A Generative Neighborhood-based Deep Autoencoder with an Extended Loss Function for Robust Imbalanced Classification

E. Troullinou*†, G. Tsagkatakis*†, A. Losonczy‡, P. Poirazi§, and P. Tsakalides*†
*Department of Computer Science, University of Crete, Heraklion, 70013, Greece
†Institute of Computer Science, Foundation for Research and Technology Hellas, Heraklion, 70013, Greece
‡Department of Neuroscience, Columbia University Medical Center, New York, USA
§Institute of Molecular Biology and Biotechnology, Foundation for Research and Technology Hellas, Heraklion, 70013, Greece

***Abstract*—Deep learning models have demonstrated remarkable performance in classification tasks; however, real-world applications often grapple with constraints such as limited labeled data and significant class imbalance. These constraints can result in unstable predictions and reduced performance. To tackle this challenge, three distinct approaches have emerged: data-level methods, model-level methods, and hybrid methods. Data-level methods make use of generative models, typically grounded in Generative Adversarial Networks, which rely on extensive data resources. In contrast, model-level methods leverage domain expertise and may be less accessible to users lacking such specialized knowledge. Hybrid methods combine elements of both these approaches. In this work, we introduce GENDA-XL, a generative neighborhood-based deep autoencoder featuring an extended loss function. GENDA-XL places emphasis on learning latent representations via supervised similarity learning and it integrates a pre-trained classification model to associate each generated sample with its corresponding label. Through comprehensive experiments conducted across various image and time-series datasets, we illustrate the effectiveness of our method.**

***Index Terms*—Data Augmentation, Generative Model, Image Data, Imbalanced Classification, Timeseries Data**

I. Introduction

Classification plays a fundamental role in artificial intelligence and machine learning, as it is instrumental in recognizing distinct patterns in the data. Within the domain of data mining and knowledge discovery, imbalanced classification emerges as a significant challenge. This problem arises when datasets exhibit uneven distributions among their classes. Such datasets comprise both majority and minority instances, representing the most and least common samples, respectively.

Imbalanced classification presents a foundational challenge in predictive modeling because the majority of machine learning algorithms are optimized to minimize overall error rates. As a result, these algorithms tend to prioritize the detection of majority samples, which exhibit extensive coverage and subsequently yield high accuracy. Meanwhile, minority samples are frequently neglected or considered as noise, resulting in their frequent misclassification. Nevertheless, in the majority of real-world scenarios, such as disease diagnosis and fraud detection, misclassifying a rare event typically incurs higher costs compared to misclassifying a common one.

Considering the serious performance degradation [1] caused by the imbalanced classification problem, the research community has proposed three major approaches including data level, model level, and hybrid level methods. Data level methods focus on generating samples or features for the minority class and are typically based on the Generative Adversarial Networks (GANs) [2] and their variants, which have become the established solutions to model the data generation mechanism with deep architectures. However, they demand substantial quantities of data, pose challenges in terms of fine-tuning, and are susceptible to mode collapse [3], rendering them unsuitable for utilization with imbalanced datasets. On the other hand, model level methods [4] introduce cost-sensitive functions and change the objective function of the classifier to increase the importance of the minority class. They require an in-depth understanding of how a given training procedure is conducted making them less accessible for users without such knowledge. Hybrid methods [5] combine the aforementioned approaches.

In an attempt to overcome the deficiencies of the aforementioned methods, we introduce a Generative Neighborhood-based Deep Autoencoder with an Extended Loss function, named GENDA-XL, which is the extended version of our previous work GENDA [6]. Similarly to GENDA, GENDA-XL is also a deep generative autoencoder based on learning latent representations that rely on the neighboring embedding space of the samples. GENDA-XL preserves all the merits of GENDA, such as the applicability to both image and time-series data making no assumption on their statistical distribution. At the same time, GENDA-XL features a more robust loss function compared to GENDA using a supervised similarity learning metric, in order to learn more efficient representations that can model the underlying distribution of the samples under high imbalance ratios. Furthermore, it leverages the insights offered by a classification model, integrating a classifier pre-trained on real-world data into its structure. This ensures that every generated sample is linked to its corresponding label. A set of experiments conducted

on diverse benchmark datasets establishes the quantitative and qualitative advantages of GENDA-XL.

II. Proposed Method

GENDA-XL encompasses a deep latent variable, denoted as $\hat{z}$, which has the capacity to capture the distribution of the training dataset X, enabling the generation of new data, denoted as $\hat{X}$, by sampling from it. GENDA-XL results from the joint optimization of a generative and a discriminative model, trained using a composite loss function comprising the following components:

- A triplet loss [7], designed to construct a representation space where the similarity between samples is minimized for similar examples and maximized for dissimilar ones;
- A reconstruction loss, which penalizes the network for producing outputs that differ from the input data;
- The classification loss from a pre-trained classifier, ensuring that the proposed autoencoder can link each generated sample to its corresponding label.

A. Model Training

Consider an imbalanced training set X consisting of M samples and let the training point $x_i \in R^D$ represent the i^{th} 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 $d < D$. Specifically, as shown in Fig. 1, the proposed encoder accepts a random reference input x_i (called anchor), a matching input $x_i^+ \in R^D$ (called positive), which is the nearest neighbor of the anchor x_i according to the Euclidean distance, also belonging to the same class, and finally accepts a non-matching input $x_i^- \in R^D$ (called negative), which is the nearest neighbor of the x_i among all the samples that have a different label from x_i. Using the samples x_i and x_i^+, the encoder outputs a latent vector $\{\hat{z}_i\}_{i=1}^M$ for each sample x_i.

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

$$p(\hat{z}_i|x_i, x_i^+) = \frac{p(\hat{z}_i, x_i, x_i^+)}{p(x_i, x_i^+)} \qquad \forall i = 1, ..., M. \tag{1}$$

The suggested encoder adopts a deep convolutional neural network design featuring a trio of identical subnetworks. These subnetworks share identical configurations, parameters, and weights, with simultaneous parameter (i.e., weight and bias) updates occurring across all three subnetworks. As demonstrated in Fig. 1, each of these subnetworks accepts a different input (i.e., the x_i, x_i^+ and x_i^-), and they work in tandem (i.e., simultaneous parameter updating) to find the similarity features z_i, z_i^+ and z_i^-, respectively . Each similarity feature is calculated by a dense layer given as follows:

$$\begin{aligned} z_i &= f(Wh_i + b), \\ z_{i^+} &= f(Wh_i^+ + b), \quad \forall i = 1, ..., M \\ z_{i^-} &= f(Wh_i^- + b), \end{aligned} \tag{2}$$

where f is the $tanh$ activation function, W the weight matrix, h_i, h_i^+ and h_i^- are the outputs of the last dense layers of each subnetwork, and b is the added bias term.

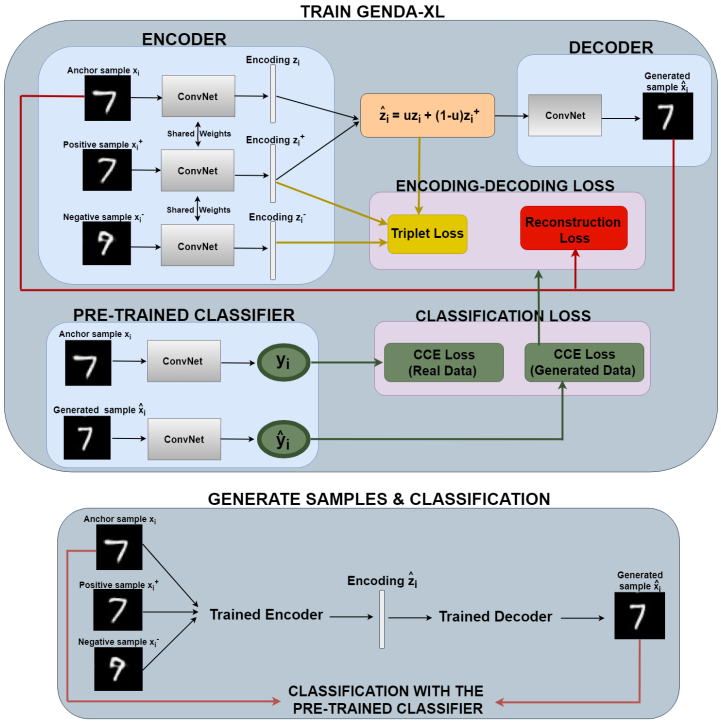

Fig. 1: Proposed generative model GENDA-XL

The latent variable $\hat{z}_i$, which corresponds to the sample x_i, is represented as the linear convex combination of each similarity feature z_i and z_i^+ as shown in the following equation:

$$\hat{z}_i = u \cdot z_i + (1 - u) \cdot z_i^+ \qquad \forall i = 1, ..., M \tag{3}$$

where u is a random number in $(0, 1)$, which follows the uniform distribution. Modelling $\hat{z}_i$ as shown in (3), causes the selection of a random vector along the line segment between two 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 results in unimodal learned representations and does not allow for different or mixed data distributions leading 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.

To ensure that GENDA-XL generates meaningful data representations in the latent space, we introduce the use of a triplet loss. Notably, GENDA-XL stands out as the first generative algorithm to include this particular loss in its objective function. The triplet loss serves as a function for supervised similarity learning, and enforces that, according to distance order, a pair of samples with the same labels are positioned closer to each other than samples with different labels by a minimum margin value denoted as m. In our case, the triplet loss function is calculated $\forall i = 1, ..., M$ as follows:

$$L_T = \max\left\{||\hat{z}_i - z_i^+||^2 - ||\hat{z}_i - z_i^-||^2 + m, 0\right\}, \tag{4}$$

where m is a margin value hyperparameter to keep negative samples far apart. As shown in Fig. 1, after the encoder outputs the latent vector $\hat{z}_i$, the proposed decoder accepts it as input 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:

$$\hat{x}_i = \sigma(W'h + b'), \qquad \forall i = 1, ..., M \tag{5}$$

where σ is the sigmoid activation function, W' is the weight matrix, h is the output of the last dense layer, and b' is the added bias term.

To achieve a useful approximation of the original x_i, the decoder minimizes a mean-squared reconstruction loss:

$$L_R = \frac{1}{M}\sum_{i=1}^{M}(x_i - \hat{x}_i)^2. \tag{6}$$

In our case, the encoder does not take only the sample x_i as input, and thus (6) can be rewritten as,

$$L_R = \frac{1}{M}\sum_{i=1}^{M}(x_i - \hat{x}_i)^2 = \frac{1}{M}\sum_{i=1}^{M}(x_i - d(e(x_i, x_i^+)))^2, \tag{7}$$

where d and e are the decoder and encoder networks, respectively. By reconstructing the sample x_i using (7), i.e. via the embedding space of its nearest neighbor, we ensure that the generated sample $\hat{x}_i$ will be a good approximation of the original sample x_i, yet not its replica. Hence, apart from its ability to generate high-quality samples, our approach effectively mitigates the issue of severe overfitting during the classification process.

In addition, we adopt a classifier, which is trained with the real as well as with the generated data, having the following classification loss:

$$L_C = L_C^1 + L_C^2 = -\sum_{i=1}^{M}\sum_{j=1}^{D} t_{ij}\log(y_{ij}) + t_{ij}\log\hat{y}_{ij} \tag{8}$$

where L_C^1 and L_C^2 correspond to the categorical cross-entropy loss values when the classifier is trained with the real and the generated data respectively. D is the number of classes, t_j is the ground truth distribution of a given sample x_i, represented by a one-hot vector, while y_j and $\hat{y}_j$ are also vectors including the predicted probability values of the classifier for each class when trained with the real and the generated data respectively.

The pre-trained classifier in GENDA-XL performs a dual function. When it comes to the ultimate classification task, it comes with pre-initialized weights, which accelerates the process of achieving more precise and resilient predictions. This is in contrast to training a classification model from the ground up, which involves starting with random weights. Lastly, the L_C^2 term of the classification loss is integrated into the overall loss function of the proposed algorithm, facilitating the autoencoder in associating each generated sample and its corresponding label. Thus, the formulation of the objective function for the proposed framework is as follows:

$$\begin{aligned}\mathcal{L} = \sum_{i=1}^{M}\max\left\{||\hat{z}_i - z_i^+||^2 - ||\hat{z}_i - z_i^-||^2 + m, 0\right\} + \\ \frac{1}{M}(x_i - \hat{x}_i)^2 - \sum_{j=1}^{D} t_{ij}\log\hat{y}_{ij}\end{aligned} \tag{9}$$

B. Data Generation and Classification

Once GENDA-XL has been trained, it can be used for the generation of new samples across all classes. In particular, one can select a point from the latent vector $\hat{z}_i$ generated by the trained encoder, and subsequently feed it through the trained decoder. This process will result in the generation of samples resembling those in the dataset. Moreover, as shown in (3), the scalar coefficient u provides the flexibility to generate an unlimited number of samples. Once the new samples have been generated, we proceed to re-train the pre-trained classifier using a balanced dataset that comprises both real and generated data. In doing so, we make use of the pre-initialized weights of the pre-trained classifier, resulting in quicker and more precise predictions compared to training a classification model from the ground up, which starts with random weights.

III. Evaluation Study

A. Datasets

In our experimental examination of the imbalanced classification problem, we employed four benchmark datasets, which encompass the single-channel MNIST [8] and Fashion-MNIST (F-MNIST) [9] image datasets, as well as the time-series datasets HAR [10] and TwoLeadECG [11]. None of these datasets inherently exhibit class imbalance, and we artificially induced imbalance by randomly selecting instances of varying sizes from different classes. Additionally, for our assessment, we tackled a neuronal cell-type classification problem utilizing a real-world scientific time-series dataset [12]. This particular dataset inherently features class imbalance and pertains to the characterization of the activity of four neuronal cell types over time in the CA1 subregion of the hippocampus, measured using the Ca^{2+} imaging technique [13]. The dataset encompasses four neuronal cell types, including the majority class of excitatory pyramidal cells (PY), as well as three GABAergic interneuronal subtypes: somatostatin-positive (SOM), parvalbumin-positive (PV), which is the minority class, and vasoactive intestinal polypeptide-positive (VIP). Specifics regarding all datasets are summarized in Table I. In the case of the F-MNIST, HAR, and TwoLeadECG datasets, each class is associated to a specific integer, which aligns precisely with the original dataset labels. For the Ca^{2+} imaging dataset, the labels 0, 1, 2 and 3 correspond to the PY cells, the SOM, PV and VIP interneurons, respectively.

B. Implementation details of the proposed method

The encoder within GENDA-XL comprises a total of five convolutional+tanh+average pooling layers and a single dense-sigmoid layer, resulting in a latent dimension of 16. A growing

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.

count of trainable kernels, specifically 16, 32, 64 and 128, has been utilized for each convolutional layer with size $(4, 1)$ for the TwoLeadECG and Ca^{2+} imaging datasets, and $(4, 4)$ for the other datasets. For the pooling operator, a $(2, 1)$ window size has been employed for the TwoLeadECG and Ca^{2+} imaging datasets across all applicable convolutional layers, while $(2, 2)$ was used for the rest of the datasets.

GENDA-XL's decoder consists of one dense layer, three transpose convolutional+BatchNormalization+LeakyReLu layers, and one transpose convolutional-sigmoid layer. In each transpose convolutional layer, a diminishing count of 128, 64, 32 and 1 trainable kernels was utilized with size $(3, 1)$ for the TwoLeadECG and Ca^{2+} imaging datasets and $(3, 3)$ for the rest of the datasets.

The pre-trained classifier integrated into GENDA-XL is composed of five convolutional+LeakyReLU+dropout layers, along with a single dense-softmax layer whose size is contingent on the number of classes specific to each dataset. For each convolutional layer, we employed 128, 64, 32 and 16 trainable kernels with size $(5, 1)$ for the TwoLeadECG and Ca^{2+} imaging datasets and $(5, 5)$ for the rest of the datasets. GENDA-XL underwent training for 40 epochs, and the Adam optimizer was applied with a learning rate of 0.0001.

In the final classification step, which occurs following the training of GENDA-XL and other data augmentation techniques, a classifier is supplied with actual data along with the necessary quantity of generated samples. This approach ensures that the classifier is trained on a balanced dataset. In the case of GENDA-XL, the pre-trained for a total of 40 epochs classifier is responsible for the classification task. However, when the classification task involves the utilization of generated data from another data augmentation method, a classifier with a matching architecture to the pre-trained classifier is employed. This classifier is trained from the beginning with both real and generated data over an 80-epoch period. In order to assess the performance of imbalanced classification, we employed three commonly-used metrics that are robust to skew: Average Class-Specific Accuracy (ACSA), which represents the average accuracy achieved for each class individually, as well as the F1-score and precision.

C. Results and Discussion

In our experiments, we evaluate GENDA-XL's performance by contrasting it with the most recent balancing techniques using real image and time-series data. We also illustrate its qualitative advantages through visualization results of the generated MNIST and F-MNIST images.

For a fair comparison, all models were provided with the same training dataset and assessed using the identical testing dataset. The comprehensive classification results for the image and time-series datasets can be found in Tables II and III, respectively. Table II highlights the degree of improvement in baseline performance for both datasets, particularly in the case of F-MNIST after the application of data augmentation. It is important to note that "baseline" refers to the classifier's performance achieved when trained with the imbalanced dataset. Our observations indicate that GENDA-XL outperforms the other models, with DGC proving to be quite competitive, approaching the performance of GENDA-XL.

Table III presents the outcomes for the time-series data. Regardless of the chosen metric, it is evident that GENDA-XL surpasses all other methods across all datasets. Notably, TimeGAN exhibits the least favorable performance when compared to the other techniques, which might be attributed to the unstable training of the GAN. Furthermore, it is worth noting that all methods, except for SMOTE, exhibit their poorest performance when trained on the Ca^{2+} imaging dataset. This is likely due to the inherent noise associated with Ca^{2+} imaging, arising from the substantial spatiotemporal information required from measurements, which suffer from a low signal-to-noise ratio and factors such as drift or cell movement, especially when dealing with living organisms.

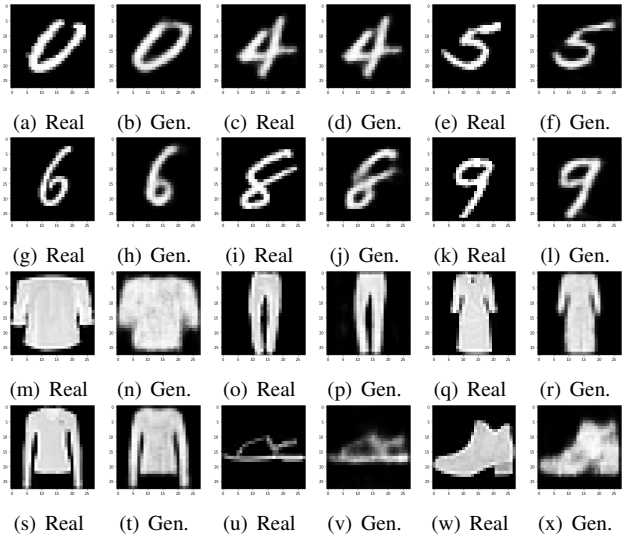

Fig. 2: Real and generated MNIST and F-MNIST images by GENDA-XL

	MNIST			F-MNIST		
Method	ACSA	F1-Score	Precision	ACSA	F1-Score	Precision
Baseline	0.579	0.563	0.54	0.499	0.475	0.454
SMOTE [14]	0.895	0.894	0.883	0.738	0.708	0.712
DGC [5]	0.948	0.947	0.911	0.836	0.831	0.781
BAGAN-GP [15]	0.863	0.85	0.841	0.731	0.729	0.69
GENDA [6]	0.925	0.922	0.926	0.811	0.801	0.794
GENDA-XL	**0.952**	**0.95**	**0.95**	**0.84**	**0.828**	**0.817**

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	0.5	0.5	0.342	0.26	0.65	0.674	0.714
SMOTE [14]	0.731	0.682	0.652	0.81	0.823	0.815	0.77	0.78	0.792
TimeGAN [16]	0.713	0.67	0.643	0.735	0.716	0.693	0.697	0.674	0.654
GENDA [6]	0.877	0.878	0.883	0.829	0.838	0.817	0.787	0.797	0.809
GENDA-XL	**0.919**	**0.918**	**0.919**	**0.938**	**0.934**	**0.927**	**0.834**	**0.846**	**0.859**

TABLE III: Comparing overall classification performance on timeseries datasets.

Fig. 2 showcases the generated images for MNIST and F-MNIST, respectively. The images in Fig. 2(a, m) and 2(k, w) depict the majority and minority class, respectively, for both the MNIST and F-MNIST datasets. Conversely, all other images (i.e., Fig. 2(c, e, g, i, o, q, s, and u)) represent randomly selected classes. The results clearly indicate that GENDA-XL produces synthetic images that are both information-rich (enhancing the classifier's discriminative capabilities and countering majority bias) and visually meaningful (even for minority classes, GENDA-XL generates realistic samples). Ultimately, this visual evidence aligns with the mathematical description of our method in terms of sample diversity, as the generated images are not exact replicas of real images, showcasing the classifier's generalization capacity.

IV. Conclusion

In this paper, we introduced an extension of GENDA, named GENDA-XL, which is a deep generative autoencoder featuring an extended loss function. Its primary purpose is to learn interpretable representations that capture the non-linear structured aspects of the underlying data. GENDA-XL effectively models the data generation process by generating synthetic samples that balance the training set. These synthetic samples can be employed for training any classifier without encountering bias-related issues.

The enhanced performance of GENDA-XL compared to GENDA, can be attributed to its updated objective function, which integrates elements such as a triplet loss, a reconstruction loss, and the classification loss from a pre-trained classifier also used in the final classification task. Experimental results exhibit the superiority of GENDA-XL compared to various other methods, highlighting its remarkable model stability even when dealing with high levels of class imbalance.

V. Acknowledgments

E.T. is co-funded by the State Scholarships Foundation I.K.Y. (MIS-5000432) and the TITAN ERA Chair project (contract no. 101086741). P.P. was supported by NIH (1R01MH124867-02) and the European Commission (H2020-FETOPEN-2018-2019-2020-01, NEUREKA GA-863245). A.L. by NIH RF1NS133381, R01AG076845, R01NS131728, R01NS121106, R01MH124867, and R01MH124047

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] I. Goodfellow, J. Pouget-Abadie, *et al.*, "Generative adversarial nets," Advances in neural information processing systems, vol. 27, 2014.

[3] Z. Pan, W. Yu, *et al.*, "Loss functions of generative adversarial networks (gans): opportunities and challenges," IEEE Trans. on Emerging Topics in Computational Intelligence, vol. 4, no. 4, pp. 500–522, 2020.

[4] X. Li, X. Sun, Y. Meng, J. Liang, *et al.*, "Dice loss for data-imbalanced nlp tasks," arXiv preprint arXiv:1911.02855, 2019.

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

[6] E. Troullinou, G. Tsagkatakis, A. Losonczy, P. Poirazi, and P. Tsakalides, "A generative neighborhood-based deep autoencoder for robust imbalanced classification," IEEE Trans. on Artificial Intelligence, 2023.

[7] X. Dong and J. Shen, "Triplet loss in siamese network for object tracking," in Proc. of the European conference on computer vision (ECCV), 2018, pp. 459–474.

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

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

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

[11] H. A. Dau, E. Keogh, K. Kamgar, C.-C. M. Yeh, *et al.*, "The ucr time series classification archive," October 2018.

[12] E. Troullinou, G. Tsagkatakis, S. Chavlis, G. F. Turi, *et al.*, "Artificial neural networks in action for an automated cell-type classification of biological neural networks," IEEE Trans. on Emerging Topics in Computational Intelligence, vol. 5, no. 5, pp. 755–767, 2020.

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

[14] 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.

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

[16] J. Yoon, D. Jarrett, *et al.*, "Time-series generative adversarial networks," Advances in Neural Information Processing Systems, vol. 32, 2019.