Improving Slow Transfer Predictions: Generative Methods Compared

Jacob Taegon Kim
University of California, Berkeley
Berkeley, CA, USA
jacobkim@berkeley.edu

Alex Sim, Kesheng Wu
Lawrence Berkeley National Laboratory
Berkeley, CA, USA
{asim, kwu}@lbl.gov

Jinoh Kim
Texas A&M University
Commerce, TX, USA
jinoh.kim@tamuc.edu

***Abstract*—Monitoring data transfer performance is a crucial task in scientific computing networks. By predicting performance early in the communication phase, potentially sluggish transfers can be identified and selectively monitored, optimizing network usage and overall performance. A key bottleneck to improving the predictive power of machine learning (ML) models in this context is the issue of class imbalance. This project focuses on addressing the class imbalance problem to enhance the accuracy of performance predictions. In this study, we analyze and compare various augmentation strategies, including traditional oversampling methods and generative techniques. Additionally, we adjust the class imbalance ratios in training datasets to evaluate their impact on model performance. While augmentation may improve performance, as the imbalance ratio increases, the performance does not significantly improve. We conclude that even the most advanced technique, such as CTGAN, does not significantly improve over simple stratified sampling.**

***Index Terms*—class imbalance, data augmentation, generative sampling, CTGAN, prediction, machine learning**

I. Introduction

Monitoring data transfer performance is one of the essential tasks in scientific computing networks [1]. While keeping track of large transfers ("elephant flows") would be crucial considering their high bandwidth share and thus their impact on overall network performance, understanding the characteristics of "slow" transfers would have significance in regard to user experience and network operations. For example, any possibility of (predicted) sluggish activities can be informed to the user in an early stage of the communication. This early notification would be beneficial since the user has a chance to consider an alternative way, and at the same time, the network can save the usage of its resources from being wasted by not-in-progress connections. In that regard, predicting data transfer performance before launching actual transfer would be essential for optimizing network usage and performance. With the predicted performance, a transfer that could be sluggish can be predicted in the initial communication phase and can be monitored in a selective fashion rather than keeping track of the entire every single transfer.

Unfortunately, however, it is highly challenging to perform the monitoring task in a per-transfer manner. One of the challenges is the class imbalance concern that often adversely impact on prediction performance [2]. In fact, class imbalance appears frequently in real-world scenarios, and is often the bottleneck to the performance of machine learning (ML) models. For instance, the data transfer log this study refers to contain very small slow transfers in quantity (0.0002% for slow vs. 0.9998% for non-slow transfers). This study focuses on a class imbalance issue to understand, characterize, and resolve the imbalance problem for enhancing prediction performance.

When dealing with class imbalance, well-known approaches are undersampling and oversampling techniques. While undersampling addresses class imbalance issue, it will lead to insufficient datasets as well as creating imperfect representation of the datasets. On the other hand, oversampling may cause overfitting, and arguably worse, provide bunch of noise. On our previous work in [3], we have studied different techniques for undersampling such as stratified sampling. In this study, we take a data augmentation approach to see its feasibility on slow transfer prediction.

Data augmentation is often considered for mitigating the class imbalance problem, which adds up minority class samples via, for example, statistical methods (e.g., Synthetic Minority Over-sampling Technique (SMOTE) [4] and Adaptive Synthetic Sampling (ADASYN) [5]) or neural network-based generative tools (e.g., Generative Adversarial Network (GAN) [6] and its variants such as Conditional Tabular GAN (CTGAN) [7]). In this study, we investigate how generating synthetic minor class samples could help improve the overall predicting accuracy in the context of slow-transfer prediction. We experiment different data augmentation techniques to evaluate their impact on performance, and present our augmentation method based on CTGAN, which outperforms the existing augmentation tools, but not significantly better than simple stratified sampling. This experimental results can be used for predicting transfer performance and classifying data transfers based on the prediction (e.g., slow or elephant flows) in scientific facilities.

II. Related Work

Monitoring network traffic is essential in network operations and management for detecting anomalous events and estimating network performance. In the realm of scientific computing, this task becomes even more critical due to the increasing demands of data-intensive exploration and computational activities. Identifying "elephant flows"—large data transfers that consume significant network capacity—has been a focal point in prior research. For instance, Basat et al. [8] introduced an algorithm that estimates the traffic volume of individual flows

to detect elephant flows based on total byte counts. They utilized two hash tables to record counters associated with flow IDs from packet traces, enabling the detection of flows exceeding a predefined threshold. Similarly, Chhabra et al. [9] approached the classification of elephant (large transfer) and mice (small transfer) flows using an unsupervised Gaussian Mixture Model clustering based on NetFlow data. While these studies emphasize the importance of identifying large flows, our research focuses on predicting slow connections, which can significantly impact the performance of data-intensive scientific applications.

Several studies have analyzed `tstat` data to improve network performance monitoring and anomaly detection. Syal et al. [10] presented a classification mechanism to detect low-throughput time intervals. Their approach involved a two-phase process: first, assigning binary labels (anomalous or not) to each time window, and second, constructing a supervised learning model using these labels. Nakashima et al. [11] evaluated various deep learning models—including Multilayer Perceptrons, Convolutional Neural Networks, Gated Recurrent Units, and Long Short-Term Memory networks—to predict network performance in terms of aggregated throughput per time interval. While these studies leveraged `tstat` data with a focus on time-windowed analysis, our study distinguishes itself by aiming for connection-level prediction, providing a more granular understanding of network performance issues.

Addressing class imbalance is a critical challenge in machine learning, especially in network anomaly detection where anomalous events are rare compared to normal traffic. Data augmentation techniques have been widely employed to mitigate this issue by artificially increasing the representation of minority classes. Methods such as SMOTE and its variants have been effectively used in various domains to balance datasets [12], [13]. However, these methods may not fully capture the complex distributions inherent in structured network data. On the other hand, generative models, especially GAN, have shown significant potential in generating realistic data and have been adapted to various fields beyond image synthesis. Despite originating from image processing, GANs have been increasingly applied to structured data like tabular datasets, though challenges remain due to the inherent differences in data types [14]. Recent studies have begun to explore the use of GANs for oversampling in structured data domains. For example, Loey et al. [15] applied CGAN to COVID-19 detection in chest CT scan image. Bourou et al. [16] utilized TGAN and CTGAN to create synthetic Intrusion Detection Systems (IDS) datasets to enhance network security tools. Wang et al. [17] applied CTGAN to traffic data, creating a new model CTTGAN for detecting malicious network traffic.

III. Data Augmentation for Data Transfer Throughput Prediction

Standard oversampling methods like SMOTE and their variations generate synthetic samples along the line segment that joins minority class samples. Therefore, these approaches are based on local information, rather than on the overall minority class distribution. Since the separation between majority and minority class clusters is not often clear, noisy samples may be generated. To avoid this, various undersampling (filtering) variations have been tried such as SMOTE-ENN [18], and Borderline-SMOTE [19]. A direct approach to the data augmentation process would be the use of a generative model that captures the actual data distribution. In this section, we describe oversampling and generative methods for augmenting slow data transfer records observed much less in real scientific computing environments.

A. Oversampling Methods

1) SMOTE: SMOTE generates samples by interpolating between randomly selected minority samples and k nearest neighbors. This technique improves on the issue of overfitting as it generates a random sample that is not a duplicate of an existing minority sample unlike random oversampling. The algorithm is as follows: 1) randomly select a sample from minority samples, 2) use KNN to select k nearest neighbors around minority sample, 3) interpolate between these points to create a random sample. Then, the algorithm repeats this process until the desired number of synthetic samples.

$$\begin{aligned} x_{\text{smote}} = x_i + (\hat{x_i} - x_i) \times \delta, \\ \delta \in [0, 1], \\ i = 1, 2, ..., k \end{aligned} \tag{1}$$

where x_i is random sample in minority class, and $\hat{x_i}$ is random KNN

2) Borderline SMOTE: Naive SMOTE might generate too much noise. Hence, a Borderline SMOTE is designed to be an improvement to SMOTE by avoiding excessive noise generation. It generates synthetic samples near the decision boundary (borderline) between classes rather than randomly in the minority class space. The algorithms is as follows: 1) Find instances of the minority class that have a mix of majority class neighbors (boundary between majority and minority space), 2) among the borderline instances, identify those that are misclassified using a k-nearest neighbor classifier, then 3) generate synthetic samples by picking a random minority class neighbor, and interpolating between the instance and its neighbor.

3) SMOTE ENN: Start SMOTE process. After oversampling is finished, start Edited Nearest Neighbor undersampling. Find the KNN of the all samples. If the majority class of the sample's KNN and the sample's class is different, then the sample and its KNN are deleted from the dataset.

4) SMOTE Tomek-Link: Start SMOTE process. After oversampling is finished, start Tomek-Link undersampling. Tomek-Link is one of a modification from Condensed Nearest Neighbors, which is defined as selecting the pair of observation a and b such that they are each other's nearest neighbor and they do not belong to the same class. After choosing a random data from the majority class, if the random data's nearest neighbor is the data from the minority class (i.e. Tomek Link), then remove the Tomek Link.

5) ADASYN: ADASYN is similar to SMOTE but it generates different number of samples depending on an estimate of the local distribution of the class to be oversampled

i.e. ADASYN adapts to the underlying data distribution by generating synthetic samples in regions where the minority class is underrepresented. The algorithm is as follows: 1) For each sample in the minority class, calculate its k nearest neighbors, 2) for each minority class sample, compute a ratio (density distribution) that dictates how many synthetic samples will be generated for that sample, depending on how many majority samples are within the KNN, then 3) generate synthetic samples for each minority class sample based on the computed ratio:

$$r_i = \frac{\Delta_i}{K}, \quad i = 1, 2, ..., C_{\text{min}} \tag{2}$$

after normalizing it, where Δ_i is the number of samples in the K nearest neighbors of $x_i \in C_{\text{min}}$ that belong to the majority class.

B. Generative Methods

1) GAN: GAN is a deep-learning unsupervised model that is based on the idea of game theory, in which a generator G and a discriminator D are trying to outsmart each other. The objective of the generator is to confuse the discriminator. The objective of the discriminator is to distinguish the instances coming from the generator and the instances coming from the original dataset. The generative model G, defined as $G : Z \rightarrow X$ where Z is the noise space of arbitrary dimension that corresponds to a hyperparameter and X is the data space, aims to capture the data distribution. The discriminative model, defined as $D : X \rightarrow [0, 1]$, estimates the probability that a sample came from the data distribution rather than G. Then the objective of the game is given as:

$$\min_G \max_D V(D, G) = E_{x \sim p_{\text{data}}(x)}[\log D(x)] + E_{z \sim p_z(z)}[\log(1 - D(G(z))] \tag{3}$$

In the equation, $p_{\text{data}}(x)$ and $p_z(z)$ refer to the distribution of real data and distribution of input noise variable, respectively. The generator G aims to produce data $G(z)$ that the discriminator D cannot distinguish from real data x, while D strives to correctly differentiate between real and generated data. Through iterative training, G and D optimize this value function, leading G to generate increasingly realistic data over time.

2) CTGAN: Traditional GAN algorithms have shown impressive capabilities in learning original images and generating synthetic ones under various conditions. However, their application to structured data is limited due to challenges such as diverse tabular data types, non-Gaussian distributions, multimodal data, sparse matrices from one-hot encoding, and highly imbalanced categorical variables. To tackle these limitations, CTGAN was developed. This model integrates conditional-GAN and tabular-GAN algorithms, addressing the common GAN issue of not effectively learning sparse categories by including conditions in the learning process. CTGAN employs mode-specific normalization and training-by-sampling to manage the issues of multimodal and non-Gaussian data distributions. During training, it normalizes numerical data using the variational Gaussian mixture, and post-training, it converts generated data back to the original scale.

IV. Experimental Setting

The datasets are provided from `tstat`, a logging tool for network traffic flows which is deployed on data transfer nodes (DTN) in National Energy Research Scientific Computing Center (NERSC[1]). For this study, we use all the data collected in 2021 as the training set and all the data from 2022 as the testing set.

For all training datasets, we divided into two sub-classes: minority C_{min} and majority C_{maj}. For minority class C_{min}, we applied augmentation methods to balance the class imbalance ratio. Class imbalance ratio, as we define is simply the proportion of minority class to majority class, $IR = |C_{\text{min}}|/|C_{\text{maj}}|$. In Section V, we use this imbalance ratio to assess the predictive power augmented in the model.

In this study, our main augmentation methods are 1) oversampling methods and 2) generative methods. In oversampling, the idea is to interpolate between a sample $x_i \in C_{\text{min}}$ and its neighbors, which then may be followed by filtering and editing. SMOTE-ENN and SMOTE-Tomek Links are latter examples. First, we applied SMOTE to minority class of training datasets, and trained models on this new augmented datasets. Similarly, we applied Borderline-SMOTE, SMOTE-ENN, SMOTE-Tomek Links, and ADASYN to the minority samples and compared the results. SMOTE-Tomek Links and SMOTE-ENN are hybrid of oversampling and undersampling methods, where after augmentation is done, it needs to satisfy criterion to be validated into resulting training sets. Hence, the resulting dataset is smaller.

We chose `train2-test2` as a Baseline, then compared the results of different data augmentation methods. Hyperparameters for Random Forest and XGBoost was fine-tuned using GridSearchCV and `train2-test2` data sets as a baseline model.

A. Evaluation Metrics

To thoroughly evaluate our models, we utilize a combination of quantitative and qualitative metrics that not only measure performance but also provide insights into the data's structure and distribution. This approach ensures a comprehensive understanding of how well our models perform and how closely synthetic data resembles real data.

1) Quantitative Metrics: **F1-Score:** Given the imbalance in our dataset—with slow connections being a minority—we avoid relying on simple accuracy, which could be misleading. Instead, we use the F1-Score, the harmonic mean of Precision and Recall, to better assess the model's ability to correctly identify slow connections. This metric provides a balanced evaluation of performance, particularly in cases of class imbalance. We also employ 10-fold cross-validation to ensure that our results are robust and not due to random chance. **Kolmogorov–Smirnov (KS) Test:** To determine how closely our synthetic data matches the real data, we use the KS Test. This statistical test measures the maximum difference between the cumulative distributions of two datasets. By applying the KS Test, we can quantitatively assess the similarity in

[1]NERSC, https://www.nersc.gov/

TABLE I: Datasets Information with Imbalance Ratio (IR)

Data	Year	Description	IR (# Samples)
train1	2021	Random sampling	1:1000
train2	2021	Keep the entire slow transfers and randomly sample the same number of normal transfers	3000:3000
train3	2021	Keep half of slow transfers and randomly sample twice the size of the slow transfers for normal transfers	1500:3000
train4	2021	Keep the entire slow transfers and sample normal transfers twice the size of the slow transfers	3000:6000
train5	2021	Sample 1,000 slow transfers and sample 10,000 normal transfers	1000:10000
test2	2022	Keep the entire slow transfers and randomly sample the same number of normal transfers	1:1

TABLE II: F1-Scores of various data augmentation methods. `Train3` was used as the baseline dataset for augmentation to balance the class ratio. Then, the performance was tested using `Test2` dataset.

Dataset	Decision Tree	Random Forests	XGBoost
Train1	0.0	0.0	0.0
Train2	0.882	0.895	0.896
SMOTE	0.884	0.9	0.895
SMOTE Borderline	0.903	0.897	0.894
SMOTE Tomek	0.884	0.9	0.896
SMOTE ENN	0.892	0.9	0.893
ADASYN	0.892	0.895	0.896
GAN	0.902	0.871	0.869
CTGAN	0.878	0.896	0.898

distributions, which is crucial for validating the effectiveness of our data generation methods.

2) Qualitative Metrics: For a deeper understanding of the data and to interpret the results more intuitively, we incorporate visualization techniques that allow us to observe the structural relationships within the data. We utilize t-Distributed Stochastic Neighbor Embedding (**t-SNE**) and Uniform Manifold Approximation and Projection (**UMAP**) for dimensionality reduction and visualization. t-SNE is effective at preserving local structures in high-dimensional data, while UMAP aims to maintain both local and global structures. By comparing visualizations from both methods, we can qualitatively assess how well the synthetic data captures the underlying patterns of the real data.

V. Results

For traditional data augmentation methods, we selected `train2-test2` as a baseline model and compared the F1-Scores of each methods. The full datasets used for training are listed in Table I. Figure 1 displays the F1-scores of various augmentation methods as the imbalance ratio varies on the training set. As the imbalance ratio increases, the performance of the model decreases. Some models like GAN and SMOTE-Borderline show a sharp decrease in performance compared to CTGAN. We observed that CTGAN is robust in the sense that it continued to perform reasonably well on datasets with larger imbalance ratios (`1:10`). Hence, we chose CTGAN to assess the quality of generated samples.

To analyze the quality of synthetic samples, we compared the distribution of different classes (`0 = majority, 1 = minority, and 2 = synthetic minority`). For CTGAN, we only input minority samples from the training dataset and generated synthetic minority samples to balance the class ratio. We assessed the quality of the synthetically generated samples via two methods: 1) visualization and 2) KS-Test. After running 2000 epochs, we saw the convergence

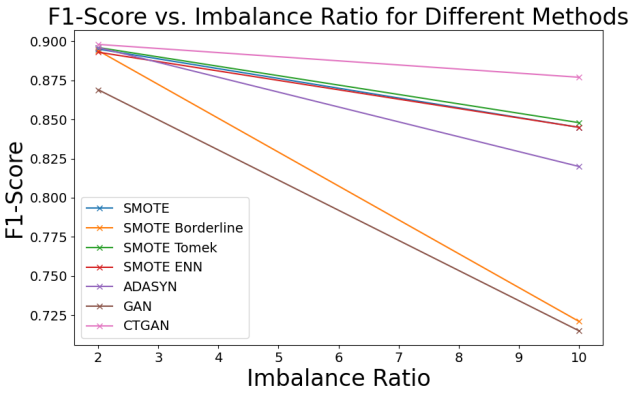

Fig. 1: Displaying the performance of data augmentation techniques across two different imbalance ratios 1:2 and 1:10. Each method is represented by a distinct line, with points marked at the respective imbalance ratios.

TABLE III: Top 6 features based KS Score. KS Score uses the inverse of Kolmogorov-Smirnov statistic to determine the distance of two distribution for original minority and synthetic minority samples. Higher score indicates higher resemblance.

Feature	KS Score
size	0.917
prev_tput	0.906
prev_size	0.892
prev_durat	0.869
prev_rtt_max	0.835
size_ratio	0.819

of the loss close to 0. The KS-Test scores result is shown in Table III. KS-Test compares two distributions and output similarity scores. For each features, the generated synthetic samples resemble true distribution of the feature if the score is high. We also compared the log distributions of the synthetic samples in Figure 2. We first grouped by class labels and overlaid it in one plot to show the contrasts of the distributions. As you can see, the density of the synthetic minority samples closely align with that of original minority samples, validating the data augmentation process.

Though the log histogram show close overlaps between real and synthetic minority samples, it does not fully capture joint behavior of the features. Looking at Figure 3, the synthetic samples may replicate individual feature distributions but fail to capture the complex relationships between features present in the original data. Using t-SNE, which considers the high-dimensional relationships between data points, effectively capturing the joint distribution of features, non-overlapping clusters may indicate that the synthetic data differs from the original data when considering the combination of features. Hence, this may explain the performance of the model not

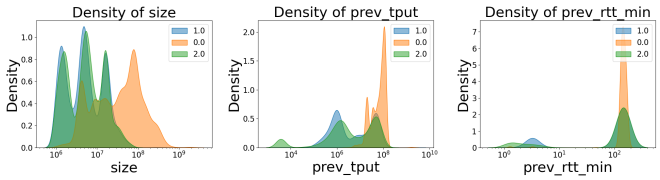

Fig. 2: Log histogram comparison of feature distributions for real and synthetic minority samples generated using CTGAN. The close overlap between the real and synthetic minority class distributions indicates a similarity in feature-level distribution.

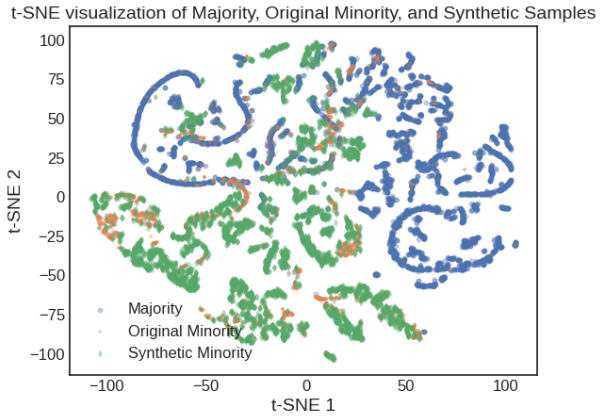

Fig. 3: t-SNE showing non-overlapping clusters of original minority and synthetic minority samples.

significantly being improved. Table II shows the performance of different data augmentation methods used.

VI. Conclusions

After comparing the results across different data augmentation techniques, we found that these methods do not consistently outperform stratified sampling. While the log histogram comparison of minority class distributions showed a close overlap between real and synthetic samples, it does not take into account of joint distributions. A more sophisticated technique (t-SNE) revealed that synthetic minority samples behave differently from real minority samples. This suggests that even advanced techniques, like CTGAN, which performed well in generating visually similar distributions, are still unable to fully replicate the true characteristics of the minority class. If the synthetic data does not accurately represent the joint feature relationships, the model may not benefit from the additional data, as it does not contribute to enhancing the model's ability to generalize to the minority class. This discrepancy between t-SNE and log histogram is likely the reason for the lack of significant performance improvement over simpler methods like stratified sampling. Even with the best-performing augmentation techniques, such as CTGAN, the synthetic data does not fully capture the behavior of real minority data, which is reflected in the comparable performance between data augmentation techniques and stratified sampling.

Acknowledgements

This work was supported by the Office of Advanced Scientific Computing Research, Office of Science, of the U.S. Department of Energy, under Contract No. DE-AC02-05CH11231. This work also used resources of the National Energy Research Scientific Computing Center (NERSC).

References

[1] Dong Lu, Yi Qiao, Peter A Dinda, and Fabián E Bustamante. Characterizing and predicting tcp throughput on the wide area network. In *25th IEEE International Conference on Distributed Computing Systems (ICDCS'05)*, pages 414–424, 2005.

[2] Chris Seiffert, Taghi M. Khoshgoftaar, Jason Van Hulse, and Amri Napolitano. Mining data with rare events: A case study. In *19th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2007)*, volume 2, pages 132–139, 2007.

[3] Robin Shao, Jinoh Kim, Alex Sim, and Kesheng Wu. Predicting slow network transfers in scientific computing. In *Fifth International Workshop on Systems and Network Telemetry and Analytics*, pages 13–20, 2022.

[4] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *J. Artif. Int. Res.*, 16(1):321–357, 2002.

[5] Haibo He, Yang Bai, Edwardo A. Garcia, and Shutao Li. Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In *IEEE International Joint Conference on Neural Networks*, pages 1322–1328, 2008.

[6] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *27th International Conference on Neural Information Processing Systems (NIPS'14)*, volume 2, page 2672–2680, 2014.

[7] Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Modeling tabular data using conditional gan. In *33rd International Conference on Neural Information Processing Systems*, 2019.

[8] Ran Ben Basat, Gil Einziger, Roy Friedman, and Yaron Kassner. Optimal elephant flow detection. In *IEEE Conference on Computer Communications*, pages 1–9, 2017.

[9] Anshuman Chhabra and Mariam Kiran. Classifying elephant and mice flows in high-speed scientific networks. *INDIS*, pages 1–8, 2017.

[10] Astha Syal, Alina Lazar, Jinoh Kim, Alex Sim, and Kesheng Wu. Automatic detection of network traffic anomalies and changes. In *ACM Workshop on Systems and Network Telemetry and Analytics*, pages 3–10, 2019.

[11] M Nakashima, A Sim, and J Kim. Evaluation of deep learning models for network performance prediction for scientific facilities. In *3rd International Workshop on Systems and Network Telemetry and Analytics*, pages 53–56, 2020.

[12] Dina Elreedy and Amir F. Atiya. A comprehensive analysis of synthetic minority oversampling technique (smote) for handling class imbalance. *Information Sciences*, 505:32–64, 2019.

[13] Alberto Fernández, Salvador Garcia, Francisco Herrera, and Nitesh V Chawla. Smote for learning from imbalanced data: progress and challenges, marking the 15-year anniversary. *Journal of artificial intelligence research*, 61:863–905, 2018.

[14] Zilong Zhao, Aditya Kunar, Robert Birke, and Lydia Y. Chen. Ctab-gan: Effective table data synthesizing. In *Proc. 13th Asian Conf. Mach. Learn.*, pages 97–112, 2021.

[15] Mohamed Loey, Gunasekaran Manogaran, and Nour Eldeen M. Khalifa. A deep transfer learning model with classical data augmentation and cgan to detect covid-19 from chest ct radiography digital images. *Neural Computing and Applications*, 2020.

[16] Stavroula Bourou, Andreas El Saer, Terpsichori-Helen Velivassaki, Artemis Voulkidis, and Theodore Zahariadis. A review of tabular data synthesis using gans on an ids dataset. *Information*, 12(9), 2021.

[17] Jiayu Wang, Xuehu Yan, Lintao Liu, Longlong Li, and Yongqiang Yu. Cttgan: Traffic data synthesizing scheme based on conditional gan. *Sensors*, 22(14), 2022.

[18] Gustavo E. A. P. A. Batista, Ronaldo C. Prati, and Maria Carolina Monard. A study of the behavior of several methods for balancing machine learning training data. *SIGKDD Explor. Newsl.*, 6(1):20–29, June 2004.

[19] 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 Advances in Intelligent Computing*, ICIC'05, page 878–887, 2005.