DeepSMOTE: Fusing Deep Learning and SMOTE for Imbalanced Data

Damien Dablain, Bartosz Krawczyk, *Member, IEEE*, and Nitesh V. Chawla, *Fellow, IEEE*

***Abstract*— Despite over two decades of progress, imbalanced data is still considered a significant challenge for contemporary machine learning models. Modern advances in deep learning have further magnified the importance of the imbalanced data problem, especially when learning from images. Therefore, there is a need for an oversampling method that is specifically tailored to deep learning models, can work on raw images while preserving their properties, and is capable of generating high-quality, artificial images that can enhance minority classes and balance the training set. We propose Deep synthetic minority oversampling technique (SMOTE), a novel oversampling algorithm for deep learning models that leverages the properties of the successful SMOTE algorithm. It is simple, yet effective in its design. It consists of three major components: 1) an encoder/decoder framework; 2) SMOTE-based oversampling; and 3) a dedicated loss function that is enhanced with a penalty term. An important advantage of DeepSMOTE over generative adversarial network (GAN)-based oversampling is that DeepSMOTE does not require a discriminator, and it generates high-quality artificial images that are both information-rich and suitable for visual inspection. DeepSMOTE code is publicly available at https://github.com/dd1github/DeepSMOTE.**

***Index Terms*— Class imbalance, deep learning, machine learning, oversampling, synthetic minority oversampling technique (SMOTE).**

I. Introduction

LEARNING from imbalanced data is among the most crucial problems faced by the machine learning community [1]. Imbalanced class distributions affect the training process of classifiers, leading to unfavorable bias toward the majority class(es). This may result in high error, or even complete omission, of the minority class(es). Such a situation cannot be accepted in most real-world applications (e.g., medicine or intrusion detection) and thus algorithms for countering the class imbalance problem have been a focus of intense research for over two decades [2]. Contemporary applications have extended our view of the problem of imbalanced data, confirming that disproportionate classes are not the sole source of learning problems. A skewed class imbalance ratio is often accompanied by additional factors, such as difficult and borderline instances, small disjuncts, small sample size [2], or the drifting nature of streaming data [3], [4]. These continuously emerging challenges keep the field expanding, calling for novel and effective solutions that can analyze, understand, and tackle these data-level difficulties. Deep learning is currently considered as the most promising branch of machine learning, capable of achieving outstanding cognitive and recognition potentials. However, despite its powerful capabilities, deep architectures are still very vulnerable to imbalanced data distributions [5], [6] and are affected by novel challenges such as complex data representations [7], the relationship between imbalanced data and extracted embeddings [8], the continually drifting nature of data [9], and learning from an extremely large number of classes [10].

A. Research Goal

We propose a novel oversampling method for imbalanced data that is specifically tailored to deep learning models and that leverages the advantages of synthetic minority oversampling technique (SMOTE) [11], while embedding it in a deep architecture capable of efficient operation on complex data representations, such as images.

B. Motivation

Although the imbalanced data problem strongly affects both deep learning models [12] and their shallow counterparts, there has been limited research on how to counter this challenge in the deep learning realm. In the past, the two main directions that have been pursued to overcome this challenge have been loss function modifications and resampling approaches. The deep learning resampling solutions are either pixel-based or use generative adversarial networks (GANs) for artificial instance generation. Both these approaches suffer from strong limitations. Pixel-based solutions often cannot capture complex data properties of images and are not capable of generating meaningful artificial images. GAN-based solutions require significant amounts of data, are difficult to tune, and may suffer from mode collapse [13]–[16]. Therefore, there is a need for a novel oversampling method that is specifically tailored to the nature of deep learning models, can work on raw images while preserving their properties, and is capable of generating artificial images that are of both of high visual quality and enrich the discriminative capabilities of deep models.

C. Summary

We propose DeepSMOTE, a novel oversampling algorithm for deep learning models based on the highly popular SMOTE method. Our method bridges the advantages of metric-based resampling approaches that use data characteristics to

Manuscript received 23 April 2021; revised 4 October 2021 and 1 December 2021; accepted 15 December 2021. Date of publication 27 January 2022; date of current version 1 September 2023. *(Corresponding authors: Bartosz Krawczyk; Nitesh V. Chawla.)*

Damien Dablain and Nitesh V. Chawla are with the Department of Computer Science and Engineering and the Lucy Family Institute for Data and Society, University of Notre Dame, Notre Dame, IN 46556 USA (e-mail: ddablain@nd.edu; nchawla@nd.edu).

Bartosz Krawczyk is with the Department of Computer Science, Virginia Commonwealth University, Richmond, VA 23284 USA (e-mail: bkrawczyk@vcu.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TNNLS.2021.3136503.

Digital Object Identifier 10.1109/TNNLS.2021.3136503

leverage their performance, with a deep architecture capable of working with complex and high-dimensional data. DeepSMOTE consists of three major components: 1) an encoder/decoder framework; 2) SMOTE-based oversampling; and 3) a dedicated loss function enhanced with a penalty term. This approach allows us to embed effective SMOTE-based artificial instance generation within a deep encoder/decoder model for a streamlined and end-to-end process, including low-dimensional embeddings, artificial image generation, and multiclass (MC) classification.

D. Main Contributions

In order for an oversampling method to be successfully applied to deep learning models, we believe that it should meet three essential criteria: 1) it should operate in an end-to-end manner; 2) it should learn a representation of the raw data and embed the data into a lower dimensional *feature space*; and 3) it should readily generate output (e.g., images) that can be visually inspected. In this article, we propose DeepSMOTE, which meets these three criteria, and also offers the following scientific contributions to the field of deep learning under class imbalance.

1) *Deep oversampling architecture:* We introduce DeepSMOTE, a self-contained deep architecture for oversampling and artificial instance generation that allows efficient handling of complex-imbalanced and high-dimensional data, such as images.
2) *Simple and effective solution to class imbalance:* Our framework is simple, yet effective in its design. It consists of only three major components responsible for low-dimensional representations of raw data, resampling, and classification.
3) *No need for a discriminator during training:* An important advantage of DeepSMOTE over GAN-based oversampling lies in the fact that DeepSMOTE does not require a discriminator during the artificial instance generation process. We propose a penalty function that ensures efficient usage of training data to prime our generator.
4) *High-quality image generation:* DeepSMOTE generates high-quality artificial images that are both suitable for visual inspection (they are of identical quality as their real counterparts) and information-rich, which allows for efficient balancing of classes and alleviates the effects of imbalanced distributions.
5) *Extensive experimental study:* We propose a carefully designed and thorough experimental study that compares DeepSMOTE with state-of-the-art oversampling and GAN-based methods. Using five popular image benchmarks and three dedicated skew-insensitive metrics over two different testing protocols, we empirically prove the merits of DeepSMOTE over the reference algorithms. Furthermore, we show that DeepSMOTE displays an excellent robustness to increasing imbalance ratios, being able to efficiently handle even extremely skewed problems.

E. Article Outline

In this article, we first provide an overview of the imbalanced data problem and the traditional approaches that have been employed to overcome this issue. Next, we discuss how deep learning methods have been used to generate data and augment imbalanced datasets. We then introduce our approach to imbalanced learning, which combines deep learning with SMOTE. Finally, we discuss our extensive experimentation, which validates the benefits of DeepSMOTE.

II. Learning From Imbalanced Data

The first works on imbalanced data came from binary classification problems. Here, the presence of majority and minority classes is assumed, with a specific imbalance ratio. Such skewed class distributions pose a challenge for machine learning models, as standard classifiers are driven by a 0–1 loss function that assumes a uniform penalty over both classes. Therefore, any learning procedure driven by such a function will lead to a bias toward the majority class. At the same time, the minority class is usually more important and thus cannot be poorly recognized. Therefore, methods dedicated to overcoming the imbalance problem aim at either alleviating the class skew or alternating the learning procedure. The three main approaches are as follows.

A. Data-Level Approaches

This solution should be viewed as a preprocessing phase that is classifier-independent. Here, we focus on balancing the dataset before applying any classifier training. This is usually achieved in one of three ways: 1) reducing the size of the majority class (undersampling); 2) increasing the size of minority class (oversampling); or 3) a combination of the two previous solutions (hybrid approach). Both under- and oversampling can be performed in a random manner, which has low complexity, but leads to potentially unstable behavior (e.g., removing important instances or enhancing noisy ones). Therefore, guided solutions have been proposed that try to smartly choose instances for preprocessing. While not many solutions have been proposed for guided undersampling [17]–[19], oversampling has gained much more attention due to the success of SMOTE [11], which led to the introduction of a plethora of variants [20]–[24]. However, recent works show that SMOTE-based methods cannot properly deal with multimodal data and cases with high intraclass overlap or noise. Therefore, completely new approaches that do not rely on k-nearest neighbors have been successfully developed [25], [26].

B. Algorithm-Level Approaches

Contrary to the previously discussed approaches, algorithm-level solutions work directly within the training procedure of the considered classifier. Therefore, they lack the flexibility offered by data-level approaches, but compensate with a more direct and powerful way of reducing the bias of the learning algorithm. They also require an in-depth understanding of how a given training procedure is conducted and what specific part of it may lead to bias toward the majority class. The most commonly addressed issues with the algorithmic approach are developing novel skew-insensitive split criteria for decision trees [27]–[29], using instance weighting for support vector machines [30]–[32], or modifying the way different layers are trained in deep learning [33]–[35]. Furthermore, cost-sensitive

solutions [36]–[38] and one-class classification [39]–[41] can also be considered as a form of algorithm-level approaches.

C. Ensemble Approaches

The third way of managing imbalanced data is to use ensemble learning [42]. Here, one either combines a popular ensemble architecture (usually based on Bagging or Boosting) with one of the two previously discussed approaches or develops a completely new ensemble architecture that is skew-insensitive on its own [43]. One of the most successful families of methods is the combination of Bagging with undersampling [44]–[46], Boosting with any resampling technique [47]–[49], or cost-sensitive learning with multiple classifiers [50]–[52]. Data-level techniques can be used to manage the diversity of the ensemble [53], which is a crucial factor behind the predictive power of multiple classifier systems. Additionally, to manage the individual accuracy of classifiers and eliminate weaker learners, one may use dynamic classifier selection [54] and dynamic ensemble selection [55], which ensures that the final decision will be based only on the most competent classifiers from the pool [56].

III. Deep Learning From Imbalanced Data

Since the imbalanced data problem has been attracting increasing attention from the deep learning community, let us discuss three main trends in this area.

A. Instance Generation With Deep Neural Networks

Recent works that combine deep learning with shallow oversampling methods do not give desirable results and traditional resampling approaches cannot efficiently augment the training set for deep models [2], [57]. This leads to an interest in generative models and adapting them to work in a similar manner to oversampling techniques [58]. An encoder/decoder combination can efficiently introduce artificial instances into a given embedding space [59]. GANs [60], variational autoencoders (VAEs) [61], and Wasserstein autoencoders (WAEs) [62] have been successfully used within computer vision (CV) [63], [64] and robotic control [65], [66] to learn the latent distribution of data. These techniques can also be extended to data generation for oversampling (e.g., medical imaging) [67].

VAEs operate by maximizing a variational lower bound of the data log-likelihood [68], [69]. The loss function in a VAE is typically implemented by combining a reconstruction loss with the Kullback–Leibler (KL) divergence. The KL divergence can be interpreted as an implicit penalty on the reconstruction loss. By penalizing the reconstruction loss, the model can learn to *vary* its reconstruction of the data distribution and thus *generate* output (e.g., images) based on a latent distribution of the input.

WAEs also exhibit generative qualities. Similar to VAEs, the loss function of a WAE is often implemented by combining a reconstruction loss with a penalty term. In the case of a WAE, the penalty term is expressed as the output of a discriminator network.

GANs have achieved impressive results in the computer vision arena [70], [71]. GANs formulate image generation as a min–max game between a generator and a discriminator network [72]. Despite their impressive results, GANs require the use of two networks, are sometimes difficult to train, and are subject to mode collapse (i.e., the repetitive generation of similar examples) [13]–[16].

B. Loss Function Adaptation

One of the most popular approaches for making neural networks skew-insensitive is to modify their loss function. This approach successfully carried over to deep architectures and can be seen as an algorithm-level modification. The idea behind modifying the loss function is based on the assumption that instances should not be treated uniformly during training and that errors on minority classes should be penalized more strongly, making it parallel to cost-sensitive learning [38]. Mean False Error [73] and Focal Loss [74] are two of the most popular approaches based on this principle. The former simply balances the impact of instances from minority and majority classes, while the latter reduces the impact of easy instances on the loss function. More recently, multiple other loss functions were proposed, such as Log Bilinear Loss [75], Cross Entropy Loss [76], and Class-Balanced Loss [77].

C. Long-Tailed Recognition

This subfield of deep learning evolved from problems where there is a high number of very rare classes that should nevertheless be properly recognized, despite their low sample size. Long-tailed recognition can be thus seen as an extreme case of the MC imbalanced problem, where we deal with a very high number of classes (hundreds) and an extremely high imbalance ratio. Due to very disproportionate class sizes, direct resampling is not advisable, as it will either significantly reduce the size of majority classes or require creation of too many artificial instances. Furthermore, classifiers need to handle the problem of small sample size, making learning from the tail classes very challenging. It is important to note that the majority of works in this domain assume that the test set is balanced. Very interesting solutions to this problem are based on adaptation of the loss function in deep neural networks, such as equalization loss [78], hubless loss [79], and range loss [80]. Recent works suggest looking closer at class distributions and decomposing them into balanced sets—an approach popular in traditional imbalanced classification. Zhou *et al.* [81] proposed a cumulative learning scheme from global data properties down to class-based features. Sharma *et al.* [82] suggested using a small ensemble of three classifiers, each focusing on majority, middle, or tail groups of classes. Meta-learning is also commonly used to improve the distribution estimation of tail classes [83].

IV. DeepSMOTE

A. Motivation

We propose DeepSMOTE, a novel and breakthrough oversampling algorithm dedicated to enhancing deep learning models and countering the learning bias caused by imbalanced classes. As discussed above, oversampling is a proven technique for combating class imbalance; however, it has traditionally been used with classical machine learning models. Several

attempts have been made to extend oversampling methods, such as SMOTE, to deep learning models, although the results have been mixed [84]–[86]. In order for an oversampling method to be successfully applied to deep learning models, we believe that it should meet three essential criteria.

1) It should operate in an end-to-end manner by accepting raw input, such as images (i.e., similar to VAEs, WAEs, and GANs).
2) It should learn a representation of the raw data and embed the data into a lower dimensional *feature space*, which can be used for oversampling.
3) It should readily generate output (e.g., images) that can be visually inspected, without extensive manipulation.

We show through our design steps and experimental evaluation that DeepSMOTE meets these criteria. In addition, it is capable of generating high-quality, sharp, and information-rich images without the need for a discriminator network.

B. DeepSMOTE Description

DeepSMOTE consists of an encoder/decoder framework, a SMOTE-based oversampling method, and a loss function with a reconstruction loss and a penalty term. Each of these features is discussed below, with Fig. 1 depicting the flow of the DeepSMOTE approach, while the pseudo-code overview of DeepSMOTE is presented in Algorithm 1.

Algorithm 1 DEEPSMOTE

Data: B: batches of imbalanced training data
(D) $B = \{b_1, b_2, \ldots, b_n\}$
Input: Model parameters: $\Theta = \{\Theta_0, \Theta_1, \ldots, \Theta_j\}$; Learning Rate: α
Output: Balanced training set.
Symbols: R_L - Reconstruction loss; P_L - Penalty loss;
T_L - Total loss;
C - Set of classes in D;
C_M - Set of minority classes in D;
G - Set of generated and encoded examples;
S - Set of generated and decoded data (balanced).
Train the Encoder / Decoder:
for $e \leftarrow epochs$ **do**
  for $b \leftarrow B$ **do**
    $E_b \leftarrow encode(b)$
    $D_b \leftarrow decode(E_b)$
    $R_L = \frac{1}{n}\sum_{i=1}^{n}(D_{bi} - b_i)^2$
    $C_D \leftarrow randomly\ sample\ a\ class\ from\ C$
    $C_b \leftarrow randomly\ sample\ |b|\ instances\ from\ C_D$
    $E_S \leftarrow encode(C_b)$
    $P_E \leftarrow permute\ order(E_S)$
    $D_P \leftarrow decode(P_E)$
    $P_L = \frac{1}{n}\sum_{i=1}^{n}(D_{Pi} - C_{Di})^2\ T_L = R_L + P_L$
    $\Theta := \Theta - \alpha\frac{\partial T_L}{\partial \Theta}$
Generate Samples:
foreach $m \leftarrow minority\ class\ (C_M)$ **do**
  $C_{md} \leftarrow select\ (C_m\ imbalanced\ data)$
  $E_m \leftarrow encode(C_{md})$
  $G_m \leftarrow SMOTE(E_m)$
  $S_m \leftarrow decode(G_m)$

C. Encoder/Decoder Framework

The DeepSMOTE backbone is based on the deep convolutional GAN (DCGAN) architecture, which was established by Radford *et al.* [87]. Radford *et al.* [87] used a discriminator/generator in a GAN, which is fundamentally similar to an encoder/decoder because the discriminator effectively encodes input (absent the final, fully connected layer) and the generator (decoder) generates output.

The encoder and decoder are trained in an end-to-end fashion. During DeepSMOTE training, an imbalanced dataset is fed to the encoder/decoder in batches. A reconstruction loss is computed on the batched data. All classes are used during training so that the encoder/decoder can learn to reconstruct both majority and minority class images from the imbalanced data. Because there are few minority class examples, majority class examples are used to train the model to learn the basic reconstruction patterns inherent in the data. This approach is based on the assumption that classes share some similar characteristics (e.g., all classes represent digits or faces). Thus, for example, although the number 9 (minority class) resides in a different class than the number 0 (majority class), the model learns the basic contours of digits.

D. Enhanced Loss Function

In addition to a reconstruction loss, the DeepSMOTE loss function contains a penalty term. The penalty term is based on a reconstruction of embedded images. DeepSMOTE's penalty loss is produced in the following fashion. During training, a class (c) is randomly selected from the set of all classes (C). A group of examples is then randomly sampled from *c* that is equal in number to the batch size. Thus, the number of sampled examples is the same as the number of examples used for reconstruction loss purposes; however, unlike the images used during the reconstruction loss phase of training, the sampled images are all from the same class. The sampled images are then reduced to a lower-dimensional feature space by the encoder. During the decoding phase, the encoded images are *not* reconstructed by the decoder in the same *order* as the encoded images. By changing the *order* of the reconstructed images, which are all from the same class, we effectively introduce *variance* into the encoding/decoding process. For example, the encoded order of the images may be D_0, D_1, D_2, and the decoded order of the images may be D_2, D_0, D_1. This variance facilitates the generation of images during inference (where an image is encoded, SMOTEd, and the decoded).

Essentially, the permutation step is necessary because DeepSMOTE uses an autoencoder (an encoder plus a decoder). The output of an autoencoder is deterministic with respect to its input, in the sense that an autoencoder can only decode or generate what it encodes. In a standard autoencoder, there is no variance in the data that is encoded and decoded. Thus, a standard autoencoder is not capable of generating examples that are different from the input data. Our goal is to introduce variance into the encoded feature space, so that the decoded example is different from the input to the autoencoder, yet constrained by the inputted data. We introduce variance into the encoding/decoding process by permuting the order of the encoded data. Thus, there is bound to be some difference between encoded image D_0 and decoded image D_1. The difference is not likely to be extremely large, since D_0 and $D_1 D_0$ are both from the same class; however, there will be

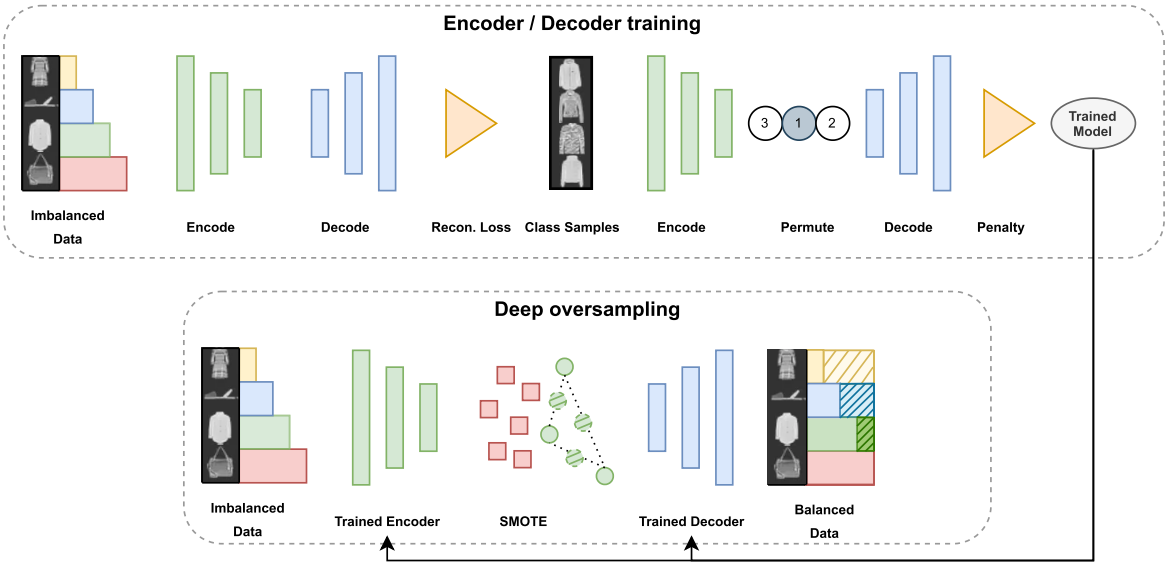


Fig. 1. Illustration of DeepSMOTE implementation. The encoder/decoder structure is trained with imbalanced data and a reconstruction and penalty loss. During training, data is sampled, encoded, and the order of examples are permuted before decoding. The trained encoder and decoder are then combined with SMOTE to produce oversampled data.

some difference. This difference becomes the penalty term. By introducing variance into the encoding process, the decoder gains "practice" at decoding examples that are different from the input data (which a standard decoder in an autoencoder is not trained to do). This "practice" is necessary because during inference, an example is encoded, then it is changed via SMOTE interpolation to a different example, which the decoder must decode.

The penalty loss is based on the mean squared error (MSE) difference between D_0 and D_1, D_1 and D_2, and so on, as if an image was oversampled by SMOTE (i.e., as if an image were generated based on the difference between an image and the image's neighbor). This step is designed to insert variance into the encoding/decoding process. We, therefore, obviate the need for a discriminator because we use training data to train the generator by simply altering the order of the encoded/decoded images.

As a refresher, the SMOTE algorithm generates synthetic instances by randomly selecting a minority class example and one of its class neighbors. The distance between the example and its neighbor is calculated. The distance is multiplied by a random percentage (i.e., between 0 and 1) and added to the example instance in order to generate synthetic instances. We simulate SMOTE's methodology during DeepSMOTE training by selecting a class sample and calculating a distance between the instance and its neighbors (in the embedding or feature space), except that the distance (MSE) during training is used as an implicit penalty on the reconstruction loss. As noted by Arjovsky *et al.* [16], many generative deep learning models effectively incorporate a penalty, or noise, term in their loss function, to impart diversity into the model distribution. For example, both VAEs and WAEs include penalty terms in their loss functions. We use permutation, instead of SMOTE, during training because it is more memory and computationally efficient. The use of the penalty term, and SMOTE's fidelity in interpolating synthetic samples during the inference phase, allows us to avoid the use of a discriminator, which is typically used by GAN and WAE models.

E. Artificial Image Generation

Once DeepSMOTE is trained, images can be generated with the encoder/decoder structure. The encoder reduces the raw input to a lower-dimensional feature space, which is oversampled by SMOTE. The decoder then decodes the SMOTEd features into images, which can augment the training set of a deep learning classifier.

The main difference between the DeepSMOTE training and generation phases is that during the data generation phase, SMOTE is substituted for the order permutation step. SMOTE is used during data generation to introduce variance, whereas during training, variance is introduced by permuting the order of the training examples that are encoded and then decoded and also through the penalty loss. SMOTE itself does not require training because it is nonparametric.

V. Experimental Study

We have designed the following experimental study in order to answer the following research questions.

- *RQ1:* Is DeepSMOTE capable of outperforming state-of-the-art pixel-based oversampling algorithms?
- *RQ2:* Is DeepSMOTE capable of outperforming state-of-the-art GAN-based resampling algorithms designed to work with complex and imbalanced data representations?
- *RQ3:* What is the impact of the test set distribution on DeepSMOTE performance?
- *RQ4:* What is the visual quality of artificial images generated by DeepSMOTE?
- *RQ5:* Is DeepSMOTE robust to increasing class imbalance ratios?
- *RQ6:* Can DeepSMOTE produce stable models under extreme class imbalance?

A. Setup

1) Overview of the Datasets: Five popular datasets were selected as benchmarks for evaluating imbalanced data oversampling: Modified National Institute of Standards and

TABLE I
CLASS DISTRIBUTIONS OF FIVE BENCHMARK DATASETS USED IN EXPERIMENTAL EVALUATION

	MNIST/FMNIST			*CIFAR/SVHN*			*CELEBA*		
	Train	Bal. Test	Imbal. Test	Train	Bal. Test	Imbal. Test	Train	Bal. Test	Imbal. Test
Class									
0	4000	1200	1000	4500	1000	1000	9000	900	1000
1	2000	1200	500	2000	1000	500	4500	900	500
2	1000	1200	250	1000	1000	250	1000	900	111
3	750	1200	187	800	1000	187	500	900	55
4	500	1200	125	600	1000	125	160	900	17
5	350	1200	87	500	1000	87			
6	200	1200	50	400	1000	50			
7	100	1200	25	250	1000	25			
8	60	1200	15	150	1000	15			
9	40	1200	10	80	1000	10			

Technology dataset (MNIST) [88], Fashion-MNIST dataset (FMNIST) [89], CIFAR-10 [90], the street view house numbers (SVHNs) [91], and Large-scale CelebFaces Attributes (CelebA) [92]. Below we discuss their details, while their class distributions are given in Table I.

1) *MNIST/FMNIST:* The MNIST dataset consists of handwritten digits and the FMNIST dataset contains Zalando clothing article images. Both training sets have 60 000 images. Both datasets contain gray-scale images ($1 \times 28 \times 28$), with ten classes each.
2) *CIFAR-10/SVHN:* The CIFAR-10 dataset consists of images, such as automobiles, cats, dogs, frogs, and birds, whereas the SVHN dataset consists of small, cropped digits from house numbers in Google Street View images. CIFAR-10 has 50 000 training images. SVHN has 73 257 digits for training. Both datasets consist of color images ($3 \times 32 \times 32$), with ten classes each.
3) *CelebA:* The CelebA dataset contains 200 000 celebrity images, each with 40 attribute annotations (i.e., classes). The color images ($3 \times 178 \times 218$) in this dataset cover large pose variations and background clutter. For purposes of this study, the images were resized to $3 \times 32 \times 32$ and five classes were selected: black hair, brown hair, blond, gray, and bald.

2) Introducing Class Imbalance: Imbalance was introduced by randomly selecting samples from each class in the training sets. For the MNIST and FMNIST, the number of imbalanced examples were: [4000, 2000, 1000, 750, 500, 350, 200, 100, 60, 40]. For the CIFAR-10 and SVHN datasets, the number of imbalanced examples were: [4500, 2000, 1000, 800, 600, 500, 400, 250, 150, 80]. For CelebA, the number of imbalanced examples were: [9000, 4500, 1000, 500, 160]. For MNIST and FMNIST, the imbalance ratio of the respective majority class compared to the smallest minority class was 100:1; and for CIFAR-10, SVHN, and CelebA, the ratio was approx. 56:1. For experiment 3, we created 20 versions of each dataset with IR in [20 400]. This imbalance ratio is the disproportion between largest and smallest classes, while all other imbalance ratios are proportionately distributed according to the number of classes. This is known as multiminority approach, where we have a single majority class and all other classes being minority ones.

3) Reference Resampling Methods: In order to evaluate the effectiveness of DeepSMOTE, we compare it to state-of-the-art shallow and deep resampling methods. We have selected four pixel-based modern oversampling algorithms: SMOTE [11], adaptive mahalanobis distance-based oversampling (AMDO) [93], combined cleaning and resampling (MC-CCR) [94], and radial-based oversampling (MC-RBO) [95]. Additionally, we have chosen two of the top performing GAN-based oversampling approaches: Balancing GAN (BAGAN) [96] and generative adversarial minority oversampling (GAMO) [97]. BAGAN initializes its generator with the decoder portion of an autoencoder, which is trained on both minority and majority images. GAMO is based on a three-player adversarial game between a convex generator, a classifier network, and a discriminator.

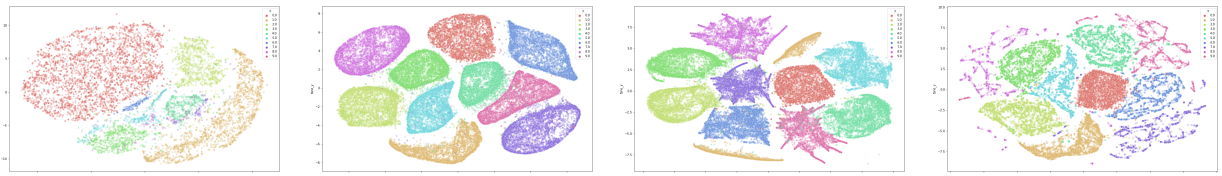
4) Classification Model: All resampling methods use an identical Resnet-18 [98] as their base classifier.

5) Performance Metrics: The following metrics were used to evaluate the performance of the various models: average class specific accuracy (ACSA), macro-averaged geometric mean (GM), and macro-averaged F1 measure (FM). Sokolova and Lapalme have demonstrated that these measures are not prejudiced toward the majority class [99].

6) Testing Procedure: A fivefold cross-validation was used for training and testing the evaluated methods. Thus, we randomly shuffled each training set and split the training sets into fivefolds. Each fold was then selected as a test group with the training examples drawn from the remaining groups. Two approaches to forming test sets were employed: imbalanced and balanced testing. For imbalanced testing, the ratio of test examples follows the same imbalance ratio that exists in the training set (this approach is common in the imbalanced classification domain). With the balanced test sets, the number of test examples was approximately equal across all classes (this approach is common in the long-tailed recognition domain). For example, with MNIST/FMNIST, there are 60 000 examples. With fivefold cross-validation, each split consists of 12 000 examples divided between ten classes or approx. 1200 examples per class.

7) Statistical Analysis of Results: In order to assess whether DeepSMOTE returns statistically significantly better results than the reference resampling algorithms, we use the Friedman test with Shaffer post-hoc test [100] and the Bayesian Wilcoxon signed-rank test [101] for statistical comparison over multiple datasets. Both tests used a statistical significance level of 0.05.

8) DeepSMOTE Implementation Details: As mentioned above, for DeepSMOTE implementation purposes, we used the DCGAN architecture developed by Radford *et al.* [87], with some modifications. The encoder structure consists of four convolutional layers, followed by batch normalization [102] and the LeakyReLu activation function [103]. Each layer consists of convolutional channels (C), with specified kernel size (K), and stride (S). For all datasets, the convolutional layers have the following parameters: $C = [64, 128, 256, 512]$, $K = [4, 4, 4, 4]$, and $S = [2, 2, 2, 2]$. The final layer is a dense layer, yielding a latent dimension of 300 for the MNIST and FMNIST and 600 for the CIFAR-10, SVHN, and CelebA datasets. The decoder structure consists of mirrored convolutional transpose layers, which use batch normalization and the rectified linear unit (ReLU) activation function [104], except for the final layer, which uses Tanh. We train the models for 50–350 epochs, depending on when the training loss plateaus.



(a) (b) (c) (d)

Fig. 2. Illustration of the distribution of the MNIST instances among classes using PCA and t-SNE. High-dimensional images were first reduced using PCA before applying t-SNE, with the x- and y-axes representing t-SNE components. (a) Original imbalanced training set distribution. (b) Balanced distribution using BAGAN. (c) Balanced distribution with GAMO. (d) Balanced distribution with DeepSMOTE. (a) Imbalanced data. (b) BAGAN. (c) GAMO. (d) DeepSMOTE.

TABLE II

PERFORMANCE OF DEEPSMOTE AND REFERENCE METHODS ON IMBALANCED TEST SET

	MNIST			*FMNIST*			*CIFAR*			*SVHN*			*CELEBA*		
	ACSA	GM	F1	ACSA	GM	F1	ACSA	GM	F1	ACSA	GM	F1	ACSA	GM	F1
SMOTE	81.48	83.99	82.44	67.94	74.84	67.12	28.02	50.08	29.58	70.18	76.33	71.80	60.29	70.48	60.03
AMDO	84.29	88.73	84.88	74.90	80.89	75.39	31.19	53.99	32.44	71.94	78.52	73.06	63.54	72.86	62.94
MC-CCR	86.19	92.04	86.46	78.58	86.17	79.03	32.83	56.68	33.91	72.01	80.94	74.26	65.23	77.14	64.88
MC-RBO	87.25	94.46	88.69	80.06	88.02	80.14	33.01	59.15	35.83	74.20	82.97	74.91	67.11	80.52	65.37
BAGAN	92.56	96.11	93.85	82.50	90.51	82.96	42.41	64.12	43.01	75.81	86.44	77.02	68.62	80.84	**68.33**
GAMO	95.45	97.61	95.11	83.05	90.76	83.00	44.72	65.72	**45.93**	75.07	86.00	76.68	66.06	79.11	64.85
DeepSMOTE	**96.16**	**98.11**	**96.44**	**84.88**	**91.63**	**83.79**	**45.26**	**66.13**	44.86	**79.59**	**88.67**	**80.71**	**72.40**	**82.91**	66.99

We use the Adam optimizer [105], with a 0.0002 learning rate. We implement DeepSMOTE in PyTorch with a NVIDIA GTX-2080 GPU. DeepSMOTE code is publicly available at https://github.com/dd1github/DeepSMOTE.

B. Experiment 1: Comparison With State-of-the-Art

1) Placement of Artificial Instances: One of the crucial elements of oversampling algorithms based on artificial instance generation lies in where in the feature space they place their instances. Random positioning is far from desirable, as we want to maintain the original properties of minority classes and enhance them in uncertain/difficult regions. Those regions are mostly class borders, overlapping areas, and small disjuncts. Therefore, the best oversampling methods focus on smart placement of instances that not only balances class distributions, but also reduces the learning difficulty. Fig. 2 depicts a 2-D projection of an imbalanced MNIST dataset, as well as the class distributions after oversampling with BAGAN, GAMO, and DeepSMOTE. In Fig. 2, we performed dimensionality reduction on the oversampled datasets by applying principal component analysis (PCA), followed by t-distributed stochastic neighborhood embedding (t-SNE) in order to better visualize the data instance distributions [106]. We can notice that both BAGAN and GAMO concentrate on saturating the distribution of each class independently, generating a significant number of artificial instances within the main distribution of each class. Such an approach balances the training data and may be helpful for some density-based classifiers. However, neither BAGAN nor GAMO focus on introducing artificial instances in a directed fashion to enhance class boundaries and improve the discrimination capabilities of a classifier trained on oversampled data. DeepSMOTE combines oversampling controlled by the class geometry with our penalty function to introduce instances in such a way that the error probability is reduced on minority classes. We hypothesize that leads to better placement of artificial instances and in result, as seen in the experimental comparison, more accurate classification.

2) Comparison With Pixel-Based Oversampling: The first group of reference algorithms is four state-of-the-art oversampling approaches. Tables II and III show their results for three metrics and two test set distribution types. We can clearly see that pixel-based oversampling is inferior to both GAN-based algorithms and DeepSMOTE. This allows us to conclude that pixel-based oversampling is not a good choice when dealing with complex and imbalanced images. Unsurprisingly, standard SMOTE performs worst of all of the evaluated algorithms, while three other methods try to offset their inability to handle spatial properties of data with advanced instance generation modules. Both MC-CCR and MC-RBO return the best results from all four tested algorithms, with MC-RBO coming close to GAN-based methods. This can be attributed to their compound oversampling solutions, which analyze the difficulty of instances and optimize the placement of new instances, while cleaning overlapping areas. However, this comes at the cost of very high computational complexity and challenging parameter tuning. DeepSMOTE returns superior balanced training sets compared to pixel-based approaches, while providing an intuitive and easy to tune architecture and, according to both nonparametric and Bayesian tests presented in Table IV, outperforms all pixel-based approaches in a statistically significant manner (**RQ1 answered**).

3) Comparison With GAN-Based Oversampling: Tables II and III show that regardless of the metric used, DeepSMOTE outperforms the baseline GAN-based models on all but two cases. Both these situations are happening with F1 measure and for different models (BAGAN displays a slightly higher F1 value on CelebA, while GAMO on CIFAR). It is important to note that for the same benchmarks, DeepSMOTE offers significantly higher ACSA and GM values than any of these reference algorithms, allowing us to conclude that F1 performance variation is not reflective on how DeepSMOTE can handle minority classes. We hypothesize that the success of DeepSMOTE can be attributed to better placement of artificial instances and empowering uncertainty

TABLE III
PERFORMANCE OF DEEPSMOTE AND REFERENCE METHODS ON BALANCED TEST SET (LONG-TAILED RECOGNITION SETUP)

	MNIST			*FMNIST*			*CIFAR*			*SVHN*			*CELEBA*		
	ACSA	GM	F1	ACSA	GM	F1	ACSA	GM	F1	ACSA	GM	F1	ACSA	GM	F1
SMOTE	87.98	89.99	85.02	70.58	76.39	68.06	27.93	42.81	25.10	68.19	74.48	64.28	48.19	56.39	42.19
AMDO	88.34	91.03	87.28	72.98	79.36	71.53	31.85	48.19	30.04	71.59	79.13	68.47	51.44	60.73	47.28
MC-CCR	90.83	93.18	91.22	75.78	81.04	74.39	33.48	51.18	32.88	74.29	81.62	72.49	58.46	65.39	57.91
MC-RBO	91.28	94.62	92.49	76.91	82.14	75.92	39.17	59.29	40.37	75.38	81.98	73.52	61.53	72.95	62.08
BAGAN	93.06	95.98	92.77	81.48	89.31	80.93	43.38	63.73	40.25	80.23	86.77	77.75	66.09	77.77	62.84
GAMO	95.52	97.47	95.47	83.03	90.26	82.50	44.89	65.30	43.35	80.53	87.17	78.21	66.00	77.71	63.01
DeepSMOTE	**96.09**	**97.80**	**96.03**	**83.63**	**90.61**	**83.27**	**45.38**	**65.30**	**43.35**	**80.94**	**87.39**	**78.73**	**69.88**	**80.38**	**69.19**

TABLE IV
RESULTS OF SHAFFER POST-HOC TESTS AND BAYESIAN WILCOXON SIGNED-RANK TESTS WITH RESPECT TO p-VALUES FOR PAIRWISE COMPARISON BETWEEN DEEPSMOTE AND THE REFERENCE OVERSAMPLING-BASED METHODS FOR THREE PERFORMANCE METRICS. WHEN A p-VALUE LOWER THAN 0.05 IS OBSERVED, WE MAY CONCLUDE THAT DEEPSMOTE DISPLAYS A STATISTICALLY SIGNIFICANTLY BETTER PERFORMANCE THAN THE REFERENCE RESAMPLING ALGORITHM. WE MERGED RESULTS FROM IMBALANCED AND LONG-TAILED RECOGNITION TEST SCENARIOS

DeepSMOTE	Shaffer post-hoc			Bayesian Wilcoxon signed-rank		
vs.	ACSA	GM	F1	ACSA	GM	F1
SMOTE	0.00001	0.00000	0.00001	0.00001	0.00000	0.00001
AMDO	0.00316	0.00048	0.00329	0.00172	0.00026	0.00188
MC-CCR	0.01042	0.00072	0.01003	0.00099	0.00018	0.00083
MC-RBO	0.02141	0.01625	0.02331	0.02007	0.01002	0.02106
BAGAN	0.03148	0.01352	0.03319	0.02581	0.01039	0.02606
GAMO	0.03204	0.01488	0.03582	0.02620	0.01721	0.02938

areas because oversampling is driven by our penalized loss function. DeepSMOTE has a potential to enhance decision boundaries, effectively reducing the classifier bias toward the majority classes. As DeepSMOTE is driven by the SMOTE-based approach for selecting and placing artificial instances, we ensure that the minority classes are enriched with diverse training data of high discriminative quality. Table IV shows that DeepSMOTE outperforms all GAN-based approaches in a statistically significant manner (**RQ2 answered**). This comes with an additional gain of directly generating higher-quality artificial images (as will be discussed in the following experiment).

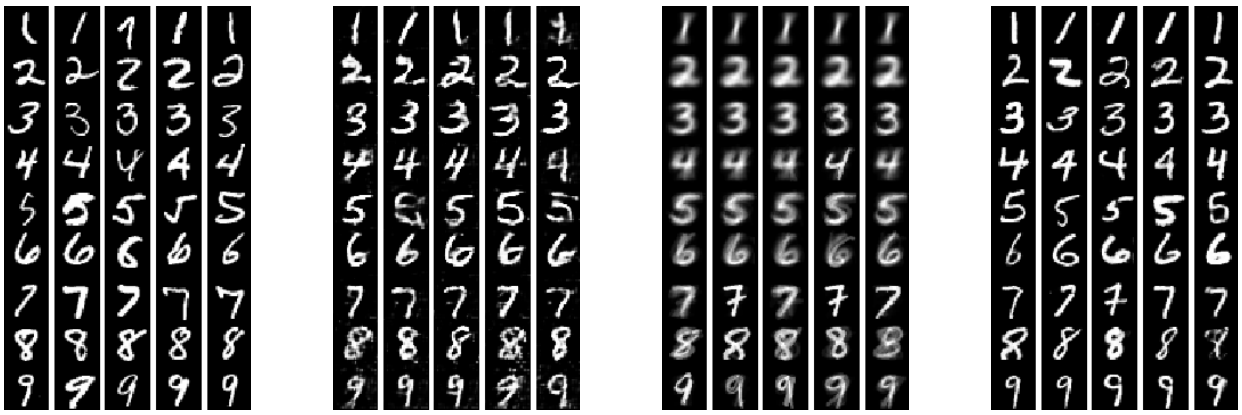
We note that the CIFAR-10 dataset was the most challenging benchmark for deep oversampling algorithms. We hypothesize that the reason why the models did not exhibit high accuracy on CIFAR-10 compared to the other datasets is because the CIFAR-10 classes do not have similar attributes. For example, in MNIST and SVHN, all classes are instances of digits and in the case of CelebA, all classes represent faces; whereas, in CIFAR-10, the classes are diverse (e.g., cat, dog, airplane, frog). Therefore, the models are not able to leverage information that they learn from the majority class (which has more examples) to the minority class (which contains fewer examples). In addition, we also noticed that, in some cases, there appears to be a significant overlap of CIFAR-10 class features.

4) Robustness to Mode Collapse: DeepSMOTE does not share some of the limitations of GAN-based oversampling, such as mode collapse. A widely used metric to determine the quality of generated images and measure mode collapse is the Frechet inception distance (FID) [107]. FID calculates a score that assesses the distance between a distribution of real and generated images based on feature activations in an Inception network [108]. A lower score, or distance between real and generated images, indicates more realistic images. Therefore, on a sample basis, we selected training images (real) and images generated by DeepSMOTE, BAGAN, and GAMO for the minority class in the CelebA dataset (class = bald). We calculated an FID score for each model and noted that DeepSMOTE's FID score (48.88) was substantially less than GAMO (213.66) and BAGAN (256.88).

5) Effects of Test Set Distribution: The final part of the first experiment focused on evaluating the role of class distributions in the test set. In the domain of learning from imbalanced data, the test set follows the distribution of the training set, in order to reflect the actual class disproportions [1]. This also impacts the calculation of several cost-sensitive measures that more severely penalize the errors on minority classes [2]. However, the recently emerging field of long-tailed recognition follows a different testing protocol [78]. In this scenario of extreme MC imbalance, the training set is skewed, but test sets for most benchmarks are balanced. As DeepSMOTE aims to be a universal approach for imbalanced data preprocessing and resampling, we evaluated its performance in both scenarios. Table II reports results for the traditional imbalanced setup, while Table III reflects the long-tailed recognition setup. We can see that DeepSMOTE excels in both scenarios, confirming our previous observations on its benefits over pixel-based and GAN-based approaches. It is interesting to see that for the long-tailed setup, DeepSMOTE returns slightly better F1 performance on the CIFAR10 and CelebA datasets. This can be explained by the way the F1 measure is calculated, as it gives equal importance to precision and recall. When dealing with a balanced test set, DeepSMOTE was able to return even better performance on these two metrics. For all other metrics and datasets, DeepSMOTE showcases similar trends for imbalanced and balanced test sets. This allows us to conclude that DeepSMOTE is a suitable and effective solution for both imbalanced and long-tailed recognition scenarios (**RQ3 answered**).

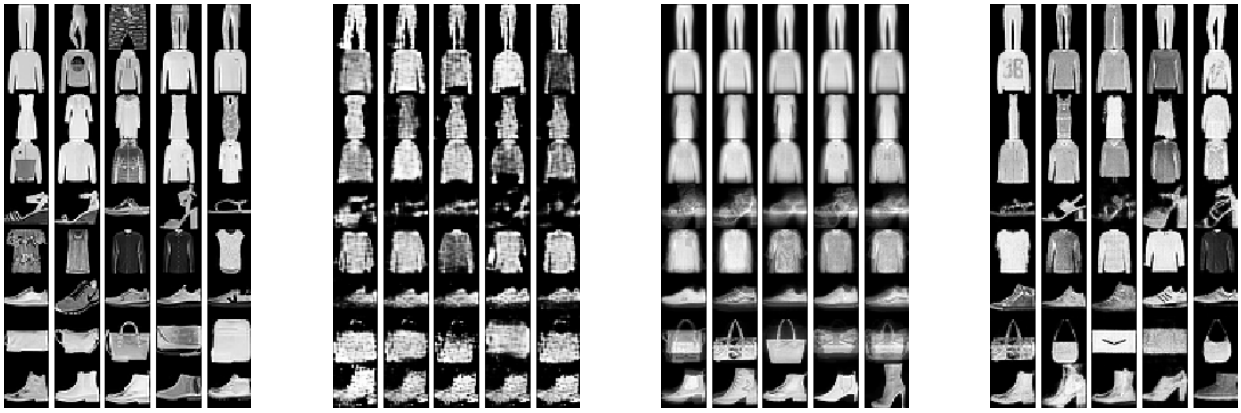
C. Experiment 2: Quality of Artificially Generated Images

1) Quality of Images Generated by DeepSMOTE: Figs. 3–7 present the artificially generated images for all five benchmark datasets by BAGAN, GAMO, and the DeepSMOTE.



(a) (b) (c) (d)

Fig. 3. MNIST minority class images, with rows corresponding to digit classes. (a) Originals. (b) BAGAN. (c) GAMO. (d) DeepSMOTE.



(a) (b) (c) (d)

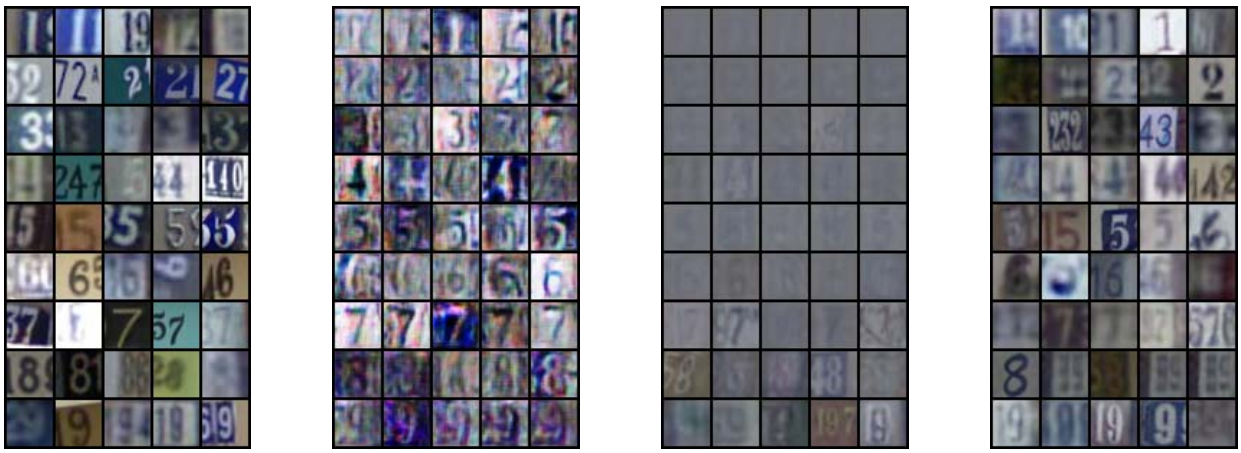
Fig. 4. FMNIST minority class images: trouser/pullover/dress/coat/sandal/shirt/sneaker/bag/ankle boot. (a) Originals. (b) BAGAN. (c) GAMO. (d) DeepSMOTE.

(a) (b) (c) (d)

Fig. 5. CIFAR-10 minority class images: automobile/bird/cat/deer/dog/frog/horse/ship/truck. (a) Originals. (b) BAGAN. (c) GAMO. (d) DeepSMOTE.

We can see the quality of DeepSMOTE-generated images. This can be attributed to DeepSMOTE using an efficient encoding/decoding architecture with an enhanced loss function, as well as preserving class topology via metric-based instance imputation. We note that in the case of GAMO, we present images that were used for classification purposes and not images generated by the GAMO2PIX method, so as to provide a direct comparison of GAMO training images to training images generated by BAGAN and DeepSMOTE. The outcomes of both experiments demonstrate that DeepSMOTE generates artificial images that are both information-rich (i.e., they improve the discriminative ability of deep classifiers and they counter majority bias) and are of high visual quality (**RQ4 answered**).

2) Insights Into DeepSMOTE Image Generation: Fig. 8 depicts the process of generating new artificial images by combining the base image with one of its nearest neighbors. The ratio of which each image influences the combination procedure is randomly established by the scaling factor of the SMOTE algorithm (which draws values 0–1 for how close



(a) (b) (c) (d)

Fig. 6. SVHN minority class images, with rows corresponding to digit classes. (a) Originals. (b) BAGAN. (c) GAMO. (d) DeepSMOTE.

(a) (b) (c) (d)

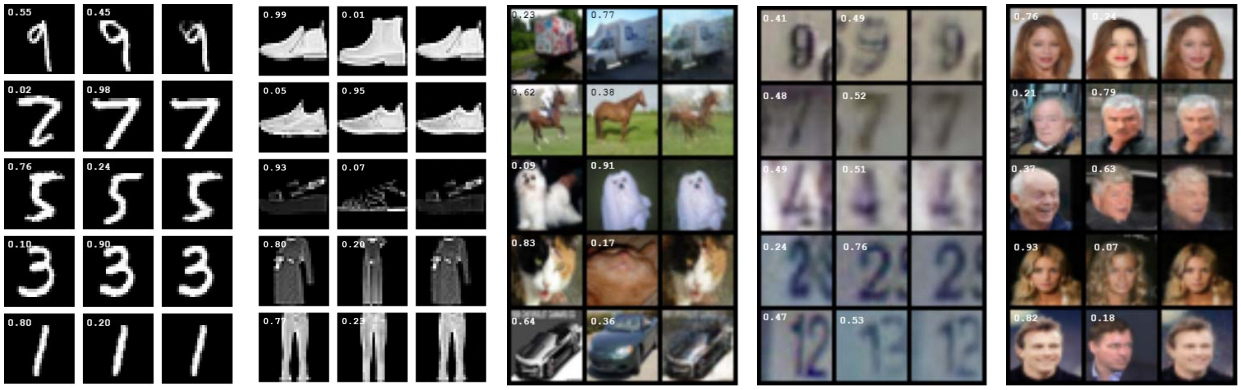
Fig. 7. CELEBA minority class images: brown hair/blond hair/gray hair/bald. (a) Originals. (b) BAGAN. (c) GAMO. (d) DeepSMOTE.

the new artificial image should resemble base and neighbor images). As DeepSMOTE operates on an encoded domain of images, the new artificial images are being generated by a convex combination of target image and its nearest neighbor. In Fig. 8, we can see how different values of the scaling factor lead to diverse types of output images—some more similar to base image, some more similar to nearest neighbor, and some bearing distinctive features of both images. We hypothesize that this diversity of generated images may be responsible for excellent performance of DeepSMOTE. It seems worthwhile to investigate in the future a directed way of controlling the scaling factor in order to obtain best artificially enriched and diversified datasets.

D. Experiment 3: Robustness and Stability Under Varied Imbalance Ratios

1) Robustness to Varying Imbalance Ratios: One of the most challenging aspects of learning from imbalanced data lies in creating robust algorithms that can manage various data-level difficulties. Many existing resampling methods return very good results only under specific conditions or under a narrow range of imbalance ratios. Therefore, in order to obtain a complete picture of the performance of DeepSMOTE, we analyze its robustness to varying imbalance ratios in the range of [20, 400]. Fig. 9 depicts the relationship between the three performance metrics and increasing imbalance ratio on five used benchmarks. This experiment allows us not only to evaluate DeepSMOTE and the reference methods under various skewed scenarios, but also offers a bird-eye view on the characteristics of the performance curves displayed by each examined resampling method. An ideal resampling algorithm should be characterized by a high robustness to increasing imbalance ratios, display stable, or small, performance degradation with increased class disproportions. Sharp and significant performance declines indicate breaking points for resampling methods and show when a given algorithm stops being capable of generating useful instances and countering class imbalance.

Analyzing Fig. 9 allows us to draw several interesting conclusions. First, Experiment 1 shows that pixel-based solutions are inferior to their GAN-based counterparts. However, we can see that this observation does not hold for extreme values of imbalance ratios. When the disproportion among classes increases, pixels-based methods (especially MC-CCR and MC-RBO) start displaying increased robustness. On the contrary, the two GAN-based methods are more sensitive to an increased imbalance ratio and we can observe a more rapid decline in their predictive power. This can be explained by two factors: the method by which resampling approaches use the original instances and the issue of small sample size. The former factor shows the limitations of GAN-based methods. While they focus on instance generation and creating high-quality images, they do not possess more sophisticated mechanisms on where to precisely inject new artificial instances. With higher imbalance ratios, this placement starts playing a crucial role, as the classifier needs to handle more and more difficult bias. Current GAN-based models use relatively simplistic mechanisms for this issue. On the contrary, pixel-based methods rely on more sophisticated mechanisms (e.g., MC-CCR uses an energy-based function, while MC-RBO uses local optimization for positioning their artificial instances). With increasing imbalance ratios, such mechanisms start to dominate simpler GAN-based solutions, making pixel-based approaches more robust to extreme imbalance ratios. The latter factor of small sample size also strongly affects GAN-based



(a) (b) (c) (d) (e)

Fig. 8. Illustration of DeepSMOTE artificial image generation by convex combination of two images on five examined datasets. Shown in the illustration are five classes with three examples each. From left to right, the examples are: 1) base image; 2) nearest neighbor selected; and 3) combined image. The combined image is based on a scaling factor between the base and nearest neighbor given by the SMOTE algorithm. (a) MNIST. (b) FMNIST. (c) CIFAR-10. (d) SVHN. (e) CELEBA.

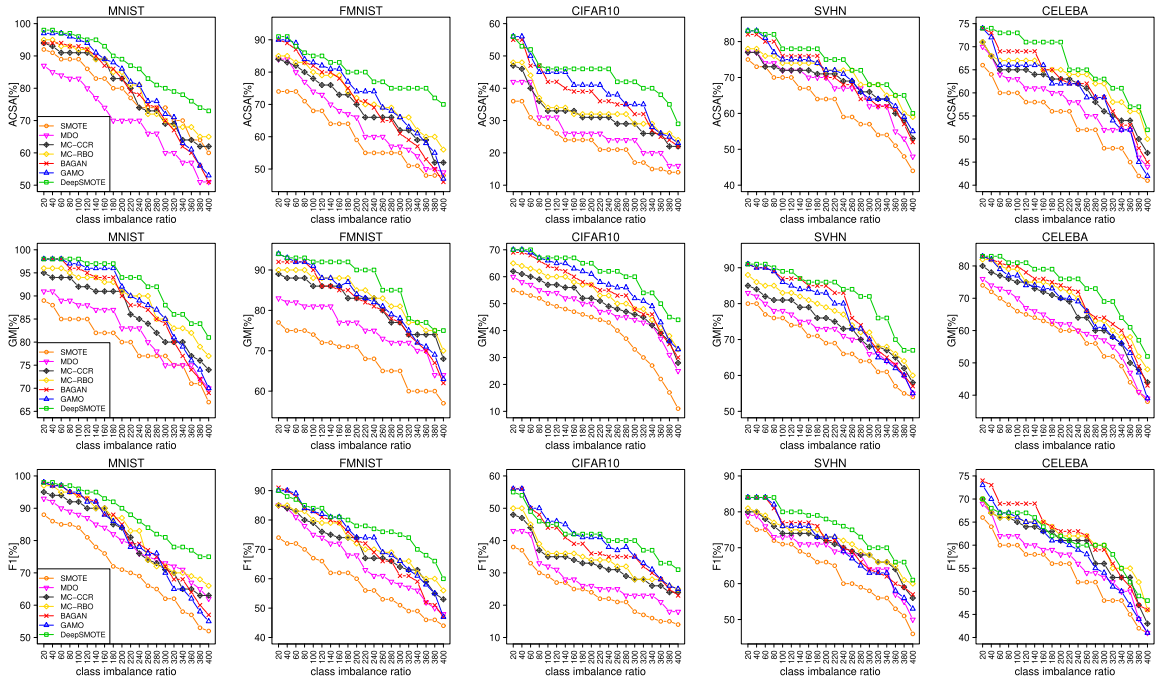


Fig. 9. Robustness to increasing imbalance ratios for DeepSMOTE and reference resampling methods.

algorithms. With extreme imbalance, we have less and less minority instances at our disposal, making it more difficult to train effective GANs.

Compared to both pixel-based and GAN-based approaches, DeepSMOTE displays an excellent robustness even to the highest imbalance ratios. We can see that DeepSMOTE is able to effectively handle such a challenging scenario, displaying the lowest decline of performance on all evaluated metrics. This can be attributed to the fact that SMOTE generates artificial instances following class geometry, while using only nearest neighbors for instance generation. This allows us to conclude that DeepSMOTE is not affected as strongly as GAN-based approaches by a small sample size and the need for smart placement of artificial instances, leading to excellent robustness (**RQ5 answered**).

2) Model Stability Under Varying Imbalance Ratios: Another important aspect of evaluating modern resampling algorithms is their stability. We need to evaluate how a given model reacts to small perturbations in data, as we want to evaluate its generalization capabilities. Models that display high variance under such small changes cannot be treated as stable and thus should not be preferred. It is especially crucial in the learning from imbalanced data area, as we want to select a resampling algorithm that will generate information-rich artificial instances under any data permutations.

In order to evaluate this, we have measured the spread of performance metrics for DeepSMOTE and GAN-based algorithms under 20 repetitions of fivefold cross validation. During each CV repetition, minority classes were created randomly from the original balanced benchmarks. This ensured that we not only measure the stability to training data permutation within a single dataset instance, but we also measure the possibility of creating minority classes with instances of varying difficulties. Fig. 10 shows the plots of three resampling methods with shaded regions denoting the standard deviation of results. GAN-based approaches display increasing variance

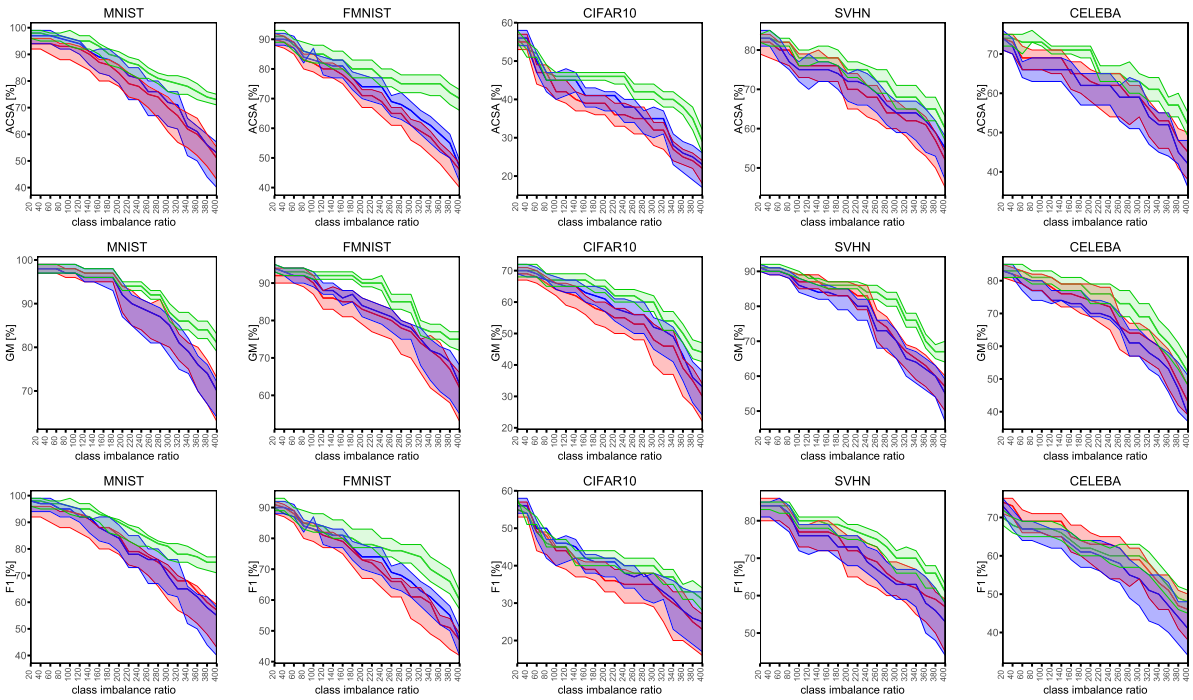


Fig. 10. Relationship between imbalance ratio and model stability (expressed as std. deviation) for DeepSMOTE and GAN-based models obtained from 20 repetitions of fivefold CV.

under higher imbalance ratios, showing that those approaches cannot be considered as stable models for challenging imbalanced data problems. DeepSMOTE returned the lowest variance within those metrics, showcasing the high stability of our resampling algorithm. This information enriches our previous observation regarding the robustness of DeepSMOTE. Joint analysis of Figs. 9 and 10 allows us to conclude that DeepSMOTE can handle extreme imbalance among classes, while generating stable models under challenging conditions (**RQ6 answered**).

VI. Discussion

1) *Simple design is effective:* DeepSMOTE is an effective approach for countering class imbalance and training skew-insensitive deep learning classifiers. It outperforms state-of-the-art solutions and is able to work on raw image representations. DeepSMOTE is composed of three components: an encoder/decoder is combined with a dedicated loss function and SMOTE-based resampling. This simplicity makes it an easy to understand, transparent, yet very powerful method for handling class imbalance in deep learning.
2) *Dedicated data encoding for artificial instance generation:* DeepSMOTE uses a two-phase approach that first trains a dedicated encoder/decoder architecture and then uses it to obtain a high-quality embedding for the oversampling procedure. This allows us to find the best possible data representations for oversampling, allowing SMOTE-based generation to enrich the training set of minority classes.
3) *Effective placement of artificial instances:* DeepSMOTE follows the geometric properties of minority classes, creating artificial instances on borders among classes. We hypothesize that this leads to improved training of discriminative models on datasets balanced with DeepSMOTE, which in turn leads to improved classification accuracy and reduced bias toward majority classes.
4) *Superiority over pixel-based and GAN-based algorithms:* DeepSMOTE outperforms state-of-the-art resampling approaches. By being able to work on raw images and extracting features from them, DeepSMOTE can generate more meaningful artificial instances than pixel-based approaches, even while using relatively simpler rules for instance generation. By using efficient and dedicated data embeddings, DeepSMOTE can better enrich minority classes under varying imbalance ratios than GAN-based solutions.
5) *Easy to use:* One of the reasons behind the tremendous success of the original SMOTE algorithm was its easy and intuitive usage. DeepSMOTE follows these steps, as it is not only accurate, but also an attractive off-the-shelf solution. Our method is easy to tune and use on any data, both as a black-box solution and as a steppingstone for developing novel and robust deep learning architectures. As deep learning is being used by a wider and wider interdisciplinary audience, such a characteristic is highly sought after.
6) *High quality of generated images:* DeepSMOTE can return high-quality artificial images that under visual inspection do not differ from real ones. This makes DeepSMOTE an all-around approach, since the generated images are both sharp and information-rich.
7) *Excellent robustness and stability:* DeepSMOTE can handle extreme imbalance ratios, while being robust to small sample size and within-data variance. DeepSMOTE is less prone to variations in training data than any of the reference methods. It is

a stable oversampling approach that is suitable for enhancing deep learning models deployed in real-world applications.

VII. Conclusion

Summary: We proposed DeepSMOTE, a novel and transformative model for imbalanced data, that fuses the highly popular SMOTE algorithm with deep learning methods. DeepSMOTE is an efficient oversampling solution for training deep architectures on imbalanced data distributions. It can be seen as a data-level solution to class imbalance, as it creates artificial instances that balance the training set, which can then be used to train any deep classifier without suffering from bias. DeepSMOTE uniquely satisfies three crucial characteristics of a successful resampling algorithm in the domain of learning from images: ability to operate on raw images, creation of efficient low-dimensional embeddings, and generation of high-quality artificial images. This was made possible by a novel architecture that combined an encoder/decoder framework with SMOTE-based oversampling and an enhanced loss function. Extensive experimental studies show that DeepSMOTE not only outperforms state-of-the-art pixel-based and GAN-based oversampling algorithms, but also offers unparalleled robustness to varying imbalance ratios with high model stability, while generating artificial images of excellent quality.

Future work: Our next efforts will focus on enhancing DeepSMOTE with information regarding class-level and instance-level difficulties, which will allow it to better tackle challenging regions of the feature space. We plan to enhance our dedicated loss function with instance-level penalties for focusing the encoder/decoder training on instances that display borderline/overlapping characteristics, while discarding outliers and noisy instances. Such a compound skew-insensitive loss function will bridge the worlds between data-level and algorithm-level approaches to learning from imbalanced data. Furthermore, we want to make DeepSMOTE suitable for continual and lifelong learning scenarios, where there is a need for handling dynamic class ratios and generating new artificial instances. We envision that DeepSMOTE may not only help to counter online class imbalance, but also help increase the robustness of lifelong learning models to catastrophic forgetting. Finally, we plan to extend DeepSMOTE to incorporate other data modalities, such as graphs and text data.

References

[1] B. Krawczyk, "Learning from imbalanced data: Open challenges and future directions," *Prog. Artif. Intell.*, vol. 5, no. 4, pp. 221–232, 2016.

[2] A. Fernández, S. García, M. Galar, R. C. Prati, B. Krawczyk, and F. Herrera, *Learning From Imbalanced Data Sets*. Switzerland: Springer, 2018, doi: 10.1007/978-3-319-98074-4.

[3] L. Korycki and B. Krawczyk, "Concept drift detection from multi-class imbalanced data streams," in *Proc. IEEE 37th Int. Conf. Data Eng. (ICDE)*, Chania, Greece, Apr. 2021, pp. 1068–1079.

[4] L. Korycki and B. Krawczyk, "Low-dimensional representation learning from imbalanced data streams," in *Proc. Adv. Knowl. Discovery Data Mining, 25th Pacific-Asia Conf. (PAKDD)*, in Lecture Notes in Computer Science, vol. 12712. Researchgate.net, 2021, pp. 629–641.

[5] F. Bao, Y. Deng, Y. Kong, Z. Ren, J. Suo, and Q. Dai, "Learning deep landmarks for imbalanced classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 8, pp. 2691–2704, Aug. 2020.

[6] L. A. Bugnon, C. Yones, D. H. Milone, and G. Stegmayer, "Deep neural architectures for highly imbalanced data in bioinformatics," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 8, pp. 2857–2867, Aug. 2020.

[7] X.-Y. Jing *et al.*, "Multiset feature learning for highly imbalanced data classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 1, pp. 139–156, Jan. 2021.

[8] Z. Wang, X. Ye, C. Wang, Y. Wu, C. Wang, and K. Liang, "RSDNE: Exploring relaxed similarity and dissimilarity from completely-imbalanced labels for network embedding," in *Proc. 32nd AAAI Conf. Artif. Intell. (AAAI), 30th Innov. Appl. Artif. Intell. (IAAI), 8th AAAI Symp. Educ. Adv. Artif. Intell. (EAAI)*, New Orleans, LA, USA, Feb. 2018, pp. 475–482.

[9] L. Korycki and B. Krawczyk, "Class-incremental experience replay for continual learning under concept drift," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2021, pp. 3649–3658.

[10] C. Wu and H. Li, "Conditional transferring features: Scaling GANs to thousands of classes with 30% less high-quality data for training," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Glasgow, U.K., Jul. 2020, pp. 1–8.

[11] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, no. 28, pp. 321–357, Jun. 2006.

[12] C. Huang, Y. Li, C. C. Loy, and X. Tang, "Deep imbalanced learning for face recognition and attribute prediction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 11, pp. 2781–2794, Nov. 2020.

[13] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," 2018, *arXiv:1802.05957*.

[14] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," 2016, *arXiv:1606.03498*.

[15] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved training of wasserstein GANs," 2017, *arXiv:1704.00028*.

[16] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 214–223.

[17] M. Koziarski, "Radial-based undersampling for imbalanced data classification," *Pattern Recognit.*, vol. 102, Jun. 2020, Art. no. 107262.

[18] W.-C. Lin, C.-F. Tsai, Y.-H. Hu, and J.-S. Jhang, "Clustering-based undersampling in class-imbalanced data," *Inf. Sci.*, vols. 409–410, pp. 17–26, Oct. 2017.

[19] P. Vuttipittayamongkol and E. Elyan, "Neighbourhood-based undersampling approach for handling imbalanced and overlapped data," *Inf. Sci.*, vol. 509, pp. 47–70, Jan. 2020.

[20] G. Douzas and F. Bação, "Geometric SMOTE a geometrically enhanced drop-in replacement for SMOTE," *Inf. Sci.*, vol. 501, pp. 118–135, Oct. 2019.

[21] H. He, Y. Bai, E. A. Garcia, and S. Li, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," in *Proc. IEEE Int. Joint Conf. Neural Netw., IEEE World Congr. Comput. Intell.*, Hong Kong, Jun. 2008, pp. 1322–1328.

[22] X. W. Liang, A. P. Jiang, T. Li, Y. Y. Xue, and G. T. Wang, "LR-SMOTE—An improved unbalanced data set oversampling based on K-means and SVM," *Knowl.-Based Syst.*, vol. 196, May 2020, Art. no. 105845.

[23] Y. Yang, Q. Zhao, L. Ruan, Z. Gao, Y. Huo, and X. Qiu, "Oversampling methods combined clustering and data cleaning for imbalanced network data," *Intell. Autom. Soft Comput.*, vol. 26, no. 5, pp. 1139–1155, 2020.

[24] Y. Xu, X. Meng, Y. Li, and X. Xu, "Research on privacy disclosure detection method in social networks based on multi-dimensional deep learning," *Comput., Mater. Continua*, vol. 62, no. 1, pp. 137–155, 2020.

[25] M. Koziarski, B. Krawczyk, and M. Wozniak, "Radial-based oversampling for noisy imbalanced data classification," *Neurocomputing*, vol. 343, pp. 19–33, May 2019.

[26] M. Koziarski and M. Wozniak, "CCR: A combined cleaning and resampling algorithm for imbalanced data classification," *Int. J. Appl. Math. Comput. Sci.*, vol. 27, no. 4, pp. 727–736, Jan. 2017.

[27] K. Boonchuay, K. Sinapiromsaran, and C. Lursinsap, "Decision tree induction based on minority entropy for the class imbalance problem," *Pattern Anal. Appl.*, vol. 20, no. 3, pp. 769–782, Aug. 2017.

[28] D. Cieslak, T. Hoens, N. Chawla, and W. Kegelmeyer, "Hellinger distance decision trees are robust and skew-insensitive," *Data Mining Knowl. Discovery*, vol. 24, no. 1, pp. 136–158, 2012.

[29] F. Li, X. Zhang, X. Zhang, C. Du, Y. Xu, and Y.-C. Tian, "Cost-sensitive and hybrid-attribute measure multi-decision tree over imbalanced data sets," *Inf. Sci.*, vol. 422, pp. 242–256, Jan. 2018.

[30] S. Datta and S. Das, "Multiobjective support vector machines: Handling class imbalance with Pareto optimality," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 5, pp. 1602–1608, May 2019.

[31] Q. Fan, Z. Wang, D. Li, D. Gao, and H. Zha, "Entropy-based fuzzy support vector machine for imbalanced datasets," *Knowl.-Based Syst.*, vol. 115, pp. 87–99, Jan. 2017.

[32] K. Qi, H. Yang, Q. Hu, and D. Yang, "A new adaptive weighted imbalanced data classifier via improved support vector machines with high-dimension nature," *Knowl.-Based Syst.*, vol. 185, Dec. 2019, Art. no. 104933.

[33] Q. Dong, S. Gong, and X. Zhu, "Imbalanced deep learning by minority class incremental rectification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 6, pp. 1367–1381, Jun. 2019.

[34] Y.-H. Liu, C.-L. Liu, and S.-M. Tseng, "Deep discriminative features learning and sampling for imbalanced data problem," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Singapore, Nov. 2018, pp. 1146–1151.

[35] P. Wang, F. Su, Z. Zhao, Y. Guo, Y. Zhao, and B. Zhuang, "Deep class-skewed learning for face recognition," *Neurocomputing*, vol. 363, pp. 35–45, Oct. 2019.

[36] C. Cao and Z. Wang, "IMCStacking: Cost-sensitive stacking learning with feature inverse mapping for imbalanced problems," *Knowl.-Based Syst.*, vol. 150, pp. 27–37, Jun. 2018.

[37] S. H. Khan, M. Hayat, M. Bennamoun, F. A. Sohel, and R. Togneri, "Cost-sensitive learning of deep feature representations from imbalanced data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 8, pp. 3573–3587, Aug. 2018.

[38] C. Zhang, K. C. Tan, H. Li, and G. S. Hong, "A cost-sensitive deep belief network for imbalanced classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 1, pp. 109–122, Jan. 2019.

[39] D. Devi, S. K. Biswas, and B. Purkayastha, "Learning in presence of class imbalance and class overlapping by using one-class SVM and undersampling technique," *Connection Sci.*, vol. 31, no. 2, pp. 105–142, 2019.

[40] B. Krawczyk, M. Wozniak, and F. Herrera, "Weighted one-class classification for different types of minority class examples in imbalanced data," in *Proc. IEEE Symp. Comput. Intell. Data Mining (CIDM)*, Orlando, FL, USA, Dec. 2014, pp. 337–344.

[41] B. Pérez-Sánchez, O. Fontenla-Romero, and N. Sánchez-Maroño, "Selecting target concept in one-class classification for handling class imbalance problem," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Killarney, Ireland, Jul. 2015, pp. 1–8.

[42] M. Woźniak, M. Graña, and E. Corchado, "A survey of multiple classifier systems as hybrid systems," *Inf. Fusion*, vol. 16, pp. 3–17, May 2014.

[43] J. F. Díez-Pastor, J. J. Rodríguez, C. García-Osorio, and L. I. Kuncheva, "Random balance: Ensembles of variable priors classifiers for imbalanced data," *Knowl.-Based Syst.*, vol. 85, pp. 96–111, Sep. 2015.

[44] J. Błaszczyński and J. Stefanowski, "Neighbourhood sampling in bagging for imbalanced data," *Neurocomputing*, vol. 150, pp. 529–542, Feb. 2015.

[45] S. Hido, H. Kashima, and Y. Takahashi, "Roughly balanced bagging for imbalanced data," *Stat. Anal. Data Mining*, vol. 2, nos. 5–6, pp. 412–426, 2009.

[46] S. E. Roshan and S. Asadi, "Improvement of bagging performance for classification of imbalanced datasets using evolutionary multi-objective optimization," *Eng. Appl. Artif. Intell.*, vol. 87, Jan. 2020, Art. no. 103319.

[47] S. Datta, S. Nag, and S. Das, "Boosting with lexicographic programming: Addressing class imbalance without cost tuning," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 5, pp. 883–897, May 2020.

[48] B. Krawczyk, M. Galar, Ł. Jeleń, and F. Herrera, "Evolutionary undersampling boosting for imbalanced classification of breast cancer malignancy," *Appl. Soft Comput.*, vol. 38, pp. 714–726, Jan. 2016.

[49] X. Zhang, Y. Zhuang, W. Wang, and W. Pedrycz, "Transfer boosting with synthetic instances for class imbalanced object recognition," *IEEE Trans. Cybern.*, vol. 48, no. 1, pp. 357–370, Jan. 2018.

[50] B. Krawczyk, M. Woźniak, and G. Schaefer, "Cost-sensitive decision tree ensembles for effective imbalanced classification," *Appl. Soft Comput.*, vol. 14, pp. 554–562, Jan. 2014.

[51] X. Tao *et al.*, "Self-adaptive cost weights-based support vector machine cost-sensitive ensemble for imbalanced data classification," *Inf. Sci.*, vol. 487, pp. 31–56, Jun. 2019.

[52] Q. Zhou, H. Zhou, and T. Li, "Cost-sensitive feature selection using random forest: Selecting low-cost subsets of informative features," *Knowl. Based Syst.*, vol. 95, pp. 1–11, Mar. 2016.

[53] J. F. Díez-Pastor, J. J. Rodríguez, C. I. García-Osorio, and L. I. Kuncheva, "Diversity techniques improve the performance of the best imbalance learning ensembles," *Inf. Sci.*, vol. 325, pp. 98–117, Dec. 2015.

[54] A. Roy, R. M. O. Cruz, R. Sabourin, and G. D. C. Cavalcanti, "A study on combining dynamic selection and data preprocessing for imbalance learning," *Neurocomputing*, vol. 286, pp. 179–192, Apr. 2018.

[55] P. Zyblewski, R. Sabourin, and M. Woźniak, "Preprocessed dynamic classifier ensemble selection for highly imbalanced drifted data streams," *Inf. Fusion*, vol. 66, pp. 138–154, Feb. 2021.

[56] M. A. Souza, G. D. C. Cavalcanti, R. M. O. Cruz, and R. Sabourin, "On evaluating the online local pool generation method for imbalance learning," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Budapest, Hungary, Jul. 2019, pp. 1–8.

[57] C. Bellinger, R. Corizzo, and N. Japkowicz, "Remix: Calibrated resampling for class imbalance in deep learning," *CoRR*, vol. abs/2012.02312, pp. 1–9, Dec. 2020.

[58] V. A. Fajardo *et al.*, "On oversampling imbalanced data with deep conditional generative models," *Expert Syst. Appl.*, vol. 169, May 2021, Art. no. 114463.

[59] C. Bellinger, C. Drummond, and N. Japkowicz, "Manifold-based synthetic oversampling with manifold conformance estimation," *Mach. Learn.*, vol. 107, no. 3, pp. 605–637, 2018.

[60] I. J. Goodfellow *et al.*, "Generative adversarial networks," 2014, *arXiv:1406.2661*.

[61] D. P Kingma and M. Welling, "Auto-encoding variational Bayes," 2013, *arXiv:1312.6114*.

[62] I. Tolstikhin, O. Bousquet, S. Gelly, and B. Schoelkopf, "Wasserstein auto-encoders," 2017, *arXiv:1711.01558*.

[63] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2223–2232.

[64] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of StyleGAN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 8110–8119.

[65] M. Watter, J. T. Springenberg, J. Boedecker, and M. Riedmiller, "Embed to control: A locally linear latent dynamics model for control from raw images," 2015, *arXiv:1506.07365*.

[66] R. Bonatti, R. Madaan, V. Vineet, S. Scherer, and A. Kapoor, "Learning visuomotor policies for aerial navigation using cross-modal representations," 2019, *arXiv:1909.06993*.

[67] X. Yi, E. Walia, and P. Babyn, "Generative adversarial network in medical imaging: A review," *Med. Image Anal.*, vol. 58, Dec. 2019, Art. no. 101552.

[68] Z. Hu, Z. Yang, R. Salakhutdinov, and E. P. Xing, "On unifying deep generative models," 2017, *arXiv:1706.00550*.

[69] C. Doersch, "Tutorial on variational autoencoders," 2016, *arXiv:1606.05908*.

[70] Y. Wu, J. Donahue, D. Balduzzi, K. Simonyan, and T. Lillicrap, "LOGAN: Latent optimisation for generative adversarial networks," 2019, *arXiv:1912.00953*.

[71] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets," 2016, *arXiv:1606.03657*.

[72] D. Pfau and O. Vinyals, "Connecting generative adversarial networks and actor-critic methods," 2016, *arXiv:1610.01945*.

[73] S. Wang, W. Liu, J. Wu, L. Cao, Q. Meng, and P. J. Kennedy, "Training deep neural networks on imbalanced data sets," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Vancouver, BC, Canada, Jul. 2016, pp. 4368–4374.

[74] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Oct. 2017, pp. 2999–3007.

[75] Y. S. Resheff, A. Mandelbom, and D. Weinshall, "Controlling imbalanced error in deep learning with the log bilinear loss," in *Proc. 1st Int. Workshop Learn. Imbalanced Domains, Theory Appl. (LIDTA PKDD/ECML)*, Skopje, Macedonia, vol. 74, Sep. 2017, pp. 141–151.

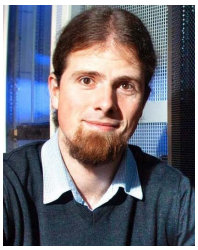
[76] Z. Zhang and M. R. Sabuncu, "Generalized cross entropy loss for training deep neural networks with noisy labels," in *Proc. Adv. Neural Inf. Process. Syst., Annu. Conf. Neural Inf. Process. Syst. (NeurIPS)*, Montréal, QC, Canada, Dec. 2018, pp. 8792–8802.

[77] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie, "Class-balanced loss based on effective number of samples," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 9268–9277.

[78] J. Tan *et al.*, "Equalization loss for long-tailed object recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Seattle, WA, USA, Jun. 2020, pp. 11659–11668.
[79] S. Abdelkarim, P. Achlioptas, J. Huang, B. Li, K. Church, and M. Elhoseiny, "Long-tail visual relationship recognition with a visiolinguistic hubless loss," *CoRR*, vol. abs/2004.00436, pp. 1–26, Jun. 2020.
[80] X. Zhang, Z. Fang, Y. Wen, Z. Li, and Y. Qiao, "Range loss for deep face recognition with long-tailed training data," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Oct. 2017, pp. 5419–5428.
[81] B. Zhou, Q. Cui, X.-S. Wei, and Z.-M. Chen, "BBN: Bilateral-branch network with cumulative learning for long-tailed visual recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Seattle, WA, USA, Jun. 2020, pp. 9716–9725.
[82] S. Sharma, N. Yu, M. Fritz, and B. Schiele, "Long-tailed recognition using class-balanced experts," *CoRR*, vol. abs/2004.03706, pp. 86–100, Oct. 2020.
[83] M. A. Jamal, M. Brown, M.-H. Yang, L. Wang, and B. Gong, "Rethinking class-balanced methods for long-tailed visual recognition from a domain adaptation perspective," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Seattle, WA, USA, Jun. 2020, pp. 7607–7616.
[84] S. Ando and C. Y. Huang, "Deep over-sampling framework for classifying imbalanced data," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, 2017, pp. 770–785.
[85] A. Fernández, S. Garcia, F. Herrera, and N. V. Chawla, "SMOTE for learning from imbalanced data: Progress and challenges, marking the 15-year anniversary," *J. Artif. Intell. Res.*, vol. 61, pp. 863–905, Apr. 2018.
[86] J. M. Johnson and T. M. Khoshgoftaar, "Survey on deep learning with class imbalance," *J. Big Data*, vol. 6, no. 1, pp. 1–54, Dec. 2019.
[87] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2015, *arXiv:1511.06434*.
[88] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
[89] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms," 2017, *arXiv:1708.07747*.
[90] A. Krizhevsky *et al.*, "Learning multiple layers of features from tiny images," Univ. Toronto, Toronto, CA, USA, 2009.
[91] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," 2011.
[92] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3730–3738.
[93] X. Yang, Q. Kuang, W. Zhang, and G. Zhang, "AMDO: An over-sampling technique for multi-class imbalanced problems," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 9, pp. 1672–1685, Sep. 2018.
[94] M. Koziarski, M. Woźniak, and B. Krawczyk, "Combined cleaning and resampling algorithm for multi-class imbalanced data with label noise," *Knowl.-Based Syst.*, vol. 204, Sep. 2020, Art. no. 106223.
[95] B. Krawczyk, M. Koziarski, and M. Wozniak, "Radial-based oversampling for multiclass imbalanced data classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 8, pp. 2818–2831, Aug. 2020.
[96] G. Mariani, F. Scheidegger, R. Istrate, C. Bekas, and C. Malossi, "BAGAN: Data augmentation with balancing GAN," 2018, *arXiv:1803.09655*.
[97] S. S. Mullick, S. Datta, and S. Das, "Generative adversarial minority oversampling," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1695–1704.
[98] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[99] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Inf. Process. Manag.*, vol. 45, no. 4, pp. 427–437, 2009.
[100] K. Stapor, P. Ksieniewicz, S. García, and M. Woźniak, "How to design the fair experimental classifier evaluation," *Appl. Soft Comput.*, vol. 104, Jun. 2021, Art. no. 107219.
[101] A. Benavoli, G. Corani, J. Demšar, and M. Zaffalon, "Time for a change: A tutorial for comparing multiple classifiers through Bayesian analysis," *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 2653–2688, Jan. 2017.
[102] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.
[103] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. ICML*, 2013, vol. 30, no. 1, p. 3.
[104] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. ICML*, 2010, pp. 807–814.
[105] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
[106] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 2579–2605, 2008.
[107] M. Lucic, K. Kurach, M. Michalski, S. Gelly, and O. Bousquet, "Are GANs created equal? A large-scale study," 2017, *arXiv:1711.10337*.
[108] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 6629–6640.

Damien Dablain is currently pursuing a Ph.D. degree with the Department of Computer Science and Engineering, University of Notre Dame, Notre Dame, IN, USA.

His research interests include generative models, imbalanced learning, adversarial examples, and explainable artificial intelligence (AI).



Bartosz Krawczyk (Member, IEEE) received the M.Sc. and Ph.D. degrees from the Wroclaw University of Science and Technology, Wrocław, Poland, in 2012 and 2015, respectively.

He is an Assistant Professor with the Department of Computer Science, Virginia Commonwealth University, Richmond, VA, USA, where he heads the Machine Learning and Stream Mining Laboratory. He has authored more than 60 journal articles and more than 100 contributions to conferences. He has coauthored the book *Learning from Imbalanced Datasets* (Springer, 2018). His current research interests include machine learning, data streams, class imbalance, continual learning, and explainable artificial intelligence.

Dr. Krawczyk is a Program Committee member for high-ranked conferences, such as KDD (Senior PC member), AAAI, IJCAI, ECML-PKDD, IEEE BigData, and IJCNN. He was a recipient of prestigious awards for his scientific achievements such as the IEEE Richard Merwin Scholarship, the IEEE Outstanding Leadership Award, and the Amazon Machine Learning Award, among others. He served as a Guest Editor for four journal special issues and as the Chair for 20 special session and workshops. He is the member of the editorial board for *Applied Soft Computing* (Elsevier).

Nitesh V. Chawla (Fellow, IEEE) is a Frank M. Freimann Professor of computer science and engineering and the Founding Director of the Lucy Family Institute for Data and Society, University of Notre Dame, Notre Dame, IN, USA.

Mr. Chawla was a recipient of the IBM Watson Faculty Award in 2012, the IBM Big Data and Analytics Faculty Award in 2013, the Rodney F. Ganey Award in 2014, the 2015 IEEE CIS Outstanding Early Career Award, and the National Academy of Engineering New Faculty Fellowship. He has also received and was nominated for a number of best paper awards. He serves on the editorial boards of a number of high-impact journals and organization/program committees of top-tier conferences.