A Universal Metric for Robust Evaluation of Synthetic Tabular Data

Vikram S Chundawat, Ayush K Tarun, Murari Mandal, Mukund Lahoti, and Pratik Narang, *Senior Member, IEEE*

***Abstract*—Synthetic tabular data generation becomes crucial when real data are limited, expensive to collect, or simply cannot be used due to privacy concerns. However, producing good quality synthetic data is challenging. Several probabilistic, statistical, generative adversarial networks and variational autoencoder-based approaches have been presented for synthetic tabular data generation. Once generated, evaluating the quality of the synthetic data is quite challenging. Some of the traditional metrics have been used in the literature, but there is lack of a common, robust, and single metric. This makes it difficult to properly compare the effectiveness of different synthetic tabular data generation methods. In this article, we propose a new universal metric, TabSynDex, for the robust evaluation of synthetic data. The proposed metric assesses the similarity of synthetic data with real data through different component scores, which evaluate the characteristics that are desirable for "high-quality" synthetic data. Being a single score metric and having an implicit bound, TabSynDex can also be used to observe and evaluate the training of neural network-based approaches. This would help in obtaining insights that was not possible earlier. We present several baseline models for comparative analysis of the proposed evaluation metric with existing generative models. We also give a comparative analysis between TabSynDex and existing synthetic tabular data evaluation metrics. This shows the effectiveness and universality of our metric over the existing metrics.**

***Impact Statement*—Tabular data synthesis plays an important role in ensuring privacy preservation in the data-driven systems. The data synthesis algorithms help in producing synthetic data that statistically resembles real data and can comply with the privacy protection regulations (such as European Union General Data Protection Regulation and California Consumer Privacy Act) due to its synthetic nature. Measuring the quality of synthetically generated tabular data is a challenging task. The effectiveness of such metrics will increase the confidence of both the user and the regulators about the privacy preservation of data. This article presents a single score universal metric for the evaluation of synthetic tabular data. The proposed TabSynDex metric in this article ensures fast computation of loss while training, consistency in the comparative evaluation of different generative methods. Moreover, it puts a more stringent criterion to measure the closeness of the synthetic data to real data. The findings in this article opens several future possibilities and highlights the need for further research on the development of better methods for tabular data synthesis.**

***Index Terms*—Evaluation metrics, generative adversarial networks (GANs), generative models, tabular data synthesis.**

I. Introduction

ARTIFICIAL data synthesis can help in alleviating several issues that are associated with the collection, obtaining consent, distribution, and scarcity of real data. Real data are difficult to obtain in many cases, such as personal data, medical data, and case-sensitive data. With the adoption of new data privacy regulations, such as European Union General Data Protection Regulation [1], California Consumer Privacy Act [2], the access of private data is increasingly becoming more cumbersome. Sometimes, it may not be possible at all to collect large amount of real data. For example, in the case of medical data for some rare disease, the available data would be very limited. The objective in synthetic data generation is to produce samples that closely represent the real data. This enables learning and inferences for real-world applications without an explicit mapping to the real data. Such generated data can be used to train machine learning (ML) models for applications where real data are not abundantly available due to any reason, such as stated previously.

A. Motivation

Measuring the quality of synthetically generated tabular data is a challenging task. Unlike image data, qualitative evaluation through visual inspection is infeasible for tabular data. Qualitative evaluation with the help of insights from experts is a highly inefficient process. While most generative adversarial networks (GANs)-based and variation autoencoder (VAE)-based tabular data generation methods [3], [4], [5], [6], [7] use the ML efficacy as one of the evaluation metrics, the remaining metrics are almost never consistent among different research articles. For example, Xu et al. [6] used a likelihood fitness metric, Zhao et al. [8] used Jensen–Shannon (JS) divergence and Wasserstein distance, and Goncalves et al. [9] used metrics, such as Kullback–Leibler (KL) divergence, pairwise correlation difference, log-cluster, and support coverage. The lack of a uniform metric makes it difficult to appropriately compare the effectiveness of different generative methods. Further, without

Manuscript received 17 July 2022; revised 28 October 2022 and 30 November 2022; accepted 10 December 2022. Date of publication 14 December 2022; date of current version 8 January 2024. This paper was recommended for publication by Associate Editor A. Etemad upon evaluation of the reviewers' comments. *(Vikram S Chundawat and Ayush K Tarun contributed equally to this work.) (Corresponding author: Murari Mandal.)*

Vikram S Chundawat, Ayush K Tarun, Mukund Lahoti, and Pratik Narang are with the Birla Institute of Technology and Science (BITS) Pilani, Pilani 333031, India (e-mail: vikram2000b@gmail.com; ayushtarun210@gmail.com; mukund.lahoti@pilani.bits-pilani.ac.in; pratik.narang@pilani.bits-pilani.ac.in).

Murari Mandal is with the School of Computer Engineering, Kalinga Institute of Industrial Technology (Deemed to be University), Bhubaneswar 751024, India (e-mail: murari.mandalfcs@kiit.ac.in).

Data is available online at https://github.com/vikram2000b/tabsyndex. This article has supplementary downloadable material available at https://doi.org/10.1109/TAI.2022.3229289, provided by the authors.

Digital Object Identifier 10.1109/TAI.2022.3229289

a bounded single score metric (such as accuracy or F1-score), it is not possible to support the training of deep learning models for analysis. This is one of the reasons why the existing deep learning-based approaches struggle to provide stable results or analysis. These methods are unable to offer clear answers to the questions given as follows: How many number of epochs are enough to converge? If the model is learning at all after a certain number of epochs? Brenninkmeijer and Hille [10] proposed an interpretable and bounded metric for evaluating synthetic data. They compare various statistics of real and fake data to generate different scores and then finally aggregate all of these scores to get a final single score metric. However, for certain scores they use correlation between the corresponding statistics of real and fake data to limit the range between [0, 1]. It means that even if the statistics corresponding to real and fake data for a particular component score are close but do not increase or decrease in the same order, the respective score will be very less. The consequence is that even if we evaluate the metrics for two different subsets of the same dataset, it shows very less similarity. This is contradictory as both the subsets belong to the same distribution. We have reported this observation through experiments in Section VI. In order to alleviate this problem, we propose a new metric in this article.

B. Our Contribution

We propose a new unified metric TabSynDex for the robust evaluation of synthetic tabular data. The TabSynDex is interpretable and bounded [0,1]. We show that our single score-based metric is sufficient to evaluate the generated synthetic data. The TabSynDex score encapsulates all the important characteristics essential to judge the quality of synthetic data. It includes comparison of basic statistics and ML efficacy with real data, fair representation of all categories of data, and also distinguishability from real data. We demonstrate its utility by evaluating the results of various synthetic tabular data generating techniques in the literature [3], [4], [5], [6]. Apart from that, we also show how this can be used (as a loss computing metric) in the training of deep learning models. While using TabSynDex to observe the training, we are also able to uncover some very interesting aspects of the training process of tabular data generating GANs.

II. Related Work

A. Traditional Methods for Tabular Data Synthesis

The earlier works have proposed numerous sampling methods to generate minority-class samples for imbalanced data. The ADASYN [11] method uses a weighted distribution of minority classes to adaptively generate samples. The SMOTE [12] method oversamples minority-class examples by k-nearest neighbors. Furthermore, the Borderline-SMOTE [13] method extend this work by oversampling the minority examples only near the borderlines. De La Calleja and Fuentes [14] used a randomized weighted distance scheme to generate minority-class data, and Yue et al. [15] used probabilistic fitting and k-means clustering to generate synthetic data for regression and classification. Garcia et al. [16] proposed variants of SMOTE that took into account proximity and the spatial distribution of the examples. For the generative modeling of relational databases, Synthetic Data Vault (SDV) [17] was proposed. The SDV uses a multivariate modeling approach to model the data. The EMERGE [18] (synthetic Electronic Medical Records Generator) method generate medical records to preserve the privacy of original records. A software package “Synthea” that simulates lifespans of synthetic patients for medical data generation was presented in [19]. Young et al. [20] used Bayesian networks to generate synthetic data, and subsequently, DataSynthesizer [21] used greedy Bayes algorithm to construct Bayesian networks that model correlated attributes.

B. GANs and VAEs for Tabular Data Synthesis

An important application of synthetic data generation is to enhance privacy preservation in the ML-based products. It can help in training an ML model even with limited set of real data. More specifically, limited data availability is quite a common problem in healthcare sector. The synthetic data generation methods have significant utility in healthcare-related applications. MedGAN [22] is one of the first method to employ GANs for tabular data synthesis. The aim of MedGAN is to produce realistic synthetic patient records to alleviate the privacy concerns of the patients regarding data sharing. It uses an encoder–decoder setup to preprocess categorical data and generates high-dimensional discrete variables. An improvement over this method is presented in medBGAN [23] that generates more realistic synthetic electronic health record (EHR) data. However, these methods [22], [23] are vulnerable to possible privacy exposure in the form of attribute and identity leaks. Another limitation is that they cannot handle the mixture of categorical and continuous columns. This limitation makes them useful only for a small subset of potential medical data types. Park et al. [3] proposed table-GAN based on deep convolutional GAN [4] to generate statistically similar data to the original ones. They show that the ML models trained with the synthetic data exhibit similar performance to the model trained on the real data. Xu and Veeramachaneni [5] proposed tabular generative adversarial networks (TGAN) to generate datasets with continuous and categorical columns. It uses a generator with a unidirectional recurrent neural network architecture with long short-term memory (LSTM) cells. The discriminator is a simple fully connected network. They also identify the problems neural networks suffer when working with non-Gaussian distributed input. Therefore, Xu and Veeramachaneni [5] used a Gaussian mixture model (GMM) with fixed number of modes on each individual continuous column to alleviate some of the issues. Xu et al. [6] made some improvements over the original TGAN [5] while also enabling conditional generation and proposed conditional tabular generative adversarial networks (CTGAN). They use a variational GMM instead of a GMM with a fixed number of modes. The generator in CTGAN uses a fully connected network instead of a sequential network used in TGAN [5]. A VAE [7] was adapted for tabular data synthesis in [6]. The generator was built by directly using the original data. More recently, Zhao et al. [8] developed a conditional table GAN to effectively model diverse data types, including a mix of continuous and categorical variables. Other notable works in the literature include [9], [10], [24], [25], [26], and [27].

C. Evaluation Metrics for Synthetic Tabular Data Generation Methods

Most of the existing works use the ML model efficacy and a variety of different metrics to evaluate the generated synthetic data. Xu et al. [6] used a likelihood fitness metric and ML efficacy. The likelihood fitness metric is only applicable for simulated data with a known probability distribution. For real-world datasets where the distribution is unknown, this metric is not suitable. Goncalves et al. [9] proposed a support coverage metric that measures whether the rarer classes have been properly represented in the synthetic data. They also use log cluster analysis, KL-divergence, and pairwise correlation distance. The KL-divergence is usually applied on an individual column. This does not account for the intercolumn statistic. Thus, it is incomplete without the correlation analysis-based metric. Snoke at al. [28] used basic statistical evaluation along with an expected value of propensity mean squared error (pmse) [29] that measures how much the synthetic data are differentiable from the real data. Zhao at al. [8] used JS-divergence and Wasserstein distance. Similar to KL-divergence, both these metrics do not account for the intercolumn statistics. Buczak et al. [18] used expert insights onto the data to asses the quality of the generated data. However, the domain expert inputs are not easily available and are usually costly. Chen et al. [25] evaluated the data generated by Synthea [19] using several clinical quality measures. Bayesian network generators [20] were evaluated using the performance of logistic regression on synthetic data. Recently, Brenninkmeijer and Hille [10] proposed a similarity score to capture the similarities between the different statistics of the real and synthesized data. They use the basic statistical properties (mean, median, and standard deviation) and other elements, such as correlation matrices, PCA, and ML efficacy, for computing the similarity score. Researchers have used different types of metrics in the literature. We notice that there is a lack of a uniform evaluation metric or a uniform single score metric that captures the intricacies and robustness of the generated synthetic data. Moreover, most of these metrics are unbounded and a comparative analysis between the existing works in the literature become difficult due to different metric adopted by different papers. We also observe that a unified metric to validate the tabular data generated from GANs and VAEs is missing in the literature. A single score metric that captures all the relevant information about the generated data would not only help in evaluating the output generated by the GANs and VAEs after training, but also in assessing the training of these GANs and VAEs.

III. TabSynDex

A. TabSynDex Metric

In order to measure the potency of the generated tabular data, we decided to focus on the quality of the data along with the statistical similarities. Therefore, in addition to the basic statistics, correlation score, and ML efficacy, we also compute the pmse and support coverage. All these scores combined can effectively represent the quality of generated data. Each of the metrics is bounded with a minimum value of 0 and a maximum of value of 1. In case a negative score is encountered, it is truncated to 0. This is done because we do not want a negative score to penalize good performance in other metrics. Similarly, the upper level for the scores is limited to value 1. The final value is an average of all the five scores. Different weights can be assigned to individual scores but equal weights are assigned for a vanilla study.

1) Basic Statistical Measures: We aggregate three basic statistics of data, i.e., *mean, median, and standard deviation*. We calculate the error for each column of the generated and real data using the following:

$$e = \frac{1}{n_c} \sum_{i=1}^{n_c} \left| \frac{R_i - F_i}{R_i} \right| \tag{1}$$

where n_c is the total number of samples in a column, and R_i and F_i are the statistical measures for a column of the real data and generated data, respectively. The relative errors are clipped to keep it at the maximum value of 1. This is done so that poor performance on a single column does not lead to an overall negative score. We also do not expect a good synthetic data generation technique to have an error of more than 100% on basic statistics. The scores for a single metric, such as mean, is calculated as

$$s_{\text{mean}} = 1 - \frac{e_1^{\text{mean}} + e_2^{\text{mean}} + \cdots .. + e_n^{\text{mean}}}{n} \tag{2}$$

where n is the number of columns and e_i^{mean} is the relative mean error for the ith column. The scores for median and standard deviation are calculated in a similar manner. Finally, the individual scores are aggregated as

$$S_{\text{basic}} = \frac{s_{\text{mean}} + s_{\text{median}} + s_{\text{std\_dev}}}{3}. \tag{3}$$

2) Log-Transformed Correlation S_{corr_l}: This statistical measure considers the correlation between the columns of each dataset. We deal with the categorical columns in the same way as in [10], i.e., we use Pearson's correlation for association between two continuous columns, correlation ratio [30] for association between categorical and continuous columns, and *Theil's U* [31] for association between two categorical columns. This approach produces two matrices consisting of n^2 values in total, where n is the number of columns in the datasets. These matrices are used to calculate the correlation scores. Similar to this, we calculate the relative errors for each entry in the correlation matrix of real data, clip it to a maximum value of 1, and subtract the average value from 1 to obtain the rescaled score.

Why log transformation?: One of the issues with directly calculating relative error is that the correlation matrix of real data can have very small values. These values may even be of the order 10^{-6}, 10^{-5}, and 10^{-4}, whereas the corresponding entries in correlation matrix for generated data can be of an order bigger or smaller. This may not necessarily mean the generated data are bad, as both the correlations are nearly 0. But the relative error can result in a higher value. Therefore, the correlation score remains bad in almost all cases. To address this issue, we use the log of the magnitudes of the entries of the association matrices while keeping the signs intact, given as follows:

$$S_{\text{corr}_l} = 1 - \frac{1}{n^2 - n}\sum_{i=1}^{n}\sum_{\substack{j=1\\ j\neq i}}^{n}\left|\frac{sln(r_{ij}) - sln(f_{ij})}{sln(r_{ij})}\right| \tag{4}$$

where $sln(x) = \text{sign}(x)ln(|x|)$, r_{ij}, and f_{ij} are corresponding entries of the association matrices of real and generated data. The $\text{sign}(x)$ equals 1 if x is positive and -1 if x is negative.

3) S_{pmse} Index: We introduce the S_{pmse} index to more robustly discriminate between the real and synthetic data. The S_{pmse} index is built upon the *pmse*, which is commonly used in the literature to analyze the difference between the distributions of real data and synthetic data. If the real data contain N_{real} samples and synthetic data contains N_{syn} samples, then the combined data contains $N_{\text{syn}} \cup N_{\text{real}}$ samples where the label Y denotes real data with 0 and synthetic data with 1. We train a logistic regression model on the combined data for classification. In our analysis, we only use the input variables and not their polynomial combinations as in [28]. Because that would make computing the similarity score highly expensive when the number of columns are large. Thus, it would render the metric unsuitable for the purpose of training where the metric needs to be quickly calculated at each epoch. Furthermore, we use the regression model to predict the probabilities $\hat{p}_i$ of each sample being synthetic or real. The *pmse* score is calculated as follows:

$$\text{pmse} = \frac{1}{N}\sum_{i=1}^{N}(\hat{p}_i - c)^2 \tag{5}$$

where N is the total number of samples after combining the synthetic and real samples, i.e., $N = N_{\text{real}} + N_{\text{syn}}$, $\hat{p}_i$ is the predicted probability of the ith sample being synthetic, and c is the proportion of the samples being synthetic, i.e., $c = N_{\text{syn}/}(N_{\text{real}} + N_{\text{syn}})$.

Snoke et al. [28] introduced pmse_0, the expected value of pmse when the synthetic data are indistinguishable from the real data, given as follows:

$$E(\text{pmse}_0) = \frac{(k-1)(1-c)^2 c}{N} \tag{6}$$

where k is the number of parameters in the logistic regression model, including the bias. The ratio between the pmse and the expected pmse_0 is calculated to evaluate the generated data in [28]. The more this ratio is close to 1, the fake data are more indistinguishable from the real data.

We further improve this metric by introducing a factor $\alpha^{-|1-\text{ratio}|}$ to this score. This also standardizes the value of the ratio within the range of [0,1]. The value of α can be chosen according to the application and after deciding what pmse value would be appropriate. For example, we observed that $\alpha = 1.2$ works quite well when the synthetic data are in the range of [0,5]. The S_{pmse} index is defined as

$$S_{\text{pmse}} = \alpha^{|1-\text{ratio}|} \tag{7}$$

where $\text{ratio} = \text{pmse}/E(\text{pmse}_0)$.

4) Regularized Support Coverage S_{cr}: We propose a support coverage [9]-based metric for histogram comparison between the real and synthetic data. The support coverage measures the amount of variables support (available in the real data) covered in the synthetic data. We penalize the overall support coverage score the most if the rarer categories of a variable are not represented well. The metric is calculated as follows for a single variable or column c

$$s_{cr} = \frac{1}{n_{\text{cat}}}\sum_{i=1}^{n_{\text{cat}}}\frac{n_i^f}{n_i^r} * \text{scaling factor} \tag{8}$$

where n_{cat} is the number of categories for a variable, n_i^f is the number of samples in that category in fake data, n_i^r is the corresponding number of samples in real data, and the scaling factor is $N_{\text{real}}/N_{\text{fake}}$, i.e., the ratio of the number of samples in real data and the number of samples in fake data. For a categorical column, n_{cat} is the number of categories. For continuous data, n_{cat} is the number of bins of equal range that we divide the data into for this analysis. In our analysis, we use 20 bins or 5% range per bin. This ensures the bins are not so large as to disregard the distribution of values in the bins themselves, and not so small that there exist categories with no samples in them. Thus, we keep a balance so that this score fairly compares the distribution of values in the real and synthetic/fake columns.

This score for a single variable is affected the most if n_i^r is small. It means the overall score may become much greater than 1 if the corresponding n_i^f is even slightly bigger, and much smaller if it is even slightly lesser. We want to see a smaller score if a rarer category is not well represented, but not disregard the representation of other categories if a single rare category is too well represented. So we limit this score to a maximum of β, which can be decided based on the desired representation of categories. In our analysis, we have used $\beta = 2$. The final score is an average of the scores for all the columns. The score of each column is limited to a maximum of 1. This is done so that a good coverage of representation of one column does not make the overall score disregard the bad coverage of other columns.

$$S_{cr} = \frac{s_{cr1} + s_{cr2} + \cdots + s_{crn}}{n} \tag{9}$$

where s_{ci} is the score for the ith column, and n is the number of columns.

5) ML Efficacy-Based Metric: We adopt the metric presented in [10], i.e., we train a common model on both real and synthetic data. The trained model is evaluated on the real and synthetic test set. We calculate the relative error with respect to the RMSE score of a regression model trained on real data. Similarly, the F1-score of the classifier trained on real data is computed. It is to be noted that all the scores are scaled within [0,1]. The average of different errors is subtracted by 1, given as follows:

$$S_{ml} = 1 - \frac{e_{m1} + e_{m2} + e_{m3} + e_{m4}}{4} \tag{10}$$

where e_{mi} is the error for the ith model, and we use four models namely, logistic regression, random forest, decision tree, and multilayer perceptron, as given in [10]. The models are trained to learn the tasks of classification on target columns and regression on target columns with random forests, lasso regression, ridge regression, and ElasticNet.

TABLE I
COMPARISON BETWEEN THE PROPOSED TABSYNDEX AND THE EXISTING METRICS FOR SYNTHETIC TABULAR DATA ANALYSIS

	Inter-column Correlation	Distribution Comparison	Rare class Coverage	Suitability to ML Applications	Bounded	Absolute
KL/JS-Divergence [9], [24]	×	✓	×	×	×	×
Correlation Difference [24]	✓	×	×	×	✓	✓
Wasserstein Distance [24]	×	✓	×	×	×	×
ML Performance [5], [6]	×	×	×	✓	×	×
pMSE [28]	×	✓	×	×	×	×
TabSynDex	✓	✓	✓	✓	✓	✓

The desirable attributes for each metric are listed in the first row.

All the abovementioned five metrics evaluate different aspects of similarity between the real and synthetic data. We further combine all these metrics and propose a unified metric TabSynDex. The proposed TabSynDex score is calculated as follows:

$$\text{TabSynDex} = \frac{S_{\text{basic}} + S_{\text{corrl}} + S_{\text{pmse}} + S_{cr} + S_{ml}}{5}. \quad (11)$$

B. Comparison With the Existing Metrics

The aim of a synthetic tabular data evaluation metric is to produce a similarity value representing the closeness of synthetic data to the real data. In this regard, Brenninkmeijer and Hille [10] combined several statistical and ML metrics. Apart from the resulting scalar value, some additional metrics are reported to highlight other characteristics of the data. It combines all the component scores to get a single metric that can be used for benchmarking the GAN architectures for data synthesis. But the methodology used to calculate this score has following shortcomings.

1) Correlation is used at multiple places to bound the scores between 0 and 1. For example, the correlation between the vectors containing the statistics of the real data and synthetic data is computed as part of the metric. We opine that nearness to the *original data* points should have been accounted for in this metric.
2) Similarly, in the ML-based metric, the nearness of the synthetic data to the real data is not considered. The metric first mirrors the column association between the real-synthetic data, and then calculates the correlation between them. This is a flawed strategy as any two random vectors would yield high correlation if they are sorted.

Table I tabulates the various desirable attributes (first row) that an ideal metric should consider. For example, intercolumn correlation, distribution comparison, rare-class coverage, implicit bound, and suitability to ML applications are some desired attributes. We compare the existing metrics in the literature (first column) with the proposed TabSynDex. It is clear that the proposed TabSynDex is the only metric that covers all the listed attributes in Table I.

Merits of the proposed TabSynDex: The proposed TabSynDex metric has several merits that make it a better choice for evaluation.

1) It can be measured on an absolute scale. Unlike the metrics, such as RMSE scores of ML models, likelihood fitness, Wasserstein Distance, and pmse, which can only be compared for two models. The TabSynDex is restricted to the range [0, 1] unlike the metrics abovementioned that restrict their usage for judging a model alone.
2) It gives a single score cumulative metric encapsulating the various desired qualities in synthetic data. This helps in benchmarking the results across different generative models. Currently, a variety of metrics are used selectively, which makes it difficult to compare the existing methods for data synthesis.
3) The proposed metric allows fast calculation after every epoch, allowing us to gain useful insights during training. For example, in CTGAN [6], the authors train each model for 300 epochs. However, our study shows that CTGAN does not learn at all during training. The model gives similar results to that of the GMM, which was used to preprocess the data. No training and 100 epochs training give almost identical results. A metric, such as the proposed TabSynDex, would make it possible to notice such behavior during training. We have shown this in our experiments in Section VI.
4) The analysis of each component metric of TabSynDex gives insights into what is happening with the model and what might happen. For example, in Section VI, a decreasing histogram comparison score can hint at an upcoming mode collapse. Similarly, a straight line histogram comparison along with 0 correlation score denotes a mode collapse.

C. Models for Tabular Data Generation

We use several GANs for tabular data synthesis. We conduct experiments on tabular VAE (TVAE) [6], [7], CTGAN [6], and TGAN [5] with GMM preprocessing. We also execute experiments on WGAN-GP [32], [33] in order to observe the results without GMM preprocessing. Furthermore, we introduce certain variations in the original TGAN and train the models to see if any minute change in the model makes any difference to the, otherwise, not so encouraging training graphs and performance. The following are the variations of TGAN that we implement in this article.

TGAN with GRU: We replaced the LSTM with gated recurrent unit (GRU) [34] in the generator. Rest of the specifics remain same to that of TGAN.

TGAN with a bidirectional generator: In the original TGAN, each column receives the context of the earlier columns for generation. We introduce a bidirectional generator [35] in which each output is generated after combining the respective column features (dense layers) from both the forward pass and backward pass models.

TGAN with Variational GMM (VGMM) preprocessing: We replace GMM with VGMM, as given in [6].

TGAN with bidirectional generator and VGMM preprocessing: We replace the GMM with VGMM in the TGAN model with a bidirectional generator.

It is to be noted that the detailed analysis of these four variants of tabular GANs and corresponding results also serve as a natural ablation study of our work.

IV. DATASETS AND EXPERIMENT SETTING

We analyze the performance of TGAN, CTGAN, WGAN-GP, TVAE, and several modifications of TGAN on datasets of various sizes and nature. We then evaluate their performance in various scenarios with five constituent metrics of the proposed TabSynDex for tabular data synthesis.

A. Datasets

We compare the performance and training progress of all the models on the following datasets:

1) concrete compressive strength data [36]—1030 rows, nine columns;
2) wine quality data [37]—4898 rows, 13 columns (we use only the white wine data);
3) online news popularity dataset [38]—39 797 rows, 61 columns;
4) combined cycle power plant dataset [39]—9568 rows, four columns.

We have used a variety of datasets with various number of data points and columns. The variety in these datasets help us analyze the results in different scenarios, given as follows:

1) which model performs well with extremely less amounts of data;
2) which model performs best when a decent number of samples are available;
3) does the number of samples required for decent performance depends on the number of columns? (as with increasing number of columns, higher number of relations needs to be covered by the model).

B. Sanity Check of the Metric for Tabular Data Synthetic

We first perform a *sanity check* on the evaluation metric by observing its behavior on two different subsets of a real dataset. The metric should ideally give a high similarity score between these two subsets. We shuffle and partition the original dataset into two equal parts, using one part as the real dataset, and treating the other as the synthesized dataset. We sample different proportions of data from the synthetic dataset to observe the effect of *sample size* on the overall score. We hypothesize that a small set of samples will show less similarity with the real data as it covers a small variety of samples. Besides, it will have high degree of randomness that would lead to less score on all components of the metric.

C. Experiment Settings

We use the GAN and VAE-based models as mentioned in Section IV-C to generate synthetic data. We evaluate their training progression to observe the evolution in the quality of data generated over the number of epochs. We also notice the highest similarity score obtained during training. The obtained similarity score is compared with the score obtained by a Bayesian-based data synthesis method DataSynthesizer [21].

For CTGAN and TGAN, the default hyperparameters are adopted. These hyperparameters are maintained across different modifications of these models throughout our experiments. In WGAN-GP, the min–max scaling is used for noncategorical columns and one hot encoding is used for categorical columns. For TVAE, we follow the hyperparameters as in [6].

TABLE II
EXPERIMENT FOR SANITY CHECK OF THE TABSYNDEX METRIC FOR TABULAR DATA SYNTHESIS EVALUATION

Data	Number of rows	TabSynDex Score			
		% of real data treated as synthetic			
		10%	25%	50%	100%
Concrete [36]	1030	0.768	0.869	0.914	0.894
News Popularity [38]	39644	0.891	0.916	0.901	0.898
Wine Quality [37]	4898	0.867	0.911	0.925	0.938
Power Plant [39]	9568	0.946	0.961	0.981	0.969

The real dataset is divided into different subsets to check the similarity between them using TabSynDex. *A higher TabSynDex score is better.*

Almost all the models either show no sign of improvement or converge by 50 epochs on each dataset. Therefore, we present the results for 50 epochs. For DataSynthesizer, we use the *correlated_attribute* mode and keep the degree of the Bayesian network as 2.

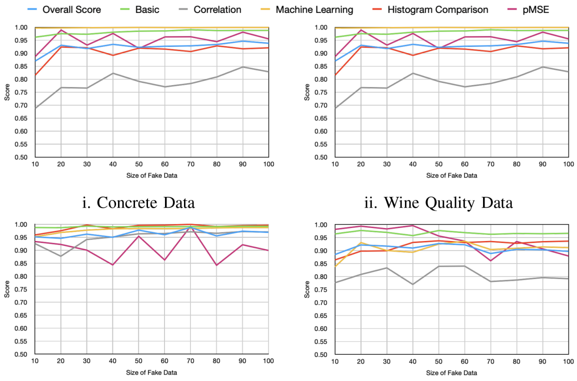
V. EXPERIMENTAL RESULTS AND ANALYSIS

A. Cross-Validation Analysis

In the first experiment, we perform a sanity check on the proposed TabSynDex metric. The idea is that if two nonoverlapping subsets of a real dataset are compared, then the TabSynDex metric should be close to 1 (the highest score). Therefore, we take different proportions of the real dataset and compute the TabSynDex between them to check the correctness of the proposed metric. We create two subsets of the real dataset: *subset1* and *subset2*. The TabSynDex score between these subsets will be computed. Table II tabulates the overall TabSynDex score when 10%, 25%, 50%, and 100% of samples are used from *subset2*. We sample various proportions of data from *subset2* to evaluate the effect of data size on the similarity metric. We observe that using a small set of data leads to poor TabSynDex score. Ideally, all the different proportion of samples (10%, 25%, 50%, and 100%) should give approximately the same similarity score as they all belong to the real dataset. However, in practice, using a small proportion of data leads to lower similarity score due to their inherent randomness. Thus, the amount of data present in *subset2* affects the similarity metric. Having more number of data points will reduce the randomness and lead to a better understanding about the characteristics of the data. In Fig. 1, the TabSynDex and its constituent submetrics are plotted for: concrete data, wine quality data, electrical power plant data, and news data. This further gives a more comprehensive view of how the number of samples affects the similarity score and its components.

B. Quantitative Analysis With Different Generative Methods

A good synthetic data generator should produce data that are similar to the data present in *subset2*, as discussed in Section VI-A. The *subset1* is used by the model to learn the distribution of the data. After fine-tuning the model, synthetic data are generated. We now use *subset2* to compare with the synthetic data. We show the results (on all four datasets) of the CTGAN [6], TGAN [5], and TGAN with GRU [34] in Fig. 2. The results of bidirectional TGAN, TGAN with VGMM



i. Concrete Data

ii. Wine Quality Data

iii. Electrical Power Plant Data

iv. News Data

Fig. 1. Cross-validation analysis with the proposed TabSynDex metric. The real dataset is divided into two sets. The synthetic data are generated by using one set of data. The generated synthetic data are then compared with the other set of real data. (i) Concrete data. (ii) Wine quality data. (iii) Electrical power plant data. (iv) News data.

preprocessing [6], bidirectional TGAN with VGMM preprocessing [6], and WGAN-GP [32], [33] are depicted in Fig. 3.

1) CTGAN: Fig. 2(a) shows the results of CTGAN [6] on four datasets. The datasets having less number of samples (i.e., *concrete data* and *wine quality data*) cause more variation. The comparison between the small test subsets leads to lower similarity score with lots of fluctuations at different epochs. The CTGAN trained of the larger datasets (i.e., *power plant data* and *news data*) give around 90% similarity. All the component scores are quite good, hinting that good quality synthetic data will score well as the distribution is very similar to real data. Furthermore, all the component scores of TabSynDex remain almost constant with respect to the increasing epochs. This leads to a constant overall score on all the four datasets. We note that the S_{pmse} score stays constant at 0 and never changes. In fact S_{pmse} never rises above 0 for any model or dataset (except for one case that we will discuss later). This means the data synthesized by all the models can easily be differentiated from real data with a linear regression model. Essentially, all the existing synthetic data generation methods fail to produce data that is nondifferentiable from real data. Our study shows that synthesizing completely nondifferentiable real data remains an unsolved problem and a challenging future research work.

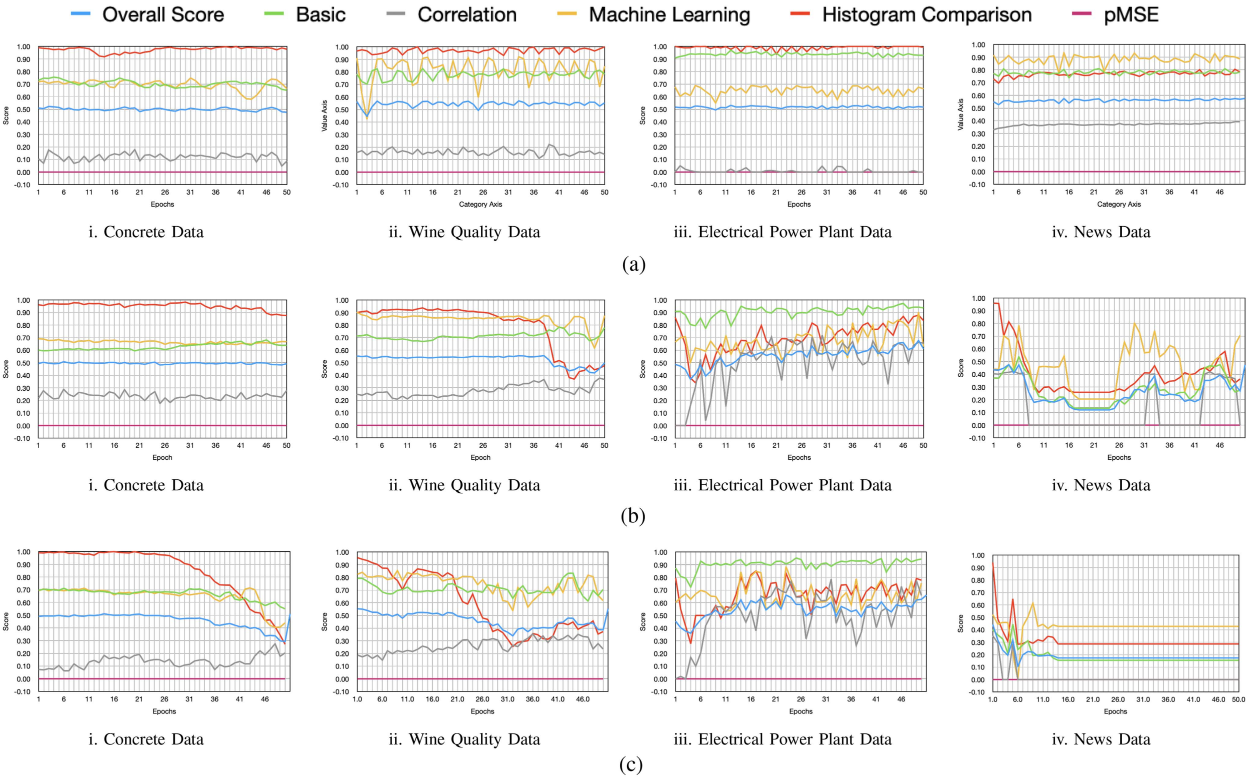
2) TGAN: The results using TGAN [5] method is shown in Fig. 2(b). On *concrete data*, the results are similar to CTGAN, i.e., the scores are constant and training does not have its intended effect of improving the quality of data generated by the model. On *wine quality data*, the scores are constant till the 37th epoch after that there is sharp decline in histogram comparison score S_{cr}. This could be due to a decrease in the diversity of data produced hinting at a potential mode collapse. We verify the effect of training further with more epochs later in this section. TGAN on *electrical power plant data* is the first instance where training has the intended effect of increasing the quality of generated data. This is visible through all the scores (except S_{pmse}). On *news data*, the correlation score S_{corr_l} goes zero for some intervals of epochs, which is mostly because of mode collapse.

3) TGAN With GRU: We modify the TGAN by replacing the LSTM cells with GRU [34] cells. Fig. 2(c) represents the results for this model. All the component scores decline with respect to TGAN. Especially, the histogram comparison score S_{cr} degrades substantially with increasing epochs. As stated earlier, this may hint at an upcoming mode collapse on *concrete* and *wine quality data*. This is evident from the graph on *news data* where the decrease in all the scores (except S_{pmse}), especially histogram comparison score S_{cr}, is followed by a mode collapse on the 6th epoch. The model obtains slightly better results on *power plant data* in comparison to the original TGAN.

4) Bidirectional TGAN: We augment another variant of TGAN with bidirectional generator [35]. Fig. 3(a) shows that a bidirectional TGAN performs better than the original TGAN on *wine quality* and *news data*. In *wine quality data*, the component scores gradually increase till the last of the 50 epochs. All the component scores are quite stable. The performance is even better on *news data*. All the component scores (except S_{pmse}) improved by a good margin in comparison to TGAN. Some of the scores are still unstable with correlation score S_{corr_l} touching zero quite a few times in the first 30 epochs. This means the model might keep collapsing and recover every now and then. The performance is almost same on the *power plant data*, and worse on the *concrete data*. On *concrete data*, the scores are stable, but they are less than those obtained with original TGAN.

5) TGAN With VGMM Preprocessing: We further study the TGAN by replacing the preprocessing method GMM with VGMM [6]. Fig. 3(b) shows the results of TGAN with VGMM preprocessing. In all four datasets, more epochs of training leads to a decrease in the quality of generated data. In the *news dataset*, we observe how the scores decrease and the model collapse at the 10th epoch. The results are also worse than that of CTGAN. This shows that adding VGMM had no positive impact on the model. This is contrary to what is shown in [6]. Xu et al. [6] claimed that introduction of VGMM leads to improved performance with traditional metrics. However, evaluating with the proposed TabSynDex gives opposite conclusion, i.e., using VGMM degrades the generated data quality. As already established through sanity check with real data comparison (in Sections V-B and VI-A), the proposed TabSynDex gives a better measurement of synthetic data quality.

6) Bidirectional TGAN With VGMM Preprocessing: We further combine VGMM preprocessing with bidirectional TGAN [6]. Fig. 3(c) shows the obtained results. The results are mixed again. The performance when compared with the original TGAN is worse on the *concrete data*. If the small dip at the end of TGAN is ignored, the histogram comparison score S_{cr} on the *wine quality data* is also lower than TGAN. Unlike in the original TGAN graph, the metrics for the *power plant data* shows no sign of any improvement with increasing epochs. Only for the *news dataset*, this model performs better than the original TGAN. But it is no better than that of bidirectional TGAN. The performances are marginally less in *wine quality* and *power plant data*. This is again contrary to the analysis presented in [6], which shows that VGMM improves the synthetic data quality. The performance of CTGAN is superior and more stable in comparison to this model.



i. Concrete Data ii. Wine Quality Data iii. Electrical Power Plant Data iv. News Data

Fig. 2. TabSynDex metric comparison for the synthetic data generated by CTGAN, TGAN, and TGAN with GRU. We observe that none of the methods generate "good quality" synthetic data, i.e., having distribution close to the real data. (a) CTGAN. (b) TGAN. (c) TGAN with GRU.

7) Wasserstein GANs With Gradient Penalty (WGAN-GP): We notice that all the previously discussed models with GMM or VGMM preprocessing usually begin with a lot of generative knowledge (due to the use of GMM) even before training starts. Therefore, the performance after 50 epochs is almost same as after 1 epoch in most of the cases. Most of the learning takes place during the preprocessing stage and not in the neural network. Thus, at last, we train a model without using GMM preprocessing. We use WGAN-GP [32], [33]. Fig. 3(d) shows the results for WGAN-GP. This gives comparatively stable performance and shows gradual improvement with training in all datasets. Despite showing improvement in the quality of data that it generates with increasing number of epochs, it fails to overcome the overall scores achieved by CTGAN. This shows the importance of GMM preprocessing in obtaining useful synthetic data.

8) TVAE: In addition to the GAN-based methods, we also use a VAE-based method named TVAE [6], [7] to evaluate the synthesized tabular data. The results of TVAE on all four datasets are given in Table III. It can be observed that TVAE performs quite well (better than all other methods) on *concrete data* but obtain average to poor performance on other three datasets. More detailed results are discussed in the Supplementary Material.

C. TabSynDex Score Comparison

We compare the TabSynDex score of eight existing tabular data generation methods and tabulate the same in Table III. The TabSynDex score is computed on the test set of four datasets. Table III tabulates the maximum similarities the various models attain over the datasets. Clearly, none of the models generate data with similarities anywhere near to those attained by real data subsets in Fig. 2(a). Also, as is visible from Table III, the GAN-based methods do not perform much better than standard statistical approaches, such as DataSynthesizer.

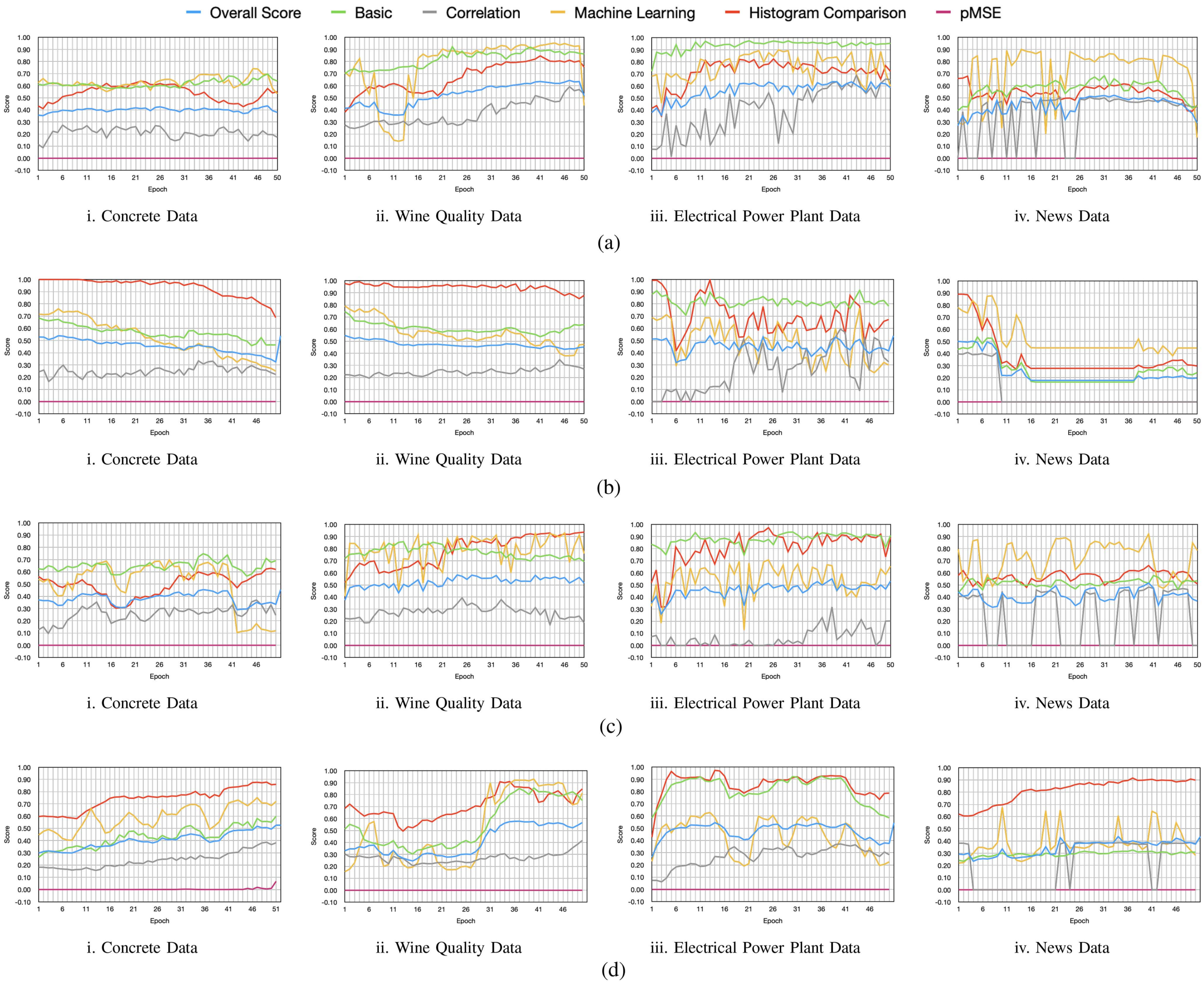
TABLE III
COMPARATIVE ANALYSIS OF THE TABULAR SYNTHETIC DATA GENERATED BY DIFFERENT MODELS

	TabSynDex Score			
Method	Concrete	Wine Quality	Power Plant	News
DataSynthesizer [21]	0.47	0.49	0.52	0.38
TVAE [6], [7]	0.72	0.45	0.65	0.40
CTGAN [6]	0.52	0.57	0.53	0.58
TGAN (LSTM) [5]	0.51	0.56	0.68	0.48
TGAN (GRU) [34]	0.51	0.55	0.66	0.44
TGAN (Bi) [6], [35]	0.43	0.64	0.64	0.52
TGAN (VGMM) [6]	0.54	0.55	0.54	0.50
TGAN (Bi, VGMM) [6]	0.45	0.58	0.55	0.52
WGAN-GP [32], [33]	0.53	0.58	0.55	0.44

The TabSynDex score on the test set is reported on four datasets.

D. Proposed and Existing Metric Score Comparison

We compare TabSynDex score with the existing metric scores in Table IV. The metric scores for TVAE [6], [7] are shown for all four datasets. The *JS-divergence* [24] denotes the average JS-divergence between the corresponding categorical columns of real and synthetic data. The *Wasserstein distance* [24] denotes the average Wasserstein distance between the corresponding continuous columns of real and synthetic data. The *correlation*



i. Concrete Data ii. Wine Quality Data iii. Electrical Power Plant Data iv. News Data

(a)

i. Concrete Data ii. Wine Quality Data iii. Electrical Power Plant Data iv. News Data

(b)

i. Concrete Data ii. Wine Quality Data iii. Electrical Power Plant Data iv. News Data

(c)

i. Concrete Data ii. Wine Quality Data iii. Electrical Power Plant Data iv. News Data

(d)

Fig. 3. TabSynDex score of synthetic tabular data generated by Bidirectional TGAN, TGAN with VGMM Preprocessing, Bidirectional TGAN with VGMM Preprocessing, and WGAN-GP. The quality of the synthesized data is not very good in most of the methods. Moreover, there is negligible learning over the epochs in almost every case except WGAN-GP. (a) Bidirectional TGAN. (b) TGAN with VGMM Preprocessing. (c) Bidirectional TGAN with VGMM Preprocessing. (d) WGAN-GP.

TABLE IV
SCORES BY DIFFERENT EVALUATION METRICS FOR SYNTHETIC DATA GENERATED BY TVAE MODEL

	Concrete [36]	Wine Quality [37]	Power Plant [39]	News [38]
JS-Divergence [9], [24]	NA	0.17	NA	0.15
Correlation Difference [24]	0.67	1.17	1.08	4.6
Wasserstein Distance [24]	0.02	0.01	0.02	0.02
ML-Real [5], [6]	7.33	0.99	0.121	3034.35
ML-Syn [5], [6]	8.7	0.85	0.124	2469.91
pMSE [28]	4.23	79.89	343.77	2402.8
TabSynDex	0.72	0.45	0.65	0.40

NA: there were no categorical columns in the data.

difference [24] denotes the L2-distance between correlation matrices of real and synthetic data. The *ML-Real* denotes the performance on the test set after training an ML model on real data and *ML-Syn* denotes the same after training on synthetic data. The ML model performance is in terms of mean absolute error for all datasets except wine quality data in which F1-score is used. The *pmse* [28] is a measure of indistinguishability of synthetic data from real data. The *TabSynDex* is the proposed metric. From Table IV, it can be observed that except *correlation difference* and *TabSynDex*, all other metrics are unbounded. Therefore, it is not possible to interpret these metrics without a reference point. The *correlation difference* does not cover many of desirable attributes of synthetic data, as given in Table I. Thus, TabSynDex is the only metric that can be used to measure the quality synthetic data on an absolute basis and without the need of a reference point.

VI. CONCLUSION

The existing tabular data synthesis methods have shown promise in generating useful synthetic data. The GANs- and VAE-based methods have outperformed the traditional statistical methods. However, the lack of a uniform metric has caused inconsistency in the comparative evaluation of different methods. The existing metrics also do not have an implicit bound. This work presents a single score bounded universal metric TabSynDex for the robust evaluation of synthetic tabular data. The proposed TabSynDex consists of five-component scores capturing various desired qualities of synthetic data. It can be used as a universal metric to compare various GANs, VAEs, or other statistical methods for generating synthetic tabular data. We also show how this metric can be used to evaluate the training of GANs and VAEs to help discover new insights. The proposed metric is compared with the existing metrics based on several desired attributes, such as intercolumn correlation, distribution comparison, rare-class coverage, implicit bound, and suitability to ML applications. In comparison with the existing metrics that cover few of these aspects, the TabSynDex covers all these attributes. In the experiments, it is observed that most of the existing tabular GAN and VAE methods are not competent for robustly synthesizing real-world data. The standard WGAN-GP and TVAE are relatively stable methods where all the component scores usually improve during training with some rare exceptions. Our work highlights the need for further research on the development of better tabular data synthesis methods. Another future work could be to study the relationship between the five constituent metrics and exploring the effect of using different weighing factor for each metric.

References

[1] P. Voigt and A. Von dem Bussche, *The EU General Data Protection Regulation (GDPR): A Practical Guide*, 1st ed., Cham, Switzerland: Springer, 2017.

[2] E. Goldman, “An introduction to the California consumer privacy act (CCPA),” Santa Clara Univ. Legal Studies Research Paper, 2020. [Online]. Available: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3211013

[3] N. Park, M. Mohammadi, K. Gorde, S. Jajodia, H. Park, and Y. Kim, “Data synthesis based on generative adversarial networks,” *Proc. VLDB Endowment*, vol. 11, no. 10, pp. 1071–1083, 2018.

[4] E. Denton, S. Chintala, A. Szlam, and R. Fergus, “Deep generative image models using a laplacian pyramid of adversarial networks,” *Adv. Neural Inf. Process. Syst.*, vol. 28, pp. 1486–1494, 2015.

[5] L. Xu and K. Veeramachaneni, “Synthesizing tabular data using generative adversarial networks,” 2018, *arXiv:1811.11264*.

[6] L. Xu, M. Skoularidou, A. Cuesta-Infante, and K. Veeramachaneni, “Modeling tabular data using conditional GAN,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 7335–7345.

[7] D. P. Kingma and M. Welling, “Auto-encoding variational Bayes,” in *Proc. Int. Conf. Learn. Representations*, 2013.

[8] Z. Zhao, A. Kunar, R. Birke, and L. Y. Chen, “CTAB-GAN: Effective table data synthesizing,” in *Proc. Asian Conf. Mach. Learn.*, 2021, pp. 97–112.

[9] A. Goncalves, P. Ray, B. Soper, J. Stevens, L. Coyle, and A. P. Sales, “Generation and evaluation of synthetic patient data,” *BMC Med. Res. Methodol.*, vol. 20, 2020, Art. no. 108.

[10] A. P. d. V. B. Brenninkmeijer and Y. Hille, “On the generation and evaluation of synthetic tabular data using GANs,” Ph.D. dissertation, Radboud Univ., Nijmegen, The Netherlands, 2019.

[11] H. He, Y. Bai, E. A. Garcia, and S. Li, “ADASYN: Adaptive synthetic sampling approach for imbalanced learning,” in *Proc. IEEE Int. Joint Conf. Neural Netw.*, 2008, pp. 1322–1328.

[12] 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, no. 1, pp. 321–357, 2002.

[13] H. Han, W.-Y. Wang, and B.-H. Mao, “Borderline-SMOTE: A new oversampling method in imbalanced data sets learning,” in *Proc. Int. Conf. Intell. Comput.*, 2005, pp. 878–887.

[14] J. De La Calleja and O. Fuentes, “A distance-based over-sampling method for learning from imbalanced data sets,” in *Proc. FLAIRS Conf.*, 2007, pp. 634–635.

[15] Y. Yue, Y. Li, K. Yi, and Z. Wu, “Synthetic data approach for classification and regression,” in *Proc. IEEE 29th Int. Conf. Appl.-Specific Syst. Architectures Process.*, 2018, pp. 1–8.

[16] V. García, J. Sánchez, and R. Mollineda, “On the use of surrounding neighbors for synthetic over-sampling of the minority class,” in *Proc. 8th Conf. Simul. Modelling Optim.*, 2008, pp. 389–394.

[17] N. Patki, R. Wedge, and K. Veeramachaneni, “The synthetic data vault,” in *Proc. IEEE Int. Conf. Data Sci. Adv. Analytics*, 2016, pp. 399–410.

[18] A. L. Buczak, S. Babin, and L. Moniz, “Data-driven approach for creating synthetic electronic medical records,” *BMC Med. Informat. Decis. Mak.*, vol. 10, no. 1, 2010, Art. no. 59.

[19] J. Walonoski et al., “Synthea: An approach, method, and software mechanism for generating synthetic patients and the synthetic electronic health care record,” *J. Amer. Med. Informat. Assoc.*, vol. 25, no. 3, pp. 230–238, 2017.

[20] J. Young, P. Graham, and R. Penny, “Using Bayesian networks to create synthetic data,” *J. Official Statist.*, vol. 25, no. 4, pp. 549–567, 2009.

[21] H. Ping, J. Stoyanovich, and B. Howe, “Datasynthesizer: Privacy-preserving synthetic datasets,” in *Proc. 29th Int. Conf. Sci. Stat. Database Manage.*, 2017, pp. 1–5.

[22] E. Choi, S. Biswal, B. Malin, J. Duke, W. F. Stewart, and J. Sun, “Generating multi-label discrete patient records using generative adversarial networks,” in *Proc. Mach. Learn. Healthcare Conf.*, 2017, pp. 286–305.

[23] M. K. Baowaly, C.-C. Lin, C.-L. Liu, and K.-T. Chen, “Synthesizing electronic health records using improved generative adversarial networks,” *J. Amer. Med. Informat. Assoc.*, vol. 26, no. 3, pp. 228–241, 2019.

[24] A. Karlsson and T. Sjoberg, “Synthesis of tabular financial data using generative adversarial networks,” *Dissertation*, 2020. [Online]. Available: http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-273633

[25] J. Chen, D. Chun, M. Patel, E. Chiang, and J. James, “The validity of synthetic clinical data: A validation study of a leading synthetic data generator (synthea) using clinical quality measures,” *BMC Med. Informat. Decis. Mak.*, vol. 19, 2019, Art. no. 44.

[26] A. Rajabi and O. O. Garibay, “TabFairGAN: Fair tabular data generation with generative adversarial networks,” *Mach. Learn. Knowl. Extraction*, vol. 4, no. 2, pp. 488–501, 2022.

[27] G. Astolfi, “Generating tabular data using generative adversarial networks with differential privacy,” in *Proc. Conf. Eur. Statisticians*, 2021, pp. 1–12.

[28] J. Snoke, G. M. Raab, B. Nowok, C. Dibben, and A. Slavkovic, “General and specific utility measures for synthetic data,” *J. Roy. Stat. Soc., Ser. A. (Statist. Soc.)*, vol. 181, no. 3, pp. 663–688, 2018.

[29] M.-J. Woo, J. P. Reiter, A. Oganian, and A. F. Karr, “Global measures of data utility for microdata masked for disclosure limitation,” *J. Privacy Confidentiality*, vol. 1, no. 1, pp. 111–124, 2009.

[30] R. A. Fisher, “Statistical methods for research workers,” in *Breakthroughs in Statistics*. Berlin, Germany: Springer, 1992, pp. 66–70.

[31] C. E. Shannon, “A mathematical theory of communication,” *Bell Syst. Tech. J.*, vol. 27, no. 3, pp. 379–423, 1948.

[32] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, “Improved training of Wasserstein GANs,” pp. 5769–5779, 2017.

[33] M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein generative adversarial networks,” vol. 70, pp. 214–223, 2017.

[34] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Empirical evaluation of gated recurrent neural networks on sequence modeling,” in *Proc. NeurIPS Workshop Deep Learn.*, 2014.

[35] J. Donahue, P. Krähenbühl, and T. Darrell, “Adversarial feature learning,” 2016, *arXiv:1605.09782*.

[36] I.-C. Yeh, “Modeling of strength of high-performance concrete using artificial neural networks,” *Cement Concrete Res.*, vol. 28, no. 12, pp. 1797–1808, 1998.

[37] P. Cortez, A. Cerdeira, F. Almeida, T. Matos, and J. Reis, “Modeling wine preferences by data mining from physicochemical properties,” *Decis. Support Syst.*, vol. 47, no. 4, pp. 547–553, 2009.

[38] K. Fernandes, P. Vinagre, and P. Cortez, “A proactive intelligent decision support system for predicting the popularity of online news,” in *Proc. Portuguese Conf. Artif. Intell.*, 2015, pp. 535–546.

[39] P. Tüfekci, “Prediction of full load electrical power output of a base load operated combined cycle power plant using machine learning methods,” *Int. J. Elect. Power Energy Syst.*, vol. 60, pp. 126–140, 2014.