On Oversampling via Generative Adversarial Networks under Different Data Difficulty Factors

Ehsan Nazari EHSAN.NAZARI@UOTTAWA.CA
Paula Branco PBRANCO@UOTTAWA.CA
School of Electrical Engineering and Computer Science, University of Ottawa
Ottawa, Ontario, Canada

Editors: Nuno Moniz, Paula Branco, Luís Torgo, Nathalie Japkowicz, Michał Woźniak and Shuo Wang.

Abstract

Over the last two decades, several approaches have been proposed to tackle the class imbalance problem which is characterized by the inability of a learner to focus on a relevant but scarcely represented class. The generation of synthetic examples to oversample the training set and thus force the learner to focus on the important cases is one of such solutions. Recently, generative adversarial networks (GANs) started to be explored as an oversampling alternative due to their capability of generating samples from an implicit distribution.

Still, data difficulty factors such as class overlap, data dimensionality or sample size, and were shown to also negatively impact the learners performance under an imbalance setting. The ability of GANs to deal with the imbalance problem and other data difficulty factors has not yet been assessed. The main goal of this paper is to understand how data difficulty factors impact the performance of GANs when they are used as an oversampling method. Namely, we study the performance of conditioned GANs (CGANs) in an image dataset with controlled levels of the following data difficulty factors: sample size, data dimensionality, class overlap and imbalance ratio. We show that CGANs are effective for tackling tasks with multiple data difficulty factors, exhibiting increased gains on the most difficult tasks.

Keywords: Conditional Generative Adversarial Network, Class Imbalance, Small Sample Size, Data Dimensionality, Class Overlap

1. Introduction

One of the main issues predictive models face in real world datasets, is the class imbalance problem, where severe performance losses occur due to the poor representativeness of one important class. This is a well studied problem for which several types of solutions have been proposed including pre-processing, post-processing and special purpose learning methods (Branco et al., 2016). Pre-processing methods are among the most versatile and thoroughly explored strategies to deal with this problem. These methods modify the training data distribution in order to bias the learner focus towards the most important and scarcely represented class. By acting before the learning stage, they enable the application of any out of the box learning algorithm, which motivates their popularity.

Many alternatives have been put forward for generating synthetic cases as a way to effectively change the training distribution. Generative modeling is among the proposed solutions to tackle the class imbalance problem (Ngwenduna and Mbuvha, 2021). Among generative models, Generative Adversarial Networks (GANs) have recently been explored as a data augmentation strategy. GANs have emerged as a way to generate realistic images (e.g. Goodfellow et al. (2014); Radford et al. (2015)) and have since been applied to a diversity of tasks. Although being still an under-explored direction (Sampath et al., 2021), approaches using GANs for tackling the class imbalance problem have been put forward (e.g. Douzas and Bacao (2018); Suh et al. (2021)).

Despite the relevance of the class imbalance problem, it was shown that, in a real world setting, several other data characteristics also severely impact the learners performance (e.g. López et al. (2013); Brzezinski et al. (2021); Branco and Torgo (2019)). Data difficulty factors, such as the presence of class overlap and the small sample size, raise obstacles to the performance and typically coexist with the imbalance problem. In particular, the minority class recognition is heavily affected when these factors are present in imbalanced datasets (Napierała et al., 2010). Still, the impact of these data difficulty factors when carrying out oversampling via GANs has not yet been studied.

The main goal of this study is to understand how other data difficulty factors present in imbalanced datasets affect the performance of upsampling strategies utilizing GANs. In particular, we focus on understanding the influence in the performance exerted by the imbalance ratio, sample size, data dimensionality, and class overlap when using a GAN architecture for data augmentation. To this end, we selected to apply conditioned GANs (CGANs) which were shown to improve the classification performance under imbalanced domains (Douzas and Bacao, 2018). We focus our experiments on a well-know image dataset (MNIST), for which we generated multiple versions with different levels of the data difficulty factors to observe their effect in a controlled way. Our main contributions are: (i) a repository with 144 datasets containing different levels of data difficulty factors; (ii) an extensive experimental study analysing the impact of each data difficulty factor as well as the combined impact of two factors in the performance of CGANs; and (iii) a repository with all the code used in our experiments to allow the reproducibility and extension of this work.

2. Related Work

Class imbalance is a critical issue for many real world applications. Pre-processing methods are a common approach to this problem whose key idea is to modify the training set distribution to force the learning algorithm to focus on the most relevant cases. The generation of synthetic minority class instances has been used as a pre-processing solution to balance the training set and this way improve the performance on the minority class.

Multiple data pre-processing alternatives have been considered that either augment the minority class or reduce the majority class cases. A popular approach is the generation of new minority class cases through the generation of synthetic examples. Strategies such as SMOTE (Chawla et al., 2002), Borderline-SMOTE (Han et al., 2005) or ADASYN (He et al., 2008) have been broadly used to face the imbalance issue. These methods use the existing minority class cases to obtain new cases by interpolating them and potentially adding a particular bias in this generation depending on the proposal.

Recently, GANs have emerged as an alternative upsampling solution. GANs are a class of generative models based on a game theory scenario that use deep learning methods to train two competing models: a generator and a discriminator. The generator is trained to generate new examples, while the discriminator's goal is to distinguish between the real and the generated cases. The two models are trained together in an adversarial game.

GANs have been used with success in several real-world applications including, computer vision, natural language processing and other fields. Although exhibiting success, their training procedure is known to be unstable and slow. To this end, Wasserstein GAN (WGAN) (Arjovsky et al., 2017) was proposed to face the stability problem. WGAN uses the Wasserstein distance, also known as the earthmover (EM) distance, as a cost function, which is a more suitable metric for measuring the distance between two distributions as it does not suffer from the vanishing gradients issue.

A straightforward output of GANs is the generation of synthetic data that approximates a given real data distribution. Considering that, in an imbalanced context, a key approach is the generation of class-conditioned examples, a straightforward solution is to use supervised GAN models such as conditional GANs (CGAN) (Mirza and Osindero, 2014). The key idea of this solution is the conditioning of the training process on the class labels for the classifier. This is achieved by simultaneously training a generative model and a fine-grained classifier. Different conditional GAN alternatives have emerged. For instance, auxiliary classifier GANs (ACGAN) (Odena et al., 2017) are an extension of class-conditional GANs that provide varying amounts of control when carrying out image generation, while classification enhancement GAN (CEGAN) (Suh et al., 2021) incorporates three networks (a generator, a discriminator and a classifier) to generate minority class examples under the WGAN-GP (Gulrajani et al., 2017) objective formulation. Douzas and Bacao (2018) showed that CGANs are able to generate data that improves the learning algorithm's performance in an imbalanced context. Still, conditional GANs, such as CGAN and ACGAN, face challenges in practice given the instability of the training process.

Overall, the use of GANs to specifically tackle the class imbalance problem by balancing the training data distribution is still a recent idea, whose applicability has not been widely tested, existing with only a few proposals that assess the effectiveness of these solutions for this problem (Sampath et al., 2021). Specific modifications of GANs were proposed in order to use them as an oversampling strategy through the incorporation or combination of other methods in the GAN architecture. For instance, to face the class imbalance problem, the application of evolutionary algorithms (e.g. Hao et al. (2020)) and the incorporation of other generative models, such as auto-encoders, was proposed (e.g. Mariani et al. (2018); Antoniou et al. (2017); Deepshikha and Naman (2020); Guo et al. (2019)).

Still, the potential of using GANs for dealing with the class imbalance is both under-explored and not well understood (Sampath et al., 2021). Moreover, to the best of our knowledge, the co-existence of other data difficulty factors, which is a frequently arising scenario in imbalanced real-world problems, has not yet been investigated. Our goal is to assess the performance of a CGAN (Mirza and Osindero, 2014) under different settings that may hinder the performance on binary imbalanced learning tasks. We will specifically study the following factors: (1) using different number of features per sample, (2) different levels of class overlap, (3) varying imbalance ratios, and (4) different sample sizes.

3. Experimental Evaluation

We conduct several experiments to shed light on different aspects of augmentative upsampling via a CGAN. Learning from an imbalanced domains is our main context. We define and generate data with multiple data difficulty factors, each of which controlling a different problem usually coexisting with class imbalanced problems. In each experiment, we train a classifier with the original imbalanced dataset, and with a dataset balanced with the help of a CGAN. Our goal is to assess the impact in the performance when carrying out upsampling with CGAN under the different data characteristics.

3.1. Datasets

We selected the MNIST dataset (LeCun and Cortes, 2010) as the base dataset for our experiments. Each MNIST case is a 28 pixels by 28 pixels gray-scale image of a hand written digit. Our goal is to generate multiple datasets with varying degrees of the following characteristics: (i) sample size; (ii) class overlap; (iii) imbalance ratio; and (iv) number of features. This allows us to control the specific difficult factors present in the data that we aim at studying. We will now explain the data generation process that we used to ensure that these characteristics are present in each one of them.

The original training set of the MNIST dataset contains 5842 instances of the number 'four'. We selected the digits of number 'four' of MNIST as the base majority class cases for all the generated datasets. We set the sample size of a dataset using the majority class as the main reference. Without changing the order of the 5842 instances of the number 'four', we selected the first 1000, 400, 200, and 100 instances of this class. These 4 sets will constitute the majority class cases of 4 base datasets with varying sample size. These datasets will be combined with the minority class cases through different combinations of characteristics as we will explain below.

In order to control the class overlap we decided to applied a transformation to the initial majority class images to generate the minority class. To obtain different levels of class overlap we generated three levels of rotation of the digits of the original number 'four'. Thus, our minority class cases are obtained by applying a rotation of 30 degrees, 45 degrees, or 90 degrees to the original majority class image. The premise is that, if the minority class images are a result of the majority class images by small degrees, it will be harder for the classifier to distinguish between them, i.e., we consider that we achieve a higher class overlap when using the original images and their rotations by 30 degrees than when using their rotations by 90 degree. Therefore, it will be comparatively easy for a model to distinguish between a normal four and a 90 degrees of rotation while it will be harder to distinguish the original image from an image rotated by 45 degrees. The binary classification of a image with a normal 'four' and an image with a 30 degrees of rotation will be hardest of the three. The image rotation is done via PIL library in Python with its default settings. We can observe examples of the minority class cases generated by different rotations in Figure 1. In this figure, the left column shows one majority class case represented with different number of features. We explain below how the majority class cases are obtained. The other three columns show the corresponding generated minority class cases for the three rotations defined.

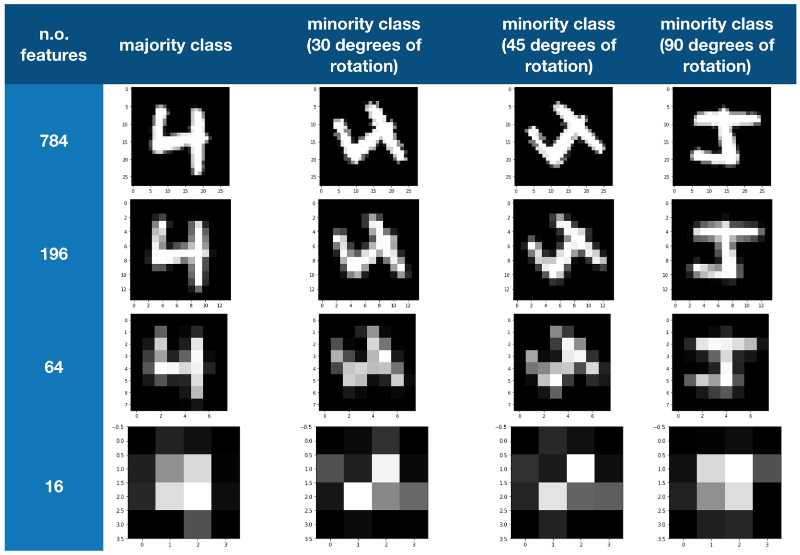

Figure 1: Illustration of the generation of different image sizes and rotations applied to an original MNIST dataset image in the top left corner.

We also created datasets with different imbalance ratios. We adopted in this paper the definition of imbalance ratio as the number of minority class images divided by the number majority class images. Three different imbalance ratios (0.4, 0.2, 0.1) are applied to the minority class. This is achieved by observing the total number of majority class cases and multiplying this number by the selected imbalance ratio to obtain the total number of minority class cases to add to the dataset. This data difficulty factor is set to discover the correlation between imbalance ratio and the CGAN's ability to upsample the minority class while improving the learner performance.

Finally, we generated datasets with different number of features by resizing into smaller sizes the original 28×28 images. Therefore, for each dataset we generated a resized version to 14×14, 8×8, and 4×4 pixels. Image resizing is done via open-cv library in Python through the INTER AREA interpolation algorithm. The main goal of reducing the original images size is to obtain a representation of the classes with a smaller number of features. In effect, the procedure described allows us to obtain datasets with 784, 196, 64 and 16 features. The study of this data difficulty factor will give us a better understanding of the effect of the number of features on CGAN's efficiency when addressing the imbalance problem. We hypothesize that a strong reduction in the number of features made available for the CGAN may be a major challenge. We can observe examples of the different sizes of the generated minority class cases in Figure 1. Table 1 summarizes all the described settings used to generate each one of the data difficulty characteristics. Overall, we generated a total of 144 ($4 \times 3 \times 3 \times 4$) different datasets by generating all combinations of the configurations for each characteristic. To ensure the reproducibility of our research, all the generated datasets are freely available at https://github.com/enazari/GAN-upsampling-LIDTA21.

Table 1: Values and brief explanation of the different data difficulty factors used.

Data Difficulty Factor	values used	explanation
Sample Size	$\{1000; 400; 200; 100\}$	Set by fixing the majority class cases used in each dataset
Class Overlap	$\{30°; 45°; 90°\}$	Rotation applied to generate the minority class cases
IR	$\{0.4; 0.2; 0.1\}$	Imbalance ratio obtained by adding or removing minority class cases
Nr. of Features	$\{28 \times 28; 14 \times 14; 8 \times 8; 4 \times 4\}$	Image resize for all cases

3.2. Experimental Setting

We carry out our experiments using the 144 datasets previously generated with different data difficulty factors. Our main goal is to test the impact of different data difficulty factors when tackling the class imbalance by upsampling the minority class cases through a CGAN. To this end, we setup two main evaluation settings: (1) without data augmentation, and (2) with data augmentation via a CGAN upsampling of the minority class that ensures a balanced training set. We selected as our learning algorithm a neural network binary classifier which is used on both settings. The main difference between the two settings is the on setting one (without data augmentation) we observe the performance achieved when using the dataset as is, while on the second experimental setting we use the conditional GAN to balance the training set before training the model. Figure 2 displays a detailed overview of the system implemented on both experimental settings. For the baseline setting we have a simple training/test flow, while for the setting with data augmentation, we have an extra upsampling module. The CGAN model is trained after which the generator is extracted and used to generate synthetic minority class cases. The new generated cases are added to the training set to balance the classes distribution. The balanced dataset is then used to train a model.

We used a 5-fold non-stratified cross validation procedure to estimate the performance and selected precision, recall and F1-score (cf. Equations 1, 2 and 3) as the performance assessment metrics. In these equations we used TP, TN, FP, FN for representing True Positive, True Negative, False Positive, and False Negative respectively.

$$Precision = \frac{TP}{TP + FP} \quad (1) \qquad Recall = \frac{TP}{TP + FN} \quad (2)$$

$$F1 - score = \frac{2 * Precision * Recall}{Precision + Recall} \quad (3)$$

Due to the imbalance nature of our problem, we cannot rely on standard performance evaluation metrics such as accuracy which can be misleading for this context (Branco et al., 2016). Moreover, we opted to report the metrics results for both the majority and minority classes instead of restricting our results to the minority class as typically done. This way, we will be able to observe the impact of the data difficulty factors and CGAN upsampling on both classes which may also provide interesting insights.

Our experiments are carried out for a total of 144 datasets. For each dataset, the performance results on the two settings (with and without using augmentative upsampling) are reported. This means a total of 288 tests are conducted.

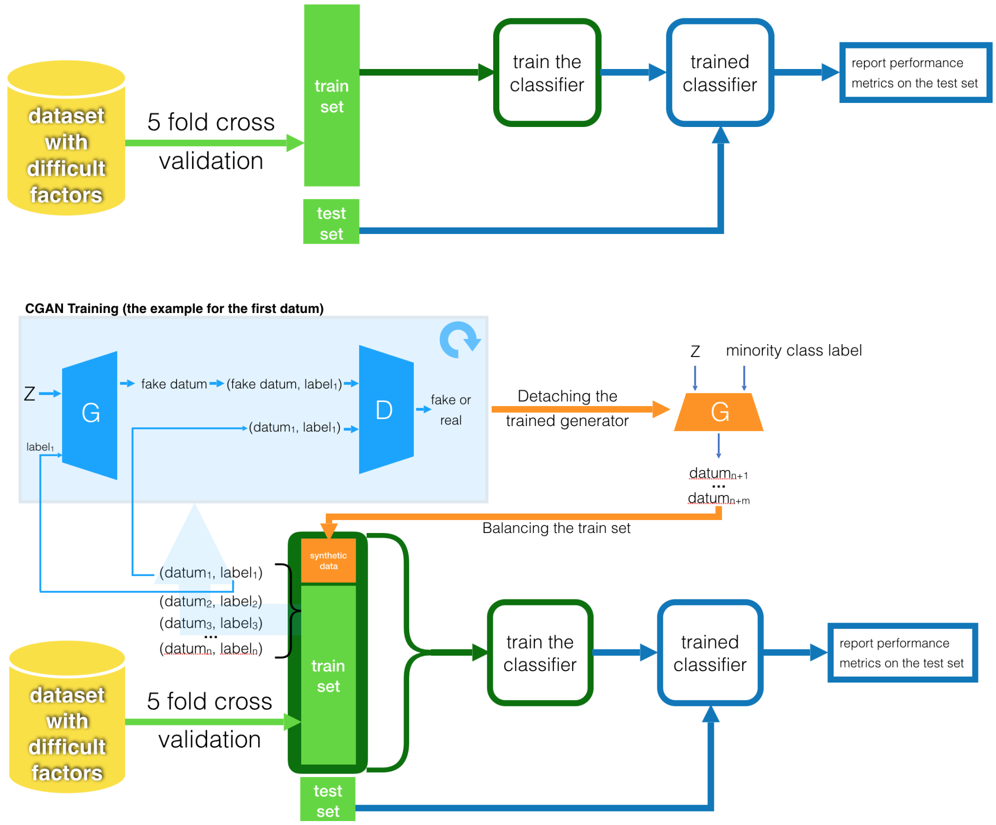

Figure 2: Detailed experimental settings: without data augmentation (top) and with data augmentation (bottom).

Regarding the classifier, an MLP with one hidden layer with 10 perceptrons is used. The activation function for the hidden layer and the output layer are ReLU and sigmoid respectively. Table 2 shows the described architecture. The loss function used is the binary crossentropy and the optimiser is Adam (Kingma and Ba, 2017) with the following parameters: learning rate=0.001, beta 1=0.9, beta 2=0.999, and epsilon=1e-07.

Regarding the CGAN used for our experiments, Figure 3 displays the CGAN architecture of the generator and the discriminator respectively for the particular case of a dataset with a total of 784 features. For other datasets with a different number of features, the output of the generator and the input of the discriminator are changed to match the number of the features of the images. For the generator and discriminator architectures, we used the implementations provided in `https://github.com/eriklindernoren/Keras-GAN/tree/master/cgan`. To ensure the reproducibility of our experiments, all the datasets and code used in these experiments are available on the following link `https://github.com/enazari/GAN-upsampling-LIDTA21`.

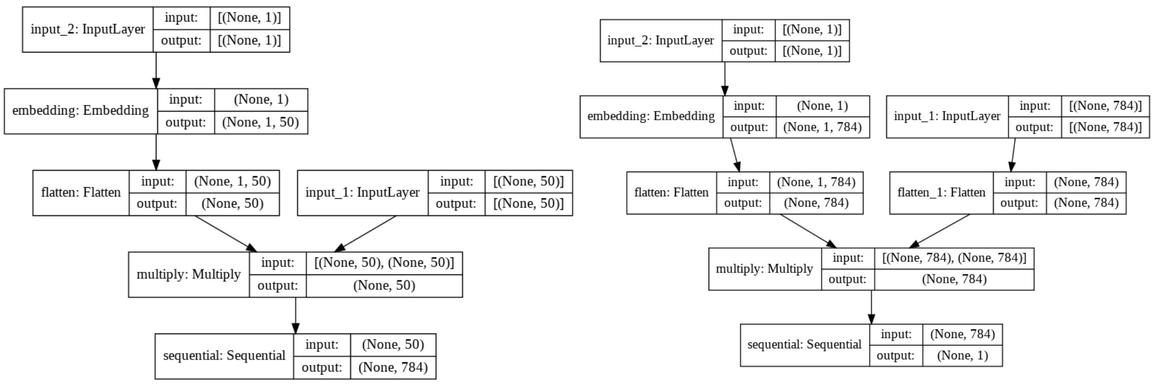

Figure 3: The architecture of the CGAN generator (on the left) and discriminator (on the right), for a dataset with 784 features.

Table 2: The architecture of the neural network binary classifier used in the experiments.

Layer Number	Nodes Count	Activation Function
first layer	one of: 784, 196, 64, 16	-
second layer	10	relu
final layer	1	sigmoid

3.3. Results and Discussion

We start by introducing the notation used throughout this section. We report precision, recall and F1-score on both the majority and minority classes. Thus, we adopted the prefix **maj** or **min** to refer to a metric calculated on the majority or minority class respectively. After the prefix, we express the metric under consideration by using **p**, **r** or **f1** to represent the precision, recall and F1-score respectively. Finally, when result is computed on the original dataset, no suffix is appended. When the result is obtained on the balanced training set, then the suffix **bal** is appended. As an example, the notation **maj_r_bal** refers to the recall metric calculated on the majority class after balancing the training set, while **min_f1** refers to the F1-score metric evaluated on the minority class when the original dataset is used. In order to compare the impact observed after applying the CGAN, for each metric, we also compute its ratio after and before balancing the training set. We used the suffix **imp** after the class and metric under consideration to represent this importance. For instance, **min_p_imp** represents the impact on the minority class precision which is measured as the ratio between min_p_bal and min_p.

Our initial analysis regards a global overview of all our results. Table 3 shows the obtained performance results when using the original imbalanced datasets and after applying CGAN as a data augmentation strategy. As we can observe in Table 3, four out of six scores are improved when applying upsampling via CGAN. All minority class scores as well as the majority class precision show improvements. The overall aggregated F1-scores after the data augmentation process show a decrease of 3% for the majority class and a remarkable 26% increase for the minority class. This shows that, considering the minority class, using

Table 3: The average of all the metrics results on the 144 tests before data augmentation and the 144 tests after data augmentation through CGAN.

metric	score without augmentation	score with augmentation
precision of the majority class	0.895806	**0.943221**
recall of the majority class	**0.994758**	0.898354
F1-score of the majority class	**0.940310**	0.912214
precision of the minority class	0.556739	**0.707897**
recall of the minority class	0.391511	**0.763900**
F1-score of the minority class	0.434325	**0.701059**

Table 4: Average of all conducted tests' results by rotation angle.

	rot_angle	maj_p	maj_p_bal	maj_p_imp	min_p	min_p_bal	min_p_imp
precision	30	0.879204	0.923472	1.05	0.499225	0.600382	1.2
	45	0.899937	0.947635	1.05	0.564149	0.733548	1.3
	90	0.908278	0.958557	1.06	0.606842	0.789759	1.3
	rot_angle	maj_r	maj_r_bal	maj_r_imp	min_r	min_r_bal	min_r_imp
recall	30	0.992280	0.863538	0.87	0.309874	0.672050	2.17
	45	0.995319	0.911442	0.92	0.413347	0.796543	1.93
	90	0.996676	0.920082	0.92	0.451313	0.823106	1.82
	rot_angle	maj_f1	maj_f1_bal	maj_f1_imp	min_f1	min_f1_bal	min_f1_imp
F1-score	30	0.930060	0.883843	0.95	0.358458	0.596278	1.66
	45	0.942688	0.923152	0.98	0.450307	0.732212	1.63
	90	0.948183	0.929648	0.98	0.494209	0.774687	1.57

an augmentative upsampling via a CGAN presents an advantage for datasets with multiple data difficulty factors. The overall gains achieved with data augmentation are significant, ranging between $\tilde{1}5\%$ and $\tilde{3}7\%$. Regarding the majority class, we observe gains only for the precision score. However, both the recall and F1-score show a small decrease for the majority class. This is however expected when using data augmentation techniques.

Table 4 shows the detailed results when considering different levels of class overlap through the use of rotated images. We observe that all metrics calculated on both the minority and majority class display lower scores for lower rotation angles. This matches our initial expectations as lower rotation angles will produce images with a higher overlap with the initial images, thus making the classification task more challenging. We also verify an increase in all results after applying the CGAN upsampling method, except for the recall and F1-score evaluated on the majority class. This shows that our method is effective even for imbalanced problems suffering from class overlap. The gains achieved on the minority class are significant reaching improvements between 120% and 217%. For the minority class, the precision scores are the ones exhibiting a lower positive impact, while both recall and F1-score display gains above 157% after balancing. The gains occur for all rotation angles applied but are higher for lower rotation angles which correspond to scenarios with more overlap between the classes.

Table 5 shows the performance results obtained for both classes aggregated by number of features of the datasets. In this case, there is a clear positive impact of the augmentative

Table 5: Average of all conducted tests' results by number of features.

	nr_features	maj_p	maj_p_bal	maj_p_imp	min_p	min_p_bal	min_p_imp
precision	16	0.825056	0.894222	1.08	0.135569	0.422668	3.12
	64	0.885336	0.956430	1.08	0.548632	0.699522	1.28
	196	0.927271	0.960783	1.04	0.724369	0.826931	1.14
	784	0.945563	0.961451	1.02	0.818384	0.882467	1.08
	nr_features	maj_r	maj_r_bal	maj_r_imp	min_r	min_r_bal	min_r_imp
recall	16	0.997253	0.748980	0.75	0.030961	0.656321	21.20
	64	0.995445	0.907104	0.91	0.325673	0.800540	2.46
	196	0.993056	0.963236	0.97	0.551118	0.787337	1.43
	784	0.993281	0.974095	0.98	0.658294	0.811400	1.23
	nr_features	maj_f1	maj_f1_bal	maj_f1_imp	min_f1	min_f1_bal	min_f1_imp
F1-score	16	0.900711	0.794932	0.88	0.045959	0.471036	10.25
	64	0.935136	0.926155	0.99	0.380603	0.717461	1.89
	196	0.957646	0.960770	1.00	0.601239	0.788395	1.31
	784	0.967748	0.966999	1.00	0.709498	0.827345	1.17

Table 6: The average of all conducted tests' results by imbalance ratio

	imbalance_ratio	maj_p	maj_p_bal	maj_p_imp	min_p	min_p_bal	min_p_imp
precision	0.1	0.928318	0.960770	1.03	0.355617	0.611449	1.72
	0.2	0.892025	0.941435	1.06	0.552048	0.708164	1.28
	0.4	0.867077	0.927459	1.07	0.762551	0.804077	1.05
	imbalance_ratio	maj_r	maj_r_bal	maj_r_imp	min_r	min_r_bal	min_r_imp
recall	0.1	0.998732	0.905932	0.91	0.234442	0.664055	2.83
	0.2	0.997305	0.890322	0.89	0.366842	0.784659	2.14
	0.4	0.988238	0.898808	0.91	0.573250	0.842985	1.47
	imbalance_ratio	maj_f1	maj_f1_bal	maj_f1_imp	min_f1	min_f1_bal	min_f1_imp
F1-score	0.1	0.961613	0.921095	0.96	0.268097	0.586425	2.19
	0.2	0.940043	0.907120	0.96	0.414639	0.707478	1.71
	0.4	0.919275	0.908428	0.99	0.620238	0.809274	1.30

strategy applied. Moreover, the magnitude of this impact is dramatically high for the scenarios with the most adverse conditions reaching an improvement of 312% 2120% and 1025% from the original imbalanced dataset for the cases with 16 features. This is an important observation as it shows that CGAN can be very effective for datasets where the number of available features is not high.

In Table 6 we can observe the impact in the performance of the different imbalance ratios tested. Similar to the previous data difficulty factors, there are important gains on the minority class results when applying CGAN data augmentation. The recall and F1-score on the majority class suffer a small negative impact. We also confirm that for the most difficult scenarios associated with a higher difference between the minority and majority class cases the upsampling strategy produces the highest benefits to the minority class. It is important to highlight that, although there is some performance loss on the majority class, that loss magnitude is much smaller that the magnitude of the gains observed.

Regarding the sample size, Table 7 displays the overall results obtained for the different scenarios considered. Similar gains are observed for the results of this data difficulty factor,

Table 7: The average of all conducted tests' results by majority class count.

	maj_count	maj_p	maj_p_bal	maj_p_imp	min_p	min_p_bal	min_p_imp
precision	100	0.854340	0.903191	1.06	0.274914	0.544614	1.98
	200	0.882872	0.939053	1.06	0.497944	0.703507	1.41
	400	0.908053	0.961276	1.06	0.636845	0.764151	1.20
	1000	0.937960	0.969365	1.03	0.817252	0.819315	1.00
	maj_count	maj_r	maj_r_bal	maj_r_imp	min_r	min_r_bal	min_r_imp
recall	100	0.994862	0.789104	0.79	0.163082	0.698022	4.28
	200	0.996836	0.921883	0.92	0.311874	0.699107	2.24
	400	0.994648	0.932106	0.94	0.449297	0.803802	1.79
	1000	0.992688	0.950323	0.96	0.641793	0.854667	1.33
	maj_count	maj_f1	maj_f1_bal	maj_f1_imp	min_f1	min_f1_bal	min_f1_imp
F1-score	100	0.916672	0.818838	0.89	0.189529	0.548599	2.89
	200	0.934167	0.926713	0.99	0.359792	0.662593	1.84
	400	0.947295	0.944452	1.00	0.498504	0.765536	1.54
	1000	0.963108	0.958853	1.00	0.689474	0.827510	1.20

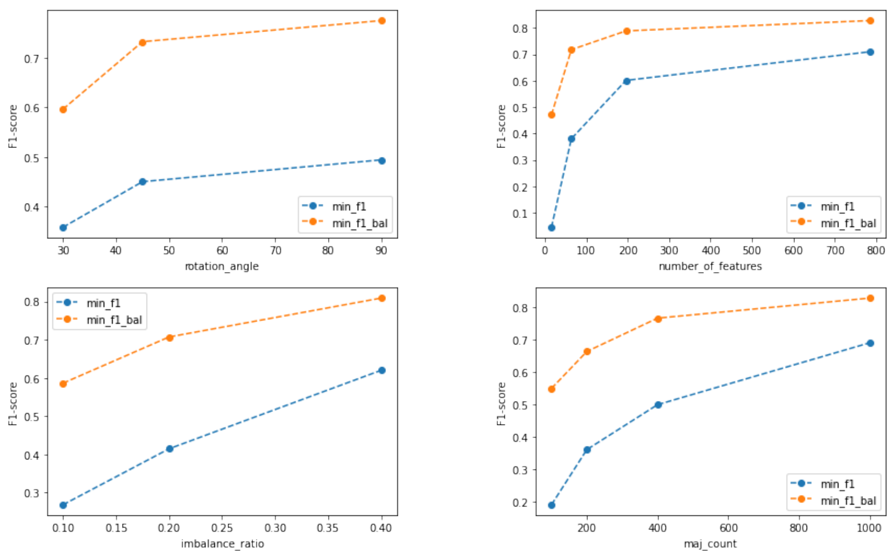

Figure 4: Minority class F1-score before (blue) and after upsampling (orange) by: rotation angle (top left), number of features (top right), imbalance ratio (bottom left), and sample size (bottom right).

confirming that CGAN is effective in this setting and provides higher performance gains for the most difficult scenarios.

Figure 4 shows the F1-score results on the minority class for the four difficult factors considered in this study: class overlap (achieved through the rotation angles applied), number of features (achieved through images resizing), imbalance ratio (achieved through minority class examples added) and sample size (achieve through the base majority class count). This figure shows that in all cases balancing the distribution via CGAN has a positive impact in the performance of the learner. It is also clear that this method has significant advantages even for the more difficult tasks where we observe lower performance results on the original imbalanced datasets.

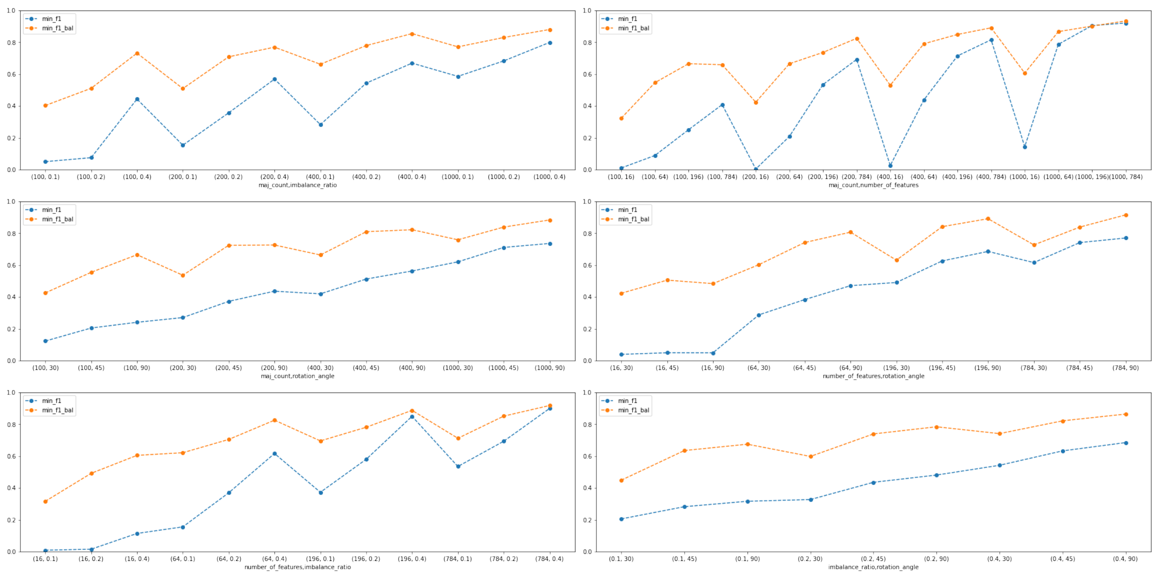

Figure 5: Minority class F1-score for two data difficulty factors (blue: original data, orange: upsampled data). From top left to bottom right: sample size-imbalance ratio, sample size-number of features, sample size-class overlap, number of features-class overlap, number of features-imbalance ratio, and imbalance ratio-class overlap.

To analyse the impact on the performance of multiple data difficult factors we formed all paired combinations of two difficult factors. Figure 5 shows the F1-score of the minority class for all these combinations. The factors are sorted by decreasing level of difficult. We observe that as the combination of the levels of the factors becomes less difficult the minority class performance improves. This happens with and without the application of CGAN upsampling. We notice that the gains in performance obtained with CGAN tend to be more reduced as the difficulty of the factors reduces. The toothed saw effect displayed on the majority of factors combinations for the dataset modified via CGAN shows that each difficult factor has an important impact when the other factor considered is fixed.

4. Conclusions

In this paper we studied the suitability of CGAN as a upsampling strategy when the predictive task includes multiple data difficulty factors. Namely, we considered the impact of class overlap, data dimensionality, imbalance ratio and sample size. Data sets with varying levels of these factors were generated using the MNIST data. Our extensive experiments show that upsampling through CGAN is an effective way of tackling the data difficulty factors studied, providing significant performance gains in the minority class in all scenarios. Overall, we found this strategy to work well under the different tested conditions. The use of

CGAN exhibited larger gains for the most extreme difficult scenarios tested associated with a lower dimensionality, higher imbalance, higher class overlap and lower sample size. The case where a low dimension data set was used, provided the most impressive gains. When analysing two difficult factors we observe a performance degradation with CGAN exhibiting increased gains for the scenarios including higher difficulty levels. Overall, our experiments show that CGAN can be effectively used to tackle multiple data difficulty factors that tend to coexist in real-world problems.

As future work we plan to explore the suitability of other GAN architectures as well as carry out more experiments with both real-world and synthetic data sets. We also consider an interesting future research direction the intelligent combination of GANs with other upsampling strategies.

References

Antreas Antoniou, Amos Storkey, and Harrison Edwards. Data augmentation generative adversarial networks. *arXiv preprint arXiv:1711.04340*, 2017.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017.

Paula Branco and Luis Torgo. A study on the impact of data characteristics in imbalanced regression tasks. In *2019 IEEE DSAA*, pages 193–202. IEEE, 2019.

Paula Branco, Luís Torgo, and Rita P Ribeiro. A survey of predictive modeling on imbalanced domains. *ACM Computing Surveys (CSUR)*, 49(2):1–50, 2016.

Dariusz Brzezinski, Leandro L Minku, Tomasz Pewinski, Jerzy Stefanowski, and Artur Szumaczuk. The impact of data difficulty factors on classification of imbalanced and concept drifting data streams. *KAIS*, 63(6):1429–1469, 2021.

Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16: 321–357, 2002.

Kumari Deepshikha and Anugunj Naman. Removing class imbalance using polarity-gan: An uncertainty sampling approach. *arXiv preprint arXiv:2012.04937*, 2020.

Georgios Douzas and Fernando Bacao. Effective data generation for imbalanced learning using conditional generative adversarial networks. *ESWA*, 91:464–471, 2018.

Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *arXiv preprint arXiv:1406.2661*, 2014.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein gans. *arXiv preprint arXiv:1704.00028*, 2017.

Ting Guo, Xingquan Zhu, Yang Wang, and Fang Chen. Discriminative sample generation for deep imbalanced learning. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 2406–2412, 7 2019.

Hui Han, Wen-Yuan Wang, and Bing-Huan Mao. Borderline-smote: a new over-sampling method in imbalanced data sets learning. In *International conference on intelligent computing*, pages 878–887. Springer, 2005.

Jingyu Hao, Chengjia Wang, Heye Zhang, and Guang Yang. Annealing genetic gan for minority oversampling. *arXiv preprint arXiv:2008.01967*, 2020.

Haibo He, Yang Bai, Edwardo A Garcia, and Shutao Li. Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In *2008 IEEE international joint conference on neural networks*, pages 1322–1328. IEEE, 2008.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2017.

Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010. URL http://yann.lecun.com/exdb/mnist/.

Victoria López, Alberto Fernández, Salvador García, Vasile Palade, and Francisco Herrera. An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. *Information sciences*, 250:113–141, 2013.

Giovanni Mariani, Florian Scheidegger, Roxana Istrate, Costas Bekas, and Cristiano Malossi. Bagan: Data augmentation with balancing gan. *arXiv preprint arXiv:1803.09655*, 2018.

Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

Krystyna Napierała, Jerzy Stefanowski, and Szymon Wilk. Learning from imbalanced data in presence of noisy and borderline examples. In *International conference on rough sets and current trends in computing*, pages 158–167. Springer, 2010.

Kwanda Sydwell Ngwenduna and Rendani Mbuvha. Alleviating class imbalance in actuarial applications using generative adversarial networks. *Risks*, 9(3), 2021. ISSN 2227-9091.

Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *ICML*, pages 2642–2651. PMLR, 2017.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

Vignesh Sampath, Iñaki Maurtua, Juan José Aguilar Martín, and Aitor Gutierrez. A survey on generative adversarial networks for imbalance problems in computer vision tasks. *Journal of big Data*, 8(1):1–59, 2021.

Sungho Suh, Haebom Lee, Paul Lukowicz, and Yong Oh Lee. Cegan: Classification enhancement generative adversarial networks for unraveling data imbalance problems. *Neural Networks*, 133:69–86, 2021.