A Generative Neighborhood-based Deep Autoencoder for Robust Imbalanced Classification

Eirini Troullinou, Gregory Tsgkatakis, Attila Losonczy, Panayiota Poirazi and Panagiotis Tsakalides

Abstract—Deep learning models perform remarkably well on many classification tasks recently. The superior performance of deep neural networks relies on the large number of training data, which at the same time must have an equal class distribution in order to be efficient. However, in most real-world applications the labeled data may be limited with high imbalance ratios among the classes and thus the learning process of most classification algorithms is adversely affected resulting to unstable predictions and low performance. Three main categories of approaches address the problem of imbalanced learning, i.e. data level, algorithmic level and hybrid methods, which combine the two aforementioned approaches. Data generative methods are typically based on Generative Adversarial Networks, which require significant amounts of data, while model level methods entail extensive domain expert knowledge to craft the learning objectives, thereby being less accessible for users without such knowledge. Moreover, the vast majority of these approaches is designed and applied to imaging applications, less to time series, and extremely rarely to both of them. To address the above issues, we introduce GENDA, a generative neighborhood-based deep autoencoder, which is simple yet effective in its design and can be successfully applied to both image and time series data. GENDA is based on learning latent representations that rely on the neighboring embedding space of the samples. Extensive experiments, conducted on a variety of widely-used real datasets demonstrate the efficacy of the proposed method.

Impact Statement—Imbalanced data classification is an actual and important issue in many real-world learning applications hampering most classification tasks. Fraud detection, biomedical imaging categorizing healthy people versus patients, and object detection are some indicative domains with an economic, social and technological impact, which are greatly affected by inherent imbalanced data distribution. However, the majority of the existing algorithms that address the imbalanced classification problem are designed with a particular application in mind,

and thus they can be used with specific datasets and even hyperparameters. The generative model introduced in this paper overcomes this limitation and produces improved results for a large class of image and time series data even under severe imbalance ratios, making it quite competitive.

Index Terms—Data augmentation, image data, imbalanced classification, latent space, timeseries data

Eirini Troullinou (corresponding author), Grigorios Tsgkatakis and Panagiotis Tsakalides are with the Department of Computer Science, University of Crete, GR 70013 Heraklion, Greece, and also with the Institute of Computer Science - FORTH, GR 70013, Heraklion, Greece (e-mail: troullinou@ics.forth.gr; greg@ics.forth.gr; tsakalid@ics.forth.gr).

Attila Losonczy is with the Department of Neuroscience, Columbia University, New York, NY 10027, and also with the Mortimer B. Zuckerman Mind Brain Behavior Institute, 3227 Broadway, New York, NY 10027 (e-mail: al2856@columbia.edu).

Panayiota Poirazi is with the Institute of Molecular Biology, and Biotechnology, Foundation for Research, and Technology Hellas, 70013 Heraklion, Greece (e-mail: poirazi@imbb.forth.gr).

This paragraph will include the Associate Editor who handled your paper.

I. INTRODUCTION

IMBALANCED classification poses a significant challenge for predictive modeling as most machine and deep learning algorithms are designed based on the assumption of an equal number of samples for each class. But imbalanced data distribution is present in many real-world applications affecting the learning process of most classification algorithms resulting to unstable predictions and low performance.

In general, a given training dataset may have a slight imbalance between majority and minority classes or it could have a severe imbalance, where there might be hundreds or thousands of examples in one class and just tens of examples in the other. In the latter case, the performance of predictive models is greatly affected, as the models are biased towards the majority classes, which may result to high error, or even complete omission of the minority classes, which are actually of greater interest, depending on the application [1]. Such a situation cannot be accepted in most real-world applications, as it could result in heavy costs (e.g. disease diagnosis, fraud detection) highlighting the importance of the imbalanced classification problem and the urgent need to be addressed.

Motivated by the serious performance degradation [1] caused by imbalanced class distribution, the research community has proposed three major approaches [2] to solve the imbalanced classification problem: data level, model level, and hybrid level. Data level approaches focus mostly on data augmentation by generating samples or features for the minority class. They include simple techniques, such as vanilla resampling [3], which is usually not preferred because although it balances the training set, it fails to provide any additional information to it, or they include more heuristic augmentation methods, such as the Synthetic Minority Oversampling Technique (SMOTE) and its extensions [4], [5], which have proved quite successful in a variety of applications making them quite competitive. Data level methods also include generative models, such as the Variational Autoencoders (VAEs) [6], the Generative Adversarial Networks (GANs) [7] and their variants, which all these have become the established solutions to model the data generation mechanism with deep architectures. GAN-based solutions though require significant

amounts of data, are difficult to tune, and may suffer from model collapse [8], which all these make them inappropriate to be applied to imbalanced datasets or even worse to long-tailed data. On the other hand, model level methods [9], [10], [11], [12] introduce cost-sensitive functions and change the objective function of the classifier in order to alleviate the bias, and thus to increase the importance of the minority class. They work directly within the training procedure of the considered classifier, and therefore they lack the flexibility offered by data-level approaches. Additionally, they require an in-depth understanding of how a given training procedure is conducted and what specific part of it may lead to bias towards the majority class, making them less accessible for users without such knowledge. Hybrid methods [13], [14], [15] combine the aforementioned approaches.

In an attempt to overcome the deficiencies of the afore-mentioned data driven and model level methods, we introduce GENDA, a deep generative autoencoding framework, which generates data that can be used to address the multiclass (as well as the binary) imbalance classification problem. Specifically, we propose an encoding-decoding mechanism modelled by a deep latent variable with the aim to capture the feature similarity between a given minority sample and its existing neighbors in latent space. In other words, the decoded (i.e. the generated) minority sample is represented via the embedding space of its neighbours. After the system has been trained, it can be used to generate as many samples as needed, so that a classification-based model can be trained with a class-balanced dataset.

In order to evaluate the efficacy of GENDA, a series of experiments have been conducted on widely-used real image and time series data. We also considered the neuronal cell-type classification problem [16] and used a real-world scientific time series dataset [17]. Specifically, the dataset describes the activity of four neuronal cell-types across time in the CA1 subregion of the hippocampus. Neuronal activity is measured using Ca^{2+} imaging, which is a powerful technique for monitoring the activity of distinct neurons in brain tissue in vivo and is currently the most popular recording technique for behaving animals [18]. This dataset is naturally imbalanced, as by construction the brain does not have the same number of cells. Additionally, neuroscientists do the labeling of the cells by using qualitative descriptors, such as the expression of specific molecular markers (proteins). Some cells however co-express the same protein, and as a result their exact type cannot be identified by marking. Neglecting the cells whose label is unknown results to cells-categories that are underrepresented. This causes an imbalance to the dataset as various minority classes are created.

Overall, the key contributions of this paper are summarized as follows:

- We introduce GENDA a novel deep generative encoding-decoding framework, which learns interpretable latent representations that can model the underlying distribution of the minority samples under high imbalance ratios.
- The proposed method is designed and successfully applied to both image and time series data highlighting its wide applicability.

- Our approach makes no assumption on the statistical distribution of the data, while most encoding-decoding algorithms consider for convenience that the data follow a Gaussian density and model the latent representation as such, which can lead to ineffective representations.
- While our proposed framework addresses primarily the imbalance classification problem, it can also be used in several other applications. Specifically, given that our approach is generative-based, it can be applied to various fields, including the medical, military and surveillance domains, where security, privacy and ethical reasons prohibit the use of original data, and thus artificially generated data are required. So, our approach has a clear advantage over model-based methods, which by construction address the imbalance classification problem without data augmentation.
- We conduct a series of experiments on a variety of benchmark datasets, including image and time series data, and we empirically prove the quantitative and qualitative merits of GENDA.
- To the best of our knowledge this is the first work that addresses the neuronal cell-type imbalance classification problem.

The remainder of the paper is organized as follows: In section II, we report the related work and in Section III, we describe and analyze the proposed approach. Experimental results are presented in Section IV and conclusions are drawn in Section V.

II. RELATED WORK

Classification is an essential process in artificial intelligence and machine learning as it is used to identify different patterns from the data. As classification results depend on the data distribution, one of the major issues arising in the area of data mining and knowledge discovery is known as the *class imbalance problem*. In general terms, any kind of dataset which shows unequal distribution between its classes comes under the category of imbalanced dataset. Existing classification algorithms cannot successfully handle imbalanced data, as their results deviate towards the majority class, which possesses bigger amount of data. In the case of highly imbalanced datasets, naive algorithms tend to ignore the smaller (minority) class as noise. Hence, researchers have devised several methods for tackling the class imbalance problem. These methods can be categorized into data-level, model-level and hybrid-level approaches.

A. Data-level Methods

Data driven approaches aim to characterize the underlying data distribution by approximating the data generation process. This mechanism, in imbalance classification, is mostly employed to augment the minority classes thus helping the classifier to determine the proper class boundaries.

A common data driven method is resampling [3], which aims to balance the class priors in two ways, namely by deleting samples from the majority class (under-sampling)

and by generating new samples in the minority class (over-sampling). Resampling is a simple mechanism to balance the training set, but it has two main drawbacks: over-sampling may cause overfitting and poor generalization to the test set while under-sampling leads to substantial loss of information from the majority class.

The Synthetic Minority Oversampling Technique (SMOTE) [4] is a popular oversampling approach, which selects examples that are close in the feature space, drawing a line between the examples in the feature space and generating a new sample at a point along that line. A general drawback of the approach is that synthetic examples are created without considering the majority class, possibly resulting in ambiguous examples if there is a strong overlap for the classes. Based on SMOTE several variants have been proposed, such as borderline-SMOTE [19] and the adaptive synthetic sampling approach (ADASYN) [20], which both focus on the minority samples that are harder to learn and classify.

Augmentative oversampling [21] is another widely used technique to inflate the size of the training dataset. Common augmentation techniques in image applications include translation, cropping, padding, rotation and flipping, operations which are amenable mostly to image data, thus restricting their applicability to other domains that face imbalance problems.

Deep generative models have gained a lot of attention in recent years due to numerous applications in deep learning. Among them, VAEs and GANs are regarded as the two most popular approaches to generative modeling. But vanilla VAEs and GANs suffer from several limitations, which lead to a poor quality of generated samples, especially when they are trained with a small amount of data.

VAEs [6] constitute the most popular class of autoencoders (AE). They can be directly applied on the given imbalanced data to capture the dimensional dependencies via latent variables, and then generate new samples from the learnt latent variables. This strategy however assumes that the data follow a single Gaussian distribution, which is not always the case, as samples may have a mixture of distributions or even follow a non-Gaussian distribution. Researchers have proposed many VAE variations [22] based on different task requirements with the goal of greatly improving the quality of the generated data.

GANs [7] learn the underlying data distributions from the available training data and then use the learned distributions to generate synthetic samples. However, training a vanilla GAN with a limited number of data is a challenging task. The key problem with having a small dataset, referred to as the *vanishing gradients problem*, is that the discriminator quickly overfits to the training examples. As a result, the generator receives very little feedback to improve its generations and the training collapses [23], [24]. To improve the performance and stability of GANs several variants have been proposed. Conditional GANs (cGANs) [25], [26] learn to sample from a conditional, $p(x|y)$, instead of marginal, $p(x)$, distribution, thus generating class-specific minority samples with desired properties [27].

Moreover, GAN-based generation methods are usually fed with a random noise, which may result in a highly entangled process and disrupt the orientation-related features [28], especially when dealing with minority classes. To solve this problem, researchers proposed Balancing GAN (BAGAN) [29] by integrating AE and cGAN via a two-step framework. The method learns the latent codes via AE and feeds them to a cGAN instead of random noise. However, attempting to over-sample the minority classes using GANs can lead to boundary distortion [30], [31], resulting to a worse performance on the majority class. To overcome the unstable issue in original BAGAN, Huang et al. proposed BAGAN with gradient penalty (BAGAN-GP) [31], where they added a gradient penalty term in the loss function. They also incorporated a supervised autoencoder with an intermediate embedding model to learn the label information directly, which helps to encode the similar but different class images separately. BAGAN-GP exhibits an improved performance compared to vanilla GAN and BAGAN, as it converges faster to better-quality generations.

B. Model-level Methods

Contrary to the data level approaches, model level solutions work directly within the training procedure of the considered classifier. Model-level methods, such as cost-sensitive learning [32] tailor task-specific loss functions, which are more focused on the minority classes during the optimization process. Essentially, these are penalized learning algorithms that increase the cost of classification mistakes on the minority classes.

Recent advances include focal loss [9] and dice loss [10]. Specifically, focal loss [9] reshapes the standard cross entropy loss, such that it down-weights the loss assigned to well-classified examples, while dice loss [10] attaches similar importance to false positives and false negatives and it is more immune to the data-imbalance issue. The two approaches have manifested a good performance in the tasks of computer vision and natural language processing, respectively.

Additionally, several studies have employed cost-sensitive learning with a focus to medical diagnosis applications. For example, breast cancer classification is a challenging task due to the skewed class distribution of the dataset. Extreme Gradient Boosting (XGBoost) is a scalable, distributed gradient-boosted decision tree (GBDT) machine learning method [33] that provides parallel tree boosting. Decision trees were shown to perform well on imbalanced data and a cost-sensitive XGBoost technique [11] was demonstrated to achieve good classification accuracy in a study utilizing four breast cancer datasets with uneven class distribution.

In another study [12], researchers developed a cost-sensitive random forest to deal with the imbalanced class problem in medical diagnosis. The study addressed the problem by assigning individual weights for each class instead of a single weight and employed several medical datasets, for which the proposed algorithm showed improved performance in accurately predicting both the minority and majority classes.

The main disadvantage of the model-level approaches is that they entail extensive domain expert knowledge to craft the learning objectives and to tune the hyperparameters, thereby being less accessible for users without such knowledge.
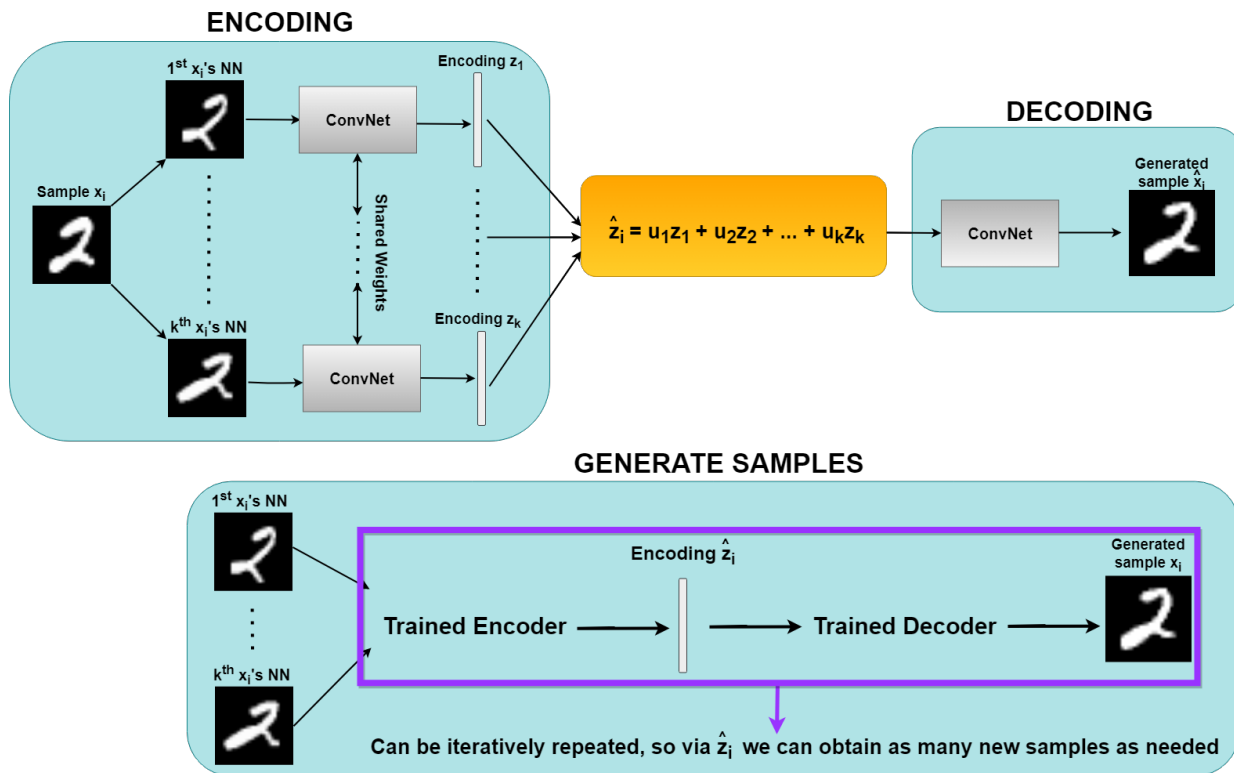
Fig. 1. Flowchart of the proposed generative model GENDA: During the encoding phase, the system takes as input the k nearest neighbors (NNs) of a random sample x_i. Each of these k inputs goes through a convolutional neural network (ConvNet), which is identical for all of them, and results to an encoding vector $\{z_j\}_{j=1}^k$. Then, the latent vector $\hat{z}_i$, which corresponds to the sample x_i is represented by the linear combination of the calculated vectors $\{z_j\}_{j=1}^k$, where the scalar coefficients $\{u_j\}_{j=1}^k$ of this combination are random numbers in $(0, 1)$. At the decoding phase, the system takes as input the latent vector $\hat{z}_i$, which goes through another ConvNet and outputs a new generated sample $\hat{x}_i$. After the system has been trained, in order to generate new samples, the trained encoder accepts k NNs of a random sample x_i, and the trained decoder generates $\hat{x}_i$. This procedure can be iteratively repeated, so via $\hat{z}_i$ (i.e. different sets of $\{u_j\}_{j=1}^k$), we can obtain as many new samples as needed.

C. Hybrid Methods

Hybrid methods combine data-level and model-level approaches. In addition to the GAN architectures discussed in Section II.A, several alternative objective functions for GANs have been proposed. Standard GANs [7] use the Jensen Shannon divergence (JSD) to measure similarity between real and GAN generated data distributions. However, JSD fails to effectively measure the distance between two distributions with negligible or no overlap. Wasserstein GAN (WGAN) [13] replaces JSD with the Earth mover Distance, also known as the Wasserstein Distance, which is smooth and can provide appropriate distance measures between close distributions with negligible or no overlap. Least square GAN (LSGAN) [14] employs a least square loss function instead of the cross entropy loss in the discriminator of the standard GAN to overcome the problem of vanishing gradient and to improve the quality of the generated data.

The Deep Generative Classifier (DGC) [15] is an end-to-end classification framework applied to imbalanced image data, whose objective function comprises three terms. It measures the distance between real and generated data via an l_2 reconstruction loss; it evaluates the difference between ground truth and generated label information via a cross entropy loss; and it adopts the maximum mean discrepancy distance measured in latent space between a conditional distribution $Q(Z|X,Y)$ and

a prior distribution $P(Z)$. To make up for the limited amount of samples in minority classes, DGC samples a set of latent codes for each minority sample by taking advantage of the reparameterization trick for the Gaussian distribution. These oversampling codes are used internally during the training of the model to generate synthetic data, and thus to infer a more robust classifier.

III. GENDA: GENERATIVE NEIGHBORHOOD-BASED DEEP AUTOENCODER

In this work, we propose a generative encoding-decoding framework modelled by a deep latent variable $\hat{z}$, which is able to learn the distribution of the training data X so that by sampling from it, we can generate new data $\hat{X}$, which is essentially an approximation of the original data X. Specifically, the proposed encoder accepts as input the k nearest neighbors of a random sample $x_i \in R^D$ and outputs a latent vector $\hat{z}_i$. This vector will be given as input to the decoder, which will generate the new sample $\hat{x}_i$.

A. Model Training

1) Encoding: Consider an imbalanced training set X consisting of M samples and let the training point $x_i \in R^D$ represent the ith sample containing feature information. Our encoder aims to learn an efficient compressed representation

of the data into a lower dimensional space R^d, also known as the latent space, where $d \ll D$. Specifically, as shown in Fig. 1, the proposed encoder takes as input the data $N(x_i)$, where N represents the neighborhood of the sample x_i. In other words, $N(x_i)$ is the set of the k nearest neighbors of x_i. Given $N(x_i)$ as input, the encoder outputs a latent vector $\{z_j\}_{j=1}^k$ for each neighbor of the given sample x_i.

From a probabilistic perspective, our encoder parameterises the following posterior conditional probability:

$$p(\hat{z}_i|N(x_i)) \equiv p(\hat{z}_i|(x_1, x_2, ..., x_k))$$
$$= \frac{p(\hat{z}_i, N(x_i))}{p(N(x_i))} \quad \forall i = 1, ..., M. \quad (1)$$

The proposed encoder is a deep convolutional neural network architecture that contains k identical subnetworks, which have the same configuration, the same parameters and weights, where parameter updating is mirrored across all k subnetworks, i.e. weight and bias updates happen simultaneously for all k subnetworks. So, each one of these k subnetworks accept a different input and the weight updates of all these subnetworks with respect to that input happen simultaneously. These subnetworks work in tandem on the k different inputs (i.e. on $x_i's$ neighbors), in order to find the similarity features and to eventually output a latent vector $\hat{z}_i$ for each sample x_i, as demonstrated in Fig. 1. Each z_j is the output of each subnetwork and is calculated by a dense layer given by the following equation

$$z_j = f(Wh_j + b) \qquad \forall j = 1, ..., k \quad (2)$$

where f is the tanh activation function, W is the weight matrix, h_j is the output of the previous layer (i.e. it is the layer, which precedes the dense layer) with each h_j coming from a subnetwork that corresponds to a specific neighbor, and b is the added bias term.

Eventually, the latent variable $\hat{z}_i$ for the specific sample x_i is represented as the linear convex combination of each $\{z_j\}_{j=1}^k$ as shown in the following equation,

$$\hat{z}_i = \sum_{j=1}^k u_j \cdot z_j = u_1 \cdot z_1 + u_2 \cdot z_2 + ... + u_k \cdot z_k$$
$$\forall i = 1, ..., M \quad (3)$$

where $\{u_j\}_{j=1}^k$ are random numbers in $(0, 1)$, which follow the uniform distribution and $\sum_{j=1}^k u_j = 1$.

Modelling $\hat{z}_i$ as shown in Eq. 3, causes the selection of a random vector along the line segment between k specific features in latent space. Our approach makes no assumption on the distribution $p(\hat{z}_i|x_i)$, whereas most encoding-decoding methods assume for convenience that $p(\hat{z}_i|x_i)$ follows the Gaussian distribution, which imposes limitations in the latent space. Assuming a Gaussian prior model leads to unimodal learnt representations and does not allow for different or mixed data distributions, which results to ineffective representations. Our approach takes advantage of the $x_i's$ local features, whose combination in latent space leads to efficient representations, as the decision region of the minority class is effectively forced to become more general.

2) Decoding: As shown in Fig. 1, the proposed decoder accepts as input the latent vector $\hat{z}_i$ and learns to reconstruct a new $\hat{x}_i$ based on this latent representation. In terms of probability models, the proposed decoder is a deep generative convolutional neural network, which parameterizes the conditional probability distribution $q(\hat{x}_i|\hat{z}_i) \forall i = 1, ..., M$, and outputs $\hat{x}_i$ via a 2D-transpose convolutional layer as shown in the following equation,

$$\hat{x}_i = \sigma(W'h + b') \quad (4)$$

where σ is the sigmoid activation function, W' is the weight matrix, h is the output of the previous layer (i.e. it is the layer, which precedes the last 2D-transpose convolutional layer) and b' is the added bias term.

In order to achieve a useful approximation of the original x_i, a decoder must minimize a mean-squared reconstruction loss given by the following equation

$$L = \frac{1}{M} \sum_{i=1}^M (x_i - \hat{x}_i)^2. \quad (5)$$

In our case though, the new sample $\hat{x}_i$ is not directly generated from the sample x_i, as the encoder does not take the sample x_i as its input, and thus Eq. 5 can be rewritten as,

$$L = \frac{1}{M} \sum_{i=1}^M (x_i - \hat{x}_i)^2 = \frac{1}{M} \sum_{i=1}^M (x_i - d(e(N(x_i))))^2 \quad (6)$$

where d and e are the decoder and encoder networks, respectively. By reconstructing the sample x_i as shown in Eq. 6, i.e. via the embedding space of its neighbours, we ensure that the generated sample $\hat{x}_i$ will be a good approximation of the original sample x_i, yet not its replica. Thus, except from the generation of high-quality samples in general, our mechanism avoids serious overfitting problems during classification. The proposed encoding-decoding framework is applied for all the samples $\{x_i\}_{i=1}^M$, accordingly.

B. Data Generation and Classification

After the proposed model has been trained, as discussed in the previous subsection, it can be used to generate new samples for all the classes. Specifically, one can sample a point from the latent vector $\hat{z}_i$ produced by the trained encoder, and then pass it through the trained decoder, which will generate samples similar to those in the dataset. Moreover, as shown in Eq. 3, the coefficients $\{u_j\}_{j=1}^k$ provide the flexibility to generate an unlimited number of samples. After the new samples have been created, we use a deep convolutional classifier, which is trained with a balanced dataset consisting of the original data and the new data generated by our proposed method. The overall algorithm for training the proposed model and generating synthetic samples is summarized in Algorithm 1.

Dataset	Shape	Classes	IR	Training Set	Testing Set
MNIST	28x28x1	10	100	4000 (0), 2000 (1), 1000 (2), 750 (3), 500 (4), 350 (5), 200 (6), 100 (7), 60 (8), 40 (9)	980 (0), 1135 (1), 1032 (2), 1010 (3), 982 (4) 892 (5), 958 (6), 1028 (7), 974 (8), 1009 (9)
Fashion-MNIST	28x28x1	10	100	4000 (0), 2000 (1), 1000 (2), 750 (3), 500 (4), 350 (5), 200 (6), 100 (7), 60 (8), 40 (9)	Each class contains 1000 samples
HAR	128x9	6	30.65	1226 (0), 800 (1), 500 (2), 300 (3), 100 (4), 40 (5)	496 (0), 471 (1), 420 (2), 491 (3), 532(4), 537 (5)
TwoLeadECG	82x1	2	14.225	569 (0), 40 (1)	12 (0), 11 (1)
Ca^{2+} Imaging	4000x1	4	7.39	5600 (0), 1183 (1), 757 (2), 3500 (3)	1400 (0), 296 (1), 190 (2), 700 (3)

TABLE I
Summarization of the experimental datasets.

Algorithm 1 GENDA

Input: $\mathcal{X} = \{x_i\}_{i=1}^{M}$: Set of training data, $\mathcal{B} = \{b_t\}_{t=1}^{n}$: Batch of training data, k: Number of NNs
Output: Balanced Training Set
Symbols: L: Loss
Encoding:
$z_j = Encoder(x_j)$, x_j : NN of x_i, $\forall x_i \in \mathcal{X}$, $\forall j = 1, .., k$
$\hat{z}_i = \sum_{j=1}^{k} u_j \cdot z_j$, $\forall i = 1, ..., M$, $\forall j = 1, ..., k$
Decoding:
$\hat{x}_i = Decoder(z_i)$
Training Step:
for $e \leftarrow epochs$ **do**
    for $b \leftarrow batches$ **do**
        $E_b \leftarrow Encoder(b)$
        $D_b \leftarrow Decoder(E_b)$
        $L = (\frac{1}{n}) \sum_{i=1}^{n} (b_i - D_{bi})^2$
    end for
end for
Generate Samples:
for $i \leftarrow$ number of classes **do**
    $X \leftarrow$ Select class data
    $E \leftarrow Encoder(X)$
    $\hat{X} \leftarrow Decoder(E)$
end for

IV. EXPERIMENTAL STUDY

In this Section, a series of experiments are conducted to evaluate GENDA across various imbalance settings for a large collection of real data sets. The models that were used in our method were implemented using the Tensorflow and Keras open-source libraries written in the Python programming language. For our experiments we used Python version 3.6.10 and Tensorflow version 2.2.0 running on a NVIDIA GeForce GTX 750 Ti GPU model under the Windows 10 operating system.

A. Datasets

Four benchmark datasets and a scientific neuronal cell dataset were selected for our experimental analysis on imbalanced classification. The benchmark datasets that we used were the image single-channel MNIST [34] and Fashion-MNIST [35], and the timeseries datasets HAR [36] and TwoLeadECG [37] from the UCI and UCR repositories,

respectively. None of these four datasets is imbalanced in nature, and thus we artificially forced imbalance by randomly selecting instances with different sizes from different classes. On the other hand, the neuronal cell dataset is naturally imbalanced and was collected during a goal oriented task in awake, behaving mice [17]. The neural signals were recorded using the two-photon Ca^{2+} imaging technique and the data were then processed in order to translate the video recordings into fluorescence signals over time. Four different neuronal types were recorded during the aforementioned task, i.e. the excitatory pyramidal cells (PY), which is the majority class and three GABAergic interneuronal subtypes, namely somatostatin-positive (SOM), parvalbumin-positive (PV), which is the minority class and vasoactive intestinal polypeptide-positive (VIP) cells making the problem a four-class imbalanced classification task.

Details for all the datasets, such as shape, number of classes, imbalance ratio and number of training as well as testing examples for each class are shown in Table I. Note that for the Fashion-MNIST, HAR and TwoLeadECG datasets, we associated each class with an integer number, as exactly assigned in the original datasets, while for the Ca^{2+} imaging dataset, 0 label corresponds to PY neurons, and labels 1, 2 and 3 correspond to SOM, PV and VIP cells respectively.

B. Setup

1) Evaluation metrics: In order to validate the imbalance classification performance, three widely-used, skew-insensitive metrics are adopted: Average class specific accuracy (ACSA), which is the averaged accuracy achieved for each class separately, also known as balanced accuracy, F1-score and precision.

2) Reference generative methods: In order to evaluate the effectiveness of GENDA both on image and time series data, we compared it with the most relevant state-of-the-art image and time series data augmentation methods. For the image datasets, we selected SMOTE [4], DGC [15] and BAGAN-GP [31], while for the time series datasets we selected TimeGAN [38] and SMOTE [4], which is an algorithm applied and designed both for image and time series datasets. The parameters of all algorithms we compared with are adopted from their original papers.

3) Implementation details of the proposed method: The encoder structure of GENDA for the image datasets consists of five 2D-convolutional layers with 16, 32, 64 and 128 filters of size $(4, 4)$. Each layer is followed by a 2D-average pooling layer of size $(2, 2)$ and the tanh activation function. The final layer is linear, yielding a latent dimension of 16. For the time series data, we used a smaller network, as we noticed that a larger network increases time and computational complexity with no gain in performance. Thus, the encoder consists of three 2D-convolutional layers with 16, 32 and 64 filters of size $(2, 1)$ for the TwoLeadECG and Ca^{2+} imaging datasets and $(2, 2)$ for the HAR dataset. Each layer is followed by a 2D-average pooling layer of size $(2, 1)$ for the TwoLeadECG and Ca^{2+} imaging datasets and $(2, 2)$ for the HAR dataset, also followed by the tanh activation function. The final layer is linear, yielding a latent dimension of 32.

Accordingly, the decoder structure for the image datasets consists of three 2D-transpose convolutional layers with 128, 64 and 32 filters of size $(4, 4)$. Each layer is followed by a 2D-average pooling layer of size $(2, 2)$ and the LeakyReLu activation function. The final layer is a 2D-transpose convolutional layer with 1 filter followed by the sigmoid activation function. For the time series data, the decoder is composed of three 2D-transpose convolutional layers with 64, 32 and 16 filters of size $(2, 1)$ for the TwoLeadECG and Ca^{2+} imaging datasets and $(2, 2)$ for the HAR dataset. Moreover, each layer is followed by a 2D-average pooling layer of size $(2, 1)$ for the TwoLeadECG dataset and $(2, 2)$ for the HAR dataset and the LeakyReLu activation function. The final layer is a 2D-transpose convolutional layer with 1 filter followed by the sigmoid activation function.

The proposed encoding-decoding system was trained for 40 epochs, and we used the Adam optimizer for both the encoder and the decoder model with a 0.001 learning rate for all the datasets. Eventually, as it is later described in Table 4, the optimal value with respect to the number of neighbors is $k = 2$.

4) Classification model: All methods except from the DGC [15], which is an end-to-end framework, use an identical 2D-convolutional network as their base classifier, which takes as input the original data and a requisite number of generated samples, so that it is trained with a balanced dataset. Specifically, the classifier consists of five 2D-convolutional layers with 128, 64, 32 and 16 filters of size $(5, 1)$ for the TwoLeadECG and Ca^{2+} imaging datasets and $(5, 5)$ for the rest of the datasets. Each layer is followed by a dropout layer and the LeakyReLu activation function. The final layer is linear, yielding a dimension that depends on the number of classes of each dataset and is followed by the softmax activation function. The classifier in all cases is trained for 80 epochs and the Adam optimizer is used with a 0.001 learning rate.

C. Results and Discussion

In our experiments, we address the following four facets of the problem: (i) we compared the performance of GENDA with that of the most recent balancing techniques on real image and time series data using established quantitative metrics; (ii) we explored the extent to which classification performance is affected with respect to several parameters, such us the u_i's distribution, the number of neighbors and the dimensionality of latent space. For these experiments we indicatively selected the image dataset MNIST and the timeseries dataset HAR; (iii) we investigated the stability of the method, and (iv) we demonstrated the qualitative merit by providing some visualization results on raw and generated MNIST and Fashion-MNIST images.

To make a fair comparison, all models were given as input the same dataset for training and were evaluated on the same testing dataset. The overall classification performance on four benchmark image and time series datasets is listed in Tables II and III, respectively. The best results are highlighted in bold.

From the results shown in Table II, we initially observe the extent to which the baseline performance improves for both datasets and especially for Fashion-MNIST after data augmentation has been applied. Note that *baseline* refers to the achieved performance when the classifier is trained with the imbalanced dataset. We observe that only DGC slightly outperforms our model with respect to the ACSA and F1-score measures, while GENDA outperforms all methods with respect to the precision metric. But the slight superiority of DGC comes with a severe time complexity and computational cost, due to the high-values assigned to the various hyperparameters. Moreover, to make up for the limitation of input data, DGC takes advantage of the reparameterization trick for Gaussian distributions, and applies an internal data augmentation only for the samples in minority classes during the training of the model. Thus, after DGC has been trained, it cannot be used to generate samples, and as a result DGC can only be used for classification applications, while our approach is designed to generate diverse samples from all classes, as many as required. So, our proposed method is a generic framework that can be used in several other applications including the medical, the military and surveillance domains, where security, privacy and ethical reasons prohibit the use of original data, and thus multiple artificially generated data are required. BAGAN-GP exhibits the worst performance compared to the other models. Although BAGAN-GP employs an enhanced autoencoder initialization to stabilize the GAN training, its performance is still unstable compared to the non-GAN models.

Overall, we observe that all methods exhibit a worse performance on Fashion-MNIST data than on MNIST data. We think that the reason that Fashion-MNIST is a more challenging dataset compared to MNIST is because of the big diversity that exists among the samples of the same class. Therefore, the models are not able to efficiently learn the basic features of each class, and especially those which belong to the minority classes.

Table III demonstrates the results on the time series data. We observe that regardless of the metric used, GENDA outperforms SMOTE and TimeGAN for all datasets. Specifically, TimeGAN has the worst performance compared to the other methods, which could be justified by the unstable training of the GAN. It is also remarkable that all methods exhibit

	MNIST			Fashion-MNIST		
Method	ACSA	F1-Score	Precision	ACSA	F1-Score	Precision
Baseline	0.579	0.563	0.45	0.499	0.475	0.454
SMOTE	0.895	0.894	0.883	0.738	0.708	0.712
DGC	**0.948**	**0.947**	0.911	**0.836**	**0.831**	0.781
BAGAN-GP	0.863	0.85	0.841	0.731	0.729	0.69
GENDA	0.925	0.922	**0.926**	0.811	0.801	**0.794**

TABLE II
Comparing overall classification performance on image datasets.

	HAR			TwoLeadECG			Ca^{2+} Imaging		
Method	ACSA	F1-Score	Precision	ACSA	F1-Score	Precision	ACSA	F1-Score	Precision
Baseline	0.605	0.536	0.555	0.5	0.342	0.26	0.65	0.674	0.714
SMOTE	0.731	0.682	0.652	0.81	0.823	0.815	0.77	0.78	0.792
TimeGAN	0.713	0.67	0.643	0.735	0.716	0.693	0.697	0.674	0.654
GENDA	**0.877**	**0.878**	**0.883**	**0.829**	**0.838**	**0.817**	**0.787**	**0.797**	**0.809**

TABLE III
Comparing overall classification performance on timeseries datasets.

	MNIST			HAR		
Distribution	ACSA	F1-Score	Precision	ACSA	F1-Score	Precision
Uniform	0.925	0.922	0.926	0.877	0.878	0.883
Normal	0.921	0.922	0.924	0.873	0.876	0.88
Lognormal	0.919	0.92	0.921	0.869	0.855	0.843

TABLE IV
Classification performance with respect to u_i's distribution.

the worst performance when trained with the Ca^{2+} imaging dataset (except from the case of SMOTE with HAR), which could be put down to the fact that Ca^{2+} imaging is an inherently noisy method due to the high spatiotemporal information desired from a sample often showing low signal-to-noise alongside drift or cell movement, particularly for living organisms.

Regarding the u_i's distribution, our method uses the uniform distribution, as there is no prior information with respect to this. Nevertheless, we also experimented with two more distributions, i.e. the normal and the lognormal distribution both with zero mean and one for the standard deviation, and as it was previously stated, we indicatively applied it to the image dataset MNIST and to the timeseries dataset HAR. So, from Table IV we observe that the obtained results are close to the initial results, where u_i followed the uniform distribution, which demonstrates the robustness of the algorithm with respect to u_i's distribution.

Table V demonstrates the classification performance with respect to the rebalancing approaches of oversamlping and undersampling. We observe that by oversampling the minority class(es) the results are slightly better compared to the performance of the baseline classifier, but still there is poor generalization performance with respect to the test set, as by randomly duplicating the minority samples the classifier does not really receive new information. On the other hand, by undersampling the rest of the classes, which do not belong to the minority class leads to substantial loss of information from these classes, and thus we observe a significant decrease in classification performance.

Table VI demonstrates the classification performance with respect to the number of neighbors (k). We observe that for $k = 2$ and $k = 3$ neighbors we get the highest performance for both datasets, while when using more neighbors (i.e. $k = 4$ or $k = 5$) the performance gradually deteriorates. This can be attributed to the fact that the fifth neighbor of the original signal can be distinctly different from the original one and its closest neighbors. Thus, the experiments of the original manuscript presented in Tables 2 and 3, and also in Figure 2 were implemented by using $k = 2$ neighbors, as we observe that as k increases the performance deteriorates and also the proposed method becomes more computationally expensive. The rest of the datasets have also the same behaviour with respect to the number of neighbors.

Table VII demonstrates the classification performance with respect to the dimension of latent space (d). We notice that by using $d = 16$ and $d = 32$ we obtain the highest classification performance for MNIST and HAR respectively. For both datasets, $d = 16$ and $d = 32$ give almost the same results, while for $d = 64$ the performance slightly deteriorates, which can be due to overfitting. Eventually, by reducing the dimensionality of latent space to $d = 8$, we observe that the performance for both datasets decreases almost by 10%. This decrease indicates that $d = 8$ is not an adequate value in order for the encoder to capture efficiently the features of the input data resulting to representations of worse quality and performance respectively. The rest of the datasets have also the same behaviour with respect to the dimensionality of latent
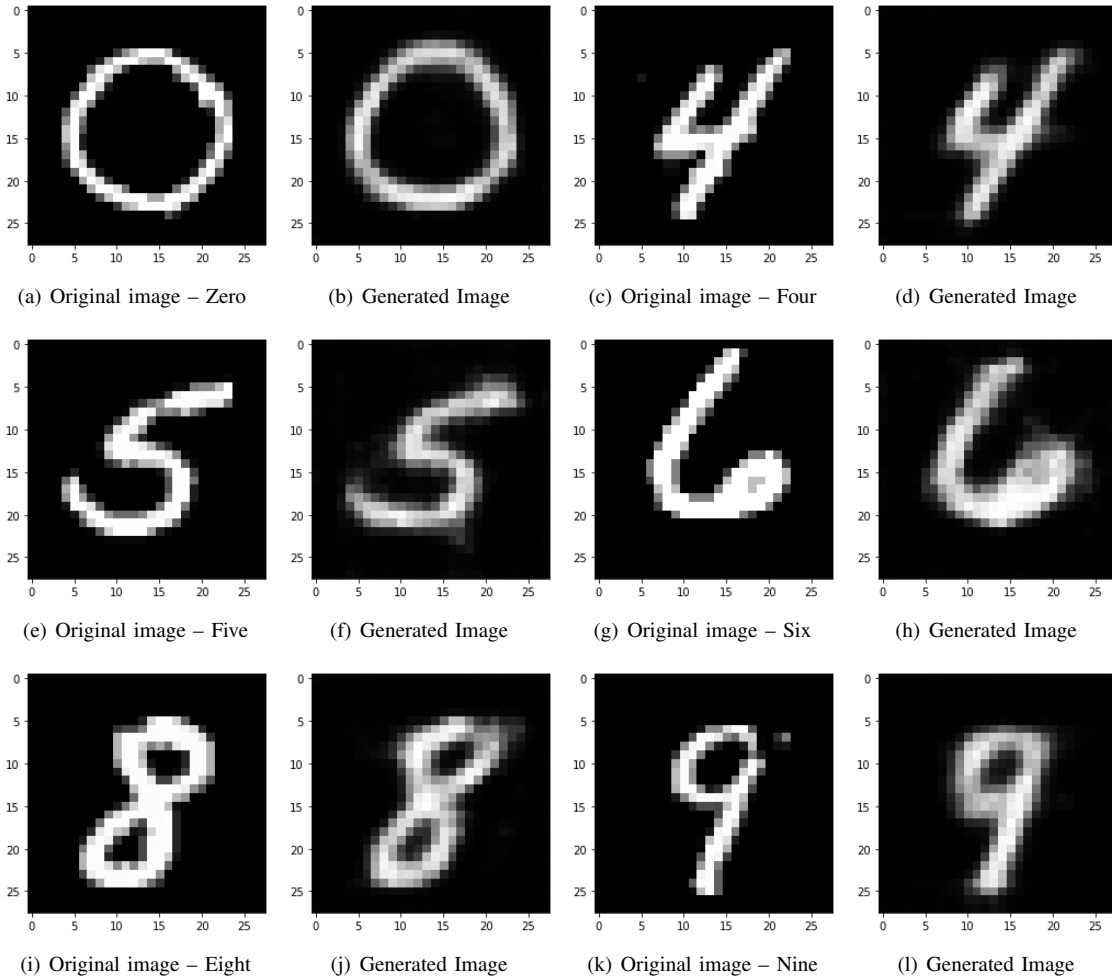
(a) Original image – Zero   (b) Generated Image   (c) Original image – Four   (d) Generated Image

(e) Original image – Five   (f) Generated Image   (g) Original image – Six   (h) Generated Image

(i) Original image – Eight   (j) Generated Image   (k) Original image – Nine   (l) Generated Image

Fig. 2. Raw and generated MNIST images by GENDA for (a)-(b) the majority class, (k)-(l) the minority class, and (c)-(j) randomly selected classes.

	MNIST			HAR		
Rebalancing Approach	ACSA	F1-Score	Precision	ACSA	F1-Score	Precision
Oversampling	0.61	0.603	0.598	0.642	0.645	0.65
Undersampling	0.522	0.52	0.51	0.53	0.52	0.525

TABLE V
Classification performance with respect to rebalancing approaches.

	MNIST			HAR		
k	ACSA	F1-Score	Precision	ACSA	F1-Score	Precision
2	0.925	0.922	0.926	0.877	0.878	0.883
3	0.927	0.926	0.925	0.875	0.877	0.88
4	0.911	0.913	0.917	0.863	0.867	0.872
5	0.89	0.893	0.895	0.833	0.841	0.85

TABLE VI
Classification performance with respect to k.

	MNIST			HAR		
d	ACSA	F1-Score	Precision	ACSA	F1-Score	Precision
8	0.826	0.829	0.835	0.789	0.793	0.796
16	0.925	0.922	0.926	0.861	0.866	0.869
32	0.926	0.927	0.928	0.877	0.878	0.883
64	0.9	0.89	0.89	0.862	0.866	0.87

TABLE VII
Classification performance with respect to d.

space.

Fig. 2 and 3 present the artificially generated images for MNIST and Fashion-MNIST, respectively. Fig. 2 (a)-(b) represent the majority class, Fig. 2 (k)-(l) represent the minority class, while all the rest are randomly selected classes. Respectively, for the Fashion-MNIST dataset, Fig. 3 (a)-(b) represent

the majority class, Fig. 3 (k)-(l) represent the minority class and all the rest are randomly selected classes. The outcomes demonstrate that the GENDA generates artificial images that are both information-rich (i.e., they improve the discriminative ability of the classifier and counter majority bias), and are also visually meaningful (i.e., even for the minority classes,
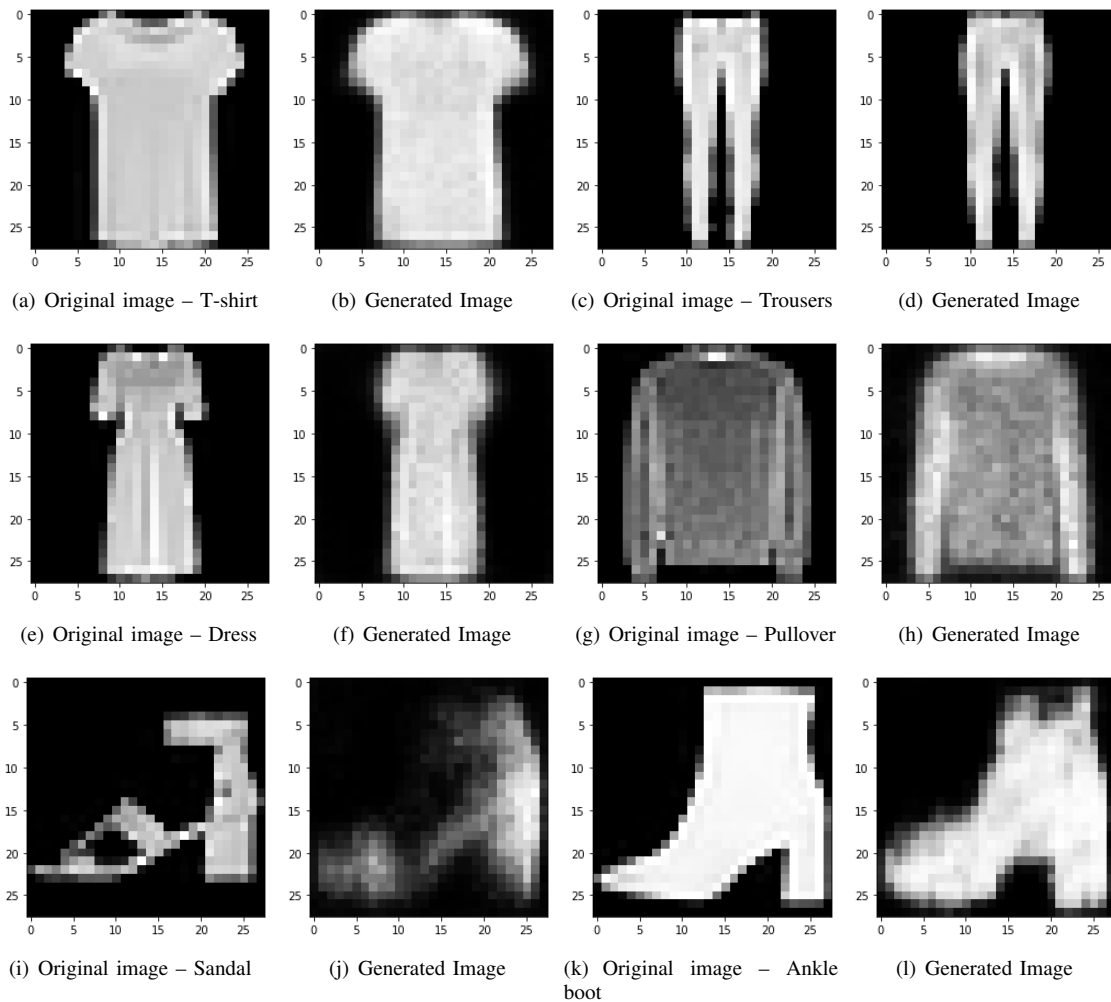
(a) Original image – T-shirt   (b) Generated Image   (c) Original image – Trousers   (d) Generated Image

(e) Original image – Dress   (f) Generated Image   (g) Original image – Pullover   (h) Generated Image

(i) Original image – Sandal   (j) Generated Image   (k) Original image – Ankle boot   (l) Generated Image

Fig. 3.   Raw and generated Fashion-MNIST images by GENDA for (a)-(b) the majority class, (k)-(l) the minority class, and (c)-(j) randomly selected classes.
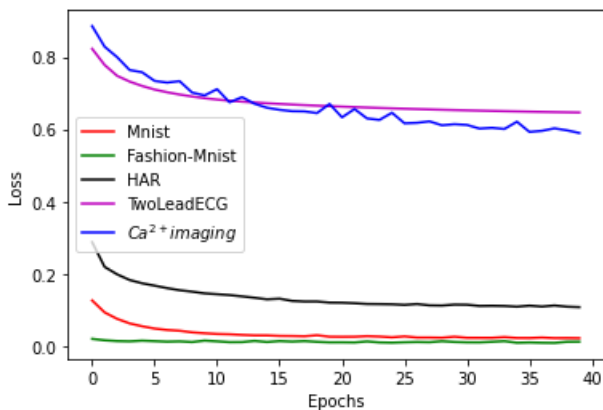

Fig. 4.   Convergence of the proposed method.

GENDA generates meaningful and realistic samples)

Fig. 4 illustrates the loss of our proposed method over epochs for all the datasets. Our algorithm exhibits a smooth and fast convergence (i.e. GENDA converges in less than 10 epochs) for all the datasets, which guarantees the stability of the proposed model. Furthermore, due to the fast convergence,

an early stopping can be applied to the training of the model, thus saving computational time and resources. Eventually, we observe a higher loss with respect to the time series datasets (i.e., HAR, TwoLeadECG and Ca^{2+} imaging) compared to the image datasets. This can be attributed to the versatility that time signals exhibit compared to static images, and thus time series data are more difficult to be modelled.

V. CONCLUSION AND FUTURE WORK

In this paper, we proposed GENDA, a deep generative encoding-decoding system, whose design lies in the learning of latent yet interpretable representations that capture the non-linear structured underlying data. It models the data generating mechanism, as it creates artificial instances that balance the training set, which can then be used to train any classifier without suffering from bias. The proposed method fulfills three crucial characteristics of a successful generative algorithm: The ability to operate on both image and timeseries data, the creation of efficient low-dimensional embeddings, and the generation of diverse and meaningful artificial instances. Experimental studies showed that our proposed method is quite competitive compared to other methods and with high model stability even under high imbalance ratios.

Our next efforts will focus on enhancing our model's loss function with instance-level penalties so that the encoder and decoder training considers instances that exhibit borderline/overlapping features while discarding outliers and noisy instances. Moreover, given that the quality of nearest neighbors gets worse as the dimensionality of the data increases, we will work on finding an efficient way, so that nearest neighbors are found in the learned latent space instead of using them in data space. Finally, the proposed method will be extended to incorporate other data modalities, such as graphs.

REFERENCES

[1] A. Somasundaram and U. S. Reddy, "Data imbalance: effects and solutions for classification of large and highly imbalanced data," in *international conference on research in engineering, computers and technology (ICRECT 2016)*, 2016, pp. 1–16.

[2] A. Fernández, S. García, M. Galar, R. C. Prati, B. Krawczyk, and F. Herrera, *Learning from imbalanced data sets*. Springer, 2018, vol. 10.

[3] M. Khushi, K. Shaukat, T. M. Alam, I. A. Hameed, S. Uddin, S. Luo, X. Yang, and M. C. Reyes, "A comparative performance analysis of data resampling methods on imbalance medical data," *IEEE Access*, vol. 9, pp. 109 960–109 975, 2021.

[4] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[5] A. Fernández, S. Garcia, F. Herrera, and N. V. Chawla, "Smote for learning from imbalanced data: progress and challenges, marking the 15-year anniversary," *Journal of artificial intelligence research*, vol. 61, pp. 863–905, 2018.

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

[7] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[8] Z. Pan, W. Yu, B. Wang, H. Xie, V. S. Sheng, J. Lei, and S. Kwong, "Loss functions of generative adversarial networks (gans): opportunities and challenges," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 4, no. 4, pp. 500–522, 2020.

[9] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[10] X. Li, X. Sun, Y. Meng, J. Liang, F. Wu, and J. Li, "Dice loss for data-imbalanced nlp tasks," *arXiv preprint arXiv:1911.02855*, 2019.

[11] M. Phankokkruad, "Cost-sensitive extreme gradient boosting for imbalanced classification of breast cancer diagnosis," in *2020 10th IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*. IEEE, 2020, pp. 46–51.

[12] M. Zhu, J. Xia, X. Jin, M. Yan, G. Cai, J. Yan, and G. Ning, "Class weights random forest algorithm for processing class imbalanced medical data," *IEEE Access*, vol. 6, pp. 4641–4652, 2018.

[13] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *International conference on machine learning*. PMLR, 2017, pp. 214–223.

[14] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. Paul Smolley, "Least squares generative adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2794–2802.

[15] X. Wang, Y. Lyu, and L. Jing, "Deep generative model for robust imbalance classification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 14 124–14 133.

[16] E. Troullinou, G. Tsagkatakis, S. Chavlis, G. F. Turi, W. Li, A. Losonczy, P. Tsakalides, and P. Poirazi, "Artificial neural networks in action for an automated cell-type classification of biological neural networks," *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2020.

[17] G. F. Turi, W.-K. Li, S. Chavlis, I. Pandi, J. O'Hare, J. B. Priestley, A. D. Grosmark, Z. Liao, M. Ladow, J. F. Zhang *et al.*, "Vasoactive intestinal polypeptide-expressing interneurons in the hippocampus support goal-oriented spatial learning," *Neuron*, vol. 101, no. 6, pp. 1150–1165, 2019.

[18] A. Birkner, C. H. Tischbirek, and A. Konnerth, "Improved deep two-photon calcium imaging in vivo," *Cell calcium*, vol. 64, pp. 29–35, 2017.

[19] H. Han, W.-Y. Wang, and B.-H. Mao, "Borderline-smote: a new over-sampling method in imbalanced data sets learning," in *International conference on intelligent computing*. Springer, 2005, pp. 878–887.

[20] H. He, Y. Bai, E. A. Garcia, and S. Li, "Adasyn: Adaptive synthetic sampling approach for imbalanced learning," in *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)*. IEEE, 2008, pp. 1322–1328.

[21] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, no. 1, pp. 1–48, 2019.

[22] R. Wei, C. Garcia, A. El-Sayed, V. Peterson, and A. Mahmood, "Variations in variational autoencoders-a comparative evaluation," *Ieee Access*, vol. 8, pp. 153 651–153 670, 2020.

[23] M. Arjovsky and L. Bottou, "Towards principled methods for training generative adversarial networks," *arXiv preprint arXiv:1701.04862*, 2017.

[24] T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen, and T. Aila, "Training generative adversarial networks with limited data," in *IEEE Conference on Neural Information Processing Systems;*, 2020.

[25] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[26] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier gans," in *International conference on machine learning*. PMLR, 2017, pp. 2642–2651.

[27] G. Douzas and F. Bacao, "Effective data generation for imbalanced learning using conditional generative adversarial networks," *Expert Systems with applications*, vol. 91, pp. 464–471, 2018.

[28] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "Infogan: Interpretable representation learning by information maximizing generative adversarial nets," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, 2016, pp. 2180–2188.

[29] G. Mariani, F. Scheidegger, R. Istrate, C. Bekas, and C. Malossi, "Bagan: Data augmentation with balancing gan," *arXiv preprint arXiv:1803.09655*, 2018.

[30] S. Santurkar, L. Schmidt, and A. Madry, "A classification-based study of covariate shift in gan distributions," in *International Conference on Machine Learning*. PMLR, 2018, pp. 4480–4489.

[31] G. Huang and A. H. Jafari, "Enhanced balancing gan: minority-class image generation," *Neural Computing and Applications*, pp. 1–10, 2021.

[32] A. Fernández, S. García, M. Galar, R. C. Prati, B. Krawczyk, and F. Herrera, "Cost-sensitive learning," in *Learning from Imbalanced Data Sets*. Springer, 2018, pp. 63–78.

[33] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.

[34] Y. LeCun, C. Cortes, and C. Burges, "Mnist handwritten digit database. at&t labs," 2010.

[35] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.

[36] D. Anguita, A. Ghio, L. Oneto, X. Parra Perez, and J. L. Reyes Ortiz, "A public domain dataset for human activity recognition using smartphones," in *Proceedings of the 21th international European symposium on artificial neural networks, computational intelligence and machine learning*, 2013, pp. 437–442.

[37] H. A. Dau, E. Keogh, K. Kamgar, C.-C. M. Yeh, Y. Zhu, S. Gharghabi, C. A. Ratanamahatana, Yanping, B. Hu, N. Begum, A. Bagnall, A. Mueen, G. Batista, and Hexagon-ML, "The ucr time series classification archive," October 2018, https://www.cs.ucr.edu/~eamonn/time_series_data_2018/.

[38] J. Yoon, D. Jarrett, and M. Van der Schaar, "Time-series generative adversarial networks," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

Eirini Troullinou received her B.Sc. (2014) from the Math Department and her M.Sc. (2018) from the Computer Science Department both from University of Crete. She is currently pursuing the Ph.D. degree in the Computer Science Department of University of Crete and is funded by the State Scholarships Foundation (IKY). Since 2015, she has been affiliated with the Signal Processing Lab at FORTH-ICS as a graduate researcher. Her main research interests include signal processing and machine learning.

Grigorios Tsagkatakis is a research associate with the Institute of Computer Science at the Foundation for Research and Technology– Hellas (FORTH) in Greece. He received his BE and MS degrees in Electronics and Computer Engineering from the Technical University of Crete, Greece in 2005 and 2007 respectively, and his Ph.D. in Imaging Science from the Rochester Institute of Technology, New York, in 2011. Between 2019 and 2021, he was a Marie Skłodowska–Curie fellow at the Department of Electrical and Computer Engineering of the University of Southern California. His research focuses on topics related to signal/image processing and machine learning with applications in remote sensing and astrophysics.

Attila Losonczy is a Professor of Neuroscience in the Mortimer B. Zuckerman Mind Brain Behavior Institute at Columbia University, New York. Dr. Losonczy's research is aimed to uncover neuronal mechanisms of learning and memory by linking synaptic, cellular and microcircuit processes with memory behaviors in the mammalian hippocampus. Thus, his research uses largescale functional imaging in combination with electrophysiology, cell-type specific manipulations and computational modeling. Currently, he is part of a team that aims to understand mechanism of memory replay.

Panayiota Poirazi is a Director of Research at the Institute of Molecular Biology and Biotechnology, Foundation for Research and Technology-Hellas (FORTH) and head of the Dendrites lab. She received the B.S. in Mathematics from the University of Cyprus in May 1996, the M.S. degree in Biomedical Engineering in May 1998 and the Ph.D. degree in Biomedical Engineering in July 2000, both from the University of Southern California. Her work focuses on understanding the role of dendrites in complex brain functions. She uses primarily computational modeling of neurons and networks, brain-inspired machine learning and recently in vivo experiments in mice. She has received several awards for academic excellence, including the EMBO Young Investigator award in 2005, two Marie Curie fellowships (2002 and 2008), an ERC Starting Grant in 2012, the Friedrich Wilhelm Bessel award of the Humboldt Foundation in 2018 and an EINSTEIN foundation visiting fellowship in 2019. She is a member of EMBO and the Secretary General elect of FENS.

Panagiotis Tsakalides Panagiotis Tsakalides received the diploma degree in electrical engineering from the Aristotle University of Thessaloniki, Greece, in 1990, and the PhD degree in electrical engineering from the University of Southern California, Los Angeles in 1995. He is a professor in computer science at the University of Crete, Heraklion, Greece, and the Head of the Signal Processing Laboratory at FORTH-ICS. His research interests include statistical signal processing, machine learning, sparse representations, and applications in remote sensing, astrophysics, audio, imaging, and multimedia systems. He has coauthored over 250 technical publications in these areas. Since 2003, he has been the project coordinator in 9 European Commission and 13 national projects with a budget in excess of 15 M € totaling more than 8 M € in actual funding for FORTH-ICS and the University of Crete.