. First, we are going to train the machine learning classifiers with real data and then compare the results with training with only synthetic data.

Synthetic data can achieve very similar results when replacing the real data in the classifier training. This experiment is described in Table 4 which represents the best DA techniques for each dataset on the Random Forest Classifier[1]. The main classifier chosen to analyze the synthetic data quality was due to the more consistent performance throughout all the techniques, as techniques such as VAE and GAN showed more volatility in the MLP and worse results on the simpler machine learning classifier, the Decision Tree.

In regards to the Adult dataset, the technique that, throughout all classifiers, performed better was the SMOTE, followed by ADASYN. This implies that these techniques may generate synthetic data with more quality than the rest of the techniques for this dataset, with more categorical features than continuous. One other important fact to take into account was the visible difference in the performance of one variation

[1] In order to abbreviate the document, the full results will be present in the Appendix section C.

of the VAE when its generated data was trained by the MLP classifier, which can be observed at Table 9. This classifier produced worse results than the other two more classical machine learning classifiers, and it was seen multiple times for the other datasets as well.

The Breast Cancer dataset had the VAE with K-means as the technique that generated the best quality of data, surpassing even the real data performance, described at Table 10. The Random Forest classifier attained a higher performance score, although it couldn't classify any minority classes correctly with GAN's synthetic data. This can be explained by the classifier overfitting when trained with those samples.

As for the Credit Card Fraud dataset, the SMOTE's synthetic data achieved superior results across all classifiers as can be seen at Table 11. It also surpassed the performance of the classifier with real data on both classes. Moreover, this dataset's classifiers were more susceptible to overfitting towards the majority class, principally with VAE variations and GAN.

Finally, in the Cerebral Stroke dataset, the clustering technique, GMM, and SMOTE were the techniques that had the closest results to the real data. GMM had a higher recall, capturing more minority samples than SMOTE, who had a superior performance in the majority class. This dataset suffered, like the previous ones, from the overfitting phenomenon in some of the generative DA techniques (VAE and GAN). Its results are described at Table 12.

These experiments demonstrated that most of the DA techniques can synthesize data in order to replace the real data in a somewhat efficient way. Note that in some cases, the use of only synthetic data as training data for the classifiers provided better results than with real data. Therefore, the use of only synthetic data could be very interesting in datasets where some data is sensitive and privacy matters.

Table 4: Classifiers performance comparison on only synthetic or real data training on the datasets

		Minority Class			**Majority Class**			
Dataset	**DA Technique**	**Precision**	**Recall**	**F1 Score**	**Precision**	**Recall**	**F1 Score**	**G-Mean**
Adult	No technique	0.705	0.6301	0.6655	0.8865	0.9164	0.9012	0.7599
	SMOTE	0.5373	0.7781	0.6356	0.918	0.7875	0.8478	0.7828
Breast Cancer	No technique	1.0	0.9063	0.9508	0.9474	1.0	0.973	0.9520
	VAE with K-means	1.0	0.9375	0.9677	0.9643	1.0	0.9818	0.9682
Credit Card	No technique	0.8824	0.6818	0.7692	0.9995	0.9998	0.9996	0.8257
	SMOTE	0.8947	0.7727	0.8293	0.9996	0.9998	0.9997	0.8790
Cerebral Stroke	No technique	0.2	0.0085	0.0163	0.982	0.9994	0.9901	0.0920
	GMM	0.0375	0.9576	0.0722	0.9986	0.5462	0.7062	0.7232

Note: The results are from the classifier Random Forest and the DA technique chosen was the technique that achieved the best scores.

With those results in mind, the following analyses focus on training the classifiers with an increased amount of data throughout multiple datasets, described in Table 5, Table 6, and Table 7. In regards to the Random Forest classifier, its results demonstrated two important points.

First, the choice of the number of samples to generate is crucial, as the classifier's performance may decrease if the final dataset has too many synthetic samples with less quality than the real data. Therefore, for each problem, it was necessary to evaluate the number of samples to synthesize. In these experiments, the adult dataset had better behavior by adding 25% more samples to the original dataset, while the rest of the datasets had higher performance by adding only minority samples. Note that while the best ratio when using only synthetic data (described in Figure 13) was 25%, this was not the same when combining real and synthetic data for the Cerebral Stroke dataset, which achieved a higher score when adding only minority samples.

Second, the combination of real and synthetic data improved the classifiers' performance. In the adult dataset, SMOTE and GAN were the techniques that incremented most of the metrics for the minority class while not decreasing the majority class performance. In the Breast Cancer dataset, the results were even more satisfactory, with an increase of 5% in the minority class f1-score. The VAE with K-means and decay did a remarkable job since it refined the performance for each class to a round 100%. The Credit Card Fraud dataset had similar results as the previous one, with a boost in the f1-score minority class by 8% in the VAE with K-means. Finally, in the Cerebral Stroke dataset, the minority results increased 8 times the initial results with no synthetic samples on the f1-score as well as by the technique SMOTE.

Moreover, these experiments permitted us to confirm some interesting facts that were mentioned previously. Datasets that contained mainly categorical features were usually associated with a difficulty in increasing the classifier results by adding new samples. Plus, the technique that adapted best to those kinds of datasets was the more classical one, SMOTE. On the other hand, for the datasets with more continuous features, the variations of the VAE, mainly VAE with K-means and VAE with K-means and decay, had very good performances.

On the other hand, the results obtained by the Decision Tree classifier, seen in Table 6. The Decision Tree classifier showed slightly worse results in comparison with the Table 5, as expected since the classifier Random Forest typically provides a more accurate performance. However, the techniques' synthetic data demonstrated similar properties. In the adult dataset GAN, followed by VAE with K-means and decay, and SMOTE were the methods that generated better quality data, increasing the classifier performance when classifying both classes. Moreover, SMOTE was the method that increased the minority class recall, capturing 62.33% of the instances (an improvement of more than 1% in comparison with the classifiers' performance when no technique is applied). As for the Breast Cancer dataset, contrarily to the Random Forest classifier results, the Decision Tree showed that GAN was the DA technique with better quality data generated. Furthermore, when compared to the non-use of DA techniques, the previous better method on Table 5, VAE with K-means and decay, demonstrated a decreasing performance. Moreover, the Credit Card Fraud dataset continued to have a variation of the VAE technique as the best generator of data. However, the SMOTE method showed a decreased performance with a much lower minority recall. Finally,

Table 5: Random Forest performance throughout the multiple DA techniques

Datasets	DA Technique	Minority Class			Majority Class			G-Mean
		Precision	Recall	F1 Score	Precision	Recall	F1 Score	
Adult[1]	No technique	0.705	0.6301	0.6655	0.8865	0.9164	0.9012	0.7599
	SMOTE	0.6801	0.6582	0.669	0.8927	0.9019	0.8973	**0.7704**
	ADASYN	0.6709	0.6259	0.6476	0.8838	0.90267	0.8932	0.7516
	GMM	0.7067	0.625	0.6634	0.8853	0.9178	**0.9012**	0.7574
	VAE with Decay	0.7003	0.6318	0.6643	0.8868	0.9143	0.9003	0.76
	VAE with K-means	0.7013	0.631	0.6643	0.8866	0.9148	0.9005	0.7597
	VAE with K-means and Decay	0.6999	0.6327	0.6646	0.887	0.914	0.9003	0.7604
	GAN	0.7018	0.6403	**0.6696**	0.889	0.9137	**0.9012**	0.7649
Breast Cancer[2]	No technique	1.0	0.9063	0.9508	0.9474	1.0	0.973	0.9520
	SMOTE	1.0	0.9688	0.9841	0.9818	1.0	0.9908	0.9843
	ADASYN	1.0	0.9688	0.9841	0.9818	1.0	0.9908	0.9843
	GMM	1.0	0.9375	0.9677	0.9643	1.0	0.9818	0.9682
	VAE with Decay	1.0	0.9375	0.9677	0.9643	1.0	0.9818	0.9682
	VAE with K-means	1.0	0.9375	0.9677	0.9643	1.0	0.9818	0.9682
	VAE with K-means and Decay	1.0	1.0	**1.0**	1.0	1.0	**1.0**	**1.0**
	GAN	1.0	0.9688	0.9841	0.9843	1.0	0.9908	0.9843
Credit Card Fraud[2]	No technique	0.8823	0.6818	0.7692	0.9995	0.9998	0.9996	0.8257
	SMOTE	0.8947	0.7727	0.8293	0.9996	0.9998	0.9997	**0.879**
	ADASYN	0.9412	0.7273	0.8205	0.9995	0.9999	0.9997	0.8528
	GMM	0.9412	0.7273	0.8205	0.9995	0.9999	0.9997	0.8528
	VAE with Decay	0.85	0.7727	0.8095	0.9996	0.9998	0.9997	0.8789
	VAE with K-means	0.9444	0.7727	**0.85**	0.9996	0.9999	**0.9998**	**0.879**
	VAE with K-means and Decay	0.8824	0.6818	0.7692	0.9995	0.9998	0.9996	0.8257
	GAN	0.8824	0.6818	0.7692	0.9995	0.9998	0.9996	0.8257
Cerebral Stroke[2]	No technique	0.2	0.0085	0.0163	0.982	0.9994	0.9906	0.092
	SMOTE	0.0486	0.2373	**0.0807**	0.9848	0.9143	0.9482	**0.4658**
	ADASYN	0.0531	0.1441	0.0776	0.9837	0.9526	0.9679	0.3705
	GMM	0.0333	0.0085	0.0135	0.9819	0.9955	0.9887	0.0918
	VAE with Decay	0.1014	0.0593	0.0749	0.9828	0.9903	0.9865	0.2424
	VAE with K-means	0.0986	0.0593	0.0741	0.9828	0.99	0.9864	0.2423
	VAE with K-means and Decay	0.125	0.0085	0.0159	0.982	0.9989	0.9904	0.092
	GAN	0.2	0.0085	0.0163	0.982	0.9994	**0.9906**	0.092

[1] The DA techniques added 25% more synthetic samples to the dataset.

[2] The DA techniques added only minority samples to the dataset.

the Cerebral Stroke Dataset shifted the best DA technique from the SMOTE to the ADASYN, while SMOTE still captured most of the minority instances.

Fundamentally, the decision tree classifier results also demonstrated the capability of augmentated data to increase the performance when classifying data and generating quality data. Moreover, it also showed the pattern of SMOTE having an increased performance with datasets with more categorical features, while the DL methods had a great performance on datasets whose features were mainly continuous.

Table 6: Decision Tree performance throughout the multiple Data Augmentation techniques

Datasets	DA Technique	Minority Class			Majority Class			G-Mean
		Precision	Recall	F1 Score	Precision	Recall	F1 Score	
Adult[1]	No Technique	0.6492	0.6122	0.6302	0.8792	0.8951	0.8871	0.7403
	SMOTE	0.6402	0.6233	0.6316	0.8816	0.8889	0.8852	**0.7444**
	ADASYN	0.6238	0.6063	0.6149	0.8763	0.8841	0.8802	0.7321
	GMM	0.6520	0.6054	0.6279	0.8777	0.8975	0.8875	0.7372
	VAE with Decay	0.6571	0.6012	0.6279	0.8769	0.9005	0.8885	0.7358
	VAE with K-means	0.6553	0.6063	0.6299	0.8781	0.8989	0.8884	0.7382
	VAE with K-means and Decay	0.6551	0.6122	0.6330	0.8796	0.8978	0.8886	0.7414
	GAN	0.6585	0.6148	**0.6359**	0.8804	0.8989	**0.8895**	0.7434
Breast[2] Cancer	No Technique	0.8750	0.8750	0.8750	0.9259	0.9259	0.9259	0.9001
	SMOTE	0.8788	0.9062	0.8923	0.9434	0.9259	0.9346	0.9160
	ADASYN	0.9375	0.9375	0.9375	0.9630	0.9630	0.9630	0.9501
	GMM	0.9000	0.8438	0.8710	0.9107	0.9444	0.9273	0.8927
	VAE with Decay	0.9062	0.9062	0.9062	0.9444	0.9444	0.9444	0.9252
	VAE with K-means	0.8611	0.9688	0.9118	0.9800	0.9074	0.9423	0.9376
	VAE with K-means and Decay	0.8235	0.8750	0.8485	0.9231	0.8889	0.9057	0.8819
	GAN	0.9677	0.9375	**0.9524**	0.9636	0.9815	**0.9725**	**0.9592**
Credit Card [2] Fraud	No Technique	0.6400	0.7273	0.6809	0.9995	0.9993	0.9994	0.8525
	SMOTE	0.4000	0.7273	0.5161	0.9995	0.9981	0.9988	0.8520
	ADASYN	0.6364	0.6364	0.6364	0.9994	0.9994	0.9994	0.7975
	GMM	0.4848	0.7273	0.5818	0.9995	0.9987	0.9991	0.8522
	VAE with Decay	0.7083	0.7727	**0.7391**	0.9996	0.9995	**0.9995**	**0.8788**
	VAE with K-means	0.6842	0.5909	0.6341	0.9993	0.9995	0.9994	0.7685
	VAE with K-means and Decay	0.6400	0.7273	0.6809	0.9995	0.9993	0.9994	0.8525
	GAN	0.6400	0.7273	0.6809	0.9995	0.9993	0.9994	0.8525
Cerebral[2] Stroke	No Technique	0.0321	0.0424	0.0365	0.9822	0.9764	0.9793	0.2034
	SMOTE	0.0396	0.2627	0.0689	0.9848	0.8825	0.9309	**0.4815**
	ADASYN	0.0449	0.1864	**0.0724**	0.9841	0.9268	0.9546	0.4157
	GMM	0.0449	0.0593	0.0511	0.9825	0.9767	**0.9796**	0.2407
	VAE with Decay	0.0440	0.1017	0.0614	0.9830	0.9592	0.9709	0.3123
	VAE with K-means	0.0484	0.1017	0.0656	0.9831	0.9631	0.9730	0.3130
	VAE with K-means and Decay	0.0533	0.0763	0.0627	0.9828	0.9750	0.9789	0.2727
	GAN	0.0321	0.0424	0.0365	0.9822	0.9764	0.9793	0.2034

[1] The DA techniques added 25% more synthetic samples to the dataset.

[2] The DA techniques added only minority samples to the dataset.

As for the MLP classifier, the results were more inconsistent, as can be seen in Table 7, mainly in the Cerebral Stroke dataset. This classifier was more susceptible to overfitting on datasets whose features were continuous, since the model focused on only classifying accurately the majority class. In the Adult dataset SMOTE demonstrated to be the method more capable when generating data. Furthermore, Breast Cancer had very good results when classifying both classes, even without augmented data. As a result, the techniques that could not maintain 100% accuracy can be labeled as producing poor data for this classifier and only this classifier, since the better DA method in the Table 5 was VAE with K-means and decay. Thirdly, the Credit Card Fraud dataset maintained the best DA method, VAE with K-means, followed by the other variations of VAE and GAN. Lastly, the Cerebral Stroke dataset DA technique with the best performance was SMOTE, with an increasing performance of 9.74% f1-score and a recall of 54.24%. Note that the classifier MLP for this dataset overfitted drastically towards the majority class, since the performance on the minority class was 0%.

As it can be seen from the previous tables, the main reasons for focusing mainly on the classifier Random Forest were the slightly better performances when compared with the Decision Tree and the overfit seen by the MLP.

Table 7: MLP performance throughout the multiple Data Augmentation techniques

		Minority Class			**Majority Class**			
Datasets	**DA Technique**	**Precision**	**Recall**	**F1 Score**	**Precision**	**Recall**	**F1 Score**	**G-Mean**
Adult[1]	No Technique	0.7120	0.6012	0.6519	0.8795	0.9229	0.9007	0.7449
	SMOTE	0.6532	0.6726	**0.6628**	0.8952	0.8868	0.8910	**0.7723**
	ADASYN	0.6741	0.6207	0.6463	0.8827	0.9048	0.8936	0.7494
	GMM	0.6895	0.5400	0.6056	0.8635	0.9229	0.8922	0.7059
	VAE with Decay	0.6758	0.6114	0.6420	0.8804	0.9070	0.8935	0.7447
	VAE with K-means	0.6802	0.5969	0.6359	0.8770	0.9110	0.8937	0.7374
	VAE with K-means and Decay	0.6785	0.5867	0.6293	0.8744	0.9118	0.8927	0.7314
	GAN	0.7133	0.5944	0.6484	0.8778	0.9242	**0.9004**	0.7412
Breast Cancer[2]	No Technique	1.000	1.0000	1.0000	1.000	1.0000	1.0000	1.0000
	SMOTE	0.9697	1.0000	0.9846	1.000	0.9815	0.9907	0.9907
	ADASYN	0.9143	1.0000	0.9552	1.000	0.9444	0.9714	0.9718
	GMM	1.000	1.0000	**1.0000**	1.000	1.0000	**1.0000**	**1.0000**
	VAE with Decay	0.9412	1.0000	0.9697	1.000	0.9630	0.9811	0.9813
	VAE with K-means	0.9375	0.9375	0.9375	0.963	0.9630	0.9630	0.9501
	VAE with K-means and Decay	0.9697	1.0000	0.9846	1.000	0.9815	0.9907	0.9907
	GAN	1.000	1.0000	**1.0000**	1.000	1.0000	**1.0000**	**1.0000**
Credit Card Fraud[2]	No Technique	0.8947	0.7727	0.8293	0.9996	0.9998	0.9997	0.8790
	SMOTE	0.6000	0.8182	0.6923	0.9997	0.9991	0.9994	**0.9041**
	ADASYN	0.6364	0.6364	0.6364	0.9994	0.9994	0.9994	0.7975
	GMM	0.0031	0.9545	0.0062	0.9998	0.4773	0.6461	0.6750
	VAE with Decay	0.8947	0.7727	0.8293	0.9996	0.9998	0.9997	0.8790
	VAE with K-means	0.9444	0.7727	**0.8500**	0.9996	0.9999	**0.9998**	0.8790
	VAE with K-means and Decay	0.8947	0.7727	0.8293	0.9996	0.9998	0.9997	0.8790
	GAN	0.8947	0.7727	0.8293	0.9996	0.9998	0.9997	0.8790
Cerebral Stroke[2]	No Technique	0.0000	0.0000	0.0000	0.9819	1.0000	0.9909	0.0000
	SMOTE	0.0535	0.5424	**0.0974**	0.9898	0.8229	0.8987	**0.6681**
	ADASYN	0.0484	0.4153	0.0867	0.9875	0.8494	0.9132	0.5939
	GMM	0.0000	0.0000	0.0000	0.9817	0.9876	0.9846	0.0000
	VAE with Decay	0.0482	0.0339	0.0398	0.9823	0.9876	0.9849	0.1830
	VAE with K-means	0.0645	0.0169	0.0268	0.9821	0.9955	0.9887	0.1299
	VAE with K-means and Decay	0.0000	0.0000	0.0000	0.9818	0.9970	0.9894	0.0000
	GAN	0.0000	0.0000	0.0000	0.9819	1.0000	**0.9909**	0.0000

[1] The DA techniques added 25% more synthetic samples to the dataset.

[2] The DA techniques added only minority samples to the dataset.

Chapter 5

Conclusion

This chapter will be centered around a summary of the results obtained during this study experiments, the future work and some final thoughts.

5.1 Experiments

In this study, we went through a benchmark of different DA techniques in multiple datasets of various domains. With the results obtained during the experiments, we can now answer the main research questions previously stated in this dissertation.

First, we found that DA could improve the results of imbalanced data problems, generating samples of the minority class or both classes (RQ1). This was observed in all datasets with multiple DA techniques.

Furthermore, with the results found during the development of this study, it was also seen that some dataset properties could influence the quality of the synthetic data generated (RQ2). Datasets that contained more categorical features were usually associated with an increased difficulty in increasing the classifiers' performance or even a decrease in the minority class classification when adding synthetic data to the original dataset. This phenomenon is visible in problems such as the Adult and Cerebral Stroke datasets, in which DA techniques such as the VAE variations and GANs had difficulties when generating synthetic data. On the other hand, other techniques (SMOTE, ADASYN, and, in some cases, GMM) generated quality synthetic samples easily. However, datasets whose features were mainly continuous had very good performances for techniques such as VAE and GAN, while maintaining good performances for SMOTE and ADASYN (even if lower than the other more complex techniques)

Relatively to the number of samples to be generated, the dataset properties also impact how to choose that parameter (RQ3). As described before, more categorical datasets achieved a greater performance when adding 25% more samples to the original dataset, while datasets whose major features are continuous should opt to add only minority samples.

Finally, these experiments helped us to determine which DA techniques provide a better quality of synthetic tabular data (RQ4). The implementation of the VAE with K-means had very good results, mainly in the minority class, as expected. Therefore, it could be chosen as a good candidate to be used as a DA technique in a dataset whose features are mostly categorical. In essence, GAN and VAE can achieve amazing results when explored and tuned for specific domains, which is one of the principal characteristics of these kinds of algorithms. Moreover, the results demonstrated that, in general, SMOTE had very good performance when generating synthetic data. As a result, SMOTE seems to be the ideal technique when the dataset properties are not taken into account, since it is very consistent in its generated samples. The GMM, the more unusual technique, showed some promise for further investigation, despite appearing to have a higher volatility in sample quality.

Although this study helped us to use DA with a lot more knowledge of how to do it and which techniques to choose, there were some obstacles during its development. The main complication was the strange overfit seen with synthetic data from techniques such as VAE and GAN. The VAE with K-means could suffer a loss in its process since the unsupervised algorithm could choose incorrectly the minority case, focusing the VAE on generating the majority class. As for the GAN, some results suggest that the noise added to the discriminator network was too high. Therefore, for that specific problem, adjusting that value could surpass the quality of data generated. Moreover, the results also suggest that the ML classifiers are susceptible to a greater or lower type of overfit depending on the quality of the data.

In regards to the development of the multiple DA techniques, the VAE and GAN proved to be somewhat challenging due to the few examples of code in a tabular data problem, especially with TensorFlow. Furthermore, the implementation of the loss function for both DA methods took the most time because it presented a number of challenges, which the use of a decay and the K-means (on the VAE) aid in resolving, as described in Section 3.2.

This dissertation's work provided two publications. The first was accepted for publication by the conference IDEAL2022 with the title of "Benchmarking Data Augmentations Techniques for Tabular Data" I.1. In this paper, we focused only on the first results of the techniques SMOTE, GMM, and a simpler version of the VAE. Following the work of the first paper, we submitted for publication an extension of those ideas to the journal Machine Learning by the title "Data Augmentation Methods for Tabular Data" I.2. In this extension work, we increased the number of datasets analyzed as well as the DA techniques, adding ADASYN, VAE with decay, VAE with K-means, VAE with K-means and decay, and GAN.

5.2 Future Work

The realization of this study answered multiple questions, which can be very useful when using DA techniques to generate synthetic data on a tabular data problem. Furthermore, this study could be followed by another one by using more datasets and other DA methods in order to gain more knowledge of how each algorithm performs under different dataset properties. Moreover, other variations of VAE and GAN could be developed since the field of generative algorithms has much interest and research behind it in the scientific community. The benchmark done could also be improved by increasing the number ML classifiers, which would result in increased computational cost and time, which were very limited during this study.

Moreover, the implementation of more and newer statistical tests, as well as other analyses, could end up providing better ways to compare real and synthetic data.

5.3 Final Thoughts

The ML field is gaining more and more attention nowadays, due to the availability of data. Therefore, the possibility of a future where synthetic data plays a huge role in decision-making is getting progressively closer.

This thesis can be viewed as one more step in that direction, since it explores multiple DA techniques and obtains very interesting results in regards to generating synthetic data in datasets with different properties and how many samples to generate.

To finish, we expect that this thesis can prove to be useful to future works on the DA field.

Bibliography

[1] H. He and E. A. Garcia. “Learning from Imbalanced Data”. In: *IEEE Transactions on Knowledge and Data Engineering* 21.9 (2009), pp. 1263–1284. doi: 10.1109/TKDE.2008.239 (cit. on pp. 1, 7, 8, 20, 21).

[2] B. Fernandes et al. “An Adjective Selection Personality Assessment Method Using Gradient Boosting Machine Learning”. In: *Processes* 8.5 (2020). issn: 2227-9717. doi: 10.3390/pr8050618. url: https://www.mdpi.com/2227-9717/8/5/618 (cit. on p. 1).

[3] B. Fernandes et al. “Traffic Flow Forecasting on Data-Scarce Environments Using ARIMA and LSTM Networks”. In: *New Knowledge in Information Systems and Technologies*. Ed. by Á. Rocha et al. Cham: Springer International Publishing, 2019, pp. 273–282 (cit. on p. 1).

[4] P. Chan et al. “Distributed data mining in credit card fraud detection”. In: *IEEE Intelligent Systems and their Applications* 14.6 (1999), pp. 67–74. doi: 10.1109/5254.809570 (cit. on pp. 1, 8).

[5] F. Perez et al. “Data Augmentation for Skin Lesion Analysis”. In: *OR 2.0 Context-Aware Operating Theaters, Computer Assisted Robotic Endoscopy, Clinical Image-Based Procedures, and Skin Image Analysis*. Ed. by D. Stoyanov et al. Cham: Springer International Publishing, 2018, pp. 303–311 (cit. on pp. 1, 9).

[6] J.-J. Lv et al. “Data augmentation for face recognition”. In: *Neurocomputing* 230 (2017), pp. 184–196. issn: 0925-2312. doi: https://doi.org/10.1016/j.neucom.2016.12.025 (cit. on pp. 1, 9).

[7] J. Nalepa, M. Marcinkiewicz, and M. Kawulok. “Data Augmentation for Brain-Tumor Segmentation: A Review”. In: *Frontiers in Computational Neuroscience* 13 (2019), p. 83. issn: 1662-5188. doi: 10.3389/fncom.2019.00083. url: https://www.frontiersin.org/article/10.3389/fncom.2019.00083 (cit. on pp. 1, 9).

[8] X. Zhu et al. “Emotion Classification with Data Augmentation Using Generative Adversarial Networks”. In: *Advances in Knowledge Discovery and Data Mining*. Ed. by D. Phung et al. Cham: Springer International Publishing, 2018, pp. 349–360 (cit. on pp. 1, 2, 9).

[9] Z. Islam et al. “Crash data augmentation using variational autoencoder”. In: *Accident Analysis & Prevention* 151 (2021), p. 105950. issn: 0001-4575. doi: https://doi.org/10.1016/j.aap.2020.105950 (cit. on pp. 1, 9, 14, 15, 18).

[10] N. V. Chawla et al. "SMOTE: Synthetic Minority Over-sampling Technique". In: *Journal of Artificial Intelligence Research* 16 (2002), pp. 321–357 (cit. on pp. 1, 7, 9, 10).

[11] H. He et al. "ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning". In: *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*. 2008, pp. 1322–1328. doi: 10.1109/IJCNN.2008.4633969 (cit. on pp. 1, 10, 18).

[12] T. Sarkar. *How to use a clustering technique for synthetic data generation*. Last Visited November 14, 2021. Sept. 2019. url: https://towardsdatascience.com/how-to-use-a-clustering-technique-for-synthetic-data-generation-7c84b6b678ea (cit. on pp. 1, 13, 14).

[13] E. Choi et al. *Generating Multi-label Discrete Patient Records using Generative Adversarial Networks*. 2018. arXiv: 1703.06490 [cs.LG] (cit. on p. 1).

[14] C. Phua, D. Alahakoon, and V. Lee. "Minority Report in Fraud Detection: Classification of Skewed Data". In: *SIGKDD Explor. Newsl.* 6.1 (June 2004), pp. 50–59. issn: 1931-0145. doi: 10.1145/1007730.1007738 (cit. on pp. 2, 8).

[15] A. Arora et al. "Data Augmentation Using Gaussian Mixture Model on CSV Files". In: Jan. 2021, pp. 258–265. isbn: 978-3-030-53035-8. doi: 10.1007/978-3-030-53036-5_28 (cit. on pp. 3, 19).

[16] G. Edwards. *Machine learning: An introduction*. Last Visited November 15, 2021. Jan. 2020. url: https://towardsdatascience.com/machine-learning-an-introduction-23b84d51e6d0 (cit. on p. 5).

[17] B. Fernandes, P. Novais, and C. Analide. "A Multi-Agent System for Automated Machine Learning". In: *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*. 2022, pp. 1899–1901 (cit. on p. 5).

[18] S. Shalev-Shwartz and S. Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014. doi: 10.1017/CBO9781107298019 (cit. on p. 5).

[19] T. M. Mitchell. "Machine Learning and Data Mining". In: *Communications of the ACM* 42 (1999), pp. 30–36 (cit. on p. 5).

[20] S. Angra and S. Ahuja. "Machine learning and its applications: A review". In: *2017 International Conference on Big Data Analytics and Computational Intelligence (ICBDAC)*. 2017, pp. 57–60. doi: 10.1109/ICBDACI.2017.8070809 (cit. on p. 5).

[21] J. M. Górriz et al. "Artificial intelligence within the interplay between natural and artificial computation: Advances in data science, trends and applications". In: *Neurocomputing* 410 (2020), pp. 237–270. issn: 0925-2312. doi: https://doi.org/10.1016/j.neucom.2020.05.078. url: https://www.sciencedirect.com/science/article/pii/S0925231220309292 (cit. on p. 5).

[22] P. Oliveira et al. "Evaluating Unidimensional Convolutional Neural Networks to Forecast the Influent pH of Wastewater Treatment Plants". In: *Intelligent Data Engineering and Automated Learning – IDEAL 2021*. Ed. by H. Yin et al. Cham: Springer International Publishing, 2021, pp. 446–457 (cit. on p. 5).

[23] B. Fernandes, J. Neves, and C. Analide. "SafeCity: A Platform for Safer and Smarter Cities". In: *Advances in Practical Applications of Agents, Multi-Agent Systems, and Trustworthiness. The PAAMS Collection*. Ed. by Y. Demazeau et al. Cham: Springer International Publishing, 2020, pp. 412–416 (cit. on p. 5).

[24] B. Mahesh. "Machine Learning Algorithms-A Review". In: *International Journal of Science and Research (IJSR).[Internet]* 9 (2020), pp. 381–386 (cit. on pp. 5–7).

[25] M. Santana. *Deep learning: Do Conceito às aplicações*. Last Visited November 15, 2021. July 2018. url: https://medium.com/data-hackers/deep-learning-do-conceito-%5C%C3%5C%A0s-aplica%5C%C3%5C%A7%5C%C3%5C%B5es-e8e91a7c7eaf (cit. on p. 6).

[26] S. Brown. *Machine Learning, explained*. Last Visited November 15, 2021. Apr. 2021. url: https://mitsloan.mit.edu/ideas-made-to-matter/machine-learning-explained (cit. on pp. 5–7).

[27] A. Géron. "Hands-on machine learning with scikit-learn and tensorflow: Concepts". In: *Tools, and Techniques to build intelligent systems* (2017) (cit. on pp. 5–7).

[28] F.-Y. Wang et al. "Where does AlphaGo go: from church-turing thesis to AlphaGo thesis and beyond". In: *IEEE/CAA Journal of Automatica Sinica* 3.2 (2016), pp. 113–120. doi: 10.1109/JAS.2016.7471613 (cit. on p. 7).

[29] M. Kubat, R. Holte, and S. Matwin. "Machine learning for the detection of oil spills in satellite radar images". English (US). In: *Machine Learning* 30.2-3 (1998). Copyright: Copyright 2020 Elsevier B.V., All rights reserved., pp. 195–215. issn: 0885-6125. doi: 10.1023/a:1007452223027 (cit. on p. 7).

[30] M. Shao, N. Gu, and X. Zhang. "Credit Card Transactions Data Adversarial Augmentation in the Frequency Domain". In: *2020 5th IEEE International Conference on Big Data Analytics (ICBDA)*. 2020, pp. 238–245. doi: 10.1109/ICBDA49040.2020.9101344 (cit. on pp. 7, 9, 19).

[31] R. B. Rao, S. Krishnan, and R. S. Niculescu. “Data Mining for Improved Cardiac Care”. In: *SIGKDD Explor. Newsl.* 8.1 (June 2006), pp. 3–10. issn: 1931-0145. doi: 10.1145/1147234.1147236 (cit. on p. 8).

[32] P. K.-F. Chan and S. Stolfo. “Toward Scalable Learning with Non-Uniform Class and Cost Distributions: A Case Study in Credit Card Fraud Detection”. In: *KDD*. 1998 (cit. on p. 8).

[33] D. A. van Dyk and X.-L. Meng. “The Art of Data Augmentation”. In: *Journal of Computational and Graphical Statistics* 10 (2001), pp. 1–50 (cit. on p. 8).

[34] N. V. Chawla et al. “SMOTEBoost: improving prediction of the minority class in boosting”. In: *In Proceedings of the Principles of Knowledge Discovery in Databases, PKDD-2003*. 2003, pp. 107–119 (cit. on p. 10).

[35] H. Guo and H. L. Viktor. “Learning from Imbalanced Data Sets with Boosting and Data Generation: The DataBoost-IM Approach”. In: *SIGKDD Explor. Newsl.* 6 (2004), pp. 30–39. doi: 10.1145/1007730.1007736 (cit. on p. 10).

[36] R. Xu and D. C. Wunsch. *Clustering*. Vol. 10. John Wiley & Sons, 2008 (cit. on p. 11).

[37] J. Brownlee. *A gentle introduction to expectation-maximization (EM algorithm)*. Last Visited November 15, 2021. Aug. 2020. url: https://machinelearningmastery.com/expectation-maximization-em-algorithm/ (cit. on p. 13).

[38] G. J. McLachlan and T. Krishnan. *The EM algorithm and extensions*. Vol. 382. John Wiley & Sons, 2007 (cit. on p. 13).

[39] J. Rocca. *Understanding variational autoencoders (VAES)*. Last Visited November 23, 2021. Mar. 2021. url: https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73 (cit. on pp. 14–16).

[40] W. Badr. *Auto-encoder: What is it? and what is it used for? (part 1)*. Last Visited November 24, 2021. July 2019. url: https://towardsdatascience.com/auto-encoder-what-is-it-and-what-is-it-used-for-part-1-3e5c6f017726 (cit. on p. 15).

[41] I. J. Goodfellow et al. *Generative Adversarial Networks*. 2014. arXiv: 1406.2661 [stat.ML] (cit. on pp. 16, 17).

[42] J. Brownlee. *A gentle introduction to generative adversarial networks (GANs)*. July 2019. url: https://machinelearningmastery.com/what-are-generative-adversarial-networks-gans/ (cit. on pp. 16, 17).

[43] C. Elkan. “The Foundations of Cost-Sensitive Learning”. In: *Proceedings of the Seventeenth International Conference on Artificial Intelligence: 4-10 August 2001; Seattle* 1 (May 2001) (cit. on p. 20).

[44] Y. Tang et al. “SVMs Modeling for Highly Imbalanced Classification”. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 39.1 (2009), pp. 281–288. doi: 10.1109/TSMCB.2008.2002909 (cit. on p. 21).

[45] R. Wirth and J. Hipp. “CRISP-DM: Towards a standard process model for data mining”. In: *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining*. Vol. 1. Manchester. 2000, pp. 29–39 (cit. on p. 22).

[46] T. pandas development team. *pandas-dev/pandas: Pandas*. Version latest. Feb. 2020. doi: 10.5281/zenodo.3509134. url: https://doi.org/10.5281/zenodo.3509134 (cit. on p. 23).

[47] C. R. Harris et al. “Array programming with NumPy”. In: *Nature* 585.7825 (Sept. 2020), pp. 357–362. doi: 10.1038/s41586-020-2649-2. url: https://doi.org/10.1038/s41586-020-2649-2 (cit. on p. 23).

[48] F. Pedregosa et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830 (cit. on p. 23).

[49] F. Chollet et al. *Keras*. https://keras.io. 2015 (cit. on p. 23).

[50] Martín Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015. url: https://www.tensorflow.org/ (cit. on p. 23).

[51] S. Lee et al. “Genetic Algorithm Based Deep Learning Neural Network Structure and Hyperparameter Optimization”. In: *Applied Sciences* 11.2 (2021). issn: 2076-3417. url: https://www.mdpi.com/2076-3417/11/2/744 (cit. on p. 24).

[52] J. Lucas et al. “Understanding Posterior Collapse in Generative Latent Variable Models”. In: *DGS*. 2019 (cit. on p. 24).

[53] W. Li et al. “Tackling mode collapse in multi-generator GANs with orthogonal vectors”. In: *Pattern Recognition* 110 (2021), p. 107646. issn: 0031-3203. doi: 10.1016/j.patcog.2020.107646 (cit. on p. 24).

[54] D. P. Kingma and M. Welling. “An Introduction to Variational Autoencoders”. In: *Foundations and Trends® in Machine Learning* 12.4 (2019), pp. 307–392. doi: 10.1561/2200000056 (cit. on pp. 24, 25).

[55] A. Singh and T. Ogunfunmi. “An overview of variational autoencoders for source separation, finance, and Bio-Signal Applications”. In: *Entropy* 24.1 (Dec. 2021), p. 55. doi: 10.3390/e24010055 (cit. on pp. 24, 25).

[56] R. Kohavi and B. Becker. *UCI Machine Learning Repository*. 1994. url: https://archive.ics.uci.edu/ml/datasets/adult (cit. on p. 26).

[57] W. H. Wolberg, W. N. Street, and O. L. Mangasarian. *UCI Machine Learning Repository*. 1998. url: https://archive.ics.uci.edu/ml/datasets/%20Breast+Cancer+Wisconsin+%5C%28Diagnostic%5C%29 (cit. on p. 26).

[58] M. L. G. ULB. *Credit Card Fraud Detection*. Mar. 2018. url: https://www.kaggle.com/mlg-ulb/creditcardfraud (cit. on p. 26).

[59] T. Liu, W. Fan, and C. Wu. *Data for: A hybrid machine learning approach to cerebral stroke prediction based on imbalanced medical-datasets*. Vol. 1. 2019. doi: https://doi:10.17632/x8ygrw87jw.1 (cit. on p. 26).

Appendix A

Detailed Statistical Tests

Table 8: Statistical Tests for each dataset

Dataset	DA Technique	Number of Continuous Features Rejected[1]	Number of Categorical Features Rejected[1]
Adult	SMOTE	1 out of 1	2 out of 7
	ADASYN	1 out of 1	0 out of 7
	GMM	1 out of 1	0 out of 7
	VAE with Decay	1 out of 1	1 out of 7
	VAE with K-means	1 out of 1	1 out of 7
	VAE with K-means and Decay	1 out of 1	0 out of 7
	GAN	1 out of 1	1 out of 7
Breast Cancer	SMOTE	17 out of 30	0 out of 0
	ADASYN	27 out of 30	0 out of 0
	GMM	30 out of 30	0 out of 0
	VAE with Decay	30 out of 30	0 out of 0
	VAE with K-means	29 out of 30	0 out of 0
	VAE with K-means and Decay	29 out of 30	0 out of 0
	GAN	30 out of 30	0 out of 0
Credit Card Fraud	SMOTE	30 out of 30	0 out of 0
	ADASYN	30 out of 30	0 out of 0
	GMM	30 out of 30	0 out of 0
	VAE with Decay	30 out of 30	0 out of 0
	VAE with K-means	30 out of 30	0 out of 0
	VAE with K-means and Decay	30 out of 30	0 out of 0
	GAN	30 out of 30	0 out of 0
Cerebral Stroke	SMOTE	3 out of 3	2 out of 7
	ADASYN	3 out of 3	0 out of 7
	GMM	3 out of 3	1 out of 7
	VAE with Decay	3 out of 3	0 out of 7
	VAE with K-means	3 out of 3	0 out of 7
	VAE with K-means and Decay	3 out of 3	0 out of 7
	GAN	3 out of 3	0 out of 7

[1] Rejection results based on a 0.05 level of significance.

Appendix B

Detailed Ratio of Generated Data Results

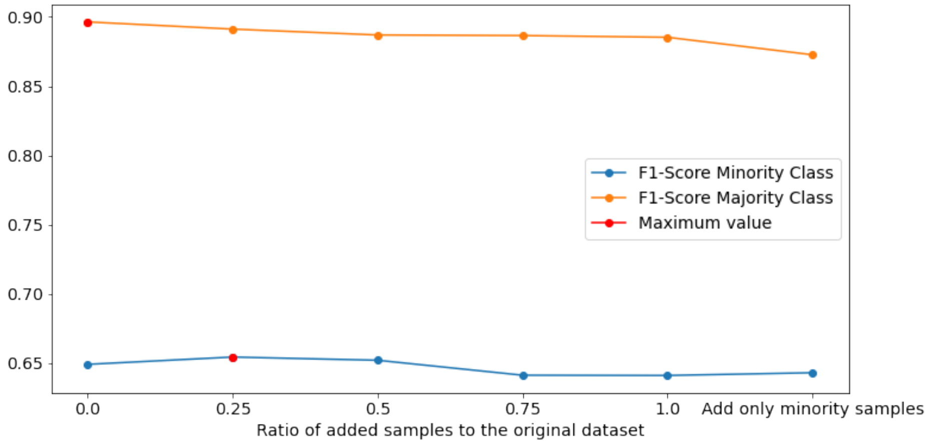


Figure 14: Ratio of data added at the Adult dataset by SMOTE

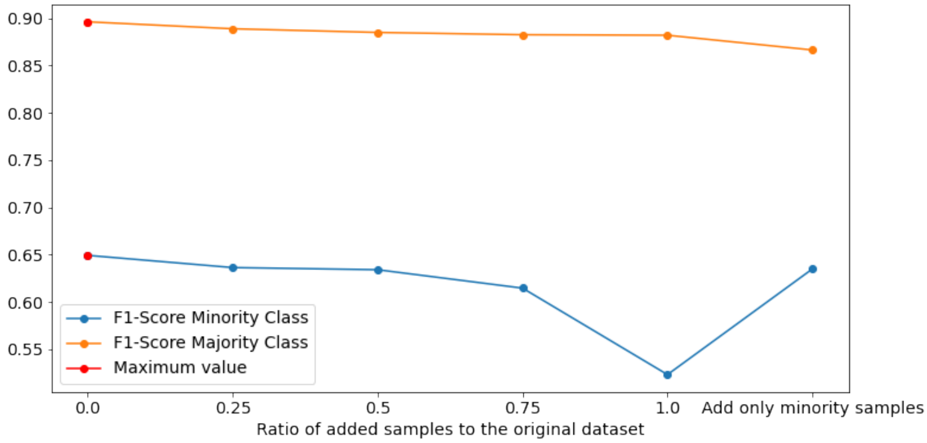


Figure 15: Ratio of data added at the Adult dataset by ADASYN

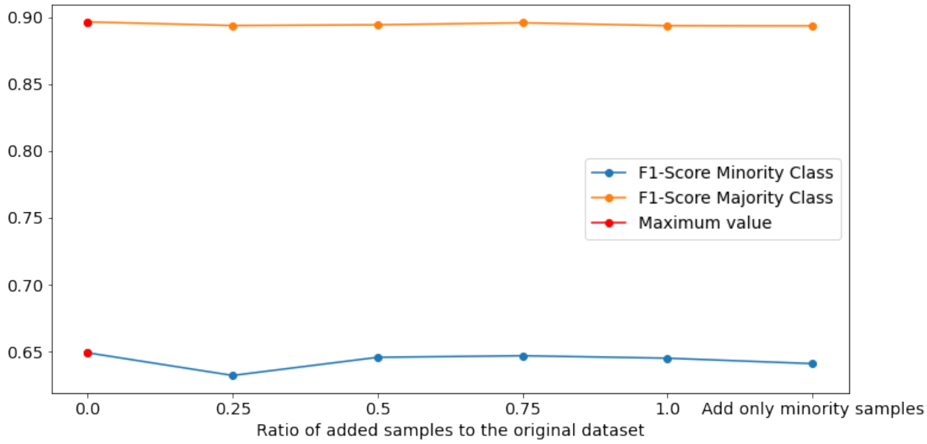


Figure 16: Ratio of data added at the Adult dataset by GMM

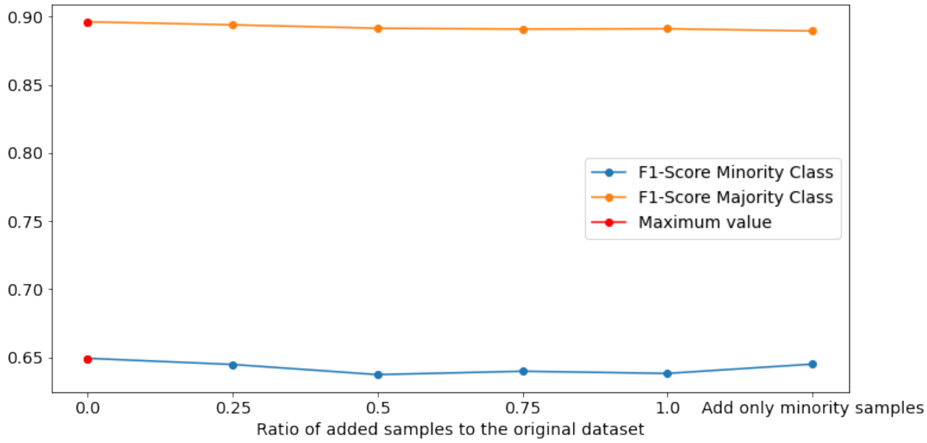


Figure 17: Ratio of data added at the Adult dataset by VAE with Decay

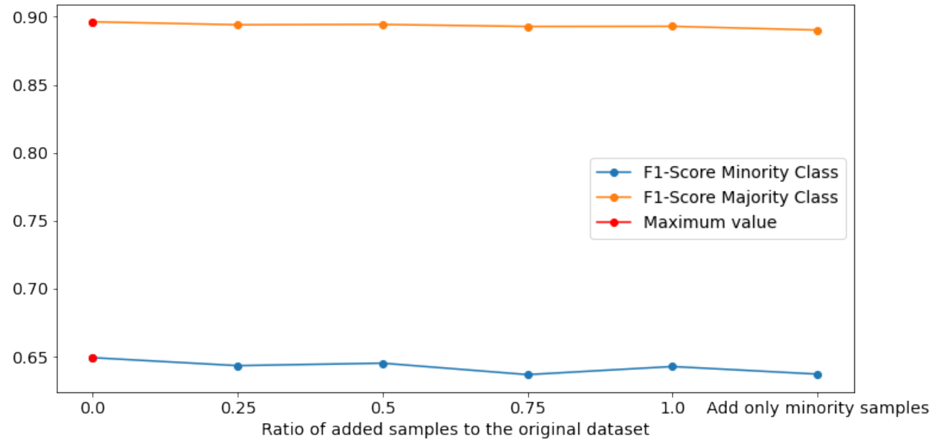


Figure 18: Ratio of data added at the Adult dataset by VAE with K-Means

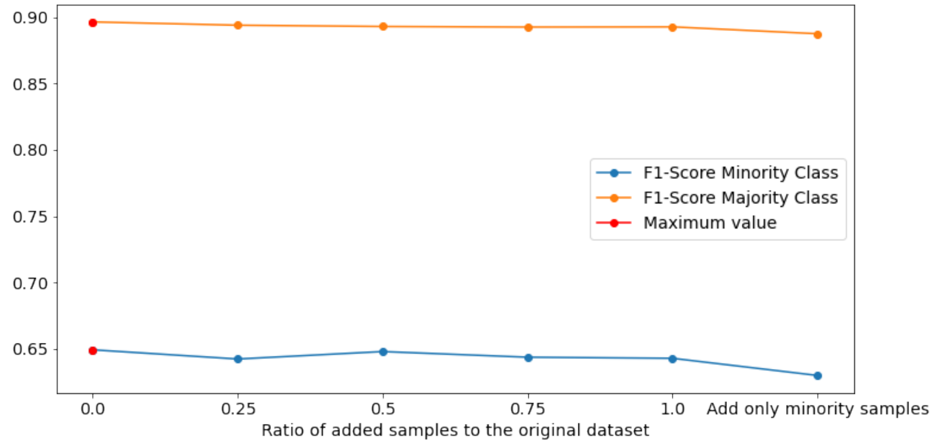


Figure 19: Ratio of data added at the Adult dataset by VAE with K-Means and Decay

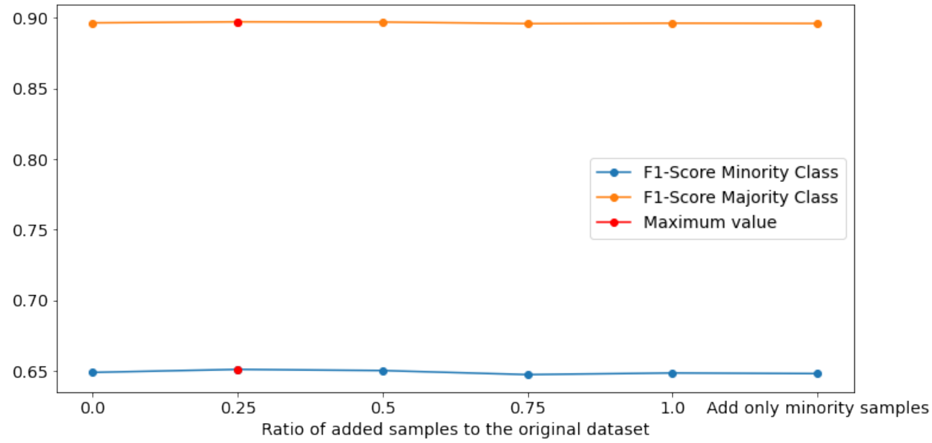


Figure 20: Ratio of data added at the Adult dataset by GAN

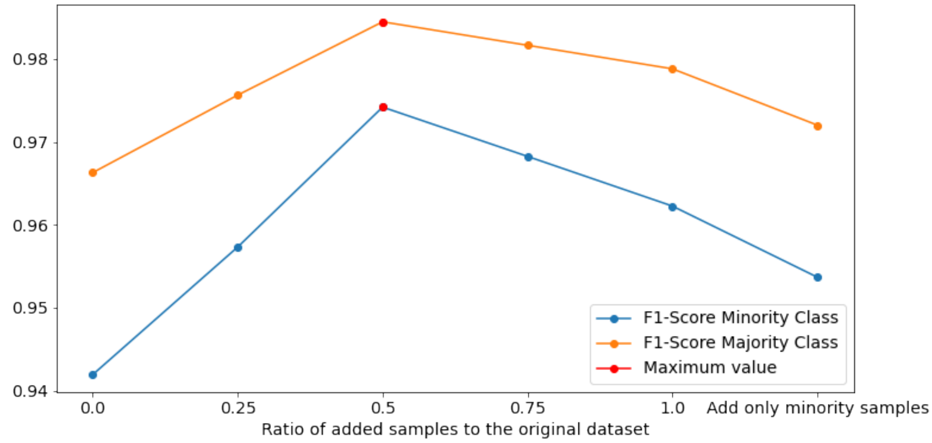


Figure 21: Ratio of data added at the Breast dataset by SMOTE

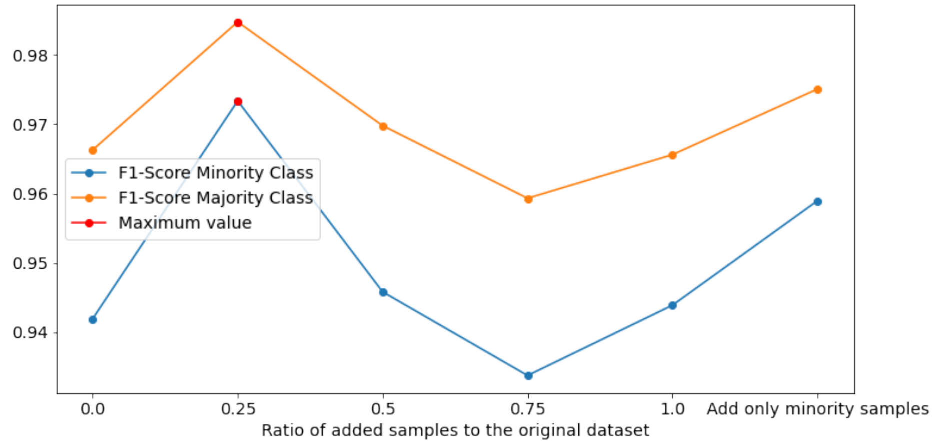


Figure 22: Ratio of data added at the Breast dataset by ADASYN

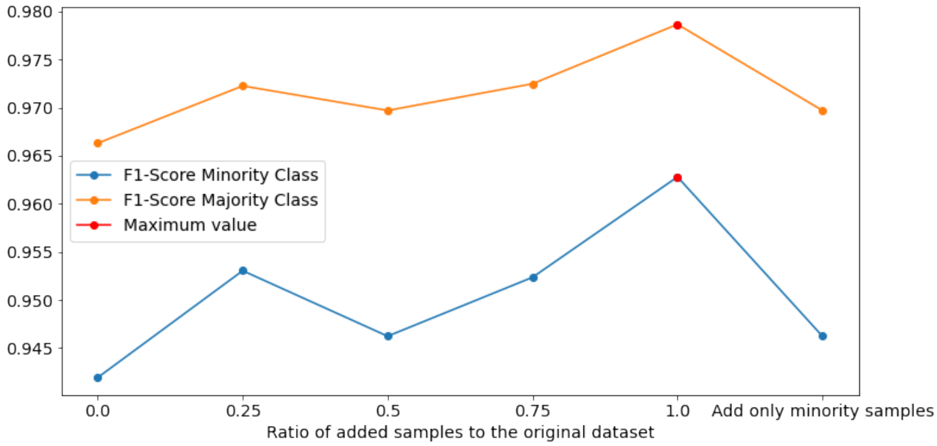


Figure 23: Ratio of data added at the Breast dataset by GMM

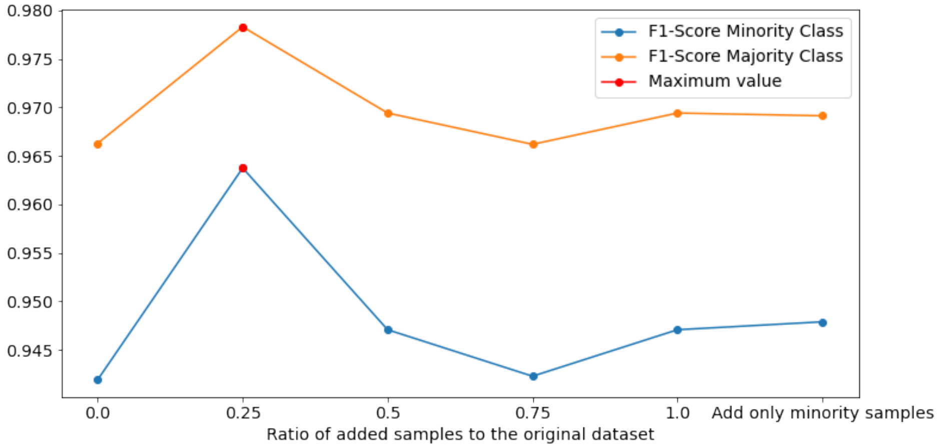


Figure 24: Ratio of data added at the Breast dataset by VAE with Decay

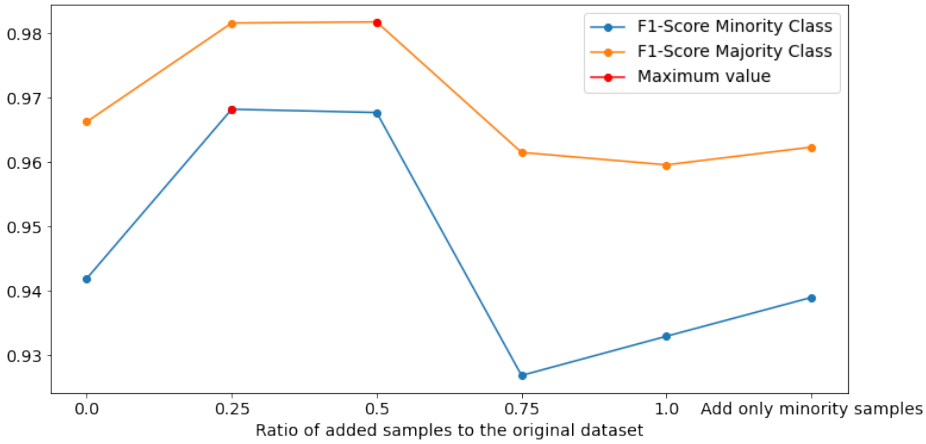


Figure 25: Ratio of data added at the Breast dataset by VAE with K-Means

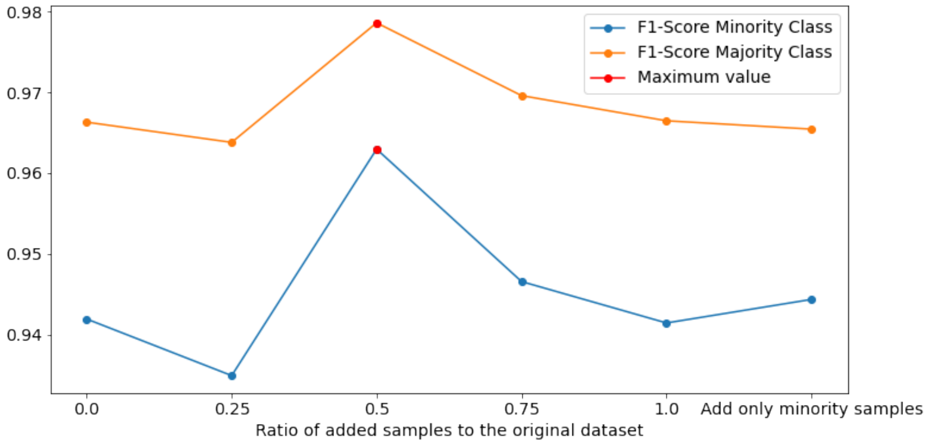


Figure 26: Ratio of data added at the Breast dataset by VAE with K-Means and Decay

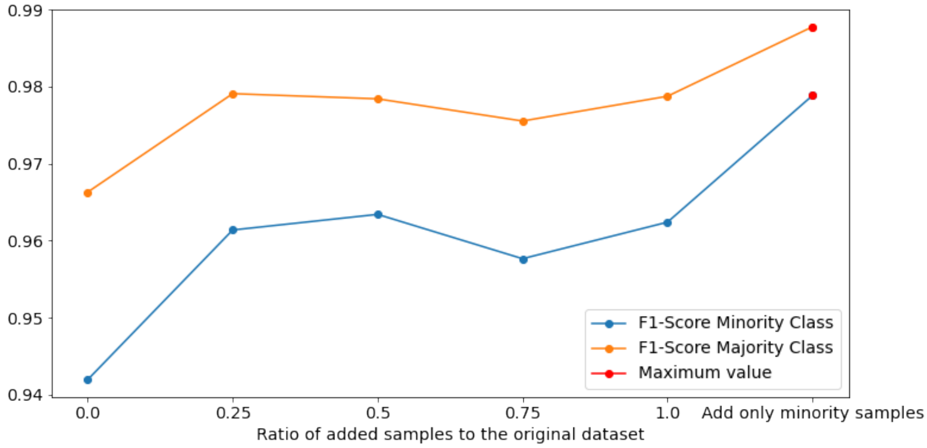


Figure 27: Ratio of data added at the Breast dataset by GAN

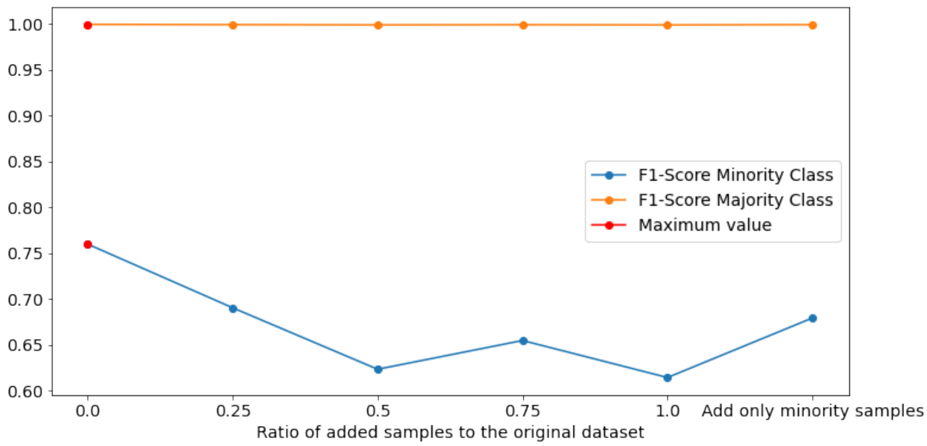


Figure 28: Ratio of data added at the Credit Card Fraud dataset by SMOTE

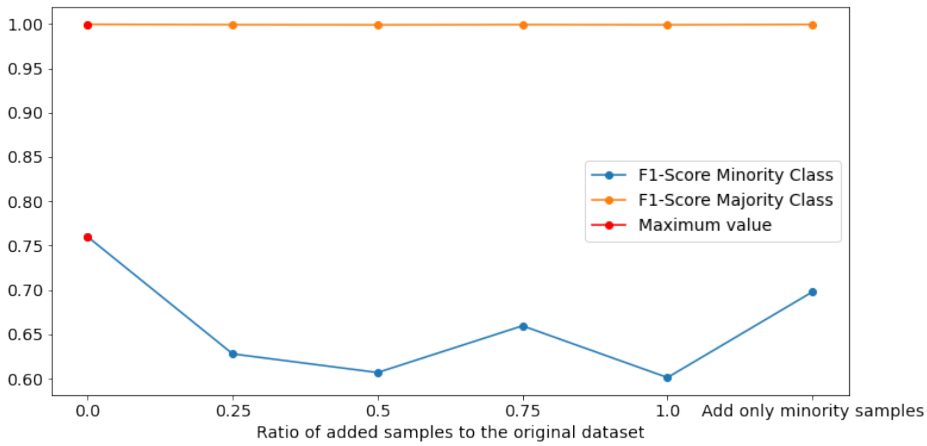


Figure 29: Ratio of data added at the Credit Card Fraud dataset by ADASYN

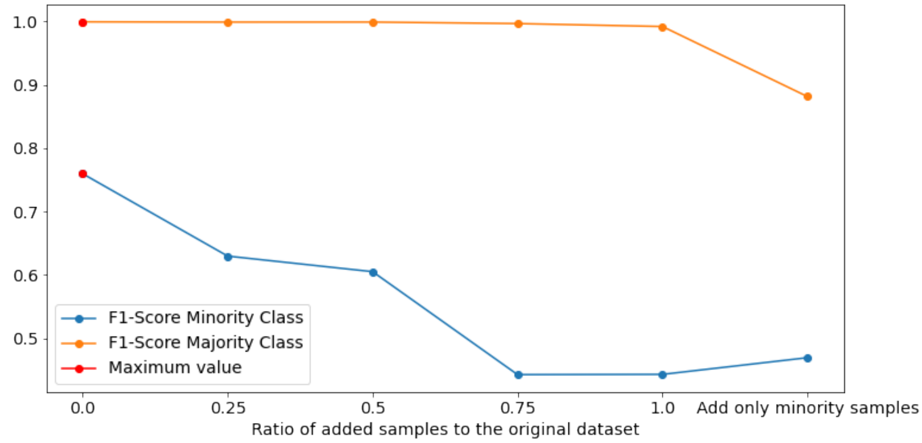


Figure 30: Ratio of data added at the Credit Card Fraud dataset by GMM

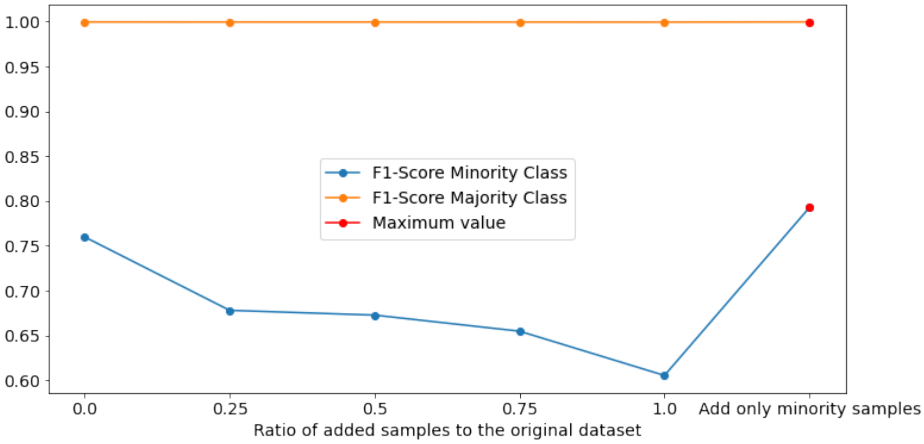


Figure 31: Ratio of data added at the Credit Card Fraud dataset by VAE with Decay

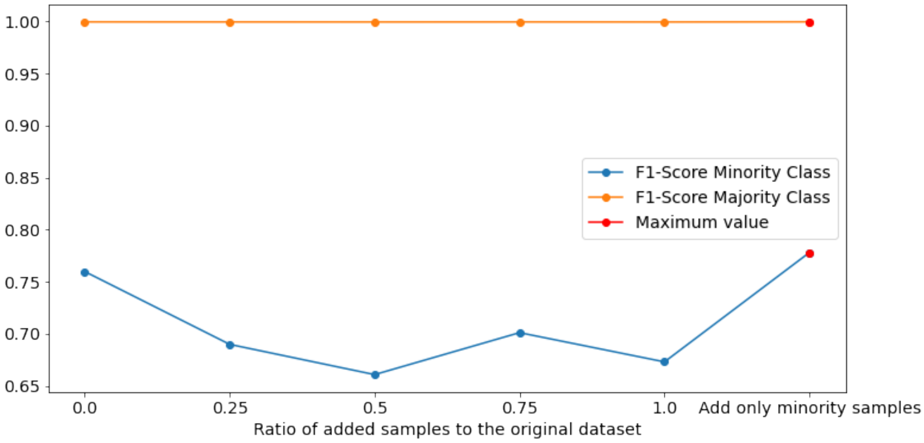


Figure 32: Ratio of data added at the Credit Card Fraud dataset by VAE with K-Means

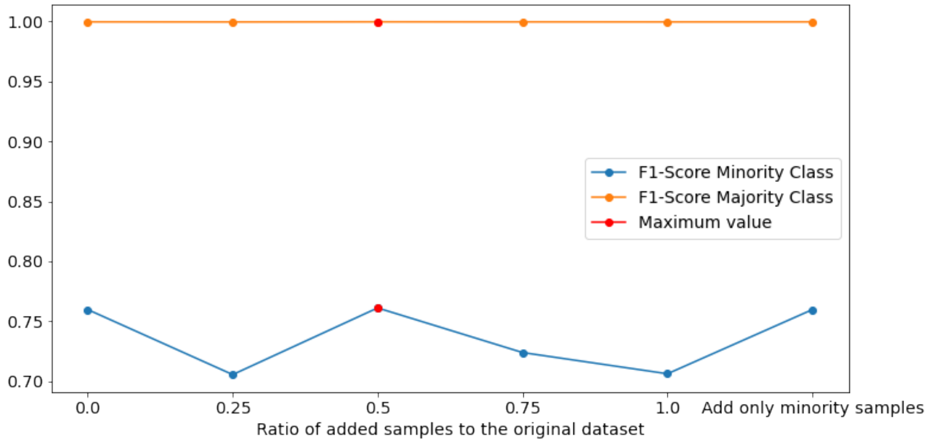


Figure 33: Ratio of data added at the Credit Card Fraud dataset by VAE with K-Means and Decay

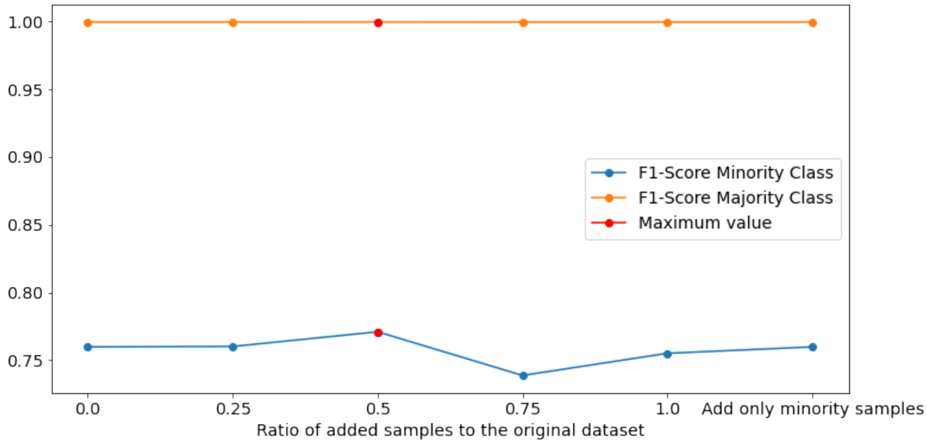


Figure 34: Ratio of data added at the Credit Card Fraud dataset by GAN

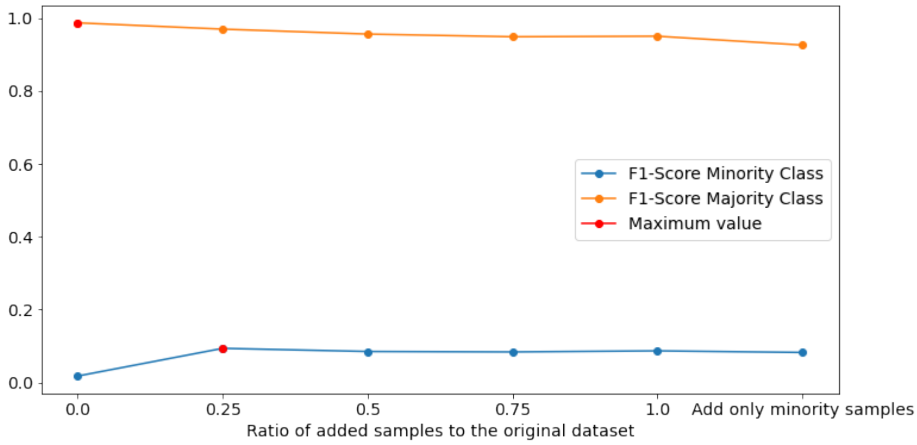


Figure 35: Ratio of data added at the Cerebral Stroke dataset by SMOTE

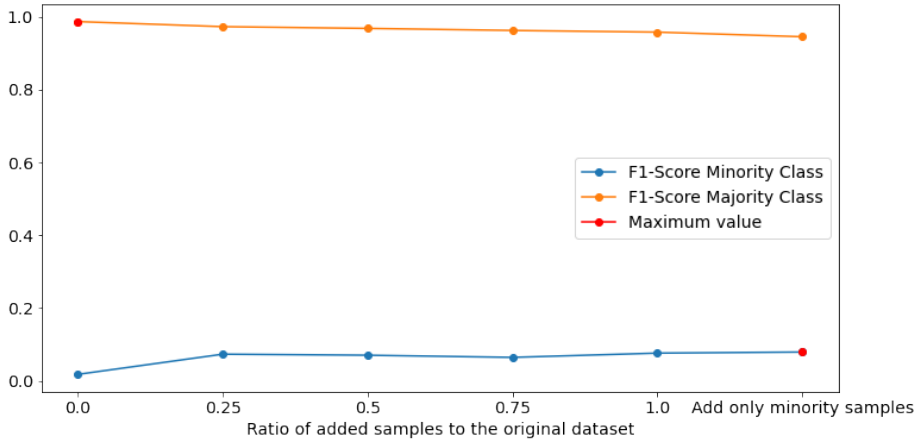


Figure 36: Ratio of data added at the Cerebral Stroke dataset by ADASYN

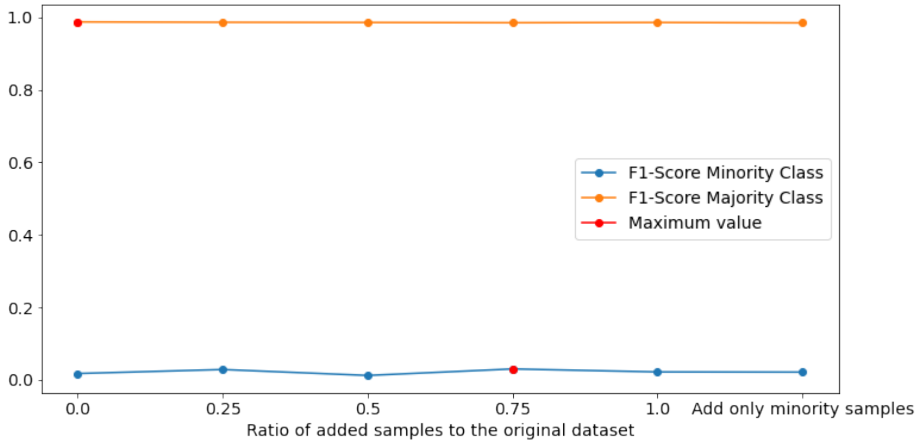


Figure 37: Ratio of data added at the Cerebral Stroke dataset by GMM

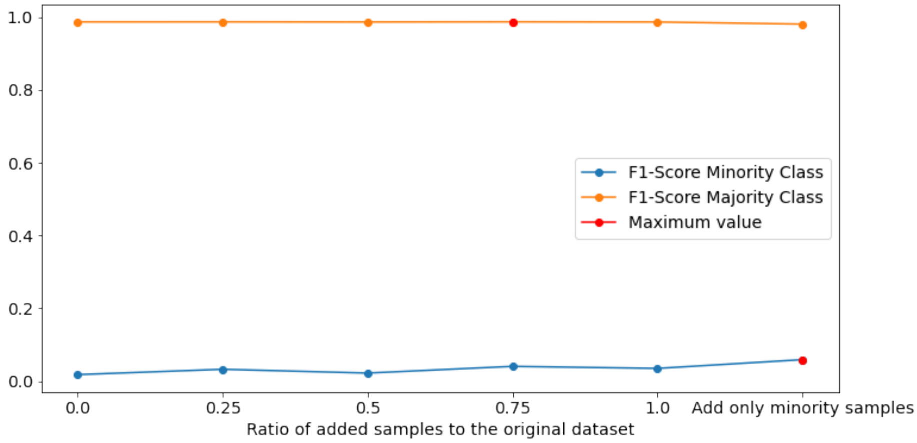


Figure 38: Ratio of data added at the Cerebral Stroke dataset by VAE with Decay

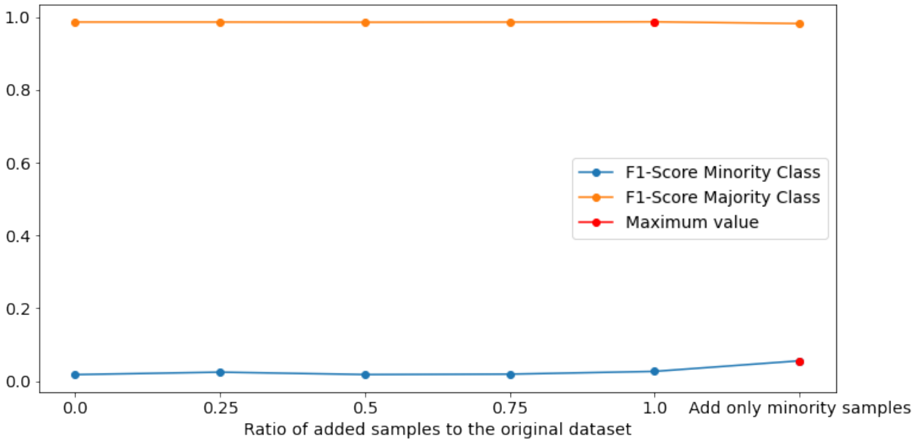


Figure 39: Ratio of data added at the Cerebral Stroke dataset by VAE with K-Means

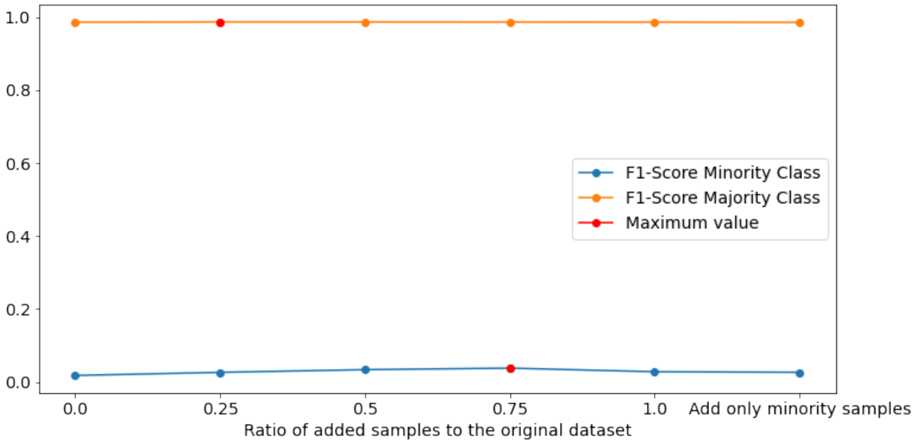


Figure 40: Ratio of data added at the Cerebral Stroke dataset by VAE with K-Means and Decay

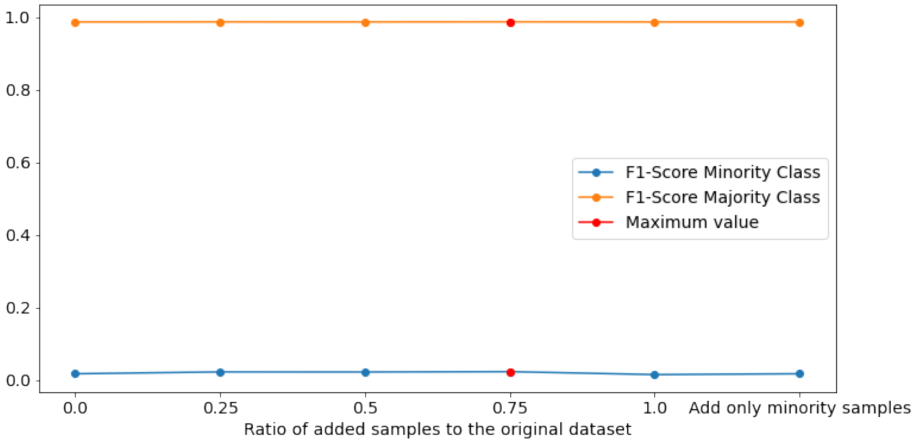


Figure 41: Ratio of data added at the Cerebral Stroke dataset by GAN

Appendix C

Detailed Results on the Classifiers Performance

Table 9: Classifiers performance comparison on only synthetic or real data training throughout different DA techniques on the Adult dataset

Classifiers	**DA Technique**	**Minority Class**			**Majority Class**			**G-Mean**
		Precision	**Recall**	**F1 Score**	**Precision**	**Recall**	**F1 Score**	
Decision Tree	No Technique	0.6492	0.6122	0.6302	0.8792	0.8951	0.8871	0.7403
	SMOTE	0.4928	0.7032	**0.5795**	0.8912	0.7706	**0.8265**	**0.7361**
	ADASYN	0.3726	0.5595	0.4473	0.8339	0.7013	0.7619	0.6264
	GMM	0.2729	0.4456	0.3385	0.7801	0.6236	0.6931	0.5271
	VAE with Decay	0.2453	0.7815	0.3734	0.7744	0.2378	0.3639	0.4311
	VAE with K-means	0.3168	0.2389	0.2724	0.7761	0.8366	0.8052	0.4471
	VAE with K-means and Decay	0.4845	0.2398	0.3208	0.7922	0.9191	0.851	0.4695
	GAN	0.2461	0.9974	0.3948	0.9748	0.03128	0.0606	0.1766
Random Forest	No Technique	0.705	0.6301	0.6655	0.8865	0.9164	0.9012	0.7599
	SMOTE	0.5373	0.7781	**0.6356**	0.918	0.7875	0.8478	**0.7828**
	ADASYN	0.4554	0.5298	0.4898	0.8428	0.7991	0.8204	0.6507
	GMM	0.2724	0.4515	0.34	0.7803	0.6177	0.6895	0.5281
	VAE with Decay	0.4101	0.227	0.2923	0.7853	0.8965	0.8372	0.4511
	VAE with K-means	0.5205	0.216	0.3053	0.7903	0.9369	0.8574	0.4498
	VAE with K-means and Decay	0.5932	0.2083	0.3084	0.7918	0.9547	**0.8657**	0.4460
	GAN	0.2472	0.9949	0.396	0.9605	0.0394	0.0756	0.1979
MLP	No Technique	0.712	0.6012	0.6519	0.8795	0.9229	0.9007	0.7449
	SMOTE	0.5098	0.8376	**0.6338**	0.9353	0.7447	0.8292	**0.7898**
	ADASYN	0.4846	0.7211	0.5796	0.8954	0.7568	0.8203	0.7387
	GMM	0.2708	0.449	0.3378	0.7792	0.6166	0.6884	0.5262
	VAE with Decay	0.2623	0.9014	0.4063	0.8624	0.196	0.3194	0.4203
	VAE with K-means	0.9539	0.1233	0.2184	0.7822	0.9981	**0.877**	0.3508
	VAE with K-means and Decay	0.0	0.0	0.0	0.7593	1.0	0.8632	0.0
	GAN	0.2553	0.9845	0.4055	0.9484	0.0892	0.1631	0.2964

Table 10: Classifiers performance comparison on only synthetic or real data training throughout different DA techniques on the Breast Cancer dataset

		Minority Class			**Majority Class**			
	DA Technique	**Precision**	**Recall**	**F1 Score**	**Precision**	**Recall**	**F1 Score**	**G-Mean**
Decision Tree	No Technique	0.875	0.875	0.875	0.9259	0.9259	0.9259	0.9001
	SMOTE	0.8611	0.9688	0.9118	0.98	0.9074	0.9423	0.9376
	ADASYN	0.8823	0.9375	0.9091	0.9615	0.9259	0.9434	0.9317
	GMM	1.0	0.1875	0.3158	0.675	1.0	0.806	0.4330
	VAE with Decay	0.6571	0.7188	0.6866	0.8235	0.7778	0.8	0.7477
	VAE with K-means	0.6512	0.875	0.7467	0.907	0.7222	0.8041	0.7949
	VAE with K-means and Decay	0.8846	0.8519	0.8679	0.7647	0.8125	0.7879	0.8319
	GAN	0.0	0.0	0.0	0.6279	1.0	0.7714	0.0
Random Forest	No Technique	1.0	0.9063	0.9508	0.9474	1.0	0.973	0.952
	SMOTE	0.9677	0.9375	0.9523	0.9636	0.9815	0.9725	0.9592
	ADASYN	1.0	0.9063	0.9508	0.9474	1.0	0.973	0.9520
	GMM	1.0	0.2813	0.439	0.7013	1.0	0.8244	0.5303
	VAE with Decay	0.7561	0.9688	0.8493	0.9778	0.8148	0.8889	0.8885
	VAE with K-means	1.0	0.9375	0.9677	0.9643	1.0	0.9818	0.9682
	VAE with K-means and Decay	0.8857	0.9688	0.9254	0.9804	0.9259	0.9524	0.9471
	GAN	0.0	0.0	0.0	0.6279	1.0	0.7714	0.0
MLP	No Technique	1.0	1.0	1.0	1.0	1.0	1.0	1.0
	SMOTE	0.9412	1.0	0.9697	1.0	0.963	0.9811	0.9813
	ADASYN	0.9063	0.9063	0.9063	0.9444	0.9444	0.9444	0.9252
	GMM	0.0	0.0	0.0	0.6279	1.0	0.7714	0.0
	VAE with Decay	0.871	0.8438	0.8571	0.9091	0.9259	0.9174	0.8839
	VAE with K-means	0.9355	0.9063	0.9206	0.9455	0.963	0.9541	0.9342
	VAE with K-means and Decay	0.8529	0.9063	0.8788	0.9423	0.9074	0.9245	0.9068
	GAN	0.7879	0.8125	0.8	0.887	0.8704	0.8785	0.8409

Table 11: Classifiers performance comparison on only synthetic or real data training throughout different DA techniques on the Credit Card Fraud dataset

		Minority Class			**Majority Class**			
	DA Technique	**Precision**	**Recall**	**F1 Score**	**Precision**	**Recall**	**F1 Score**	**G-Mean**
Decision Tree	No Technique	0.64	0.7273	0.6809	0.9995	0.9993	0.9994	0.8525
	SMOTE	0.3137	0.7273	0.4384	0.9995	0.9973	0.9984	0.8516
	ADASYN	0.0068	0.8182	0.0134	0.9996	0.7937	0.8849	0.8059
	GMM	0.0031	0.8636	0.0061	0.9995	0.515	0.6798	0.6669
	VAE with Decay	0.2727	0.4091	0.3273	0.999	0.9981	0.9986	0.639
	VAE with K-means	0.8333	0.2273	0.3571	0.9987	0.9999	0.9993	0.4767
	VAE with K-means and Decay	0.0	0.0	0.0	0.9983	1.0	0.9991	0.0
	GAN	0.0	0.0	0.0	0.9983	1.0	0.9991	0.0
Random Forest	No Technique	0.8824	0.6818	0.7692	0.9995	0.9998	0.9996	0.8257
	SMOTE	0.8947	0.7727	0.8293	0.9996	0.9998	0.9997	0.879
	ADASYN	0.0215	0.9091	0.042	0.9998	0.9288	0.963	0.9189
	GMM	0.0026	0.8636	0.0052	0.9995	0.4335	0.6047	0.6119
	VAE with Decay	0.8	0.3636	0.5	0.9989	0.9998	0.9994	0.603
	VAE with K-means	0.0	0.0	0.0	0.9983	1.0	0.9991	0.0
	VAE with K-means and Decay	0.0	0.0	0.0	0.9983	1.0	0.9991	0.0
	GAN	0.0	0.0	0.0	0.9983	1.0	0.9991	0.0
MLP	No Technique	0.8947	0.7727	0.8293	0.9996	0.9998	0.9997	0.879
	SMOTE	0.9998	0.9971	0.9984	0.3393	0.8636	0.4872	0.928
	ADASYN	0.0111	0.7727	0.022	0.9996	0.882	0.9371	0.8256
	GMM	0.0025	0.7727	0.005	0.9992	0.4717	0.6408	0.6037
	VAE with Decay	0.9	0.4091	0.5625	0.999	0.9999	0.9995	0.6396
	VAE with K-means	0.0	0.0	0.0	0.9983	1.0	0.9991	0.0
	VAE with K-means and Decay	0.0	0.0	0.0	0.9983	1.0	0.9991	0.0
	GAN	0.0	0.0	0.0	0.9983	1.0	0.9991	0.0

Table 12: Classifiers performance comparison on only synthetic or real data training throughout different DA techniques on the Cerebral Stroke dataset

		Minority Class			**Majority Class**			
	DA Technique	**Precision**	**Recall**	**F1 Score**	**Precision**	**Recall**	**F1 Score**	**G-Mean**
Decision Tree	No Technique	0.0321	0.0424	0.0365	0.9822	0.9764	0.9793	0.2034
	SMOTE	0.0426	0.3136	0.075	0.9856	0.8699	0.9241	0.5223
	ADASYN	0.0127	0.3644	0.0245	0.976	0.4761	0.64	0.4165
	GMM	0.0379	0.9322	0.0728	0.9978	0.5631	0.7199	0.7245
	VAE with Decay	0.0217	0.178	0.0386	0.9825	0.8517	0.9124	0.3893
	VAE with K-means	0.0208	0.0085	0.012	0.9819	0.9926	0.9872	0.0917
	VAE with K-means and Decay	0.0159	0.0085	0.0111	0.9819	0.9903	0.9861	0.0916
	GAN	0.0	0.0	0.0	0.9819	1.0	0.9909	0.0
Random Forest	No Technique	0.2	0.0085	0.0163	0.982	0.9994	0.9906	0.092
	SMOTE	0.0461	0.2627	0.0784	0.9851	0.8996	0.9404	0.4861
	ADASYN	0.0129	0.339	0.0248	0.9771	0.5199	0.6787	0.4198
	GMM	0.0375	0.9576	0.0722	0.9986	0.5462	0.7062	0.7232
	VAE with Decay	0.0	0.0	0.0	0.9819	0.9986	0.9902	0.0
	VAE with K-means	0.0	0.0	0.0	0.9819	0.9986	0.9902	0.0
	VAE with K-means and Decay	0.0	0.0	0.0	0.9819	0.9986	0.9902	0.0
	GAN	0.0	0.0	0.0	0.9819	0.9986	0.9902	0.0
MLP	No Technique	0.0	0.0	0.0	0.9819	1.0	0.9909	0.0
	SMOTE	0.0518	0.5508	0.0948	0.9899	0.814	0.8934	0.6696
	ADASYN	0.0166	0.3475	0.0318	0.981	0.621	0.7605	0.4645
	GMM	0.0361	0.9915	0.0696	0.9997	0.5107	0.6761	0.7116
	VAE with Decay	0.0	0.0	0.0	0.9819	1.0	0.9909	0.0
	VAE with K-means	0.0	0.0	0.0	0.9819	1.0	0.9909	0.0
	VAE with K-means and Decay	0.0	0.0	0.0	0.9819	1.0	0.9909	0.0
	GAN	0.0	0.0	0.0	0.9819	1.0	0.9909	0.0

Annex I

Publications

I.1 Benchmarking Data Augmentations Techniques for Tabular Data

Authors: Pedro Machado, Bruno Fernandes, and Paulo Novais

Title: Benchmarking Data Augmentations Techniques for Tabular Data

Conference: International Conference on Intelligent Data Engineering and Automated Learning (IDEAL) 2022

Date submission: 2022, July

Abstract: Imbalanced learning and small-sized datasets are usual in machine learning problems, even with the increased data availability provided by recent developments. The performance of learning algorithms in the presence of unbalanced data and significant class distribution skews is known as the "imbalanced learning problem". The models' performance on such problems can drastically decrease for certain classes with an uneven distribution because the models do not learn the distributive features of the data and present accuracy too favorable for a specific set of classes of data. As an example, this can have negative consequences when talking about cancer detection since the model may poorly identify unhealthy patients. Hence, data augmentation techniques are usually conceived to evaluate how models would behave in non-data-scarce environments, generating synthetic data that mimics the characteristics of real data. By applying those techniques, the amount of available data can be increased, balancing the class distributions. However, there are no standardized data augmentation processes that can be applied to every domain of tabular data. Therefore, this study aims to identify which characteristics of a dataset provide a better performance when synthesizing samples by a data augmentation technique in a tabular data environment.

Keywords: Data Augmentation, Imbalanced Data, Machine Learning

State of Publication: Accepted for publication

I.2 Data Augmentation Methods for Tabular Data

Authors: Pedro Machado, Bruno Fernandes, and Paulo Novais

Title: Data Augmentation Methods for Tabular Data

Journal: Machine Learning

Date submission: August, 2022

Abstract: Despite the increasing data availability brought by recent breakthroughs, machine learning difficulties frequently involve imbalanced learning and small datasets. The ”imbalanced learning problem“ refers to how learning algorithms perform when there are significant skewed class distributions and unbalanced data. Due to the fact that the models do not learn the distributive characteristics of the data and present accuracy that is overly advantageous for a specific set of classes, their performance on such issues can substantially decrease for some classes with an uneven distribution. Hence, data augmentation methods are usually developed to examine how models behave in non-data-scarce contexts, providing synthetic data that resembles the features of real data. By applying those techniques, the amount of available data can be increased, balancing the class distributions. However, there are no standardized data augmentation processes that can be applied to every domain of tabular data. Our results show the ability of SMOTE to properly augment tabular data throughout all domains of datasets, while our combination of the Variational Autoencoder with K-means, as well as GAN, demonstrated an increased capability when augmenting datasets whose features are mainly continuous.

Keywords: Data Augmentation, Generative Models, Imbalanced Data, Machine Learning

State of Publication: Submitted for publication

Benchmarking Data Augmentation Techniques for Tabular Data

Pedro Machado[0000−0002−1697−1667], Bruno Fernandes[0000−0003−1561−2897], and Paulo Novais[0000−0002−3549−0754]

ALGORITMI Center, University of Minho, Braga, Portugal
pedrofcmachado26@gmail.com, bruno.fernandes@algoritmi.uminho.pt, pjon@di.uminho.pt

Abstract. Imbalanced learning and small-sized datasets are usual in machine learning problems, even with the increased data availability provided by recent developments. The performance of learning algorithms in the presence of unbalanced data and significant class distribution skews is known as the "imbalanced learning problem". The models' performance on such problems can drastically decrease for certain classes with an uneven distribution because the models do not learn the distributive features of the data and present accuracy too favorable for a specific set of classes of data. As an example, this can have negative consequences when talking about cancer detection since the model may poorly identify unhealthy patients. Hence, data augmentation techniques are usually conceived to evaluate how models would behave in non-data-scarce environments, generating synthetic data that mimics the characteristics of real data. By applying those techniques, the amount of available data can be increased, balancing the class distributions. However, there are no standardized data augmentation processes that can be applied to every domain of tabular data. Therefore, this study aims to identify which characteristics of a dataset provide a better performance when synthesizing samples by a data augmentation technique in a tabular data environment.

Keywords: data augmentation · imbalanced data · machine learning

1 Introduction

In recent years, the imbalanced learning problem has become a highly frequent topic among academia, industry, and government funding agencies. The fundamental issue with the imbalanced learning problem is the ability of imbalanced data to significantly decrease the performance of machine learning algorithms [4]. These algorithms, when faced with imbalanced data, do not learn the distributive features of the data and present accuracies too favorable to a specific set of classes of data, in this case, the majority classes, compromising the performance of the other classes (the minority classes) because of that bias. In fairness, a dataset is considered imbalanced when it exhibits an unequal distribution between its classes. Nevertheless, the community usually considers that imbalanced data corresponds to a large unequal distribution and, in some cases, extremes.

In an imbalanced data problem, the real problem can't be solved with data treatment and/or model changes since the limitation is in the data itself. The same goes for a small-sized dataset, since the model cannot learn enough features to classify the problem in a real-time situation. Therefore, data augmentation appears as a way to surpass that limitation [3].

The present article, by identifying more favorable properties in a dataset when synthesizing samples, aims to conceive and benchmark several candidate models to overcome the imbalanced learning problem as well as increase the amount of available data without a loss in quality. With this in mind, we used classical techniques, such as SMOTE, a very uncommon clustering technique like Gaussian Mixture Model (GMM), and deep learning ones, such as Variational Autoencoder (VAE). Finally, this manuscript is structured as follows: the next section describes the literature review on the addressed domains; the third section presents the conducted experiments as well as the achieved results for this benchmark; the last section summarizes the obtained conclusions and outlines future work.

2 State of Art

Data Augmentation refers to methods for constructing iterative optimization or sampling algorithms via the introduction of unobserved data or latent variables [2]. With these techniques, we can increase the amount of data, thus balancing the target variable in an imbalanced dataset. Data augmentation can be applied to images or tabular data, but this article will focus on the latter. When used alongside images, techniques tend to apply transformations to samples of datasets like geometric transformations, flipping, color modification, cropping, etc. One other way is to introduce new synthetic images created by machine learning algorithms, for example, VAEs. Instead, if we are dealing with tabular data, we cannot apply simple transformations to samples, but instead synthesize samples (new or duplicated) based on the class distributions and features.

In [11], Data Augmentation is utilized to surpass the data limitations of the minority class, in this case, fraudulent transactions. The classification performance improved considerably and overfitting was alleviated, demonstrating the benefits of using a these techniques. These techniques can also be applied to automated skin lesion analysis by applying traditional color and geometric transformations, and more unusual augmentations such as elastic transformations, random erasing, and a novel augmentation that mixes different lesions, as stated in [8]. They prove the importance of data augmentation techniques in both training and testing, leading to more performance gains than simply obtaining new images.

In this study, we focused on some of the most popular data augmentation techniques, namely, SMOTE, GMM, and VAE. These techniques are present in multiple data augmentation studies and are going to be developed and evaluated in order to augment the used datasets.

SMOTE. One of the classic data augmentation techniques is SMOTE. It oversamples the minority class by creating synthetic samples [1]. 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. Therefore, instead of duplicating minority samples, SMOTE synthesizes new examples from that class.

This technique synthesizes the minority class by operating in the feature space. It selects examples that are close in the feature space and introduces synthetic samples along the line drawn from these examples. SMOTE is effective because the new synthetic samples from the minority class are somewhat close in feature space to real samples from that same class. This makes the created samples plausible.

Gaussian Mixture Model The GMM's generative nature provides an opportunity to explore its performance as a data augmentation technique, contrarily to other clustering algorithms, such as K-means. The use of a simple radial distance metric by k-means to assign cluster membership results in poor performance and a typical circular form for the clusters. This algorithm has no built-in way of accounting for non-circular clusters (oblong or elliptical), which do not represent the true shape of the data points sometimes. Moreover, this algorithm does not have a probabilistic nature when forming clusters.

Therefore, GMMs are an extension of the ideas behind k-means. This algorithm aims to model the data as a combination of multiple multi-dimensional Gaussian probability distributions and it works on the basis of the Expectation-Maximization algorithm. Because of this, the EM algorithm finds the maximum likelihood, i.e., finds a set of parameters that results in the best fit for the joint probability of the data sample [7]. Due to the generative nature of GMM, it can generate synthetic data close to the distribution of the fitted data [10]. After the algorithm fits the data and learns its distribution, it can generate an arbitrary number of samples from the learned distribution.

Variational Autoencoder. Nowadays, deep learning has gained a lot of interest and has made some amazing improvements regarding its performance. From the deep learning models, the family of generative models has also increased in popularity, showing a magnificent ability to produce highly realistic samples of various kinds, such as images, text, and sounds. These families of models, like all deep learning models, rely on huge amounts of data, well-structured architectures, and smart training techniques. One of these popular deep learning generative models is the Variational Autoencoder. In short, a VAE is an autoencoder whose encoding distribution is regularized during the training in order to ensure that its latent space[1] has good properties, allowing us to generate some new data [9].

[1] Latent space is a representation of compressed data in which similar data points are closer together in space. It is useful to learn the features.

A VAE consists of an encoder and a decoder, just like an autoencoder, but the loss term and the encoded layers of the autoencoder are altered in order for the model to be used as a generative model [5]. Its training is adjusted to avoid overfitting, making sure that the latent space has good properties that enable the generative process. On the other hand, an autoencoder is trained to encode and decode with as few losses as possible, making no difference how the latent space is organized. The main distinction between the two encoding layer algorithms is that they encode an input as a distribution throughout the latent space rather than a single point [9]. With this in mind, the VAE avoids having some points in the latent space that would provide meaningless information once decoded.

3 Experiments

The data generated was used in two different ways in order to evaluate the data augmentation techniques. First, it was added to the original training data that trained the classifiers and then evaluated based on the test data. The second approach is to train the classifiers only with synthetic data and evaluate them with the test data. Note that all the test data is real.

3.1 Data

In this experiment, multiple datasets were chosen to perform a good comparison of these data augmentation techniques in generating new data. Moreover, these datasets are inserted into different domains, such as health and fraud detection, and are imbalanced. Furthermore, these datasets were also chosen due to being mainly composed by continuous or categorical features. The chosen datasets were the following:

- **Adult**. The adult dataset was extracted from the census bureau and has information about multiple adults. This dataset serves as a binary classification, predicting if a certain adult has an income superior to fifty thousand in a year. The target class is clearly imbalanced, as the majority class (income superior to fifty thousand) is three times more frequent than the minority class. The dataset has over $30K$ instances.
- **Breast Cancer**. Another health domain analyzed in this experiment was breast cancer prediction. This dataset was obtained from the University of Wisconsin Hospitals, Madison by Dr. William H. Wolberg and contains samples of clinical cases gathered periodically. The dataset contains a target class imbalanced with 66% of the instances belonging to benign cases and the rest being malignant. This dataset, beyond being imbalanced, is also small in size, since it only has 570 instances.
- **Credit Card Fraud**. Fraud detection is also a recurrent domain where imbalanced data is present. Therefore, in this dataset, transactions made by credit cards in September 2013 by European cardholders are analyzed. This dataset originally had more than $280K$ instances but was reduced (while maintaining the target class ratio) to $85K$ due to computational reasons.

3.2 Assessment Metrics

In order to compare the performances of all the data augmentation techniques, it is required to define how their performances can be compared. Therefore, we need to define which metrics fit better into an imbalanced data problem. Traditionally, the most often used metrics are *accuracy* and *error rate*. Although accuracy provides an easy way to describe the model's performance, it can mislead in certain situations. Therefore, accuracy and error rate do not provide enough information about a classifier's functionality in terms of the sort of classification required.

As a means to provide comprehensive assessments of imbalanced learning problems, the research community adopted other evaluation metrics, such as *precision*, *recall*, *F-measure*[2], and *G-mean*.

First, precision is a metric that measures how many correct positive predictions the model makes (a measure of exactness)[3]. Therefore, precision calculates the accuracy of the positive class and is sensitive to data distribution. Second, recall is a metric that measures how many correct positive predictions were produced out of all possible positive predictions. Unlike precision, which only gives information on the correct positive predictions of all positive predictions, recall indicates the missed positive predictions and it is not sensitive to data distributions. Moreover, recall is also known as sensitivity. When used correctly, recall and precision can evaluate an imbalanced learning problem adequately. Nevertheless, the F-measure metric combines the two previous metrics as a weighted focus on either recall or precision. Finally, the G-mean (Geometric mean) metric evaluates the balance of classification between the majority and minority classes. Even if the negative cases are accurately identified, a low G-Mean suggests poor performance in the classification of positive cases.

3.3 Experimental Results

Regarding the synthetic data generated by all data augmentation techniques, the experiments have produced a variety of findings. These analyses consist, mainly, of:

1. Comparing each feature's distribution throughout statistical methods;
2. Training the classifiers only with synthetic data;
3. Training the classifiers with real and synthetic data.

In all cases, all the techniques implemented generated the same amount of synthetic data. In this case, this amount is the size of the training dataset (i.e., seventy percent of the entire dataset). As a result, all the synthetic data generated can be compared to one another and to the original data.

In order to compare the synthetic data of each data augmentation technique, we first compared how each feature of the real and synthetic datasets behaves,

[2] It is also known as *F-score*.

[3] In this case, the positive class is considered the minority.

i.e., if they possess the same distribution. Ideally, a synthetic dataset should have properties very similar to the original one. Therefore, we implemented some statistical methods to compare each feature distribution on the real and synthetic datasets. However, due to the very different behavior of continuous and categorical features, it was necessary to apply different statistical tests. The categorical feature distributions were analyzed by the chi-square test and the continuous features by the Kolmogorov-Smirnov test.

In the adult dataset, most of the features are categorical, with only one continuous feature. The statistical tests showed that the data augmentation techniques had almost no difficulty representing the original categorical features in the synthetic data. However, the techniques couldn't represent the continuous feature distributions since all of the techniques failed the test. In regards to the Breast Cancer dataset, all features are continuous since they are medical measures. SMOTE showed as the best technique to represent the feature distributions as the other techniques couldn't. Finally, the credit card fraud dataset had similar properties to the previous dataset, containing only continuous features. However, all the data augmentation techniques had difficulties representing similar continuous feature distributions.

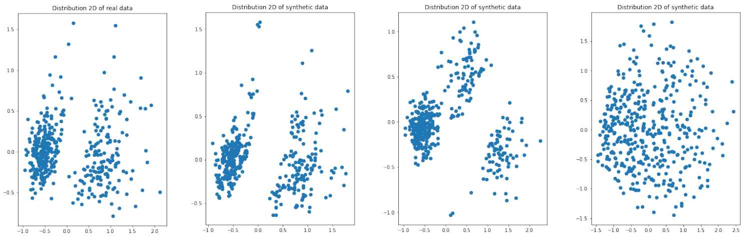
The results of the data augmentation techniques throughout all datasets indicated the increased difficulty in representing continuous feature distributions, with SMOTE being the technique with the best representations. However, the categorical feature distributions were much easier to represent, with GMM and VAE being the ones with better results.

Moreover, one important factor noticed during the training of more complex and computationally resource-demanding techniques, such as VAE, was the fact that continuous features should be normalized in order to reduce computational cost and avoid crashes during training. These kinds of crashes are detectable when the loss is *NaN* during training.

In regards to the data distribution of one of the datasets, as we can see in Figure 1, most data augmentation techniques can represent the entirety of the data distribution. Also, VAE seems to be the technique with the most difficulty in separating what seems to be the two target classes, but it doesn't appear to affect the classifier's performance as we will observe.

We can still perform two additional crucial analyses to determine how reliable the generated data is after the analysis of the synthetic data properties. First, we are going to train the machine learning classifiers with real data and then compare the results with training with only synthetic data. Note that there were multiple classification models, but we only represented the best model for each case.

During the experiments and implementation of the data augmentation techniques, there were some obstacles with regard to the performance of some techniques, mainly the Variational Autoencoder. The implemented VAE suffered from the phenomenon called *posterior collapse* [6]. As a result, the minority class was unable to be synthesized, and the solution was to add a weight decay on the loss function as well as change the latent space dimension.



(a) Real Data (b) SMOTE (c) GMM (d) VAE

Fig. 1: Comparison of two dimension data throughout all data augmentation techniques on the Breast Cancer Dataset.

As observed in Table 1, synthetic data can achieve similar training scores in comparison with training with real data. SMOTE and VAE demonstrated better performance in generating samples on the three datasets. In this experiment, the VAE showed better performance for datasets with fewer categorical features (in this experiment, the Breast Cancer and Credit Card fraud datasets), while GMM indicated difficulties in generating good synthetic data. These experiments demonstrated that data augmentation techniques such as SMOTE or VAE can synthesize data in order to replace the real data in an efficient way. This could be very interesting in datasets where some data is sensitive and privacy matters.

Table 1: Best classifier performance on different kinds of training data.

Dataset	**DA Technique**	**Minority Class**			**Majority Class**			**G-Mean**
		Precision	**Recall**	**F1 Score**	**Precision**	**Recall**	**F1 Score**	
Adult	—	0.6957	0.6327	0.6626	0.8868	0.9123	0.8993	0.7597
	SMOTE	0.6088	0.6173	**0.6130**	0.8781	0.8742	**0.8762**	**0.7643**
	GMM	0.2352	0.6033	0.3385	0.7504	0.3780	0.5028	0.4776
	VAE	0.4565	0.2742	0.3427	0.7958	0.8965	0.8431	0.4958
Breast Cancer	—	1.0	0.9524	0.9756	0.9737	1.0	0.9867	0.9759
	SMOTE	1.0	0.8095	0.8947	0.9024	1.0	0.9487	0.8997
	GMM	0.6111	0.5238	0.5641	0.7500	0.8108	0.7792	0.6517
	VAE	0.9090	0.9524	**0.9302**	0.9722	0.9459	**0.9589**	**0.9492**
Credit Card Fraud	—	1.0	0.8667	0.9286	0.9998	1.0	0.9999	0.9309
	SMOTE	0.9286	0.8667	0.8966	0.9998	0.9998	0.9998	0.9309
	GMM	0.0027	0.8667	0.0054	0.9995	0.4411	0.6121	0.6005
	VAE	0.9286	0.8667	**0.8967**	0.9998	0.9999	**0.9999**	**0.9309**

With those results in mind, the following analyses focus on training the classifiers with an increased amount of data (in this case, twice the original data size). At Table 2, we get to see that the data augmentation techniques that achieved better results in Table 1 got the best results with the addition of real data into the classifier's training. We can also observe that these techniques increase or maintain the classifier's performance in both major and minor classes. One example of that is the dataset Breast Cancer, where the application of a Variational Autoencoder made the classifier's performance go up in both classes' f1-score and g-mean metrics. This implies that the data augmentation can increase not only the minority class's performance but all classes' performances as well. Another interesting finding was the performance gained by the GMM technique when joining its synthetic and real data. This may be explained by the variety of generated samples that, in this case, benefited the classifier training.

Table 2: Best classifier performance while training with real and synthetic data.

Dataset	**DA Technique**	**Minority Class**			**Majority Class**			**G-Mean**
		Precision	**Recall**	**F1 Score**	**Precision**	**Recall**	**F1 Score**	
Adult	SMOTE	0.6935	0.6263	0.6583	0.8850	0.9123	0.8884	0.7559
	GMM	0.7149	0.6301	**0.6698**	0.8870	0.9203	**0.9034**	**0.7615**
	VAE	0.7032	0.6199	0.6589	0.8839	0.9171	0.9002	0.7540
Breast Cancer	SMOTE	1.0	0.9048	0.9500	0.9487	1.0	0.9737	0.9512
	GMM	1.0	0.9048	0.9500	0.9487	1.0	0.9734	0.9512
	VAE	0.9545	1.0	**0.9767**	1.0	0.9730	**0.9863**	**0.9864**
Credit Card Fraud	SMOTE	1.0	0.8000	0.8889	0.9996	1.0	0.9998	0.8944
	GMM	1.0	0.7333	0.8462	0.9996	1.0	0.9998	0.8563
	VAE	1.0	0.8667	**0.9286**	0.9998	1.0	**0.9999**	**0.9308**

4 Conclusion

In this study, we went through a benchmark of different data augmentation techniques in multiple datasets of various domains. We observed that classical techniques such as SMOTE are competitive with more recent and powerful techniques like VAE. Also, the introduction of a not so frequent technique like GMM gave a new look to cluster models as a possibility to generate samples. Even though the Variational Autoencoder is more complex and susceptible to training problems such as the *posterior collapse*, it is a very powerful technique.

Furthermore, VAE was shown to be a better solution for a dataset with more continuous features. On the contrary, SMOTE had a better performance in a dataset with more categorical features. One other important factor to take into account is the normalization of continuous features during the preprocessing of the data. This avoids higher losses that may stall the training process.

Regarding the obtained results, the data augmentation techniques showed a great capability to create almost identical datasets to the real ones and have very similar scores. Moreover, these techniques can combat the imbalanced data problem by increasing the performance of the minority class, and they can also increase the size of a dataset without a classifier's performance loss. Future work will focus on further benchmarking techniques and new analyses of the classifier's performances.

Acknowledgements This work has been financed by *FCT – Fundação para a Ciência e Tecnologia* within the scope of project DSAIPA/AI/0099/2019.

References

1. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research **16**, 321–357 (2002)
2. van Dyk, D.A., Meng, X.L.: The art of data augmentation. Journal of Computational and Graphical Statistics **10**, 1–50 (2001)
3. Fernandes, B., Silva, F., Alaiz-Moretón, H., Novais, P., Analide, C., Neves, J.: Traffic flow forecasting on data-scarce environments using arima and lstm networks. In: Rocha, Á., Adeli, H., Reis, L.P., Costanzo, S. (eds.) New Knowledge in Information Systems and Technologies. pp. 273–282. Springer International Publishing, Cham (2019)
4. He, H., Garcia, E.A.: Learning from imbalanced data. IEEE Transactions on Knowledge and Data Engineering **21**(9), 1263–1284 (2009). https://doi.org/10.1109/TKDE.2008.239
5. Islam, Z., Abdel-Aty, M., Cai, Q., Yuan, J.: Crash data augmentation using variational autoencoder. Accident Analysis & Prevention **151**, 105950 (2021). https://doi.org/https://doi.org/10.1016/j.aap.2020.105950
6. Lucas, J., Tucker, G., Grosse, R.B., Norouzi, M.: Understanding posterior collapse in generative latent variable models. In: DGS@ICLR (2019)
7. McLachlan, G.J., Krishnan, T.: The EM algorithm and extensions, vol. 382. John Wiley & Sons (2007)
8. Perez, F., Vasconcelos, C., Avila, S., Valle, E.: Data augmentation for skin lesion analysis. In: OR 2.0 Context-Aware Operating Theaters, Computer Assisted Robotic Endoscopy, Clinical Image-Based Procedures, and Skin Image Analysis. pp. 303–311. Springer International Publishing, Cham (2018)
9. Rocca, J.: Understanding variational autoencoders (vaes) (03 2021), `https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73`, last Visited July 11, 2022
10. Sarkar, T.: How to use a clustering technique for synthetic data generation (9 2019), `https://towardsdatascience.com/how-to-use-a-clustering-technique-for-synthetic-data-generation-7c84b6b678ea`, last Visited July 10, 2022
11. Shao, M., Gu, N., Zhang, X.: Credit card transactions data adversarial augmentation in the frequency domain. In: 2020 5th IEEE International Conference on Big Data Analytics (ICBDA). pp. 238–245 (2020). https://doi.org/10.1109/ICBDA49040.2020.9101344

Data Augmentation Methods for Tabular Data

Pedro Machado[1*], Bruno Fernandes[1] and Paulo Novais[1]

[1]ALGORITMI Center, University of Minho, Braga, Portugal.

*Corresponding author(s). E-mail(s): pedrofcmachado26@gmail.com;
Contributing authors: bruno.fernandes@algoritmi.uminho.pt; pjon@di.uminho.pt;

Abstract

Despite the increasing data availability brought by recent breakthroughs, machine learning difficulties frequently involve imbalanced learning and small datasets. The "imbalanced learning problem" refers to how learning algorithms perform when there are significant skewed class distributions and unbalanced data. Due to the fact that the models do not learn the distributive characteristics of the data and present accuracy that is overly advantageous for a specific set of classes, their performance on such issues can substantially decrease for some classes with an uneven distribution. Hence, data augmentation methods are usually developed to examine how models behave in non-data-scarce contexts, providing synthetic data that resembles the features of real data. By applying those techniques, the amount of available data can be increased, balancing the class distributions. However, there are no standardized data augmentation processes that can be applied to every domain of tabular data. Our results show the ability of SMOTE to properly augment tabular data throughout all domains of datasets, while our combination of the Variational Autoencoder with K-means, as well as GAN, demonstrated an increased capability when augmenting datasets whose features are mainly continuous.

Keywords: Data Augmentation, Generative Models, Imbalanced Data, Machine Learning

1 Introduction

The issue of imbalanced learning has recently gained significant attention from the industry, academic community, and government funding organizations. The ability of imbalanced data to dramatically reduce the performance of machine learning algorithms is the underlying problem with the imbalanced learning problem (He & Garcia, 2009). These algorithms, when presented with imbalanced data, fail to learn the distributive characteristics of the data and present accuracy that is overly favorable to one group of data classes, in this case, the majority classes. As a result of this bias, the performance of the other classes (the minority classes), which are less well represented in the data, is negatively impacted. In fairness, a dataset is labeled as imbalanced when the distribution of its classes is uneven. However, the general consensus is that imbalanced data corresponds to a significant unequal distribution and, occasionally, extremes.

Since the limitation in an imbalanced data problem is within the data itself, the real problem cannot be resolved through data treatment and/or model adjustments. The same holds true for short datasets, as the model cannot learn enough features to accurately categorize the issue in a real-time setting. Therefore, data augmentation (DA) seems to be a means to get around that limit (Fernandes et al., 2019).

The present article aims to conceive and benchmark several candidate models to overcome the imbalanced learning problem as well as increase the amount of available data without a loss in quality, extending previous work of the authors in this domain (Machado, Fernandes, & Novais, 2022). With this in mind, we use classical techniques, such as SMOTE and ADASYN, a very uncommon clustering technique like Gaussian Mixture Model (GMM), and deep learning ones, such as Variational Autoencoder (VAE) and Generative Adversarial Network (GAN). Furthermore, this study aims to answer four specific research questions (RQ) in regard to the use of DA in tabular settings, mainly:

RQ1) Does data augmentation improves the performance of machine learning classifiers?

RQ2) Does the dataset properties influence the quality of generated synthetic samples?

RQ3) How many samples should a DA technique generate for a certain problem?

RQ4) Which DA technique provides better quality in terms of synthetic tabular data?

Finally, the organization of this document is as follows: the next section discusses the state of the art on the subject areas; the third section presents the conducted experiments; the fourth section presents the obtained results; and, finally, the last section summarizes the obtained conclusions.

2 State of Art

DA refers to techniques for building iterative optimization or sampling algorithms with the addition of latent variables or unobserved data (van Dyk & Meng, 2001). These methods allow us to add more data, balancing the target variable in an imbalanced dataset. Images or tabular data can be subject to DA. However, this article will concentrate on the latter. Techniques typically apply transformations to samples of datasets like geometric transformations, flipping, color modification, cropping, etc. when employed alongside images. Adding fresh synthetic images made by machine learning algorithms, such as VAEs and GANs, is another option. Instead, when working with tabular data, we are unable to simply apply simple transformations; rather, we must create new or duplicate samples based on the class distributions and features.

In Shao, Gu, and Zhang (2020), DA is used to overcome the data constraints of the minority class, in this case, fraudulent transactions. The effectiveness of these strategies was demonstrated by the significant improvement in classification performance and the reduction of overfitting. As stated in Perez, Vasconcelos, Avila, and Valle (2018), these techniques can also be used for automated skin lesion analysis by applying conventional color and geometric transformations, as well as more unusual augmentations like elastic transformations, random erasing, and a novel augmentation that combines various lesions. They demonstrate the value of DA methods in both testing and training, resulting in greater performance gains than merely acquiring new images.

Moreover, by expanding the training dataset and including images from DA approaches, it is possible to improve face recognition datasets by reducing overfitting, posture variation, lighting changes, and partial occlusions (Lv, Shao, Huang, Zhou, & Zhou, 2017). Finally, in Nalepa, Marcinkiewicz, and Kawulok (2019), image segmentation of magnetic resonance images of brain tumors also applied DA to expand the dataset and strengthen a machine learning model.

In this study, we focused on some of the most popular DA techniques, namely, SMOTE, ADASYN, GMM, VAE, and GAN. These techniques are present in multiple DA studies and are going to be developed and evaluated in order to augment the used datasets.

2.1 SMOTE

One of the classic DA techniques is SMOTE. It oversamples the minority class by creating synthetic samples (Chawla, Bowyer, Hall, & Kegelmeyer, 2002). 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. Therefore, SMOTE synthesizes new examples from that class rather than reproducing minority samples.

As shown in Figure 1, this method operates in the feature space to synthesize the minority class. In order to introduce synthetic samples along the line inferred from these examples, it chooses instances that are reasonably similar in the feature space. SMOTE works well because the new synthetic samples from the minority class are somewhat similar to the real samples from the same class in terms of feature space. This makes the created samples plausible.

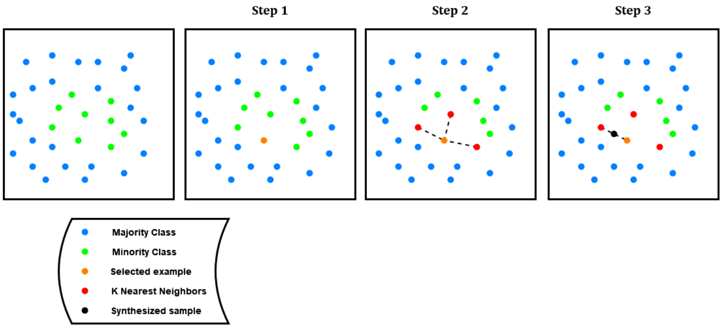


Fig. 1 SMOTE algorithm

2.2 ADASYN

According to (He, Bai, Garcia, & Li, 2008), the ADASYN strategy, in contrast to SMOTE, focuses on synthesizing minority samples that drive the learning algorithm to concentrate on those harder to learn samples. Therefore, the resulting dataset after the application of the technique ADASYN is not a balanced representation of the class distributions because the technique forces the learning algorithm to focus on those difficult samples to learn.

2.3 Gaussian Mixture Model

In contrast to other clustering algorithms like K-means, the generative aspect of the GMM offers a possibility to investigate its performance as a DA approach. Oblong or elliptical clusters, which occasionally represent the true geometry of the data points, are automatically taken into consideration by GMM. This approach, which is based on the Expectation-Maximization algorithm, seeks to describe the data as a combination of many multidimensional Gaussian probability distributions.

Because of this, the EM algorithm finds the maximum likelihood, i.e., finds a set of parameters that results in the best fit for the joint probability of the data sample (McLachlan & Krishnan, 2007). Due to the generative nature of GMM, it can generate synthetic data close to the distribution of the fitted data (Sarkar, 2019). After the algorithm fits the data and learns its distribution, it can generate an arbitrary number of samples from the learned distribution.

2.4 Variational Autoencoder

Nowadays, deep learning has gained a lot of interest and has made some amazing improvements regarding its performance.

The family of generative models, which derive from deep learning models, has gained popularity due to its amazing capacity to create samples of many kinds, including images, text, and sounds, that are incredibly realistic. These families of models, like all deep learning models, rely on huge amounts of data, well-structured architectures, and smart training techniques. The Variational Autoencoder is one of these well-liked deep learning generative models and it is described in Figure 2. In short, a VAE is an autoencoder whose encoding distribution is regularized during the training in order to ensure that its latent space[1] has good properties, allowing us to generate some new data (Rocca, 2021).

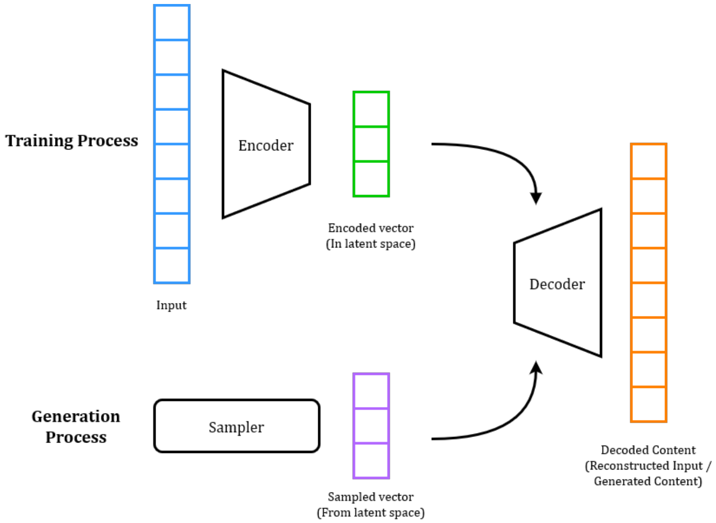


Fig. 2 Variational Autoencoder algorithm [Adapted from (Rocca, 2021)]

2.5 Generative Adversarial Network

GANs, along side VAEs, are a famous deep learning generative model and were proposed in (Goodfellow et al., 2014). The GAN model architecture involves two neural networks: a generator and a discriminator. The generator is a model that generates new plausible samples for the problem, while the discriminator is a model that classifies examples as real (from the domain) or fake (generated) (Brownlee, 2019). Therefore, GANs are based on a game-theoretic scenario in which the generator network must compete against an adversary, the discriminator (Goodfellow et al., 2014), as can be observed in Figure 3.

On one hand, the Generator model takes a random vector, drawn from a Gaussian distribution, as input and generates a sample in the domain. This

[1] Latent space is a representation of compressed data in which similar data points are closer together in space.

vector serves as a seed for the generative process. When the training is complete, the generator model is kept and used to generate new samples. On the other hand, the Discriminator model receives as input an example that can be real or generated and predicts whether it is real or fake (generated). Therefore, the real inputs are received from the dataset, whereas the generated examples are output by the Generator model. Moreover, the Discriminator is a normal classification model and, after the training, it is discarded as we are interested in the final Generator.

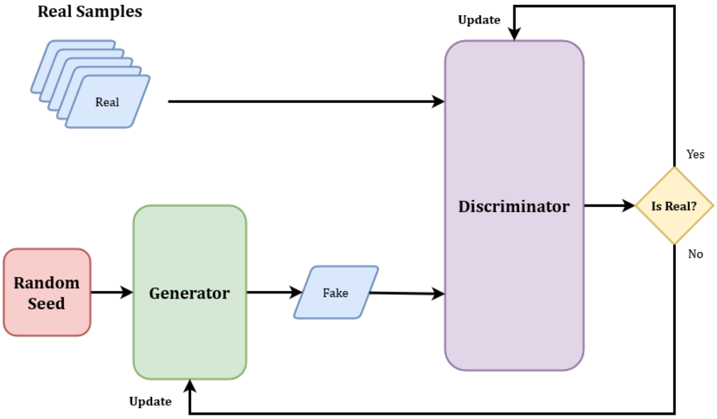


Fig. 3 Generative Adversarial Network algorithm

3 Experiments

The data generated was used in two different ways in order to evaluate the DA techniques. First, we train the classifiers only with synthetic data and evaluate them with the test data, which is composed only of real data not used in the training. The second approach is to add the generated data to the original training data to train the classifiers and then evaluate them.

Furthermore, during the experiments and implementation of the DA techniques, there were some obstacles with regard to the performance of some techniques, mainly the Variational Autoencoder and the Generative Adversarial Network. The implemented VAE suffered from the phenomenon called *posterior collapse* (Lucas, Tucker, Grosse, & Norouzi, 2019), as well as the GAN suffered from *mode collapse*, (Li, Fan, Wang, Ma, & Cui, 2021). The posterior collapse happens when the information contained in the learned latent space is rendered useless. As for GAN, its generator only learns to generate a small set of outputs, making the generator over-optimizing for a particular discriminator. As a result, the minority class was unable to be synthesized. Therefore, it was crucial to find a way to surpass this limitation. As for the

GAN, the solution passed through adding noise to the discriminator's inputs and incrementing the latent space dimension.

Moreover, we explored a bit more of the possible solutions for the VAE. The first solution was to add a weighted decay to its loss. The VAE loss is composed of two factors. The first forces the decoded samples to resemble the input by penalizing the latent representation with a reconstruction loss (RC_{loss}), (Kingma & Welling, 2019; Singh & Ogunfunmi, 2021). Consequently, the RC_{loss} can be explained as:

$$RC_{loss} = -\sum_{bs=1}^{BS}\sum_{i=1}^{N} x_{bs,i} \cdot log(x'_{bs,i}) \tag{1}$$

where x is the input data, N the dimensions of the input data, BS the batch size, and x' the reconstructed input. Keep in mind that this loss is predicated on the probability of the Binary Cross-Entropy.

The second loss is the Kullback-Leibler divergence term (KL_{loss}), which serves as a regularization term to aid the model's learning of well-formed latent spaces:

$$KL_{loss} = \sum_{bs=1}^{BS}\sum_{i=1}^{N} \frac{\sigma_{bs,i}^2 + \mu_{bs,i}^2 - 1 - 2log(\sigma_{bs,i})}{2} \tag{2}$$

where μ, σ are the mean and standard deviation of a Gaussian distribution, respectively, BS is the batch size, N the dimensions of the input data, (Kingma & Welling, 2019; Singh & Ogunfunmi, 2021). The network weights are controlled by the weight decay (W_{decay}), which penalizes the KL_{loss} more as the number of training epochs increases[2] and causes the model to become more regular. The VAE network is thus prevented from overfitting the training data, which is typically towards the majority class, by this weight decay. In essence, the VAE loss function is:

$$VAE_{loss} = RC_{loss} + KL_{loss} \cdot W_{decay} \tag{3}$$

As for the other solution, we explored the latent space properties, adding a k-means algorithm to be applied to the latent space. This cluster algorithm would be fitted after the VAE training in order to identify the minority and majority clusters. Therefore, before the VAE generated samples, the cluster algorithm would readjust the points of the latent space[3], making them closer to the centroid of the minority class cluster. An example of the application of this cluster algorithm to the process of synthesizing by the VAE can be seen in the section 4 by the Figure 4. Finally, the last version of VAE implemented in regards to surpassing the posterior collapse was the combination of the weight decay and the k-means.

Furthermore, the use of more computationally demanding DA techniques, such as VAE and GAN, demonstrated a greater importance for each dataset

[2] The weight decay has a proportional inverse behavior in regards to the number of training epochs.

[3] In this case, the latent space is the input vector that the decoder uses to generate samples.

preprocessing, particularly for continuous features. These continuous features were an issue during the training of these techniques because of the calculated loss. Another occurring phenomenon was that the loss had *NaN* values during the training due to high computations when faced by continuous features. As a result, a simple solution to this problem was to perform a normalization on each continuous feature.

3.1 Data

In this experiment, multiple datasets were chosen to perform a good comparison of these DA techniques in generating new data. Additionally, these datasets are unbalanced[4] and inserted into many domains, including fraud detection and health. The chosen datasets were the following:

- **Adult**. The adult dataset was extracted from the census bureau and has information about multiple adults, (Kohavi & Becker, 1994). This dataset serves as a binary classification, predicting if a certain adult has an income superior to fifty thousand in a year. Regarding its features, it has seven categorical features and one continuous. Furthermore, the target class is clearly imbalanced, as the majority class (income superior to fifty thousand) is three times more frequent than the minority class. The dataset has over $30K$ instances.
- **Breast Cancer**. Another health domain analyzed in this experiment was breast cancer prediction, (Wolberg, Street, & Mangasarian, 1998). This dataset was obtained from the University of Wisconsin Hospitals, Madison by Dr. William H. Wolberg and contains samples of clinical cases gathered periodically. The dataset contains a target class imbalanced with 66% of the instances belonging to benign cases and the rest being malignant. This dataset, beyond being imbalanced, is also small in size, since it only has 570 instances. Its features are all continuous.
- **Credit Card Fraud**. Fraud detection is also a recurrent domain where imbalanced data is present, (ULB, 2018). Therefore, in this dataset, transactions made by credit cards in September 2013 by European cardholders are analyzed. This dataset originally had more than $280K$ instances but was reduced (while maintaining the target class ratio) to $85K$ due to computational reasons. Moreover, this dataset contains only continuous attributes.
- **Cerebral Stroke**. This dataset contains features regarding individuals that may or may not suffer a cerebral stroke, (Liu, Fan, & Wu, 2019). A cerebral stroke is when part of the brain loses its blood supply and the part of the body that the blood-deprived brain cells control stops working. Therefore, it is very useful to predict if a person may or may not suffer a stroke. This dataset is highly imbalanced, having only 2% of strokes and more categorical features than continuous. The dataset has more than $40K$ instances.

[4] Note that the chosen datasets have a binary target.

3.2 Assessment Metrics

In order to compare the performances of all the DA techniques, it is required to define how their performances can be compared. Therefore, we need to define which metrics fit better into an imbalanced data problem. Traditionally, the most often used metrics are *accuracy* and *error rate*. Although accuracy provides an easy way to describe the model's performance, it can mislead in certain situations. Therefore, accuracy and error rate do not provide enough information about a classifier's functionality in terms of the sort of classification required.

Imbalanced data problems are examples of that kind of deceiving, because if a minority class has 5 percent of examples and the majority has the rest of the data classes, a model that classifies all instances as being in the majority class has 95 percent accuracy. At first glance, this value appears to be an excellent classifier for the problem at hand, but it fails to identify any of the minority examples. Therefore, accuracy and error rate do not provide enough information about a classifier's functionality in terms of the sort of classification required.

As a means to provide comprehensive assessments of imbalanced learning problems, the research community adopted other evaluation metrics, such as *precision*, *recall*, *F-measure*[5], and *G-mean*.

First, precision is a metric that measures how many correct positive predictions the model makes (a measure of exactness)[6]. Therefore, precision calculates the accuracy of the positive class and is sensitive to data distribution. Second, recall is a metric that measures how many correct positive predictions were produced out of all possible positive predictions. Unlike precision, which only gives information on the correct positive predictions of all positive predictions, recall indicates the missed positive predictions and it is not sensitive to data distributions. Moreover, recall is also known as sensitivity. When used correctly, recall and precision can evaluate an imbalanced learning problem adequately. Nevertheless, the F-measure metric combines the two previous metrics as a weighted focus on either recall or precision. Finally, the G-mean (Geometric mean) metric evaluates the balance of classification between the majority and minority classes. Even if the negative cases are accurately identified, a low G-Mean suggests poor performance in the classification of positive cases.

4 Results and Discussion

This benchmark involved multiple analyses of the implemented DA techniques throughout various datasets with various classifiers, such as Decision Trees, Random Forests, and MLPs (Multilayer Perceptrons). These machine learning algorithms were chosen based on their current scientific popularity and the typically good performance associated with them.

[5] It is also known as *F-score*.

[6] In this case, the positive class is considered the minority.

Regarding the synthetic data generated by all DA techniques, the experiments have produced a variety of findings. These analyses consist, mainly, of:

1. Comparing each feature's distribution throughout statistical methods;
2. Training the classifiers only with synthetic data;
3. Comparing different number of generated samples added to the real data;
4. Training the classifiers with real and synthetic data.

First, in order to compare the quality of the synthetic data generated by each augmentation technique, we performed statistical analyses on both real and synthetic data, i.e., to determine if they possess the same distribution. Ideally, and when talking about the same number of samples, the synthetic data should have properties very similar to the real one. Therefore, we implemented some statistical methods to compare each feature distribution on the real and synthetic datasets. However, due to the different behavior of continuous and categorical features, it was necessary to apply different statistical tests. On that account, the categorical features distributions were analyzed by the *chi-square test* and the continuous features by the *Kolmogorov-Smirnov test.*

These statistical tests showed that the DA techniques generally had difficulties representing the original continuous features distributions in the generated data. SMOTE was the technique with better representation, followed by ADASYN and the variations of VAE, namely VAE with K-means and VAE with K-means and decay on its loss. However, representing the categorical feature distributions was something easier for the DA techniques. Although SMOTE had a good capability to represent continuous features, it was the worst technique in regards to representing categorical features. As for the other techniques, ADASYN and the VAE with K-means and weighted decay on its loss were the ones more capable of producing similar distributions.

Before analyzing the classifiers' performance with synthetic data, it was observed how using a k-means to alter the latent space on a VAE could affect the synthesized samples, and if, as expected, the number of minority samples generated would be higher. Figure 4 demonstrates that the VAE with K-means, as expected, can produce way more minority samples through a 2D visualization of the target variable on the Breast Cancer dataset. Note that this dataset was chosen to analyze the effects of the utilization of K-means on the latent space of a VAE due to the lower number of samples on this dataset, which facilitates the visualization.

Although the DA choice is extremely important, the number of samples to generate is a crucial factor too. Therefore, we analyzed which ratio of synthesized samples provided the best results for the classifier performance. As a basis of comparison, we compared the various ratios to the classifiers' performance with no addition of synthetic data. The DA ratios are percentages of the size of the dataset for each problem, so when comparing a ratio of zero to one, we are comparing the classifier's performance when training with the original dataset versus a dataset with the same original data plus the same number of

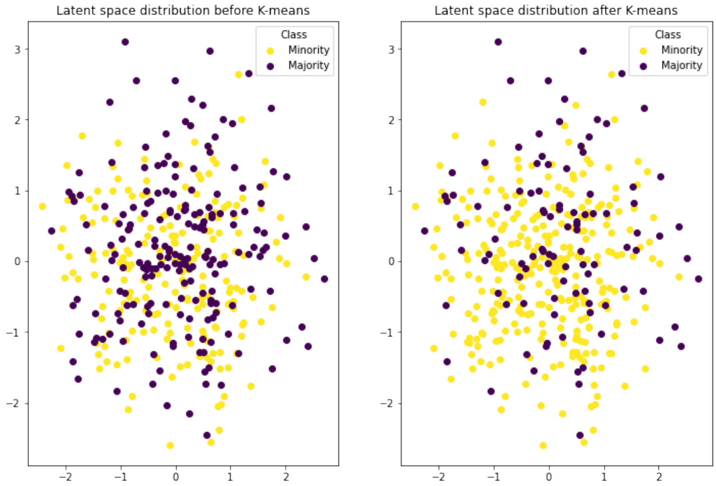


Fig. 4 Application of the K-means on the process of generating samples of VAE

synthetic samples. Another fact to take into account is that the metric values on the following plots are averages of the various classifiers used.

As seen in Figure 5, the adult dataset showed the best results with the SMOTE technique. The classifiers' performance, with the addition of synthetic data, tended to maintain or decrease a little throughout all DA techniques. However, SMOTE has the technique with better performance, making the minority class performance increase more with the addition of 25% of synthetic samples. As for the rest of the techniques, most of them showed difficulties in improving the classifiers' performance, except for the GAN.

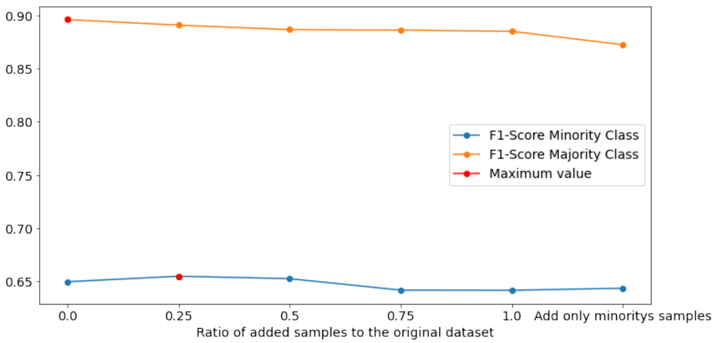


Fig. 5 Ratio of data added at the Adult dataset by SMOTE

On the contrary to the adult dataset, the addition of synthetic data to the Breast cancer dataset improved both classes classification performances as described at the Figure 6. Although the addition of 25% of synthetic samples increased by a lot the performance, the experiments showed that adding only minority samples was the best choice. The techniques that demonstrated better performance were the GAN and VAE with K-means.

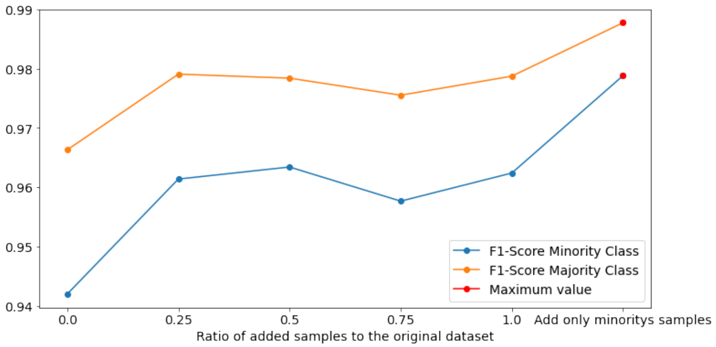


Fig. 6 Ratio of data added at the Breast dataset by GAN

As for the Credit Card Fraud dataset, the ratio performance, seen in Figure 7, increased more when adding only minority samples. The VAE with K-means was the technique that improved the training data for the classifiers, followed by the other VAE variations and the GAN. The overall results were a lot similar to the Breast dataset, since the ratio and techniques that achieved better performance were, in essence, the same ones. This could be explained by the properties of the two datasets that are very similar since they are only composed of continuous features.

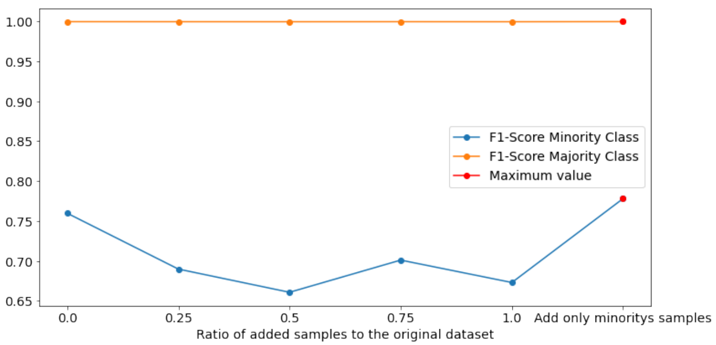


Fig. 7 Ratio of data added at the Credit Card Fraud dataset by VAE with K-means

Finally, for the Cerebral Stroke dataset, the results were also almost identical to the dataset with similar properties (Adult), as the best ratio of synthetic samples to add was 25% too, and the technique that resulted in a better performance by the classifiers was SMOTE. The results may be observed at Table 8.

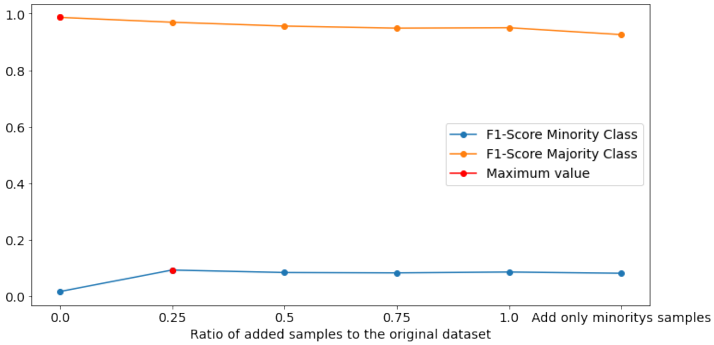


Fig. 8 Ratio of data added at the Cerebral Stroke dataset by SMOTE

In essence, this experiment showed that more categorical datasets could achieve greater performance by adding 25% of samples. As for the continuous datasets, the best ratio was to add only minority samples. The chosen techniques seemed to indicate a certain pattern.

We can still perform two additional crucial analyses to determine how reliable the generated data is after the analysis of the synthetic data properties. First, we are going to train the machine learning classifiers with real data and then compare the results with training with only synthetic data.

Synthetic data can achieve very similar results when replacing the real data in the classifier training. This experiment is described in Table 1 which represents the best DA techniques for each dataset on the Random Forest Classifier[7]. The main classifier chosen to analyze the synthetic data quality was due to the more consistent performance throughout all the techniques, as techniques such as VAE and GAN showed more volatility in the MLP and worse results on the simpler machine learning classifier, the Decision Tree.

In regards to the Adult dataset, the technique that, throughout all classifiers, performed better was the SMOTE, followed by ADASYN. This implies that these techniques may generate synthetic data with more quality than the rest of the techniques for this dataset, with more categorical features than continuous. One other important fact to take into account was the visible difference in the performance of one variation of the VAE when its generated data was trained by the MLP classifier, which can be observed at Table A1 in the appendix A. This classifier produced worse results than the other two more classical machine learning classifiers, and it was seen multiple times for the other datasets as well.

The Breast Cancer dataset had the VAE with K-means as the technique that generated the best quality of data, surpassing even the real data performance, described at Table A2. The Random Forest classifier attained a higher performance score, although it couldn't classify any minority classes correctly with GAN's synthetic data. This can be explained by the classifier overfitting when trained with those samples.

[7] In order to abbreviate the document, the full results will be present in the Appendix section A.

As for the Credit Card Fraud dataset, the SMOTE's synthetic data achieved superior results across all classifiers as can be seen at Table A3. It also surpassed the performance of the classifier with real data on both classes. Moreover, this dataset's classifiers were more susceptible to overfitting towards the majority class, principally with VAE variations and GAN.

Finally, in the Cerebral Stroke dataset, the clustering technique, GMM, and SMOTE were the techniques that had the closest results to the real data. GMM had a higher recall, capturing more minority samples than SMOTE, who had a superior performance in the majority class. This dataset suffered, like the previous ones, from the overfitting phenomenon in some of the generative DA techniques (VAE and GAN). Its results are described at Table A4.

These experiments demonstrated that most of the DA techniques can synthesize data in order to replace the real data in a somewhat efficient way. Note that in some cases, the use of only synthetic data as training data for the classifiers provided better results than with real data. Therefore, the use of only synthetic data could be very interesting in datasets where some data is sensitive and privacy matters.

Table 1 Classifiers performance comparison on only synthetic or real data training on the datasets

		Minority Class			**Majority Class**			
Dataset	**DA Technique**	**Precision**	**Recall**	**F1 Score**	**Precision**	**Recall**	**F1 Score**	**G-Mean**
Adult	No technique	0.705	0.6301	0.6655	0.8865	0.9164	0.9012	0.7599
	SMOTE	0.5373	0.7781	0.6356	0.918	0.7875	0.8478	0.7828
Breast Cancer	No technique	1.0	0.9063	0.9508	0.9474	1.0	0.973	0.9520
	VAE with K-means	1.0	0.9375	0.9677	0.9643	1.0	0.9818	0.9682
Credit Card Fraud	No technique	0.8824	0.6818	0.7692	0.9995	0.9998	0.9996	0.8257
	SMOTE	0.8947	0.7727	0.8293	0.9996	0.9998	0.9997	0.8790
Cerebral Stroke	No technique	0.2	0.0085	0.0163	0.982	0.9994	0.9901	0.0920
	GMM	0.0375	0.9576	0.0722	0.9986	0.5462	0.7062	0.7232

Note: The results are from the classifier Random Forest and the DA technique chosen was the technique that achieved the best scores.

With those results in mind, the following analyses focus on training the classifiers with an increased amount of data throughout multiple datasets[8], described in Table 2. These results demonstrated two important points.

First, the choice of the number of samples to generate is crucial, as the classifier's performance may decrease if the final dataset has too many synthetic samples with less quality than the real data. Therefore, for each problem, it was necessary to evaluate the number of samples to synthesize. In these experiments, the adult dataset had better behavior by adding 25% more samples to the original dataset, while the rest of the datasets had higher performance by adding only minority samples. Note that while the best ratio when using only synthetic data (described in Figure 8) was 25%, this was not the same

[8] For the same reason as was previously indicated in Table 1, take note that the results in Table 2 are just for the Random Forest classifier.

when combining real and synthetic data for the Cerebral Stroke dataset, which achieved a higher score when adding only minority samples.

Second, the combination of real and synthetic data improved the classifiers' performance. In the adult dataset, SMOTE and GAN were the techniques that incremented most of the metrics for the minority class while not decreasing the majority class performance. In the Breast Cancer dataset, the results were even more satisfactory, with an increase of 5% in the minority class f1-score. The Variational Autoencoder with K-means and decay did a remarkable job since it refined the performance for each class to a round 100%. The Credit Card Fraud dataset had similar results as the previous one, with a boost in the f1-score minority class by 8% in the VAE with K-means. Finally, in the Cerebral Stroke dataset, the minority results increased 8 times the initial results with no synthetic samples on the f1-score as well as by the technique SMOTE.

Moreover, these experiments permitted us to confirm some interesting facts that were mentioned previously. Datasets that contained mainly categorical features were usually associated with a difficulty in increasing the classifier results by adding new samples. Plus, the technique that adapted best to those kinds of datasets was the more classical one, SMOTE. On the other hand, for the datasets with more continuous features, the variations of the Variational Autoencoder, mainly VAE with K-means and VAE with K-means and decay, had very good performances.

5 Conclusion

In this study, we went through a benchmark of different DA techniques in multiple datasets of various domains. With the results obtained during the experiments, we can now answer the main questions previously stated in this paper.

First, we found that DA could improve the results of imbalanced data problems, generating samples of the minority class or both classes (RQ1). This was observed in all datasets with multiple DA techniques.

Furthermore, with the results found during the development of this study, it was also seen that some dataset properties could influence the quality of the synthetic data generated (RQ2). Datasets that contained more categorical features were usually associated with an increased difficulty in increasing the classifiers' performance or even a decrease in the minority class classification when adding synthetic data to the original dataset. This phenomenon is visible in problems such as the Adult and Cerebral Stroke datasets, in which DA techniques such as the VAE variations and GANs had difficulties when generating synthetic data. On the other hand, other techniques (SMOTE, ADASYN, and, in some cases, GMM) generated quality synthetic samples easily. However, datasets whose features were mainly continuous had very good performances for techniques such as VAE and GAN, while maintaining good performances for SMOTE and ADASYN (even if lower than the other more complex techniques)

Table 2 Random Forest performance throughout the multiple DA techniques

		Minority Class			Majority Class			
Datasets	**DA Technique**	**Precision**	**Recall**	**F1 Score**	**Precision**	**Recall**	**F1 Score**	**G-Mean**
Adult[1]	No technique	0.705	0.6301	0.6655	0.8865	0.9164	0.9012	0.7599
	SMOTE	0.6801	0.6582	0.669	0.8927	0.9019	0.8973	**0.7704**
	ADASYN	0.6709	0.6259	0.6476	0.8838	0.90267	0.8932	0.7516
	GMM	0.7067	0.625	0.6634	0.8853	0.9178	**0.9012**	0.7574
	VAE with Decay	0.7003	0.6318	0.6643	0.8868	0.9143	0.9003	0.76
	VAE with K-means	0.7013	0.631	0.6643	0.8866	0.9148	0.9005	0.7597
	VAE with K-means and Decay	0.6999	0.6327	0.6646	0.887	0.914	0.9003	0.7604
	GAN	0.7018	0.6403	**0.6696**	0.889	0.9137	**0.9012**	0.7649
Breast Cancer[2]	No technique	1.0	0.9063	0.9508	0.9474	1.0	0.973	0.9520
	SMOTE	1.0	0.9688	0.9841	0.9818	1.0	0.9908	0.9843
	ADASYN	1.0	0.9688	0.9841	0.9818	1.0	0.9908	0.9843
	GMM	1.0	0.9375	0.9677	0.9643	1.0	0.9818	0.9682
	VAE with Decay	1.0	0.9375	0.9677	0.9643	1.0	0.9818	0.9682
	VAE with K-means	1.0	0.9375	0.9677	0.9643	1.0	0.9818	0.9682
	VAE with K-means and Decay	1.0	1.0	**1.0**	1.0	1.0	**1.0**	**1.0**
	GAN	1.0	0.9688	0.9841	0.9843	1.0	0.9908	0.9843
Credit Card Fraud[2]	No technique	0.8823	0.6818	0.7692	0.9995	0.9998	0.9996	0.8257
	SMOTE	0.8947	0.7727	0.8293	0.9996	0.9998	0.9997	**0.879**
	ADASYN	0.9412	0.7273	0.8205	0.9995	0.9999	0.9997	0.8528
	GMM	0.9412	0.7273	0.8205	0.9995	0.9999	0.9997	0.8528
	VAE with Decay	0.85	0.7727	0.8095	0.9996	0.9998	0.9997	0.8789
	VAE with K-means	0.9444	0.7727	**0.85**	0.9996	0.9999	**0.9998**	**0.879**
	VAE with K-means and Decay	0.8824	0.6818	0.7692	0.9995	0.9998	0.9996	0.8257
	GAN	0.8824	0.6818	0.7692	0.9995	0.9998	0.9996	0.8257
Cerebral Stroke[2]	No technique	0.2	0.0085	0.0163	0.982	0.9994	0.9906	0.092
	SMOTE	0.0486	0.2373	**0.0807**	0.9848	0.9143	0.9482	**0.4658**
	ADASYN	0.0531	0.1441	0.0776	0.9837	0.9526	0.9679	0.3705
	GMM	0.0333	0.0085	0.0135	0.9819	0.9955	0.9887	0.0918
	VAE with Decay	0.1014	0.0593	0.0749	0.9828	0.9903	0.9865	0.2424
	VAE with K-means	0.0986	0.0593	0.0741	0.9828	0.99	0.9864	0.2423
	VAE with K-means and Decay	0.125	0.0085	0.0159	0.982	0.9989	0.9904	0.092
	GAN	0.2	0.0085	0.0163	0.982	0.9994	**0.9906**	0.092

[1] The DA techniques added 25% more synthetic samples to the dataset.

[2] The DA techniques added only minority samples to the dataset.

Relatively to the number of samples to be generated, the dataset properties also impact how to choose that parameter (RQ3). As described before, more categorical datasets achieved a greater performance when adding 25% more samples to the original dataset, while datasets whose major features are continuous should opt to add only minority samples.

Finally, these experiments helped us to determine which DA techniques provide a better quality of synthetic tabular data (RQ4). The implementation of the VAE with K-means had very good results, mainly in the minority class, as expected. Therefore, it could be chosen as a good candidate to be used as a DA technique in a dataset whose features are mostly categorical. In essence, GAN and VAE can achieve amazing results when explored and tuned for specific domains, which is one of the principal characteristics of these kinds of algorithms. Moreover, the results demonstrated that, in general, SMOTE had very good performance when generating synthetic data. As a result, SMOTE seems to be the ideal technique when the dataset properties are not taken into account, since it is very consistent in its generated samples. The GMM, the more unusual technique, showed some promise for further investigation, despite appearing to have a higher volatility in sample quality.

Although this study helped us to use DA with a lot more knowledge of how to do it and which techniques to choose, there were some obstacles during its development. The main complication was the strange overfit seen with synthetic data from techniques such as VAE and GAN. The VAE with K-means could suffer a loss in its process since the unsupervised algorithm could choose incorrectly the minority case, focusing the VAE on generating the majority class. As for the GAN, some results suggest that the noise added to the discriminator network was too high. Therefore, for that specific problem, adjusting that value could surpass the quality of data generated. Moreover, the results also suggest that the machine learning classifiers are susceptible to a greater or lower type of overfit depending on the quality of the data.

Authors' contributions: All authors contributed to the study conception and design. Material preparation, data collection and analysis were performed by Pedro Machado and Bruno Fernandes. The first draft of the manuscript was written by Pedro Machado and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Funding: This work is financed by National Funds through the Portuguese funding agency, FCT - Fundação para a Ciência e a Tecnologia, within the R&D Units project UIDB/00319/2020 and DSAIPA/AI/0099/2019.

Conflict of interest: The authors declare no conflict of interest.

Ethics approval: Not applicable.

Consent to participate: Not applicable.

Consent for publication: Not applicable.

Availability of data and material: This paper contains the vast majority of the data.

Code availability: The code will be available later on GitHub.

Appendix A Supplementary results

Table A1 Classifiers performance comparison on only synthetic or real data training throughout different DA techniques on the Adult dataset

		Minority Class			**Majority Class**			
Classifiers	**DA Technique**	**Precision**	**Recall**	**F1 Score**	**Precision**	**Recall**	**F1 Score**	**G-Mean**
Decision Tree	No technique	0.6492	0.6122	0.6302	0.8792	0.8951	0.8871	0.7403
	SMOTE	0.4928	0.7032	**0.5795**	0.8912	0.7706	**0.8265**	**0.7361**
	ADASYN	0.3726	0.5595	0.4473	0.8339	0.7013	0.7619	0.6264
	GMM	0.2729	0.4456	0.3385	0.7801	0.6236	0.6931	0.5271
	VAE with Decay	0.2453	0.7815	0.3734	0.7744	0.2378	0.3639	0.4311
	VAE with K-means	0.3168	0.2389	0.2724	0.7761	0.8366	0.8052	0.4471
	VAE with K-means and Decay	0.4845	0.2398	0.3208	0.7922	0.9191	0.851	0.4695
	GAN	0.2461	0.9974	0.3948	0.9748	0.03128	0.0606	0.1766
Random Forest	No technique	0.705	0.6301	0.6655	0.8865	0.9164	0.9012	0.7599
	SMOTE	0.5373	0.7781	**0.6356**	0.918	0.7875	0.8478	**0.7828**
	ADASYN	0.4554	0.5298	0.4898	0.8428	0.7991	0.8204	0.6507
	GMM	0.2724	0.4515	0.34	0.7803	0.6177	0.6895	0.5281
	VAE with Decay	0.4101	0.227	0.2923	0.7853	0.8965	0.8372	0.4511
	VAE with K-means	0.5205	0.216	0.3053	0.7903	0.9369	0.8574	0.4498
	VAE with K-means and Decay	0.5932	0.2083	0.3084	0.7918	0.9547	**0.8657**	0.4460
	GAN	0.2472	0.9949	0.396	0.9605	0.0394	0.0756	0.1979
MLP	No technique	0.712	0.6012	0.6519	0.8795	0.9229	0.9007	0.7449
	SMOTE	0.5098	0.8376	**0.6338**	0.9353	0.7447	0.8292	**0.7898**
	ADASYN	0.4846	0.7211	0.5796	0.8954	0.7568	0.8203	0.7387
	GMM	0.2708	0.449	0.3378	0.7792	0.6166	0.6884	0.5262
	VAE with Decay	0.2623	0.9014	0.4063	0.8624	0.196	0.3194	0.4203
	VAE with K-means	0.9539	0.1233	0.2184	0.7822	0.9981	**0.877**	0.3508
	VAE with K-means and Decay	0.0	0.0	0.0	0.7593	1.0	0.8632	0.0
	GAN	0.2553	0.9845	0.4055	0.9484	0.0892	0.1631	0.2964

Table A2 Classifiers performance comparison on only synthetic or real data training throughout different DA techniques on the Breast Cancer dataset

		Minority Class			**Majority Class**			
	DA Technique	**Precision**	**Recall**	**F1 Score**	**Precision**	**Recall**	**F1 Score**	**G-Mean**
Decision Tree	No technique	0.875	0.875	0.875	0.9259	0.9259	0.9259	0.9001
	SMOTE	0.8611	0.9688	0.9118	0.98	0.9074	0.9423	0.9376
	ADASYN	0.8823	0.9375	0.9091	0.9615	0.9259	0.9434	0.9317
	GMM	1.0	0.1875	0.3158	0.675	1.0	0.806	0.4330
	VAE with Decay	0.6571	0.7188	0.6866	0.8235	0.7778	0.8	0.7477
	VAE with K-means	0.6512	0.875	0.7467	0.907	0.7222	0.8041	0.7949
	VAE with K-means and Decay	0.8846	0.8519	0.8679	0.7647	0.8125	0.7879	0.8319
	GAN	0.0	0.0	0.0	0.6279	1.0	0.7714	0.0
Random Forest	No technique	1.0	0.9063	0.9508	0.9474	1.0	0.973	0.952
	SMOTE	0.9677	0.9375	0.9523	0.9636	0.9815	0.9725	0.9592
	ADASYN	1.0	0.9063	0.9508	0.9474	1.0	0.973	0.9520
	GMM	1.0	0.2813	0.439	0.7013	1.0	0.8244	0.5303
	VAE with Decay	0.7561	0.9688	0.8493	0.9778	0.8148	0.8889	0.8885
	VAE with K-means	1.0	0.9375	0.9677	0.9643	1.0	0.9818	0.9682
	VAE with K-means and Decay	0.8857	0.9688	0.9254	0.9804	0.9259	0.9524	0.9471
	GAN	0.0	0.0	0.0	0.6279	1.0	0.7714	0.0
MLP	No technique	1.0	1.0	1.0	1.0	1.0	1.0	1.0
	SMOTE	0.9412	1.0	0.9697	1.0	0.963	0.9811	0.9813
	ADASYN	0.9063	0.9063	0.9063	0.9444	0.9444	0.9444	0.9252
	GMM	0.0	0.0	0.0	0.6279	1.0	0.7714	0.0
	VAE with Decay	0.871	0.8438	0.8571	0.9091	0.9259	0.9174	0.8839
	VAE with K-means	0.9355	0.9063	0.9206	0.9455	0.963	0.9541	0.9342
	VAE with K-means and Decay	0.8529	0.9063	0.8788	0.9423	0.9074	0.9245	0.9068
	GAN	0.7879	0.8125	0.8	0.887	0.8704	0.8785	0.8409

Table A3 Classifiers performance comparison on only synthetic or real data training throughout different DA techniques on the Credit Card Fraud dataset

		Minority Class			**Majority Class**			
	DA Technique	**Precision**	**Recall**	**F1 Score**	**Precision**	**Recall**	**F1 Score**	**G-Mean**
Decision Tree	No technique	0.64	0.7273	0.6809	0.9995	0.9993	0.9994	0.8525
	SMOTE	0.3137	0.7273	0.4384	0.9995	0.9973	0.9984	0.8516
	ADASYN	0.0068	0.8182	0.0134	0.9996	0.7937	0.8849	0.8059
	GMM	0.0031	0.8636	0.0061	0.9995	0.515	0.6798	0.6669
	VAE with Decay	0.2727	0.4091	0.3273	0.999	0.9981	0.9986	0.639
	VAE with K-means	0.8333	0.2273	0.3571	0.9987	0.9999	0.9993	0.4767
	VAE with K-means and Decay	0.0	0.0	0.0	0.9983	1.0	0.9991	0.0
	GAN	0.0	0.0	0.0	0.9983	1.0	0.9991	0.0
Random Forest	No technique	0.8824	0.6818	0.7692	0.9995	0.9998	0.9996	0.8257
	SMOTE	0.8947	0.7727	0.8293	0.9996	0.9998	0.9997	0.879
	ADASYN	0.0215	0.9091	0.042	0.9998	0.9288	0.963	0.9189
	GMM	0.0026	0.8636	0.0052	0.9995	0.4335	0.6047	0.6119
	VAE with Decay	0.8	0.3636	0.5	0.9989	0.9998	0.9994	0.603
	VAE with K-means	0.0	0.0	0.0	0.9983	1.0	0.9991	0.0
	VAE with K-means and Decay	0.0	0.0	0.0	0.9983	1.0	0.9991	0.0
	GAN	0.0	0.0	0.0	0.9983	1.0	0.9991	0.0
MLP	No technique	0.8947	0.7727	0.8293	0.9996	0.9998	0.9997	0.879
	SMOTE	0.9998	0.9971	0.9984	0.3393	0.8636	0.4872	0.928
	ADASYN	0.0111	0.7727	0.022	0.9996	0.882	0.9371	0.8256
	GMM	0.0025	0.7727	0.005	0.9992	0.4717	0.6408	0.6037
	VAE with Decay	0.9	0.4091	0.5625	0.999	0.9999	0.9995	0.6396
	VAE with K-means	0.0	0.0	0.0	0.9983	1.0	0.9991	0.0
	VAE with K-means and Decay	0.0	0.0	0.0	0.9983	1.0	0.9991	0.0
	GAN	0.0	0.0	0.0	0.9983	1.0	0.9991	0.0

Table A4 Classifiers performance comparison on only synthetic or real data training throughout different DA techniques on the Cerebral Stroke dataset

		Minority Class			**Majority Class**			
	DA Technique	**Precision**	**Recall**	**F1 Score**	**Precision**	**Recall**	**F1 Score**	**G-Mean**
Decision Tree	No technique	0.0321	0.0424	0.0365	0.9822	0.9764	0.9793	0.2034
	SMOTE	0.0426	0.3136	0.075	0.9856	0.8699	0.9241	0.5223
	ADASYN	0.0127	0.3644	0.0245	0.976	0.4761	0.64	0.4165
	GMM	0.0379	0.9322	0.0728	0.9978	0.5631	0.7199	0.7245
	VAE with Decay	0.0217	0.178	0.0386	0.9825	0.8517	0.9124	0.3893
	VAE with K-means	0.0208	0.0085	0.012	0.9819	0.9926	0.9872	0.0917
	VAE with K-means and Decay	0.0159	0.0085	0.0111	0.9819	0.9903	0.9861	0.0916
	GAN	0.0	0.0	0.0	0.9819	1.0	0.9909	0.0
Random Forest	No technique	0.2	0.0085	0.0163	0.982	0.9994	0.9906	0.092
	SMOTE	0.0461	0.2627	0.0784	0.9851	0.8996	0.9404	0.4861
	ADASYN	0.0129	0.339	0.0248	0.9771	0.5199	0.6787	0.4198
	GMM	0.0375	0.9576	0.0722	0.9986	0.5462	0.7062	0.7232
	VAE with Decay	0.0	0.0	0.0	0.9819	0.9986	0.9902	0.0
	VAE with K-means	0.0	0.0	0.0	0.9819	0.9986	0.9902	0.0
	VAE with K-means and Decay	0.0	0.0	0.0	0.9819	0.9986	0.9902	0.0
	GAN	0.0	0.0	0.0	0.9819	0.9986	0.9902	0.0
MLP	No technique	0.0	0.0	0.0	0.9819	1.0	0.9909	0.0
	SMOTE	0.0518	0.5508	0.0948	0.9899	0.814	0.8934	0.6696
	ADASYN	0.0166	0.3475	0.0318	0.981	0.621	0.7605	0.4645
	GMM	0.0361	0.9915	0.0696	0.9997	0.5107	0.6761	0.7116
	VAE with Decay	0.0	0.0	0.0	0.9819	1.0	0.9909	0.0
	VAE with K-means	0.0	0.0	0.0	0.9819	1.0	0.9909	0.0
	VAE with K-means and Decay	0.0	0.0	0.0	0.9819	1.0	0.9909	0.0
	GAN	0.0	0.0	0.0	0.9819	1.0	0.9909	0.0

References

Brownlee, J. (2019, 07). *A gentle introduction to generative adversarial networks (gans).* Retrieved from https://machinelearningmastery.com/what-are-generative-adversarial-networks-gans/

Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P. (2002). Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, *16*, 321–357.

Fernandes, B., Silva, F., Alaiz-Moretón, H., Novais, P., Analide, C., Neves, J. (2019). Traffic flow forecasting on data-scarce environments using arima and lstm networks. Á. Rocha, H. Adeli, L.P. Reis, & S. Costanzo (Eds.), *New knowledge in information systems and technologies* (pp. 273–282). Cham: Springer International Publishing.

Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., … Bengio, Y. (2014). *Generative adversarial networks.*

He, H., Bai, Y., Garcia, E.A., Li, S. (2008). Adasyn: Adaptive synthetic sampling approach for imbalanced learning. *2008 ieee international joint conference on neural networks (ieee world congress on computational* *intelligence)* (pp. 1322–1328). 10.1109/IJCNN.2008.4633969

He, H., & Garcia, E.A. (2009). Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, *21*(9), 1263–1284.

10.1109/TKDE.2008.239

Kingma, D.P., & Welling, M. (2019). An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, *12*(4), 307–392.

10.1561/2200000056

Kohavi, R., & Becker, B. (1994). *UCI machine learning repository.* Retrieved from https://archive.ics.uci.edu/ml/datasets/adult

Li, W., Fan, L., Wang, Z., Ma, C., Cui, X. (2021). Tackling mode collapse in multi-generator gans with orthogonal vectors. *Pattern Recognition*, *110*, 107646.

10.1016/j.patcog.2020.107646

Liu, T., Fan, W., Wu, C. (2019). *Data for: A hybrid machine learning approach to cerebral stroke prediction based on imbalanced medical-datasets* (Vol. 1). https://doi:10.17632/x8ygrw87jw.1

Lucas, J., Tucker, G., Grosse, R.B., Norouzi, M. (2019). Understanding posterior collapse in generative latent variable models. *Dgsiclr.*

Lv, J.-J., Shao, X.-H., Huang, J.-S., Zhou, X.-D., Zhou, X. (2017). Data augmentation for face recognition. *Neurocomputing*, *230*, 184–196.

10.1016/j.neucom.2016.12.025

Machado, P., Fernandes, B., Novais, P. (2022). Benchmarking data augmentation techniques for tabular data. Submitted for publication in IDEAL.

McLachlan, G.J., & Krishnan, T. (2007). *The em algorithm and extensions* (Vol. 382). John Wiley & Sons.

Nalepa, J., Marcinkiewicz, M., Kawulok, M. (2019). Data augmentation for brain-tumor segmentation: A review. *Frontiers in Computational Neuroscience*, *13*, 83.

10.3389/fncom.2019.00083

Perez, F., Vasconcelos, C., Avila, S., Valle, E. (2018). Data augmentation for skin lesion analysis. D. Stoyanov et al. (Eds.), *Or 2.0 context-aware operating theaters, computer assisted robotic endoscopy, clinical image-based procedures, and skin image analysis* (pp. 303–311). Cham: Springer International Publishing.

Rocca, J. (2021, 03). *Understanding variational autoencoders (vaes).* Towards Data Science. Retrieved from https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73 (Last Visited November 23, 2021)

Sarkar, T. (2019, 9). *How to use a clustering technique for synthetic data generation.* Towards Data Science. Retrieved from https://towardsdatascience.com/how-to-use-a-clustering-technique-for-synthetic-data-generation-7c84b6b678ea (Last Visited November 14, 2021)

Shao, M., Gu, N., Zhang, X. (2020). Credit card transactions data adversarial augmentation in the frequency domain. *2020 5th ieee international conference on big data analytics (icbda)* (pp. 238–245). 10.1109/ICBDA49040.2020.9101344

Singh, A., & Ogunfunmi, T. (2021, Dec). An overview of variational autoencoders for source separation, finance, and bio-signal applications. *Entropy*, *24*(1), 55.

10.3390/e24010055

ULB, M.L.G. (2018, Mar). *Credit card fraud detection.* Retrieved from https://www.kaggle.com/mlg-ulb/creditcardfraud

van Dyk, D.A., & Meng, X.-L. (2001). The art of data augmentation. *Journal of Computational and Graphical Statistics*, *10*, 1–50.

Wolberg, W.H., Street, W.N., Mangasarian, O.L. (1998). *UCI machine learning repository.* Retrieved from https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29