Examining Effects of Class Imbalance on Conditional GAN Training

Yang Chen*, Dustin J. Kempton, and Rafal A. Angryk

Georgia State University, Atlanta, GA 30302, USA
{ychen113}@student.gsu.edu, {dkempton1, angryk}@cs.gsu.edu

Abstract. In this work, we investigate the impact of class imbalance on the accuracy and diversity of synthetic samples generated by conditional generative adversarial networks (CGAN) models. Though many studies utilizing GANs have seen extraordinary success in producing realistic image samples, these studies generally assume the use of well-processed and balanced benchmark image datasets, including MNIST and CIFAR-10. However, well-balanced data is uncommon in real world applications such as detecting fraud, diagnosing diabetes, and predicting solar flares. It is well known that when class labels are not distributed uniformly, the predictive ability of classification algorithms suffers significantly, a phenomenon known as the "class-imbalance problem." We show that the imbalance in the training set can also impact sample generation of CGAN models. We utilize the well known MNIST datasets, controlling the imbalance ratio of certain classes within the data through sampling. We are able to show that both the quality and diversity of generated samples suffer in the presence of class imbalances and propose a novel framework named Two-stage CGAN to produce high-quality synthetic samples in such cases. Our results indicate that the proposed framework provides a significant improvement over typical oversampling and undersampling techniques utilized for class imbalance remediation.

Keywords: class-imbalance issue · generative adversarial networks · synthetic data

1 Introduction

Most classification algorithms assume that training data classes are distributed uniformly. When this assumption is questioned, regular algorithms suffer from the class-imbalance problem, i.e., their ability to predict minority classes decreases significantly. This well-known issue can also have a profound effect on training generative adversarial networks (GAN). So much so, that the authors of [1] state that traditional GANs cannot be employed to generate minority-class images from an imbalanced dataset. There have been few studies conducted to address this imbalance issue, one of which being BAGAN [2]. In the BAGAN

* Corresponding author: Yang Chen
ychen113@student.gsu.edu

work, an augmentation tool for generating high-quality images of minority classes was developed by achieving the following: (1) Using an autoencoder to initiate the GAN training, allowing the model to learn accurate class-conditioning information in the latent space. (2) Combining the real/fake loss and classification loss at the discriminator into a single output. Based on BAGAN, authors of [1] utilize the supervised Autoencoder and gradient penalty to solve the instability problem when images from different classes appear similar. Nevertheless, the aforementioned works attempt to address the imbalance issue at the algorithm level, either by employing Autoencoder to learn latent features or by modifying objective functions during the training procedure. In this work, we investigate the issue of class imbalance inherent to GAN training at the data-level, and develop a solution through the following contributions:

- Show how the imbalance in the training set has a negative effect on the performance of GANs.
- Show the ineffectiveness of common remedies for training GANs on imbalanced datasets, such as oversampling and undersampling.
- Propose a novel solution, Two-stage CGAN, to enhance the quality of samples from minority classes when training GAN models on imbalanced datasets.
- Show that the proposed framework can generate synthetic samples of higher quality than scenarios that use the original imbalanced set or sets that are rebalanced by oversampling or undersampling.

2 Related Work

In this section, we begin with an overview of the issue of class imbalance and the traditional methods used to overcome it. Next, we introduce the concept of a generative adversarial network (GAN) and its many variants, including the conditional GAN (CGAN) employed in this study. Additionally, we present Fréchet Inception Distance, or FID, as the standard measure for assessing the quality of generative models.

2.1 Imbalance issue

Class imbalance typically occurs when there are more instances of some classes than others. It is common to use special remedies to address the class imbalance if it is present, since standard classifiers can be overwhelmed by the majority classes and neglect the minority ones. In typical class-imbalance situations, the minority class is the class of interest and therefore cannot be ignored. As a result, two approaches to overcoming the imbalance issue are established: either reduce the class skew at the data level or alternate the learning procedure at the algorithm level. As the representative method of data level, resampling is a classifier-independent technique for addressing imbalanced data, and it is accomplished in one of three ways: (1) Oversampling: selecting and duplicating samples of the minority class; (2) Undersampling: removing samples of the majority class; or (3)

Hybrid: coupling the oversampling and undersampling methods when multi-class data are present [3]. The authors of [4] show that the classification performance improves when the above class-imbalance remedies are applied to a solar flare benchmark dataset, namely SWAN-SF [5]. However, random undersampling can jeopardize the preservation of important concepts because it removes the most samples from the majority classes [6]. Random oversampling is susceptible to the risk of overfitting because it neither introduces nor utilizes new data. To reduce such risks in the image domain, we can perform transformation-based data augmentation, a heuristic oversampling strategy for dealing with the lack of data. To achieve this, the current examples are subjected to one or more data transformations, such as random rotation, translation, reflection, cropping, blurring, sharpening, and hue adjustment. These transformations are not applicable in all circumstances. A reflection or affine transformation, for instance, would alter the chirality of a picture of a solar filament. In addition, it is challenging to apply transformation-based data augmentation to feature-based data points or sequential data such as time series and text data [7]. To deal with such a situation, SMOTE [8], a heuristic oversampling method, is introduced by constructing new synthetic samples between minority instances and their nearest neighbors of the same class, but it may suffer when the separation between majority and minority clusters is not always obvious, resulting in noisy samples [6]. In addition, the method is based on information from the local area, not the overall distribution of minority classes [9]. Generative Adversarial Networks provide an alternative method for addressing the lack of data by learning the underlying distribution of real samples and then generating new realistic samples [10, 11].

In contrast to the solutions discussed at the data level, algorithm-level approaches are promptly implemented within the training procedures of the classifiers under consideration in three ways: (1) Classifier adaptation: adapting existing machine learning algorithms to a particular imbalanced dataset [12]; (2) Ensemble learning: combining several base models to construct an optimal predictive model. One example is dividing the sample set of majority classes into multiple small portions that are balanced with minority classes, and then, training multiple individual classifiers to classify the data, yielding the final decision through a voting mechanism [13]; (3) Cost-sensitive learning: designating a high misclassification cost to minority classes with the objective of minimizing the total cost [14]. There are a variety of approaches to the class-imbalance problem, but resampling methods that manipulate existing data or generate synthetic data to accomplish a balanced class distribution are more versatile than algorithm modifications. In this work, we therefore place greater emphasis on data-level solutions.

2.2 Generative Adversarial Network

Generative Adversarial Network is an emerging method for modeling implicitly the high-dimensional distributions of actual samples [15]. Originally proposed in [10], the GAN learns to generate plausible data by training two adversarial components, the generator and the discriminator. First, the generator is used to capture the data distribution by sampling random vectors from a latent space as inputs and producing samples that resemble the actual data. Next, the discriminator receives both generated and actual samples as inputs and estimates the probability that the input originated from the real data space. By simultaneously training the generator and the discriminator, a generator can generate progressively more realistic samples under the supervision of actual samples. This procedure is repeated until the discriminator is unable to distinguish between generated and actual samples. Depending on the actual data source, either the generator or the discriminator can typically be implemented by arbitrary multilayer neural networks consisting of fully connected networks, convolutional neural networks, and recurrent neural networks.

The vanilla GAN has limitations regarding the stability of model training and the diversity of the samples it generates [16]. Consequently, a number of studies have investigated the design of novel architectures to mitigate training issues and enhance the quality of generated samples. Deep Convolutional GAN (DCGAN) replaces pooling layers with strided convolutions (discriminator) and fractional-strided convolutions (generator) to enhance training stability [17]. The Wasserstein GAN implements the Earth-Mover distance to enhance learning stability and provide a meaningful learning curve for hyperparameter tuning [16]. Conditional information is incorporated into the Conditional GAN (CGAN) to enhance the quality of the generated samples and control the classes of synthetic samples [18]. Class labels are the most common type of conditional information.

2.3 Fréchet Inception Distance

Introduced in 2017, the Fréchet Inception Distance (FID) score is the current standard metric for evaluating the quality of generative models. Using the feature vectors derived from the Inception v3 model [19], FID calculates the distance between real and generated images. Specifically, the final pooling layer preceding the classification of output images is used to capture computer-vision-specific features of an input image. In practice, each input image is represented as a vector of 2048 units. Suppose that if we select 1,000 real samples and 1,000 synthetic samples, X and Y are feature vectors of the real and synthetic samples with the same shape $[1000, 2048]$. Then, multivariate FID can be computed based on the formulation in Eq. 1. μ_X and μ_Y are the vector magnitudes X and Y, respectively. $Tr(.)$ is the trace of the matrix, while Σ_X and Σ_Y are the covariance matrices of X and Y. Lower FID values indicate higher quality and diversity in synthetic samples.

$$FID = ||\mu_X - \mu_Y||^2 + Tr(\Sigma_X + \Sigma_Y - 2\sqrt{\Sigma_X \Sigma_Y}) \tag{1}$$

3 Methodology

3.1 Recap: CGAN

In this project, we use the Conditional Generative Adversarial Network (CGAN), and there are two main justifications for doing so: To begin, CGAN allows us to control the category of generated samples, enabling us to generate samples of minority classes to alleviate the class imbalance problem. Second, when compared to the vanilla GAN [20], it can provide more stable and quicker training. Figure 1 depicts the design of CGAN. The generator's (G) ultimate goal is to produce output that is similar to the real data. The method begins by taking a random input vector (Z) and a conditional vector (C). The generator's outputs, known as generated or synthetic samples, are computed by feeding them through the LSTM and Dense layers pipelines.

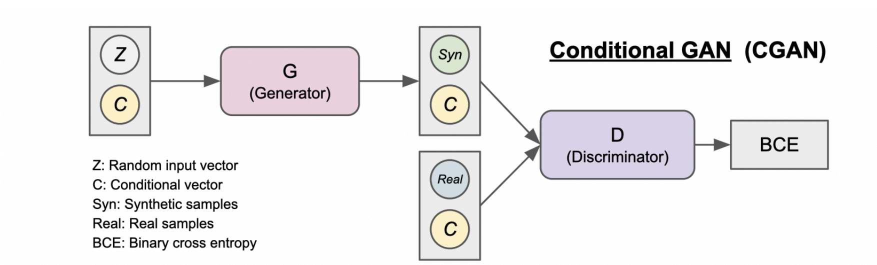

Fig. 1: This is the framework of the CGAN model, including components of the generator (G) and the discriminator (D). The inputs of the generator are random input vectors concatenated with conditional vectors. The inputs of the discriminator are either synthetic or real samples with conditional vectors. The binary cross-entropy is the criterion for optimizing the model.

A discriminator (D) is responsible for classifying inputs as either real or generated samples produced by the generator. The discriminator accepts as inputs both the real and generated samples. By inputting C into D, the discriminator determines whether the sample is generated or real and assesses whether the generated sample's category corresponds to its conditional information. Backpropagation is then used to adjust the weighting parameters of the generator and discriminator based on the binary cross-entropy loss calculated between the predicted and actual values.

3.2 Two-stage CGAN

After describing how CGAN is constructed, we present a new framework named Two-stage CGAN for addressing the class imbalance problem in CGAN model training. The proposed pipeline consists of three steps. To begin with, if the original set is unbalanced, we use random undersampling to reduce it to a smaller, more balanced set (i.e., Training-set-1 in Figure 2) and then train the first CGAN model ($CGAN_1$) on it. After completing $CGAN_1$ training, we can generate synthetic minority class samples, resulting in Synthetic-set-1. The reason for performing undersampling and generating the Synthetic-set-1 dataset based on it is that we discovered that the $CGAN_1$ can generate synthetic samples of minority classes with acceptable diversity. In the intermediary step, the original set and Synthetic-set-1 are merged to create Training-set-2 in Figure 2, a balanced and much larger set. This dataset is then used to train the second CGAN model ($CGAN_2$). Again, we generate synthetic minority class samples to construct the Synthetic-set-2. In the final step, the Original-set and Synthetic-set-2 will be combined to form the final training set (i.e., Final-set in Figure 2) for subsequent applications.

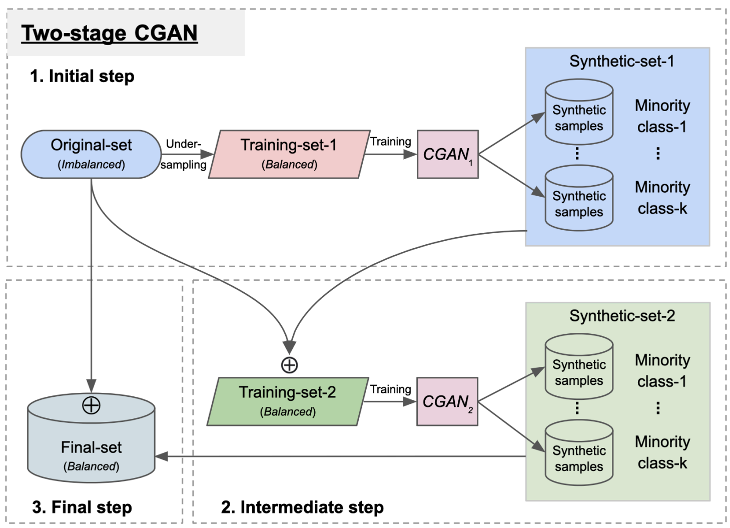

Fig. 2: The Two-stage CGAN framework consists of three steps: (1) undersampling Original-set and training the $CGAN_1$ model on it to form Synthetic-set-1 for minority classes; (2) merging Original-set and Synthetic-set-1 to training the $CGAN_2$ model to produce Synthetic-set-2 for minority classes; and (3) combining Original-set and Synthetic-set-2 to obtain Final-set for subsequent applications.

4 Experiments and Results

4.1 Dataset

MNIST is a benchmark database of handwritten digits that is frequently used to train and evaluate machine learning algorithms [21]. The original dataset consists of 10 classes, which are distributed evenly across 60,000 training images

Table 1: The table lists five datasets intended to assess the performance of CGAN training. A is directly taken from the original MNIST. B is produced by reducing the minority classes of $'3'$ and $'4'$ to 500 and 100 samples, respectively, based on A. C and D are obtained by employing oversampling and undersampling strategies to B. E is the dataset that has been augmented on B using Two-stage CGAN.

Dataset	Type	Digit Class					Total
		0	1	2	3	4	
A	Balanced	5923	6742	5958	6131	5842	30596
B	Imbalanced	5923	6742	5958	500	100	19223
C	Oversampling (OS)	5923	6742	5958	6000	6000	30623
D	Undersampling (US)	100	100	100	100	100	500
E	Two-stage CGAN	5923	6742	5958	6000	6000	30623

and 10,000 testing images. For the sake of brevity, we use only a subset of the original MNIST and perform the necessary resampling operations to meet the experimental requirements. More specifically, we select five digit classes out of ten, and we consider $\{'0',' 1',' 2'\}$ to be the majority classes and $\{'3',' 4'\}$ to be the minority classes, as shown in Table **??**. In addition, we manually generate five different datasets to evaluate the efficacy of CGAN models trained on them. The dataset-A is derived directly from the original MNIST, which has approximately 6,000 samples per class and is balanced. The dataset-B is created based on A by reducing the minority classes of $'3'$ and $'4'$ to 500 and 100 samples, respectively. We chose 500 and 100 because we wish to examine two distinct imbalance ratios, which are approximately 1:12 and 1:60. If the assumption that the class imbalance issue affects the performance of CGAN models holds true, we consider two common resampling strategies in practice: oversampling and undersampling. The dataset C is created by duplicating and rebalancing existing samples of classes $'3'$ and $'4'$ with majority classes. We can also determine if the overfitting issue resulting from oversampling the underrepresented classes is affecting the sample quality. The dataset D is generated by removing the existing samples of majority classes to align their size with the size of minority classes. The dataset E differs from the dataset C in that it was oversampled using Two-stage CGAN, a newly devised framework. Instead of duplicating existing samples, we rebalance the dataset by adding 5,500 and 5,900 synthetic samples, respectively, to the minority classes of $'3'$ and $'4'$.

4.2 Experimental settings

We evaluate the performance of CGAN model with the same hyper-parameter configuration across different experiments, setting the latent space dimension to 3, the learning rates to 0.1, the batch size to 32, and the LSTM hidden size of 100. The models were trained with 500 epochs. Empirically, we use the Adam Optimizer for the generator and the Gradient Descent Optimizer for the discriminator. The CGAN model is implemented based on the TensorFlow 2.1 library [22].

For the sake of simplicity, we only display the FID score distribution of the CGAN trained on dataset-A in Figure 3 when performing model selection based on FID scores. We review the checkpoints every 25 epochs, between the 200th and 500th epochs. We conclude that the 300th epoch is a reasonable option given the trade-off between performance and computational cost. We use the 300th checkpoint in evaluation with experiments A, B, C, and E because the dataset size variation is insignificant. Since the experiment D dataset is much smaller than other datasets, we repeat the FID-based model selection process for it, and we choose to use the 900th checkpoint in the evaluation that follows.

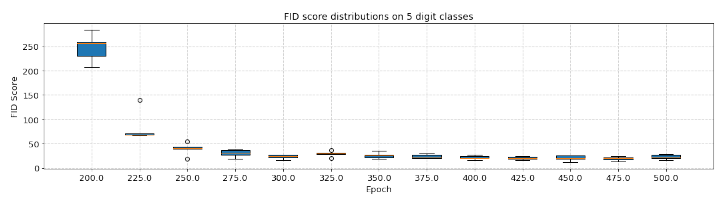

Fig. 3: The box plots depict the distributions of FID scores for five digit classes (i.e., $'0'$, $'1'$, $'2'$, $'3'$, and $'4'$) as calculated by the CGAN model trained on dataset-A. The x-axis represents the models per 25 epochs between the 200th and 500th epochs, and the y-axis represents the corresponding FID scores for each class. This metric is considered the selection criterion for models.

4.3 Results and Analysis

Using the setup shown in Table 1, we trained CGAN models on each of the five datasets separately. Fig.4 shows examples of the output from these models. By looking at the outputs in subplot (A), it is evident that a CGAN trained on a balanced training set is capable of producing acceptable synthetic samples for all classes. However, in the subplot (B), we discovered that the generated samples of class $'4'$ are of lesser quality, whereas we can generate samples of comparable quality for other classes using A. This confirms the assumption that the imbalance ratio in the training set can affect the performance of a GAN, i.e. that GANs give more attention to the majority classes in practice. In scenario (C), we observe that both $'3'$ and $'4'$ synthetic samples have low diversity and low quality. This may be because the random oversampling strategy typically involves duplicating samples exactly to expand the data space, which may lead to the overfitting issue. Therefore, balancing the training set by randomly oversampling minority class samples cannot enhance the performance of the CGAN and

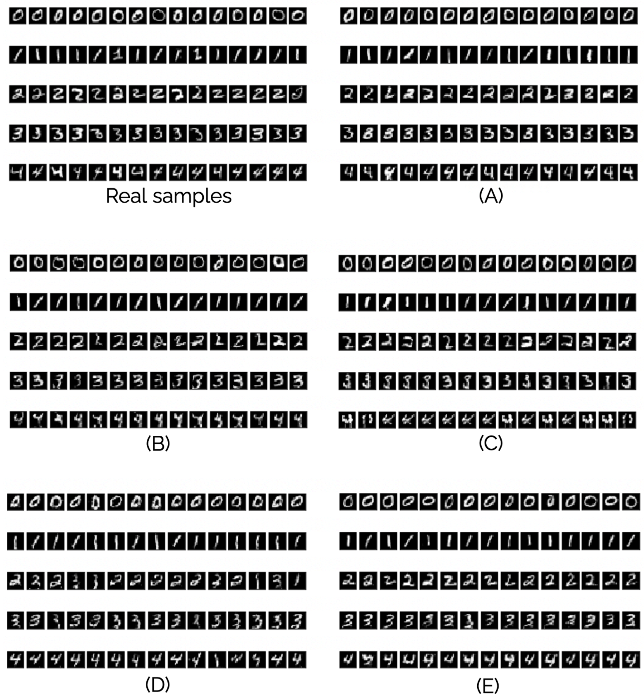

Fig. 4: The diagram shows real samples and synthetic samples generated by CGAN models trained on five datasets listed in Section 4.1.

generate high-quality synthetic samples. In (D), we can see that the diversity of minority classes is better than the results in (C), although this strategy can put the loss of important concepts at risk. The lower-quality outputs are caused by insufficient training data. The subplot (E) shows the synthetic samples generated by Two-stage CGAN, which improve quality and diversity simultaneously. In (D), we can observe that the diversity of minority classes is greater than in (C), despite the fact that this strategy puts at risk the loss of important concepts. Insufficient training data is responsible for the lower-quality results. The subplot (E) depicts the synthetic samples generated by Two-stage CGAN, which enhance both quality and diversity simultaneously.

The FID score is then utilized to quantitatively assess the similarity between the real and synthetic samples. Specifically, 1,000 real samples per class are selected at random, and 1,000 synthetic samples per class are generated using CGAN models trained on five training sets. The results of the FID are shown in Table 2. Row-A displays the FID scores of five classes for a balanced training set, which can be interpreted as the baseline similarity between real and synthetic samples. Row B is the result of an imbalanced training set. We discovered that the FID scores for the majority classes (i.e., $'3'$ and $'4'$) are lower than A while the scores for the minority classes (i.e., $'3'$ and $'4'$) are higher than A, with means

Table 2: The table provides a summary of the FID evaluation outcomes from five experiments. Each FID score is determined by comparing 1,000 actual and 1,000 synthetic samples of the same class. Five separate simulations are performed to calculate the final results, guaranteeing the correctness of the assessment.

$\text{FID}_{(\text{mean}\pm\text{std})}$	Digit - 0	Digit - 1	Digit - 2	Digit - 3	Digit - 4
A	21.44 ± 0.32	15.81 ± 0.20	27.47 ± 0.36	26.62 ± 0.45	24.99 ± 0.55
B	15.28 ± 0.38	11.07 ± 0.26	15.11 ± 0.15	32.89 ± 0.23	86.79 ± 0.50
C	15.12 ± 0.19	14.16 ± 0.36	25.46 ± 0.44	51.75 ± 0.81	142.59 ± 0.59
D	55.36 ± 0.48	30.31 ± 0.28	59.81 ± 0.39	53.73 ± 0.57	45.32 ± 0.46
E	12.44 ± 0.28	10.03 ± 0.39	17.11 ± 0.52	10.31 ± 0.23	9.81 ± 0.19

of 32.89 and 86.79, respectively. The "4" digit class, which has the greatest imbalance in our design (approximately 1:60), is especially affected. There are two possible explanations for why the digit class of $'3'$ is not significantly affected: (1) Because the class of $'3'$ is not the rarest, the weights in the generator for generating $'3'$s receive more training opportunities than the weights for generating $'4'$s; (2) because the digits $'2'$ and $'3'$ are naturally more similar, feeding sufficient samples of '2' into the training process can aid the training process of $'3'$. The FID scores of the oversampling strategy as a remedy for class imbalance are displayed in Row C. The increased FID scores of both minority classes (i.e., $'3'$ and $'4'$) indicate less similarity between real and synthetic samples. This is more evident for $'4'$, indicating that oversampling cannot mitigate the data deficiency issue and result in a well-trained synthetic data generator for minority classes. Row D displays the FID score of using the undersampling strategy as the class imbalance remedy, which results in higher FID scores than Row A. However, given that the training set is balanced, the variances in FID are not that great. In D, the FID score of the digit $'4'$ is 45.81, which is lower than in B and C, indicating that the synthetic samples of $'4'$ are more similar to real samples of $'4'$. Therefore, we are considering utilizing this advantage to generate synthetic samples of minority classes to supplement the imbalance dataset (i.e., dataset B), and then training a final CGAN model on a larger and more balanced training set, yielding the result of Row E. Observing the FID results for Row E, we can see that it achieves the lowest FID scores of all classes, indicating the proposed framework provides a significant improvement over typical oversampling and undersampling techniques utilized for class imbalance remediation.

5 Conclusion

In this study, we show how the imbalance in the training set has a negative effect on the performance of GANs. In addition, we show the ineffectiveness of common remedies for training GANs on imbalanced datasets, such as oversampling and undersampling. We propose a novel solution named Two-stage CGAN, to improve the quality of samples from minority classes in two stages. Our experimental results show that the proposed framework can generate synthetic samples of higher quality than scenarios that use the original imbalanced set or sets that are rebalanced by oversampling or undersampling. In our future work, we plan to improve the algorithm in multiple ways. The first is to investigate heuristic-based undersampling techniques to preserve as much diversity as possible for the majority classes in the original set. The second is to extend the usage of Two-stage CGAN to time series data, including univariate and multivariate time series. A FID-like score is expected to be implemented using representation learning methods such as Autoencoder and dictionary learning to evaluate the quality of generated time series. Furthermore, we would like to investigate if adding more stages to the current framework would increase the quality and diversity of synthetic samples for minority classes.

Acknowledgment

This project has been supported in part by funding from the Division of Advanced Cyberinfrastructure within the Directorate for Computer and Information Science and Engineering, the Division of Atmospheric & Geospace Sciences within the Directorate for Geosciences, under NSF awards #1931555 and # 1936361. It has also been supported by NASA's Space Weather Science Application Research-to-Operations-to-Research program grant #80NSSC22K0272.

References

1. G. Huang and A. H. Jafari, "Enhanced balancing gan: Minority-class image generation," *Neural Computing and Applications*, pp. 1–10, 2021.
2. G. Mariani, F. Scheidegger, R. Istrate, C. Bekas, and C. Malossi, "Bagan: Data augmentation with balancing gan," *arXiv preprint arXiv:1803.09655*, 2018.
3. A. Ahmadzadeh *et al.*, "Challenges with extreme class-imbalance and temporal coherence: A study on solar flare data," in *2019 IEEE International Conference on Big Data (Big Data)*, 2019, pp. 1423–1431. [Online]. Available: https://doi.org/10.1109/BigData47090.2019.9006505
4. A. Ahmadzadeh *et al.*, "How to train your flare prediction model: Revisiting robust sampling of rare events," *The Astrophysical Journal Supplement Series*, vol. 254, no. 2, p. 23, may 2021. [Online]. Available: https://doi.org/10.3847/1538-4365/abec88
5. R. A. Angryk *et al.*, "Multivariate time series dataset for space weather data analytics," *Scientific Data*, vol. 7, no. 1, Jul. 2020. [Online]. Available: https://doi.org/10.1038/s41597-020-0548-x
6. G. Douzas, F. Bacao, and F. Last, "Improving imbalanced learning through a heuristic oversampling method based on k-means and smote," *Information Sciences*, vol. 465, pp. 1–20, 2018.
7. Q. Wen, L. Sun, F. Yang, X. Song, J. Gao, X. Wang, and H. Xu, "Time series data augmentation for deep learning: A survey," *arXiv preprint arXiv:2002.12478*, 2020.

8. N. Chawla, K. Bowyer, L. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, 2002. [Online]. Available: https://doi.org/10.1613/jair.953
9. 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. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417417306346
10. Goodfellow *et al.*, "Generative adversarial nets," in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, ser. NIPS'14. Cambridge, MA, USA: MIT Press, 2014, p. 2672–2680. [Online]. Available: https://dl.acm.org/doi/10.5555/2969033.2969125
11. Y. Chen, D. J. Kempton, A. Ahmadzadeh, and R. A. Angryk, "Towards synthetic multivariate time series generation for flare forecasting," Cham, pp. 296–307, 2021. [Online]. Available: https://doi.org/10.1007/978-3-030-87986-0_26
12. K. Chen, B.-L. Lu, and J. T. Kwok, "Efficient classification of multi-label and imbalanced data using min-max modular classifiers," in *The 2006 IEEE International Joint Conference on Neural Network Proceedings*. IEEE, 2006, pp. 1770–1775.
13. M. A. Tahir, J. Kittler, and F. Yan, "Inverse random under sampling for class imbalance problem and its application to multi-label classification," *Pattern Recognition*, vol. 45, no. 10, pp. 3738–3750, 2012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320312001471
14. Y. Sun, M. S. Kamel, A. K. Wong, and Y. Wang, "Cost-sensitive boosting for classification of imbalanced data," *Pattern recognition*, vol. 40, no. 12, pp. 3358–3378, 2007.
15. A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 53–65, 2018. [Online]. Available: https://doi.org/10.1109/MSP.2017.2765202
16. M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proceedings of the 34th International Conference on Machine Learning - Volume 70*. JMLR.org, 2017, p. 214–223. [Online]. Available: https://dl.acm.org/doi/10.5555/3305381.3305404
17. A. Radford *et al.*, "Unsupervised representation learning with deep convolutional generative adversarial networks," *CoRR*, vol. abs/1511.06434, 2016.
18. M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014. [Online]. Available: http://arxiv.org/abs/1411.1784
19. C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2818–2826.
20. J. Brownlee, *Generative Adversarial Networks with Python: Deep Learning Generative Models for Image Synthesis and Image Translation*. Machine Learning Mastery, 2019. [Online]. Available: https://books.google.com/books?id=YBimDwAAQBAJ
21. Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
22. M. Abadi *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org.