2023 2nd International Conference for Innovation in Technology (INOCON)
Bangalore, India. Mar 3-5, 2023

A Comparative Analysis of GAN and VAE based Synthetic Data Generators for High Dimensional, Imbalanced Tabular data

Kiran A
Reaserach scholar, Department of CSE,
SOET CMR Unieversity,
Bangalore 562149, Karnataka, India
kiranarnady@gmail.com

Dr. S Saravana Kumar
Professor, Department of CSE
SOET CMR Unieversity,
Bangalore 562149, Karnataka, India
saravanakumars81@gmail.com

2023 2nd International Conference for Innovation in Technology (INOCON) | 979-8-3503-2092-3/23/$31.00 ©2023 IEEE | DOI: 10.1109/INOCON57975.2023.10101315

***Abstract*— Synthetic data has emerged as an acceptable solution in machine learning that overcomes the constraints of data availability due to data privacy restrictions. Other major challenge with machine learning is dealing with imbalanced dataset. Several techniques exist to deal with the data imbalance, however, the problem continues to exist when using synthetic data generators dealing with highly imbalanced datasets. Generative Adversarial Network is already proven to be an excellent model to generate synthetic data, especially for high-dimensional datasets. There are other deep learning models that use Variational Autoencoders and Recurrent Neural Networks which are also being explored. To understand how these generators perform when presented situations dealing with highly imbalanced datasets, we experimentally evaluate two deep-learning synthetic data generators, one is based on Generative Adversarial Network (CTGAN) and the other is on Variational Auto Encoder (TVAE). We assess how each of these performs when presented with two datasets of distinct characteristics. The datasets used are high dimensional, highly imbalanced tabular data with one dataset having 19.3% minority class and the other having only 5.68% of minority class. Our test results find that TVAE fails to generate minority data when the minority class is very small in number.**

Keywords—CTGAN, TVAE, Synthetic Data Vault, ADASYN

I. INTRODUCTION

Data imbalance and fairness are one of the major challenges in machine learning classifier models. Fraud detection or patient diagnostic, detecting anomaly in finished products, or spam detection, the majority class is always skewed. Synthetic sampling techniques such as SMOTE, Borderline-SMOTE, and ADASYN have been able solutions to handle class imbalance. However, they can only be supplemental and work on simply oversampling the data by generating more data or understanding by removing the available data. They lack the capability to maintain the statistical characteristics of the dataset. Secondly, they cannot generate any dataset on their own from the available sample. Due to disclosure limitation, data availability has become a challenge, especially in the field of healthcare and financial sectors, which has led researchers to focus on synthetically generated data. To overcome the challenge of disclosure limitation, D Rubin[1] introduced synthetic data based on the theory of imputing missing information. T. Raghunathan, J.P Reiter and D.B Rubin [2] validated that it is not only a good substitute for real data but also covers the statistical disclosure limitation. The initial theory was focused on census and public dataset that were mostly tabular in nature and used statistical approach to generate artificial data which has a similar statistical distribution to resemble the original dataset. However, Organ imagery data in the field of healthcare, sensory data coming out of sensors in the manufacturing plant or information from an autonomous vehicle are all high dimensional which needed a different technique to generate synthetic dataset. Deep learning neural network generators namely Generative Adversarial Network and Variational AutoEncoder have emerged as an area of focus to handle the dimensionality as well the fairness challenges.

Generative adversarial network (GAN) introduced by **Ian Goodfellow** et al [3] in 2014, uses a neural network architecture with two modules, one known as a generator and the other known as the discriminator. As depicted in "Fig 1", the generator tries to produce data that is as close to the real data by using the random information from the noise source. The discriminator compares the data produced by the generator with real data and outputs the data loss information. This information is fed back to the generator. The generator uses feedback to improve the data quality. Both modules continuously learn together, in the process generate the output which is very close to the original data. This generative module is suitable for unsupervised learning.

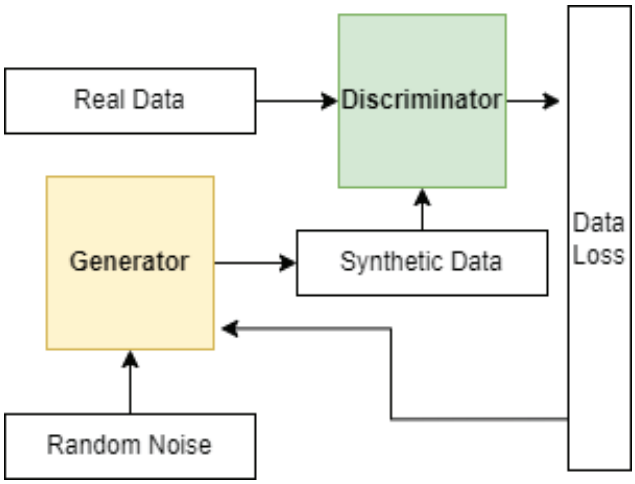

Fig 1 The modular architecture representing GAN

Variational autoencoder (VAE) introduced by Diederik P. Kingma and Max Welling [4] , also uses two modules, encoder, and decoder as shown in "Fig 2". The encoder takes the data from the input and encodes it into a low-dimensional latent space. The decoder picks the encoded data from latent

space and tries to decode it and attempts to re-create the input.

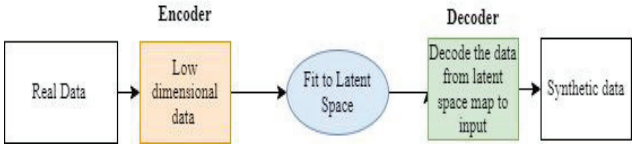

Fig 2 The representing the architecture of VAE

II. RELATED WORK

SMOTE [5] uses the technique of oversampling the minority data and under sampling the majority data to overcome the challenges with data imbalance. Borderline-SMOTE [6] introduced a technique to oversample only the borderline cases instead of oversampling the far outliers. Adaptive synthetic sampling ADASYN [7] use the weighted average to oversample the minority classes based on the difficulty in learning for classifiers. These models have the limitations as they do not use the full statistical information of the dataset, instead keep adding extra samples of the minority class [8] as highlighted by C. Zhang *et al.* VAE [9] has emerged as a better option to generate synthetic minority data for an imbalanced dataset in comparison with SMOTE, Borderline-SMOTE, and ADASYN. These models do have further limitation, they don't guarantee data privacy or statistical disclosure control. To address these challenges, synthetic data generators have emerged as the most preferred option. The most advanced field in the usage of the synthetic dataset is the medical field. Due to the non-availability of the right samples and the stringent data privacy constraints, scientists and researchers have adopted synthetic data generators. Available literature shows that GAN is more popular among researchers due to its capability of handling very high-dimensional data such as image [10]. GAN-based synthetic data generator used to generate synthetic data for augmenting the limited availability of the image for the classification of the liver lesions [11] has shown favorable results. James Jordan, Jinsung Yoon and Mihaela van der Schaar [12] introduced GAN coupled with Private Aggregation of Teacher Ensembles called PATE-GAN to generate tabular synthetic data with differential privacy. FairGAN [13] focuses on generating synthetic data on the tabular dataset (Adult income) that generates fair models without bias.

"Modeling Tabular Data using Conditional GAN" [14] by Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante and Kalyan Veeramachaneni introduce Conditional GAN (CTGAN), a novel technique to generate synthetic data with imbalanced data having continuous and discrete variables. The objective is to generate data with columns that have discrete imbalanced data. Another approach introduced by them is based on VAE, which is named TVAE. The experiments on the chosen dataset show favorable results. However, this work looks at how effectively CTGAN and VAE handle the varied types (categorical, continuous, and discrete) of data in a dataset and benchmark the performance of generators. It has not considered the minority dataset those are minuscule.

All the previous works focus more on data generation and privacy. Tests are carried out on just a single dataset. There is no significance evidence on how these generators perform when the data is highly imbalanced or how do they treat constants in the dataset, in such cases do they retain those features or ignore. Objective of our work is to compare CTGAN and TVAE-based generators for a very high dimensional tabular dataset. We use two distinct datasets from Kaggle. Wafer anomaly is the first dataset that has only 5.68% of the minority class and another is the malware detection dataset which has 19.3% of the minority class. We assess how GAN and VAE-based generators perform. We look at 1) the performance of two generators for the very high dimensional dataset, 2) what is the ratio of the minority class in the resulting synthetic data, 3) how well the data is reproduced and represented for all the features of the dataset and 4) how does the resulting dataset perform when used in machine learning classifier model.

III. EXPERIMENT

The experiments are conducted with the publicly available dataset from Kaggle using python 3.8.13 and Synthetic Data Vault [15] (SDV) 0.17.1 library. As the data sets are high dimensional, Azure ML studio on cloud is used for synthetic data generation. 4core CPU, 16 GB RAM compute instance is used for the data generation. In Table 1, we tabulate the key features of the two datasets used in our experiment.

A. Data

TABLE I. DATASET FROM KAGGLE USED IN ASSESSMENT OF CTGAN AND TVAE SYNTHETIC DATA GENERATORS

Dataset Name	*Features*	*Rows*	*Data types*	*Major Class*	*Minor class*	*Minor class %*
Detecting Anomalies in Wafer Manufacturing	1159	2519	categorical and continuous	2376	143	5.68%
Malware Executable Detection	532	374	Only categorical	301	72	19.3%

Data Source :
https://www.kaggle.com/datasets/arbazkhan971/anomaly-detection
https://www.kaggle.com/datasets/piyushrumao/malware-executable-detection

These datasets were chosen as a primary dataset for the experiment because of their high dimensionality, the combination of data types, and high-class imbalance, especially in the wafer anomaly dataset. Apart from wafer anomaly and malware dataset, the stroke dataset, cervical cancer dataset, and credit card fraud dataset were finally used for cross-verification of the findings. Their results are not tabulated as it was used only to cross verify and confirm the observations.

B. Experiemental setup

The wafer manufacturing anomaly data set has 3 separate comma-separated files. Train.csv has 1559 columns whereas Test.csv has 1558 columns and Sample_Submision.csv has only the target column. All 3 files were concatenated into one file and named it as anomaly.csv. No other data modification was required for this dataset. Target class of Malware dataset had to be encoded to numeric data. Beside these no other preprocessing was required for the malware

dataset. For Synthetic data vault (SDV) is available as a Python library. CTGAN and TVAE are the classes available in the SDV library that was used to generate the synthetic dataset. In the first pass, the sample output rows were restricted to the size of the original dataset or 2000 rows incase the original data had more number of rows. Data was generated using two different tuning parameters namely EPOCH and BATCH_SIZE as these were the most influencing parameters. EPOCH values were set to 10, 100, 200, and 300 in each pass. Similarly, BATCH_SIZE was set to 50, 100, 200, and 500. To assess the variation in the generated dataset, the hyperparameter values varied from low to high. To determine if the number of sampled rows made any difference to the minority class ratio, the number of sample rows in the generated data were increased to 10000. The hyperparameter that produced the best result was used in this step. The dataset was then used to train the machine learning (ML) classifier models. Classification results obtained by training the model using synthetic data was compared with results obtained from the real data. XGBoost, Support Vector, Logistic, KNN, Decision Tree, and Random Forest classifiers are used. Testing and comparison was carried out in following steps. 1) training and testing the machine learning models with real data 2) training and testing the machine learning model with synthetic data 3) training with synthetic data and testing on real data. The objective was to compare how close the results produced by synthetic data were in comparison with real data and not to find best classifier model that gave accurate results.

Finally, oversampling was done using ADASYN with the sampling_strategy set to "*auto*" to oversample all features except the majority class for both the real dataset as well as the synthetic dataset. The oversampled data was tested using the Decision Tree classifier model to assess how close are the results produced by CTGAN generated data and TVAE generated data in comparison with the real data

C. Results

Results of generated data for malware detection dataset and wafer anomaly dataset are tabulated in Table II to Table V. Tabulations show the percentage of minority class that were generated for different hyperparameter setting. The output rows were sampled at 2000 rows for wafer anomaly, and 374 rows for malware dataset which was the actual size of the original dataset. When the sampled rows are restricted to 2000, the percentage of minority class in the original dataset went slightly higher to 7.14% as against 5.68%. The row count in original anomaly dataset is 2519.

TABLE II. PERCENTAGE OF MINORITY CLASS AND COULMNS WITH JUST ONE CATEGORICAL DATA FOR SYNTHETICALLY GENERATED MALWARE DATASET USING CTGAN

CTGAN : Malware Detection				
EPOCH	***BATCH_SIZE***	***Generation time in mins***	***Minority class %***	***Columns with single categorical value***
10	50	1.2	19.57%	28
100	100	2.46667	27.61%	35
200	200	2.31667	27.07%	34
300	500	5.46667	17.15%	31
Real Data			19.30%	28

TABLE III. PERCENTAGE OF MINORITY CLASS AND COULMNS WITH JUST ONE CATEGORICAL DATA FOR SYNTHETICALLY GENERATED MALWARE DATASET USING TVAE

TVAE : Malware Detection				
EPOCH	***BATCH_SIZE***	***Generation time in mins***	***Minority class %***	***Columns with single categorical value***
10	50	0.54	0%	**429**
100	100	0.6667	23.59%	**367**
200	200	0.76667	18.49%	**399**
300	500	0.7	14.74%	**410**
Real Data			19.30%	28

TABLE IV. PERCENTAGE OF MINORITY CLASS AND COULMNS WITH JUST ONE CATEGORICAL DATA FOR SYNTHETICALLY GENERATED ANOMALY DATASET USING CTGAN

CTGAN : Wafer Anomaly				
EPOCH	***BATCH_SIZE***	***Generation time in mins***	***Minority class %***	***Columns with single categorical value***
10	50	10.73	10.80%	**131**
100	100	71.11667	4.95%	1
200	200	109.6	9.85%	0
300	500	142.76667	5.00%	1
Real Data			7.14%	22

TABLE V. PERCENTAGE OF MINORITY CLASS AND COULMNS WITH JUST ONE CATEGORICAL DATA FOR SYNTHETICALLY GENERATED ANOMALY DATASET USING TVAE

TVAE : Wafer Anomaly				
EPOCH	***BATCH_SIZE***	***Generation time in mins***	***Minority class %***	***Columns with single categorical value***
10	50	2.2	0%	**1548**
100	100	10.73333	0%	**1287**
200	200	16.3	0.35%	**1029**
300	500	19.45	0%	**1504**
Real Data			7.14%	22

Prediction score, Precision, Recall, F1 -Score, and Model score were the metrics captured. Table VI tabulates these metrics for malware dataset with real data and Table VII has the metrics captured for TVAE-based data generator. Hyperparameters are EPOCH = 200, BATCH_SIZE =200 and number of sampled rows = 10000. This was the best result that was produced by the TVAE generator which was equal to the real data. For the anomaly dataset, the TVAE generator produced only marginal minority data of 0.35% with EPOCH and BATCH_SIZE of 200 when the sampled rows were 2000. No other hyperparameter setting produced any minority class for the anomaly dataset even when the sampled output rows were increased to 10000. Hence there is no resultant comparison.

TABLE VI. CONFUSION MATRIX SCORE FOR THE REAL DATASET

Dataset : Malware Detection (Real)					
	Prediction score	***Precision***	***Recall***	***f1 score***	***Model Score***
XGBoost	1.0	1.0	1.0	1.0	0.9961
SVM	0.991	1.0	0.95	0.97	1.0
Logistic	0.991	1.0	0.95	0.97	1.0
KNNNeigbors	0.9953	1.0	0.74	0.85	0.954
Decision Tree	0.991	0.95	1/0	0.97	1.0
Random Forest	0.991	1.0	0.95	0.97	1.0

TABLE VII. CONFUSION MATRIX SCORE FOR SYNTHETICALLY GENERATED MALWARE DATASET WITH OUTPUT ROWS SAMPLED FOR 10000

Generator : TVAE **Dataset : Malware Detection** **EPOCH : 200** **BATCH_SIZE = 200** **SAMPLE_ROWS = 10000**					
	Prediction score	***Precision***	***Recall***	***f1 score***	***Model Score***
XGBoost	0.963	0.97	0.85	0.91	0.9678
SVM	0.9646	0.94	0.89	0.91	0.969
Logistic	0.9626	0.96	0.86	0.91	0.9651
KNNNeigbors	0.9636	0.96	0.86	0.91	0.971
Decision Tree	0.9643	0.94	0.89	0.91	0.9691
Random Forest	0.966	0.95	0.89	0.92	0.9698

"Fig 3" is a comparison of PR and ROC curve performance for synthetically generated malware dataset at EPOCH=300 and BATCH_SIZE=500 using CTGAN data generator. Similarly, "Fig 4" is the performance curve for data generated using TVAE generator. Source information and the dataset used to generate the synthetic data and the corresponding results are available in https://github.com/kiranPHD/SyntheticGenerators

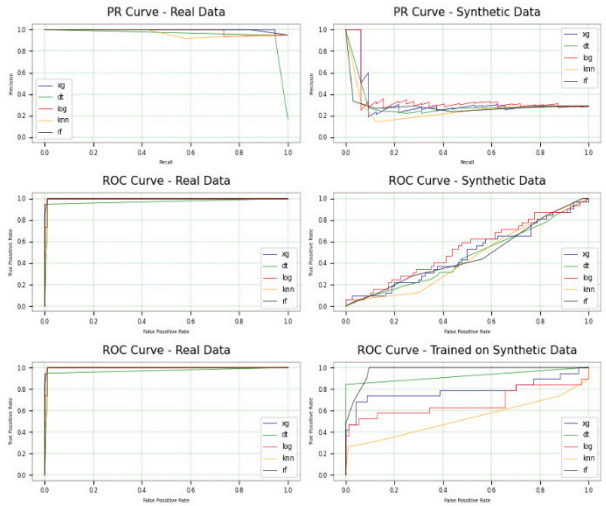

Fig 3 PR and ROC Curve comparison for malware dataset generated with CTGAN generator

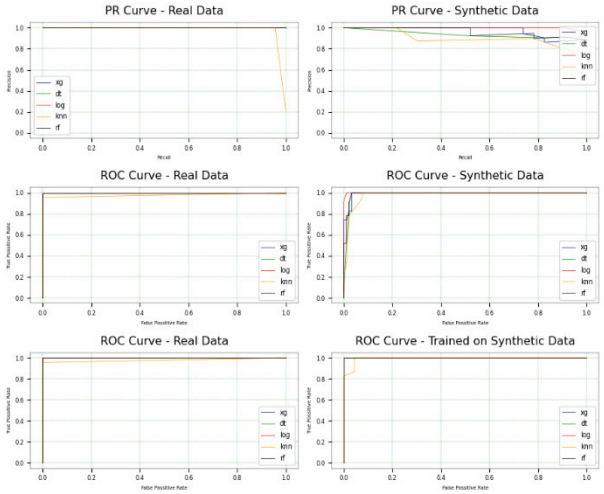

Fig 4 PR and ROC Curve comparison for malware dataset generated with TVAE generator

To assess how well these datasets can be used to train the machine learning classifier, both the real data set as well as synthetic dataset were fit into ADASYN with the sampling strategy parameter set to *"auto"*. The anomaly and malware dataset generated using EPOCH and BATCH_SIZE as 200 were opted for comparison as TVAE was able to generate 0.35% for anomaly dataset. PR and ROC curves in "Fig 5" show the comparison between the real data, data generated using CTGAN and data generated using TVAE for anomaly dataset. "Fig 6" provide the same view for malware dataset

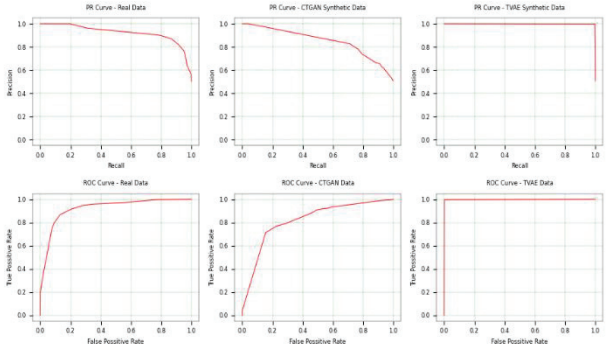

Fig 5 PR and ROC Curve comparison for anomaly dataset

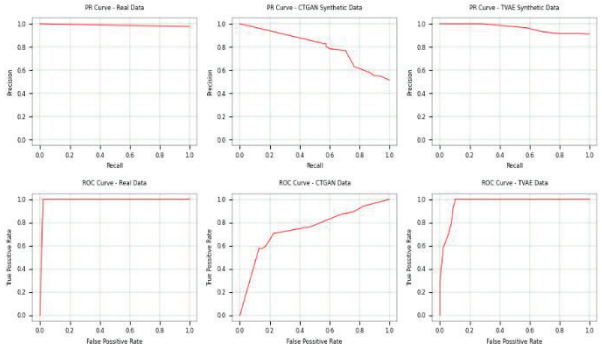

Fig 6 PR and ROC Curve comparison for malware dataset

D. Discussion

Looking at the objectives of this work, following observations are made, 1) TVAE is take less time for data generation as compared to CTGAN. 2) TVAE ignores the minority class completely when the minority percentage is very small as in the case of the anomaly dataset. This was consistent when TVAE was used on the other three datasets, i.e. Stroke Dataset, Cervical Cancer dataset, and Credit Card Fraud dataset that were used to confirm the observation. 3) When the column has categorical values, TVAE ignores the category and substitutes the column with a single value. This behavior was observed in several instances for different hyperparameters which are documented in Table II to Table V. 4) TVAE generates dataset comparable with real dataset, if there is enough minority sample available as observed in the case of the malware dataset. 5) CTGAN is more consistent and produces the minority classes even when the sample population is very minimum. 6) Classifier models trained on synthetic data and tested on real data did not show any improvement.

For the wafer anomaly dataset generated using CTGAN using hyperparameter setting of EPOCH=200 and BATCH_SIZE=200 and oversampled using ADASYN, Decision Tree classifier produced better output with **78%** accuracy score which is close to real wafer anomaly dataset at **85%** accuracy. Overfitting was observed in case of data generated using TVAE which had generated a small amount of minority class. For the malware dataset, TVAE performed much better with the accuracy score of **86.6% as** compared to CTGAN which **scored 74.1%**. These scores were observed for Random Forest classifier**.** When the sampling rows were increased to 10000, TVAE produced the best result. The waw synthetic data itself, produced an accuracy score of **96.66%** with Random Forest classifier. When the data was fit into ADASYN for oversampling and used on Decision Tree classifier model, it produced an accuracy score of **93%** as compared to **54%** for CTGAN. All the scores were considered against the minority class.

Based on the above observations, very noticeable advantage TVAE has is it is many times faster than CTGAN in data generation. CTGAN is the choice when 1) the minority class percentage is very small, 2) if the dataset has many categorical columns which are important and meaningful to be retained. If the minority class has a considerable presence, then TVAE is a good option. TVAE scored the best for malware detection dataset for hyperparameters (EPOCH=100, BATCH_SIZE=100 and EPOCH=200, BATCH_SIZE=200). It also scored better than CTGAN when the sampling records were increased. However, increasing the output sample rows had no effect on the generated dataset when the minority class was less in number.

E. Future work

The observation in this paper is based on the initial exploratory outcome. The experiments must be repeated with multiple datasets of varying minority class ratios and correlated continuous column variables, to assess the consistency of the outcome of these two synthetic data generators in discussion. As TVAE outscored CTGAN with malware dataset, it is to be further explored by using the technique introduced by C. Zhang *et al.* and Y. Zhao et.al to improve the sampling of minority class and then check if VAE based generator is able to generate better dataset when the minority class is real minority.

IV. CONCLUSION

Considering the availability of many options to generate synthetic data, we felt the need of assessing some of them for unique and specific requirements so that it helps academia's, researchers and also industries to choose the right generators based on their data needs. We picked up two generators, CTGAN and TVAE available in python libraries bundled as a part of synthetic data vault. Both are deep learning based synthetic data generators. We evaluate them against two specific imbalanced datasets. Our experiments find that TVAE would be a better choice when there is a fair amount of representation of minority class in the dataset and resources to generate the synthetic data are constrained. CTGAN was more reliable and consistent as compared to TVAE. TVAE generator completely ignores the minority class when their numbers are too small and introduces far too many constants into the categorical column.

REFERENCES

[1] D. B and Rubin, "Statistical Disclosure Limitation (SDL)," *Journal of Official Statistics*. pp. 461–468, 1993. doi: 10.1007/978-0-387-39940-9_3686.

[2] T. Raghunathan, "Multiple imputation for statistical disclosure limitation," *J. Off. Stat.*, vol. 19, no. 1, pp. 1–16, 2003, [Online]. Available: http://hbanaszak.mjr.uw.edu.pl/TempTxt/RaghunathanEtAl_2003_Multiple Imputation for Statistical Disclosure Limitation.pdf

[3] S. Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley and and Y. B. Ozair, Aaron Courville, "Generative adversarial nets," *Adv. Neural Inf. Process. Syst.*, vol. 27, pp. 2672–2680, 2014, [Online]. Available: https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf

[4] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *2nd Int. Conf. Learn. Represent. ICLR 2014 - Conf. Track Proc.*, no. Ml, pp. 1–14, 2014.

[5] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, 2002, doi: 10.1613/jair.953.

[6] H. Han, W. Y. Wang, and B. H. Mao, "Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning," *Lect. Notes Comput. Sci.*, vol. 3644, no. PART I, pp. 878–887, 2005, doi: 10.1007/11538059_91.

[7] S. He, H., Bai, Y., Garcia, E., & Li, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In IEEE International Joint Conference on Neural Networks, 2008," *IJCNN 2008.(IEEE World Congr. Comput. Intell. (pp. 1322– 1328)*, no. 3, pp. 1322–1328, 2008.

[8] C. Zhang *et al.*, *Over-Sampling Algorithm Based on VAE in Imbalanced Classification*, vol. 10967 LNCS. Springer International Publishing, 2018. doi: 10.1007/978-3-319-94295-7_23.

[9] Y. Zhao, K. Hao, X. song Tang, L. Chen, and B. Wei, "A conditional variational autoencoder based self-transferred algorithm for imbalanced classification," *Knowledge-Based Syst.*, vol. 218, p. 106756, 2021, doi: 10.1016/j.knosys.2021.106756.

[10] E. Choi, S. Biswal, B. Malin, J. Duke, W. F. Stewart, and J. Sun, "Generating Multi-label Discrete Patient Records using Generative Adversarial Networks," vol. 68, pp. 1–20, 2017, [Online]. Available: http://arxiv.org/abs/1703.06490

[11] M. Frid-Adar, E. Klang, M. Amitai, J. Goldberger, and H. Greenspan, "Synthetic data augmentation using GAN for improved liver lesion classification," *Proc. - Int. Symp. Biomed. Imaging*, vol. 2018-April, pp. 289–293, 2018, doi: 10.1109/ISBI.2018.8363576.

[12] J. Jordon, J. Yoon, and M. Van Der Schaar, "PATE-GaN: Generating synthetic data with differential privacy guarantees," *7th Int. Conf. Learn. Represent. ICLR 2019*, pp. 1–21, 2019.

[13] D. Xu, S. Yuan, L. Zhang, and X. Wu, "FairGAN: Fairness-aware Generative Adversarial Networks," *Proc. - 2018 IEEE Int. Conf. Big Data, Big Data 2018*, pp. 570–575, 2019, doi: 10.1109/BigData.2018.8622525.

[14] L. Xu, M. Skoularidou, A. Cuesta-Infante, and K. Veeramachaneni, "Modeling tabular data using conditional GAN," *Adv. Neural Inf. Process. Syst.*, vol. 32, no. NeurIPS, 2019.
[15] N. Patki, R. Wedge, and K. Veeramachaneni, "The synthetic data vault," *Proc. - 3rd IEEE Int. Conf. Data Sci. Adv. Anal. DSAA 2016*, pp. 399–410, 2016, doi: 10.1109/DSAA.2016.49.