DATA GENERATION WITH STRUCTURE ENFORCING ADVERSARIAL LEARNING

Indu Solomon[⋆] *Uttam Kumar*[⋆] *Senthilnath Jayavelu*[†]

[⋆]International Institute of Information Technology Bangalore, India
[†]Institute for Infocomm Research, Agency for Science, Technology and Research (A*STAR), Singapore

ABSTRACT

Class imbalance issues are very common among real-world datasets. Traditional oversampling approaches are interpolation based and are not well suited for image datasets. These techniques lead to class overlapping and the generation of visually unappealing minority class images. Lately, Generative Adversarial Network (GAN)-based models are used widely for oversampling of image data; however, the learning bias towards the majority classes lead to generation of majority classes in excess and minority classes in rarity. Most of the existing oversampling techniques work on data space, whereas low-dimensional latent space for oversampling is less explored. To tackle these issues, we propose a novel latent space oversampling framework called Structure Enforcing Adversarial Learning (SEAL). In the proposed architecture, the generator is trained by additionally minimizing the structure loss. This boosts the generation of synthetic samples, which retain the covariance structure of each class. The proposed model is evaluated on four image datasets and is compared with the state-of-the-art methods.

***Index Terms*—** Class imbalance, oversampling, generative models, latent space, structure loss

1. INTRODUCTION

Class imbalance has always been a major issue for the machine learning (ML) community. A class imbalance arises due to the unequal number of training samples for different classes in a dataset. The important class in a real-world imbalanced dataset is the minority class(es). However, the learning bias of most ML algorithms towards the majority class affects the generalization of unseen data and tends to wrongly classify the minority class(es). Examples of real-world imbalanced datasets are medical diagnosis, credit fraud detection, and wafer defect detection.

Indu Solomon and Uttam kumar are grateful to International Institute of Information Technology Bangalore, India for the infrastructure support and acknowledges Mphasis Cognitive Computing Centre of Excellence for the financial assistance under Grant No. 7111.

Senthilnath Jayavelu acknowledge funding from the Accelerated Materials Development for Manufacturing Program at A*STAR via the AME Programmatic Fund by the Agency for Science, Technology and Research under Grant No. A1898b0043.

Class imbalance issues are usually countered using data-level [1–4], algorithm-level [5] and ensemble learning techniques [6, 7]. In data-level methods, samples are generated/removed by oversampling/undersampling techniques or randomly re-sampling of available data. Algorithm-level or cost-sensitive techniques assign different costs to different classes according to their distribution. Misclassification of minority class(es) is given a higher weightage compared to misclassification of majority class(es). Ensemble techniques combine several classifiers to improve the classification performance of imbalanced data.

These traditional techniques work well for classical ML models and tabular data. However, image data is more complex compared to tabular data. Conventional oversampling techniques cause class overlap and tend to generate meaningless pixel values [8]. Data augmentation techniques are widely used to balance the imbalanced image datasets. Deep generative models are capable of learning the data distribution and they can generate real looking synthetic images. Two of the most popular generative models are Generative Adversarial Networks (GANs) [9], and Variational AutoEncoders (VAEs) [10]. These models can generate new samples closer to the original data, but when these models are trained on imbalanced data, they suffer from learning bias, mode collapse, stability issues and generation of blurred images.

Most of the existing oversampling techniques generate new minority samples either by interpolating between the selected minority class samples or as a convex combination of existing minority class samples. These techniques are neither exploiting the minority class data distribution nor the structure of the data. These oversampling techniques result in poor utilization of data space and this leads to class overlap. To overcome these drawbacks, Structure Enforcing Adversarial Learning (SEAL) architecture for multi-class imbalance learning is proposed in this paper. SEAL minimizes an additional penalty term to boost the generation of minority class synthetic samples. The proposed method is evaluated on four highly imbalanced datasets and it outperforms the selected state-of-the-art methods. Unlike most other oversampling techniques, the proposed oversampling is performed in low dimensional latent space.

The important contributions of this paper are as follows: i) Structure Enforcing Adversarial Learning (SEAL) architecture, in which SEAL operates on a low-dimensional latent space and performs oversampling by generating latent space embeddings for the minority classes, ii) An additional penalty term called structure loss, which boosts the generation of synthetic samples that can retain the covariance structure of the data, iii) The proposed method is applied on a real-world wafer defect map dataset.

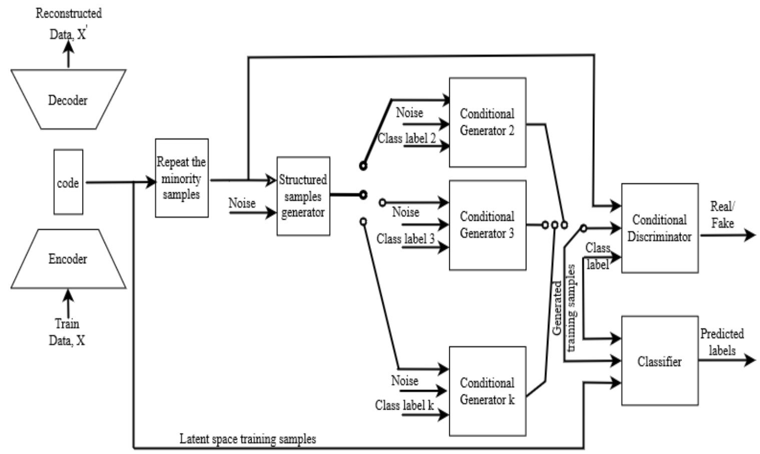
2. RELATED WORKS

The recent literature extends the GAN models for minority class oversampling. Generative Adversarial Minority Oversampling (GAMO) [11] handles multi-class imbalance by a three player adversarial game between a convex generator, a discriminator and a classifier. This helps to generate samples near the minority class boundaries and this leads to generation of blurred samples which are difficult to classify. GAMO also suffers from mode collapse [12]. Deep SMOTE [13] consists of an encoder-decoder framework, a SMOTE-based oversampling method, and a loss function with a reconstruction loss and a penalty term. Deep SMOTE is intended to address the class imbalance issue of image data, but it suffers from slow convergence issues. There are oversampling models, which make use of both GANs and VAEs/autoencoders. BAGAN [14] model for imbalance learning combines an autoencoder and a GAN for data oversampling. However, BAGAN model has limitations, as it suffers from stability issues and also experience slow convergence [15]. Discriminative Variational Autoencoding Adversarial Network (DVAAN) [8] combines a Discriminative VAE (DVAE) and a GAN. DVAAN learns a two component class conditioned mixture distribution, DVAAN model is targeted to address binary class imbalance [8] and not for multi-class imbalance.

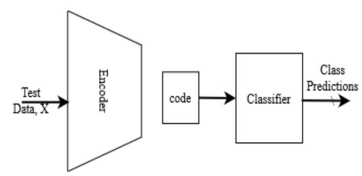
3. SEAL ARCHITECTURE

The proposed SEAL architecture addresses the issue of class imbalance and performs oversampling to generate a balanced dataset to improve the classification performance. Our architecture takes advantage of encoder-decoder framework of VAE to bring down the dimensionality of the data and to capture an uncorrelated class conditional distribution in the low dimensional latent space. The low dimensional latent data is augmented with SEAL, which tries to effectively maintain the structure of the data by minimizing the structure loss along with the generator and discriminator losses. The performance of oversampled data is evaluated by training a classifier. Fig 1(a) shows the SEAL architecture.

In the proposed architecture, a conditional GAN is adopted and additionally penalized to perform oversampling on a low dimensional latent space. The "structured data generator" (Fig 1(a)) block takes lower dimensional latent data as input and computes the covariance structure of the same.



(a) Training phase of the proposed model



(b) Testing phase of the proposed model

Fig. 1: SEAL architecture. Fig1 (a) shows the training phase of SEAL. Encoder-decoder framework helps in bringing down the dimensionality of the data and "repeat samples" block is used for stabilizing the GAN training process. "Structured samples generator" block generates samples which preserve the structure of the data. These samples are used for CGAN training to compute the structure loss and thus to boost the GAN generator to generate samples which maintain the covariance structure of the minority class training data. Fig1(b) shows the testing phase of SEAL.

This block generates new samples by sampling from a standard normal distribution and mapping them to the required covariance structure of the minority class(es). Training of GAN with imbalanced data may lead to instability and mode collapse. This problem is dealt with random re-sampling (Fig 1(a) "repeat minority samples" block) of minority classes. This re-sampled balanced dataset is used for training. Let z_k, μ_k be the sample array and sample mean of k^{th} minority class latent data, then the covariance matrix Σ_k is given by,

$$\Sigma_k = \frac{1}{N}\sum_{k}(z_k - \mu_k)(z_k - \mu_k)^T \tag{1}$$

Eigen spectrum of k^{th} minority class is given by,

$$D_k = V_k^T \Sigma_k V_k \tag{2}$$

where D_k and V_k are the eigen value and eigen vector matrices of k^{th} minority class respectively. Convert to a unit variance distribution by pre and post multiplying the eigen value matrix D_k with $D_k^{-1/2}$.

$$I = (V_k D_k^{-1/2})^T \Sigma_k (V_k D_k^{-1/2}) = F_k^T \Sigma_k F_k \tag{3}$$

The transformation $F_k = V_k D_k^{-1/2}$ has a unit covariance structure. This property is helpful in sampling from a standard normal distribution and apply F^{-1} transform to transport these samples to the required minority class covariance structure. These samples give SEAL generator the required boost to generate more samples that have the same covariance structure as the original data. In effect, "the structured samples block" generated samples act as exemplary samples. SEAL generator has to minimize the structure loss ($\mathcal{L}_{struct}$) between the SEAL generated samples and structured samples. The loss terms minimized during the training phase is given by,

$$\mathcal{L} = \lambda_1 \mathcal{L}_{ED} + \lambda_2 \mathcal{L}_{SEAL} + \lambda_3 \mathcal{L}_{CLF} \tag{4}$$

$$\mathcal{L}_{SEAL} = \lambda_{struct} \mathcal{L}_{struct} + \mathcal{L}_{Gen} + \mathcal{L}_{Dis} \tag{5}$$

$$\mathcal{L}_{struct} = \sum_k \sum_i \mathbb{E}_{z_{ks} \sim S_{\psi_k}(Z_{k_{is}} - G_k(Z_{in}))^2} \tag{6}$$

$$\mathcal{L}_{Gen} = \sum_k \mathbb{E}_{z \sim P_z(z)} (1 - log(D(G_k(z_n|y_k)))) \tag{7}$$

$$\mathcal{L}_{Dis} = \sum_k \mathbb{E}_{x \sim P_{data}(x)} (log(D(x|y))) + \mathbb{E}_{z \sim P_z(z)} (1 - log(D(G_k(z_n|y_k)))) \tag{8}$$

$$\mathcal{L}_{CLF}(f_{\theta_c}, Z, Y) = -\mathbb{E}_{(z_i, y_i) \sim (Z,Y)} \sum_i \sum_{k=1}^{K} \mathbb{1}_{[k=y_i]} \log(f_{\theta_c}(z_i)) \tag{9}$$

$$\mathcal{L}_{ED}(\theta, \phi) = \sum_{i \in X} -\mathbb{E}_{z \sim Q_\theta(z|x_i)} [log P_\phi(x_i|z)] + KL(Q_\theta(z|x_i)||p(z)) \tag{10}$$

Here x, y, z are the data sample, label and latent space sample respectively. Equation (4) refers to the total loss minimized during the training process, the $\mathcal{L}_{ED}$, $\mathcal{L}_{SEAL}$ and $\mathcal{L}_{CLF}$ refers to the encoder-decoder block loss, SEAL loss and classifier loss respectively and λ is a tunable parameter. Equation (5) refers to the total SEAL loss function, SEAL loss is a combination of structure loss and min-max loss of GAN. Equation (6) refers to the structure loss, $\mathcal{L}_{struct}$ is a squared error loss computed between the GAN generator synthesised samples and structured samples and i refers to the training batch. Minimizing the $\mathcal{L}_{struct}$ loss compels the GAN generator to learn the covariance structure of the minority class and to generate structured samples. Equations (7) and (8) refers to generator and discriminator loss functions respectively. Equation (9) refers to the cross entropy loss function of the classifier, where f_{θ_c} refers to the neural network classifier parameterised with θ_c. Z and Y are latent space data and ground truth respectively. Equation (10) refers to the encoder-decoder block loss function, where θ and ϕ refer to the parameters of encoder and decoder blocks respectively. The first term of $\mathcal{L}_{ED}$ is a reconstruction loss and the second term is KL divergence between the encoder distribution and the standard normal distribution. The goal of the encoder is to learn the probability distribution $Q_\theta(z|X)$, given the input data X. The goal of the decoder is to learn the probability distribution $P_\phi(X|z)$. $P(z)$ is the standard normal distribution. KL divergence term of loss function helps in pushing the learnt encoder distribution $Q_\theta(z|X)$ as close as possible to the standard normal distribution.

Fig1(b) shows the testing phase, unseen test data is passed through the encoder block and sent to the classifier to predict the class labels.

4. EXPERIMENTAL RESULTS

The performance of SEAL model is evaluated using four image datasets. The benchmark datasets used for evaluation are MNIST [16], Fashion MNIST [17] and Kuzushiji MNIST [18]. The wafer defect map dataset [19] is collected from the wafer manufacturing plant and it contains 38000 mixed mode wafer map defect data. There are 38 classes in the mixed-type wafer map defect dataset, including 1 normal type, 8 single defect types, and 29 mixed defect types. Imbalance in these datasets is artificially introduced by selecting different number of samples from various classes. The dataset imbalance is defined by imbalance factor IF [20] and it can vary from perfectly balanced ($IF = 0.0$) to extreme imbalance ($IF = 1.0$).

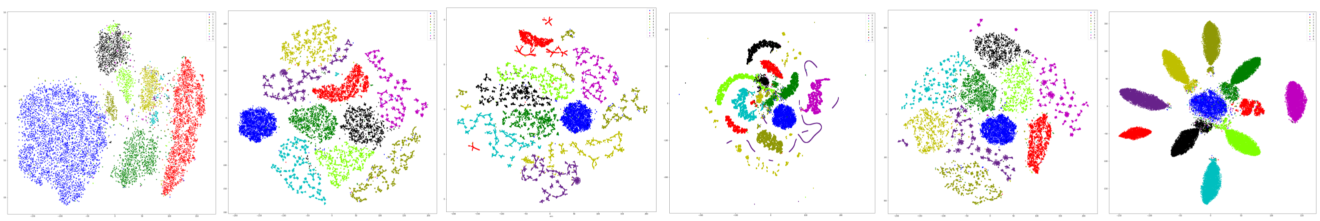
Traditional (SMOTE [1], borderline SMOTE [2]) and generative techniques (GAMO [11], Deep SMOTE [13]) are chosen for conducting a comparison study. To ensure a fair comparison, all chosen techniques are re-run on the same environment with same train and test datasets. Evaluation metrics Average Class Specific Accuracy (ACSA) [20] and Geometric Mean (GM) [20] are used for evaluating the classifier trained on the oversampled dataset.

MNIST [16], FMNIST [17] and KMNIST [18] are multi-class balanced datasets with 60000 training images and 10000 test images of dimension 28×28. The training dataset consists of randomly selected {4000, 2000, 1000, 750, 500, 350, 200, 100, 60, 40} samples ($IF = 0.96$) from the classes {0,1,2…,9} respectively. The testing of the trained model is carried out with both balanced and imbalanced test datasets. Balanced test dataset is formed by 500 non-overlapping samples from each {0,…,9} class. Imbalanced test dataset is formed with {800, 600, 500, 400, 350, 250, 200, 150, 100, 50} non-overlapping samples from classes {0,…,9}.

Wafer defect map imbalanced training dataset consists of randomly selected {900, 800, 700, 600, 500, 400, 300, 200, 100, 50} samples ($IF = 0.89$) from 10 selected classes. Wafer samples are single channel grayscale images of size 52×52. Balanced test dataset is non-overlapping and formed by selecting 100 samples from each {0,…,9} class. Imbalanced test dataset contains {180, 150, 120, 100, 80, 60, 50, 35, 28, 15} non-overlapping samples from classes {0,…,9} respectively. Table 1 shows the experimental results for four

	MNIST		FMNIST		KMNIST		WAFER	
Balanced test dataset								
	ACSA	GM	ACSA	GM	ACSA	GM	ACSA	GM
SMOTE	0.90±0.01	0.89±0.01	0.81±0.01	0.79 ±0.01	0.70±0.01	0.67±0.01	0.80±0.01	0.76 ±0.01
BSMOTE	0.89±0.01	0.88±0.01	0.80±0.01	0.77±0.01	0.70±0.01	0.67±0.01	0.8±0.01	0.75±0.02
GAMO	0.89±0.01	0.88±0.01	0.81±0.01	0.78±0.01	0.73±0.01	0.71±0.01	0.79±0.01	0.74±0.01
Deep SMOTE	0.90±0.01	0.90±0.004	0.81±0.01	0.79±0.01	0.70±0.01	0.67±0.01	0.8±0.01	0.75±0.02
SEAL(Ours)	**0.92±0.01**	**0.91±0.01**	**0.82±0.01**	**0.80±0.01**	**0.75±0.01**	**0.73±0.01**	**0.81±0.01**	**0.78±0.01**
Imbalanced test dataset								
SMOTE	0.90±0.01	0.90±0.01	0.81±0.01	0.78 ±0.01	0.68±0.01	0.65±0.01	0.79±0.01	0.75 ±0.02
BSMOTE	0.90±0.01	0.89±0.01	0.80±0.01	0.77±0.01	0.67±0.01	0.64±0.01	0.78±0.01	0.73±0.02
GAMO	0.89±0.01	0.88±0.01	0.80±0.01	0.77±0.01	0.71±0.01	0.69±0.01	0.78±0.01	0.75±0.01
Deep SMOTE	0.90±0.01	0.90±0.01	0.81±0.01	0.78±0.01	0.68±0.01	0.65±0.01	0.78±0.01	0.72±0.02
SEAL(Ours)	**0.92±0.01**	**0.92±0.01**	**0.82±0.01**	**0.80±0.01**	**0.73±0.01**	**0.70±0.01**	**0.81±0.01**	**0.79±0.01**

Table 1: Performance Comparison of SEAL with other models for MNIST, FMNIST, KMNIST and Wafer defect datasets



(a) Imbalanced data (b) SMOTE (c) BSMOTE (d) GAMO (e) Deep SMOTE (f) SEAL

Fig. 2: t-SNE visualization of (a) Imbalanced data distribution ($IF = 0.96$), (b) SMOTE, (C) Borderline SMOTE, (d) GAMO, (e) Deep SMOTE and (f) SEAL balanced distributions respectively

selected datasets. All experiments were performed for 10 different runs and the means and standard deviations of the evaluation metrics are reported. Oversampled data visualization with the help of t-SNE plot is shown in Fig 2. Tabulated results show that SEAL technique outperforms the selected traditional and generative oversampling techniques for both balanced and imbalanced test data. Fairness of comparative study is ensured by training a similar MLP classifier for all selected methods. We can see that SMOTE group of techniques, both traditional and generative, produced comparable results and t-SNE plots of the oversampled data distribution also look visually very similar. However, Deep SMOTE[1] has the advantage of oversampling in lower dimensional space and hence lesser complexity. Convex generator of GAMO [2] deteriorates with increase in class imbalance factor (IF), leading to class overlap and misclassification. t-SNE visualization evidently shows that our technique is capable of maintaining the structure of the data and retains the elliptical fit. SEAL ablation study was conducted by setting $\lambda_{struct} = 0$ and this resulted in a drop in performance and evaluation metrics. Ablation study results for MNIST and Wafer defect map datasets are tabulated in Table 2. The structure loss has improved the generated minority class data and the improved ACSA, GM values proves that the generated samples are superior.

The hardware and software setup for the experiments are as follows: CPU - Intel® Xeon® CPU E5-2630 v4 2.20 GHz, GPU - Nvidia Quadro RTX 8000, GPU memory - 48 GB, CUDA cores - 4608, python version 3.10.4, NumPy version 1.22.4, pytorch version 1.12.1, torchvision version 0.13.1 and sklearn version 1.1.2. The parameter selection is as follows: latent space dimension is selected between 32 to 64, learning rate is chosen between 0.0002 to 0.0015, activation functions used are ReLU, leaky ReLU and sigmoid, the encoder-decoder has 2 hidden layers, cGAN generator, discriminator has 3 hidden layers, MLP classifier has a three layer architecture and $\lambda_{struct}, \lambda_{1,..,3}$ are set as 0.5, 1.0 respectively.

	MNIST		WAFER	
Method	ACSA	GM	ACSA	GM
SEAL\λ_{struct}	0.912	0.904	0.81	0.77
SEAL	**0.925**	**0.92**	**0.82**	**0.79**

Table 2: Ablation study on MNIST and WAFER dataset

5. CONCLUSIONS

In this paper, we propose a novel Structure Enforcing Adversarial Learning (SEAL) architecture. SEAL performs oversampling and generates low-dimensional latent space samples for balancing the minority classes. SEAL generator is trained with an additional penalty term called structure loss and this added loss term helps in retaining the covariance structure of the generated data. We compared the performance of our technique on four multi-class image datasets and our technique outperforms all the selected baselines.

[1]https://github.com/dd1github/DeepSMOTE
[2]https://github.com/SankhaSubhra/GAMO

6. REFERENCES

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

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

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

[4] Hong Cao, Xiao-Li Li, Yew-Kwong Woon, and See-Kiong Ng, "Spo: Structure preserving oversampling for imbalanced time series classification," in *2011 IEEE 11th International Conference on Data Mining*. IEEE, 2011, pp. 1008–1013.

[5] Nguyen Thai-Nghe, Zeno Gantner, and Lars Schmidt-Thieme, "Cost-sensitive learning methods for imbalanced data," in *The 2010 International joint conference on neural networks (IJCNN)*. IEEE, 2010, pp. 1–8.

[6] Chris Seiffert, Taghi M Khoshgoftaar, Jason Van Hulse, and Amri Napolitano—RUSBoost, "A hybrid approach to alleviating class imbalance‖ ieee transactions on systems," *Man, And Cybernetics—Part A: Systems And Humans*, vol. 40, no. 1, 2010.

[7] Tahira Alam, Chowdhury Farhan Ahmed, Sabit Anwar Zahin, Muhammad Asif Hossain Khan, and Maliha Tashfia Islam, "An effective ensemble method for multi-class classification and regression for imbalanced data," in *Industrial Conference on Data Mining*. Springer, 2018, pp. 59–74.

[8] Ting Guo, Xingquan Zhu, Yang Wang, and Fang Chen, "Discriminative sample generation for deep imbalanced learning," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI-19), August 10-16 2019, Macao, China*, 2019.

[9] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[10] Diederik P Kingma and Max Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[11] Sankha Subhra Mullick, Shounak Datta, and Swagatam Das, "Generative adversarial minority oversampling," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1695–1704.

[12] Seulki Park, Youngkyu Hong, Byeongho Heo, Sangdoo Yun, and Jin Young Choi, "The majority can help the minority: Context-rich minority oversampling for long-tailed classification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 6887–6896.

[13] Damien Dablain, Bartosz Krawczyk, and Nitesh V Chawla, "Deepsmote: Fusing deep learning and smote for imbalanced data," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

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

[15] Gaofeng Huang and Amir Hossein Jafari, "Enhanced balancing gan: Minority-class image generation," *Neural Computing and Applications*, pp. 1–10, 2021.

[16] Yann LeCun, "The mnist database of handwritten digits," *http://yann. lecun. com/exdb/mnist/*, 1998.

[17] Han Xiao, Kashif Rasul, and Roland Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.

[18] Tarin Clanuwat, Mikel Bober-Irizar, Asanobu Kitamoto, Alex Lamb, Kazuaki Yamamoto, and David Ha, "Deep learning for classical japanese literature," *arXiv preprint arXiv:1812.01718*, 2018.

[19] Junliang Wang, Chuqiao Xu, Zhengliang Yang, Jie Zhang, and Xiaoou Li, "Deformable convolutional networks for efficient mixed-type wafer defect pattern recognition," *IEEE Transactions on Semiconductor Manufacturing*, vol. 33, no. 4, pp. 587–596, 2020.

[20] Abeegithan Jeyasothy, Savitha Ramasamy, and Suresh Sundaram, "Meta-neuron learning based spiking neural classifier with time-varying weight model for credit scoring problem," *Expert Systems with Applications*, vol. 178, pp. 114985, 2021.