Artificial Intelligence 340 (2025) 104292

Contents lists available at ScienceDirect

Artificial Intelligence

journal homepage: www.elsevier.com/locate/artint

TTVAE: Transformer-based generative modeling for tabular data generation

Alex X. Wang [a], Binh P. Nguyen [a,b,*]

[a] School of Mathematics and Statistics, Victoria University of Wellington, Wellington 6012, New Zealand
[b] Faculty of Information Technology, Ho Chi Minh City Open University, 97 Vo Van Tan, District 3, Ho Chi Minh City 70000, Viet Nam

ARTICLE INFO

Keywords:
Generative AI
Tabular data
Transformer
Latent space interpolation

ABSTRACT

Tabular data synthesis presents unique challenges, with Transformer models remaining underexplored despite the applications of Variational Autoencoders and Generative Adversarial Networks. To address this gap, we propose the Transformer-based Tabular Variational AutoEncoder (TTVAE), leveraging the attention mechanism for capturing complex data distributions. The inclusion of the attention mechanism enables our model to understand complex relationships among heterogeneous features, a task often difficult for traditional methods. TTVAE facilitates the integration of interpolation within the latent space during the data generation process. Specifically, TTVAE is trained once, establishing a low-dimensional representation of real data, and then various latent interpolation methods can efficiently generate synthetic latent points. Through extensive experiments on diverse datasets, TTVAE consistently achieves state-of-the-art performance, highlighting its adaptability across different feature types and data sizes. This innovative approach, empowered by the attention mechanism and the integration of interpolation, addresses the complex challenges of tabular data synthesis, establishing TTVAE as a powerful solution.

1. Introduction

Synthetic data proves valuable for tasks like data augmentation, class balancing, and ensuring data privacy in real datasets [1]. It has wide applications, including face recognition, text classification, graph analysis, disease prediction, anomaly detection, and statistical disclosure [2,3]. A comprehensive exploration of synthetic data use cases is available in [4].

Tabular data synthesis (TDS) is challenging, as tabular datasets often contain heterogeneous data and lack inherent structure. Moreover, defining a well-structured evaluation framework for synthetic tabular data remains unclear. While traditional machine learning (ML) methods are recognized for their superiority over deep learning (DL) techniques in handling such data [5], there is a need to develop a robust deep representation for tabular data, which remains an open problem [6]. Several adapted models, including Variational Autoencoders (VAEs), Generative Adversarial Networks (GANs), and Diffusion models, have gained prevalence [7–9]. Some approaches, like CTGAN and TVAE, rely on conventional Multilayer Perceptrons (MLPs), while others, such as CTABGAN+ and TabDDPM, incorporate more advanced architectures, including Convolutional Neural Networks (CNNs) and Diffusion models. Despite these improvements, these techniques still struggle to capture complex feature interactions in tabular data effectively. Contextual embedding, which focuses on high-level feature interactions during training, has demonstrated considerable success in enhancing tabular feature representation and improving deep learning model performance [10]. Research shows that contextual embedding

* Corresponding author at: School of Mathematics and Statistics, Victoria University of Wellington, Wellington 6012, New Zealand.
E-mail address: binh.p.nguyen@vuw.ac.nz (B.P. Nguyen).

https://doi.org/10.1016/j.artint.2025.104292
Received 6 May 2024; Received in revised form 13 January 2025; Accepted 18 January 2025

Available online 20 January 2025
0004-3702/

consistently outperforms not only traditional MLP-based models but also more complex architectures designed for solving challenging tasks [1]. Moreover, these models often employ stochastically generated latent variables in data generation, limiting control over the generated data. Although limited comparative studies have been conducted with the Synthetic Minority Over-sampling Technique (SMOTE) [11], the results consistently indicate that SMOTE outperforms these models [12,9].

To bridge these gaps, we propose a Transformer-based tabular Variational Autoencoder (TTVAE) designed to utilize the Transformer architecture for model training and employ latent space interpolation for data generation. Our approach leverages a Transformer with an attention mechanism to effectively capture contextual relationships among heterogeneous features [13], which is a key factor in TTVAE's strong performance. Additionally, we incorporate the shallow interpolation mechanism of SMOTE in our data generation process, recognizing its effectiveness in the tabular data domain [14]. This integration allows for improved data quality with minimal overhead. The main contributions of our work are summarized below:

1. We conduct a comprehensive evaluation of key tabular data generation models, assessing their relative performance across various tasks and datasets.
2. We present TTVAE, a streamlined tabular data generation algorithm capable of synthesizing mixed data types, including numerical and categorical features.
3. We demonstrate TTVAE's superior performance over alternative TDS models, including statistical, traditional ML, and recent DL models. Our analysis identifies key factors contributing to its success.
4. Building on previous research, we recognize the efficacy of SMOTE in generating competitive synthetic data. However, our study consistently demonstrates the superior performance of TTVAE compared to SMOTE, particularly when dealing with large datasets.

2. Related work

2.1. Data generation

Data generation is the process of creating synthetic data resembling real data [15]. It serves wide applications, including data augmentation, data balancing, enhancing privacy, and training/testing ML models [16]. Tabular data generation, among other modalities, has emerged as a significant topic due to its widespread usage in daily activities and across diverse industries [17]. However, generating synthetic tabular data is non-trivial due to its inherent complexities. Firstly, tabular data collection can be diverse and manual. It can also originate from various systems, leading to arbitrary dataset lengths and inconsistent features with missing values. Secondly, the heterogeneous nature of tabular data, which includes both categorical and numerical features, presents challenges in both model training and evaluation. The encoding of categorical variables into a numeric format adds complexity, especially when features are derived from unrelated sources. Moreover, traditional metrics borrowed from other domains, like the "Inception Score", tailored for evaluating 2D image data, may not be suitable for measuring the quality of synthetic tabular data [18]. Finally, the generation of tabular synthetic data is often driven by specific business questions, requiring different preprocessing and postprocessing steps. For example, it is crucial to recognize that synthetic data created to enhance ML models differs significantly from data created for privacy preservation; each represents a distinct trade-off between data accuracy and privacy [19,20].

2.2. Variational autoencoder

The Variational Autoencoder is a probabilistic generative model designed for learning latent representations of data [21]. It consists of two primary components: an encoder network and a decoder network. Let $\mathbf{X}$ represent the observed data and $\mathbf{Z}$ the latent variable. The encoder function $q_\phi(\mathbf{Z}|\mathbf{X})$ estimates the posterior distribution of the latent variable given the data, while the decoder function $p_\theta(\mathbf{X}|\mathbf{Z})$ models the conditional distribution of the data given the latent variable. Both q_ϕ and p_θ are parametrized by neural networks with parameters ϕ and θ, respectively. The objective of the VAE is to maximize the Evidence Lower Bound (ELBO) on the log-likelihood of the observed data, where the ELBO is expressed as:

$$\log p_\theta(\mathbf{X}) \geq \mathbb{E}_{q_\phi(\mathbf{Z}|\mathbf{X})}[\log p_\theta(\mathbf{X}|\mathbf{Z})] - \mathrm{KL}[q_\phi(\mathbf{Z}|\mathbf{X})||p(\mathbf{Z})]. \tag{1}$$

Here, the first term on the right-hand side is the reconstruction term, representing the log-likelihood of the data given the latent variable. The second term, $\mathrm{KL}[q_\phi(\mathbf{Z}|\mathbf{X})||p(\mathbf{Z})]$, is the Kullback-Leibler (KL) divergence between the approximate posterior $q_\phi(\mathbf{Z}|\mathbf{X})$ and the prior distribution $p(\mathbf{Z})$, which is typically chosen as a multivariate standard normal distribution. To enable differentiability during training, the VAE employs the reparameterization trick. This trick involves sampling a latent variable $\mathbf{z}$ from the approximate posterior in a way that allows backpropagation. For a standard normal prior, the reparameterization trick can be expressed as:

$$\mathbf{Z} = \mu + \sigma \odot \epsilon, \tag{2}$$

where μ and σ represent the mean and standard deviation predicted by the encoder, and ϵ is sampled from a standard normal distribution. In summary, VAEs are trained to learn a probabilistic mapping from the data space to a lower-dimensional latent space. This learned mapping enables both efficient data compression and data generation.

2.3. Transformer

The Transformer architecture proposed by Vaswani et al. in 2017 [22], has played a pivotal role in natural language processing. Its fundamental strength lies in the utilization of the self-attention mechanism, which enables parallel processing and efficient capture of dependencies within sequences. This capability to model relationships and learn long-range dependencies has contributed significantly to its widespread success.

Given a sequence of input vectors, typically denoted as $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$, where $\mathbf{x}_i \in \mathbb{R}^d$ represents the i-th element in the sequence with a d-dimensional vector, the self-attention mechanism computes three linear projections for each element: queries ($\mathbf{Q}$), keys ($\mathbf{K}$), and values ($\mathbf{V}$). These are defined as:

$$\mathbf{Q} = \mathbf{X}W_Q, \quad \mathbf{K} = \mathbf{X}W_K, \quad \mathbf{V} = \mathbf{X}W_V,$$

where W_Q, W_K, and W_V are learnable weight matrices. The attention mechanism calculates the attention score between each query and key by performing a scaled dot-product:

$$\text{Attention}(\mathbf{K}, \mathbf{Q}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q} \cdot \mathbf{K}^T}{\sqrt{d_k}}\right) \cdot \mathbf{V},$$

where d_k is the dimensionality of the keys and queries. The Softmax function normalizes these scores so they sum to 1, effectively weighting the importance of each value in the sequence. The Transformer applies this self-attention mechanism across multiple layers, with each layer focusing on different aspects of the input sequence. To account for the sequential nature of data—like word order in sentences or temporal dependencies in time series—the model uses positional encodings that indicate the position of each element in the sequence. Importantly, the self-attention framework isn't limited to NLP; its flexibility makes it applicable to various data types, including tabular datasets. For instance, in tabular data with a mix of numerical and categorical features, the self-attention mechanism can capture both local and global interactions by learning dependencies between features, regardless of their type. This adaptability allows the Transformer to tackle the complexities and nonlinear relationships often found in tabular data.

Transformers provide significant advantages in the tabular data domain, such as capturing complex feature interactions and learning meaningful representations in latent space [13]. They have proven effective in supervised learning tasks [23,24]. For instance, AutoInt, introduced by Song et al. [25], emphasizes the importance of self-attention mechanisms and automatically learns high-order feature interactions for subsequent predictive tasks. TabTransformer [26] employs a transformer encoder to derive contextual embeddings for categorical features, while FT-Transformer [27] extends a similar mechanism to both categorical and numerical features. These approaches have further been extended to time series data and integrated with advanced DL algorithms. FATA-Trans [23], introduces a field- and time-aware Transformer model for sequential tabular data, enhancing its ability to model temporal patterns. Yan et al. propose T2G-FORMER, incorporating a Graph Estimator to automatically estimate relations among tabular features, guiding Transformer-based tabular learning [24]. However, while Transformers have shown effectiveness in supervised learning with tabular data, their potential in tabular data generation remains unexplored. In response, we introduce TTVAE, a simple adaptation of the Transformer architecture tailored for TDS.

Algorithm 1 Transformer-based tabular variational autoencoder.

Input: Real data samples $\{x(i)\}_{i=1}^m$, learning rates $\alpha_{\text{enc}}, \alpha_{\text{dec}}$, MMD-based regularization coefficient β
Initialize: Autoencoder parameters ϕ, θ
for each training iteration **do**
  Sample $x(i)$ from the training set
  Sample $z'(i)$ from the true prior p_z
  Sample z from $q_\phi(z|x)$
  Train the encoder/decoder (ϕ, θ)**:**
  Compute the reconstruction loss:
  $L_{\text{rec}} \leftarrow -\frac{1}{m}\sum_{i=1}^m \log p_\theta(x(i)|z(i))$
  Compute the regulation loss:
  $L_{\text{reg}} \leftarrow \beta \cdot \text{MMD}(q(z), p(z))$
  Update encoder and decoder parameters:
    $\phi \leftarrow \phi - \alpha_{\text{enc}} \nabla_\phi (L_{\text{rec}} + L_{\text{reg}})$
    $\theta \leftarrow \theta - \alpha_{\text{dec}} \nabla_\theta (L_{\text{rec}} + L_{\text{reg}})$
end for

3. TTVAE: transformer-based tabular variational AutoEncoder

This section introduces our proposed tabular data generation model, TTVAE, which leverages Transformer-based neural networks. TTVAE comprises two key stages: (1) fine-tuning a Transformer-based VAE to capture feature interactions in the latent space; and (2) sampling from latent space interpolations derived from the fine-tuned VAE to generate synthetic tabular data. The complete architectural overview is provided in Fig. 1. We then elaborate on each component, covering fine-tuning and sampling procedures, and conclude with a brief summary of our approach.

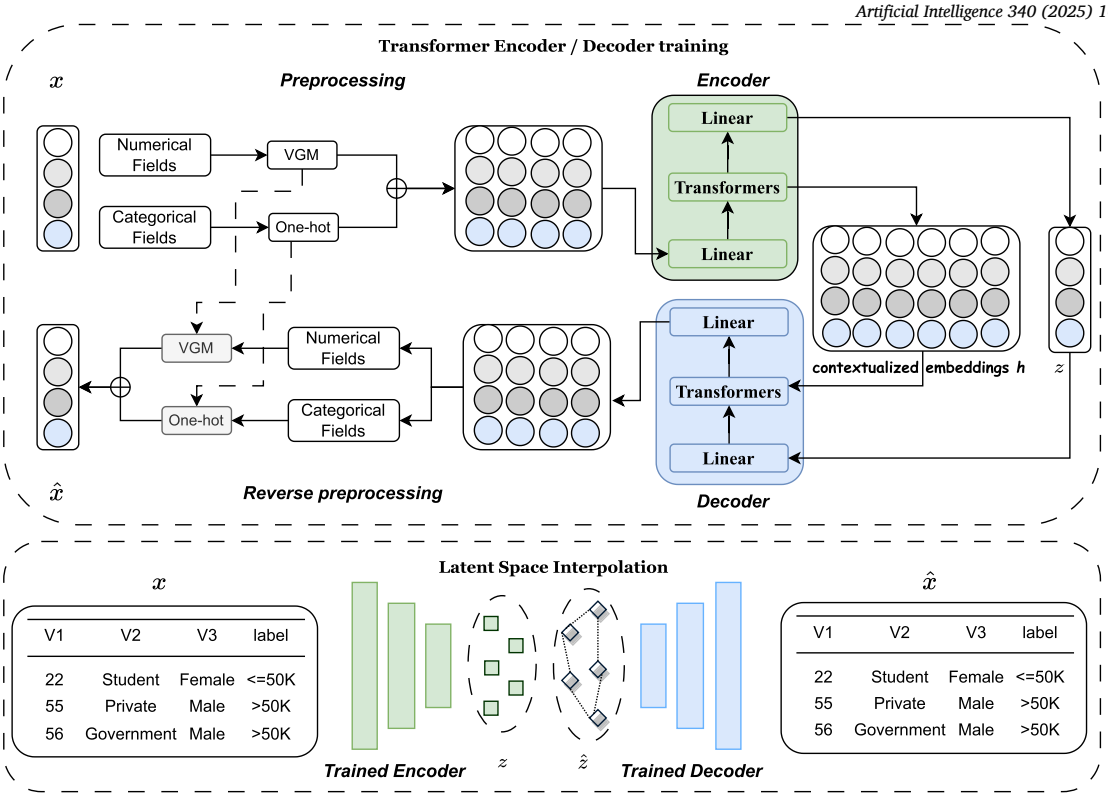

Fig. 1. Illustration of TTVAE with latent space interpolation. During training, each data sample x undergoes a transformation, mapping it to contextual embeddings h and a latent space z via column-wise preprocessing and a Transformer-based encoder. These embeddings h and z are used as inputs for a Transformer-based decoder, incorporating reconstruction and a penalty loss. In the generation phase, a synthetic latent space $\hat{z}$ is generated through latent space interpolation on the encoded latent space z. This, along with contextual embeddings h, is fed into the trained decoder to generate synthetic data, followed by reverse processing to return the data to its original format, $\hat{x}$.

3.1. Problem definition of TDS

For a dataset of $\{x_i | x_i \in R, i = 1, \ldots, N\}$ as samples from a true data distribution $q(x)$, the goal of a DGM is to build deep neural networks with parameters $\theta \in R$, to describe a distribution $p_\theta(x)$ so that the parameters θ can be trained to ensure $p_\theta(x)$ match $q(x)$ the best [16].

3.2. Preprocessing

Following previous studies [7,8], we employed different preprocessing steps for continuous and categorical variables. For continuous variables, a variational Gaussian mixture model (VGM) is applied to handle non-Gaussian and multimodal distributions. The model estimates the number of modes, normalizes values within each mode, and represents them as a concatenation of normalized values $\alpha_{i,j}$ and one-hot encoded modes $\beta_{i,j}$ for each row j. Categorical variables are encoded using one-hot encoding. As a result, the initial jth row, r_j can be represented as the concatenation of processed numerical and categorical variables:

$$r_j = \alpha_{1,j} \oplus \beta_{1,j} \oplus \ldots \oplus \alpha_{n_c,j} \oplus \beta_{n_c,j} + d_{1,j} \oplus \ldots \oplus d_{n_d,j}. \tag{3}$$

3.3. Transformer variational AutoEncoder

The preprocessing steps above project heterogeneous features from tabular datasets into a common latent space. However, this method often misses the relationships and interactions between these features, which are important for understanding the data structure. To address this, we introduce a contextual embedding mechanism that utilizes the attention mechanism from Transformer architectures, inspired by previous work in supervised learning [26,27].

We propose a new extension of the VAE framework that incorporates the Transformer's capabilities to enhance structured data generation for tabular datasets. By adapting the Transformer encoder-decoder architecture, known for its success in sequence-to-sequence tasks [13], we improve the model's ability to capture complex feature interactions and dependencies during training. Central to this approach is the Transformer's self-attention mechanism, which computes attention scores to model dependencies between features. Given a dataset $\mathbf{X}$ with n features, $\mathbf{X} = \{x_1, x_2, \ldots, x_n\}$, the self-attention mechanism learns weighted representations of

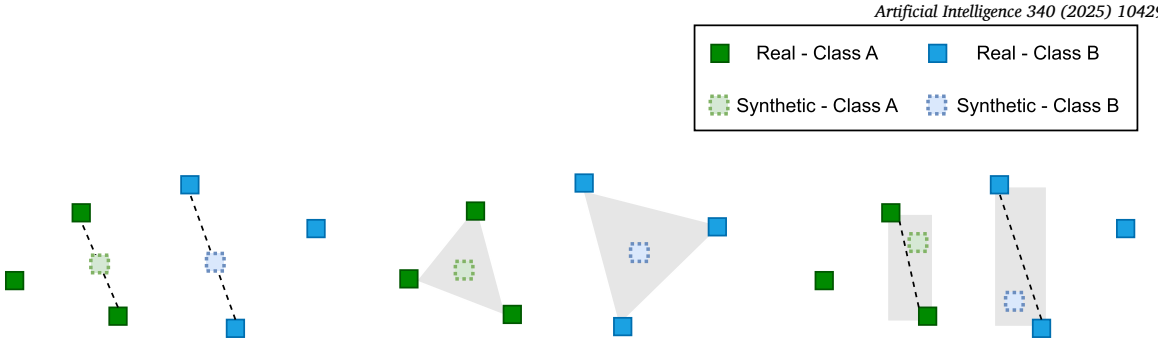

Fig. 2. Illustration of latent interpolation discussed in this study. **a** within-class interpolation, **b** triangular interpolation, **c** hyper-rectangle interpolation.

each feature. This allows the model to capture both local and global interactions, making it adept at handling diverse data types in tabular datasets and seamlessly integrating categorical and numerical features. Ultimately, this results in more coherent and structured synthetic data. Let $\mathbf{Z}$ denote the latent variables. The encoder, defined as $q_\phi(\mathbf{Z}|\mathbf{X})$, uses a Transformer-based mechanism to attend to different parts of the input data. Through the application of self-attention, the encoder generates contextual embeddings h, which capture feature relationships in a compressed format [26]. These learned contextual embeddings are crucial for effectively encoding interactions between features and will be leveraged in the data generation step. The decoder, $p_\theta(\mathbf{X}|\mathbf{Z})$, also utilizes the Transformer architecture to map the latent variables back to the original data space. The learned contextual embeddings h enable the decoder to model dependencies between features during the generation process, resulting in synthetic data that maintains the inherent relationships found in the original dataset.

3.4. Loss function

In a standard VAE, the loss function is typically expressed as the negative ELBO shown in Equation (1). Here are two main components: the reconstruction term $\mathbb{E}_{q(z|x)}[\log p(x|z)]$ and the KL divergence term $\mathrm{KL}[q(z|x)||p(z)$, which encourages the model's latent space distribution to approximate a prior distribution. While the ELBO primarily captures the first two moments (mean and variance) of the data distribution, we adopt Maximum Mean Discrepancy (MMD) [28], following [29], to capture higher-order moments and better represent complex data distributions. The resulting loss function is defined as:

$$\mathcal{L}_{\text{TTVAE}} = -\mathbb{E}_{q(z|x)}[\log p(x|z)] + \beta \cdot \text{MMD}(q(z), p(z)). \tag{4}$$

Here, $\text{MMD}(q(z), p(z))$ represents the Maximum Mean Discrepancy between the approximate posterior $q(z)$ and the prior $p(z)$, calculated as follows [30]. The parameter β serves as a hyperparameter governing the intensity of the MMD regularization.

$$\begin{aligned} &\text{MMD}^2(q(z), p(z)) = \|\mu_q - \mu_p\|_{\mathcal{F}}^2 \\ &= \frac{1}{n(n-1)}[\sum_{i \neq j} k(z_i, z_j) + \sum_{i \neq j} k(z'_i, z'_j)] - \frac{2}{nn}\sum_{i,j} k(z_i, z'_j), \end{aligned} \tag{5}$$

where $k(.,.)$ is a positive definite kernel function:

$$k(q, p) = \exp\left(-\frac{\|z - z'\|^2}{2\sigma^2}\right). \tag{6}$$

A higher β value emphasizes the MMD loss, promoting a more disentangled latent space. Conversely, a lower β value promotes the reconstruction loss, potentially leading to better data reconstruction.

3.5. Latent space interpolation for data generation

Latent space interpolation is a well-known concept in generative modeling, particularly useful in computer vision [31]. Various interpolation techniques can be employed to introduce noise into the latent space [32]. One basic approach, similar to *SMOTE*, involves generating synthetic latent variables by interpolating between a randomly selected point and one of its neighbors. This is analogous to *mixup* in supervised learning, where input samples are linearly interpolated to augment data, encouraging smoother transitions between classes and promoting better generalization [33]. Another approach is the *triangular mechanism*, where n observations z are combined to generate a new latent variable $\hat{z} = \lambda_1 z_1 + \cdots + (1 - \lambda_1 - \cdots - \lambda_{n-1})z_n$, with $\sum_{i=1}^{n-1} \lambda_i = 1$. Similarly, the *hyper-rectangle mechanism*, generates a new latent variable by sampling within a rectangular area formed between two latent variables. These methods help explore the latent space and introduce smooth transitions between data points, as illustrated in Fig. 2.

In the context of VAEs, *latent mixup* facilitates interpolation within the learned latent space instead of the input space. This shift allows for the synthesis of new samples that possess more meaningful latent features, which has been shown to promote smoother decision boundaries in the latent space. Therefore, this results in more realistic and semantically coherent data generation [34]. By

Table 1
Datasets used in this study. #S, #N and #C stand for the number of samples, numerical and categorical columns.

Abbr	Name	#S	#N	#C	Task type
CR	credit-g	1000	3	18	Classification
SI	sick	3103	5	18	Classification
AB	abalone	4177	7	2	Regression
NA	wine-quality	4898	11	1	Regression
PA	parkinsons-telemonitoring	5875	18	2	Regression
NC	NewspaperChurn	15855	3	14	Classification
CO	compass	16644	5	13	Classification
LA	law-school-admission	20800	2	9	Classification
KI	King	21613	13	6	Regression
NO	nomao	34465	56	63	Classification
NW	news	39644	40	18	Regression
MV	mv	40768	6	4	Classification
BE	Beijing	43824	7	5	Regression
BM	Bank-Marketing	45211	7	10	Classification
AD	adult	48842	5	10	Classification
CI	Census-Income	299285	9	33	Classification

enabling smooth transitions between samples, latent space interpolation not only fosters the exploration of learned representations but also preserves critical data semantics throughout the process [31]. Building on the concept of *latent mixup*, we specifically employ the *triangular interpolation* method applied to the encoded latent space derived from the Transformer encoder. This approach significantly enhances continuity in data generation and increases the meaningfulness and realism of the synthesized samples by leveraging learned representations, in contrast to using randomly generated variables [14].

4. Experiments

In this section, we experimentally assess TTVAE's performance using diverse data quality metrics, aiming to establish a unified and comprehensive framework for evaluating TDS techniques. Our code is made publicly available in our GitHub repository.[1] We implemented TTVAE and all the baseline methods using PyTorch. All the experiments were conducted on an NVIDIA T4 2560 GPU with 16GB of memory.

4.1. Datasets

To systematically assess tabular generative models, we chose a diverse set of 16 real-world public datasets with variations in size, characteristics, features, and distributions. These datasets, previously used in evaluating tabular models, are detailed in Table 1.

4.2. Baselines

A variety of methods exist for generating synthetic tabular data, each with its own unique strengths and limitations [35]. Statistical distribution-based approaches simulate data based on known distributions, capturing statistical properties but often assuming feature independence [36]. Distance-based methods, typified by SMOTE, are effective for oversampling minority classes, with variants like ADASYN and SMOTENC addressing categorical data. DGMs are versatile in generating diverse data types but encounter difficulties when handling discrete and categorical data. Recent research has produced numerous models, including tabular VAEs, GAN-based models, and Diffusion-based models, such as TVAE and TabDDPM. In this study, we conduct a comprehensive comparison against state-of-the-art tabular synthetic data models from ML and DL models, including SMOTE [37], and its variants: ADASYN [38], SMOTENC [39], along with TVAE, CTGAN [7], CTABGAN+ [8], TabDDPM [9], TabSyn [40] and Tabula [41].

4.3. Evaluation measures

In alignment with prior studies [15,42,43], we assess the quality of synthetic data across three dimensions: ML utility, statistical similarity, and privacy. We also include visualization aids for each evaluation aspect. For ML utility, we employ various supervised learning algorithms to train classification or regression models, evaluating their performance on a separate holdout test dataset to prevent information leakage. To ensure a fair comparison, default hyperparameters are utilized [8,9]. The objective is to demonstrate that models trained on high-quality synthetic data can be as competitive as, or even outperform, those trained on real data. We calculate an average efficiency across a diverse set of ML models, including logistic regression, decision trees, and others, following common practices in the literature.

[1] https://github.com/coksvictoria/TTVAE.

For statistical similarity, we employed three groups of metrics commonly used in previous studies: column-wise metrics (univariate), pair-wise column correlation (bivariate), and table-level metrics (multivariate) [8,44,9]. To assess synthetic data from a univariate distribution perspective, we utilized Kullback-Leibler divergence (KLD), Jensen-Shannon divergence (JSD), and Wasserstein distance (WD) [45,8]. Additionally, to determine if the synthetic data captures the expected interrelationships in the ground truth, a correlation analysis is performed. Finally, we adopted the metrics proposed in [46] to evaluate data fidelity (realism) using density and data diversity (variability) using coverage. While coverage is bounded between 0 and 1, density has no upper limit, with higher values indicating better performance in both cases. When examining distributions at the table level, we use the Maximum Mean Discrepancy (MMD) [28], a kernel-based statistical test designed to discern disparities between the distribution in synthetic data and that of real data, with lower values indicating higher quality of synthetic data.

In privacy evaluation, we use a variety of methods to assess the privacy levels achieved by the synthetic data. However, we acknowledge that the selection and standardization of privacy evaluation metrics are still open research challenges. To begin, we use the mean Distance to Closest Record (DCR) [9], which calculates the average minimum L2 distance from each synthetic sample to real records. Low DCR values suggest that synthetic samples closely emulate real data points, raising potential privacy concerns. However, it is important to note that this might also indicate a close resemblance between synthetic and real data. To directly evaluate whether real data points are merely being replicated, we calculate the number of duplications [16] and divide it by the total sample size. We define this as the Copy Ratio (CR), which quantifies the proportion of real data instances duplicated within the synthetic dataset. This metric helps quantify the degree to which synthetic data replicates records from the original dataset. To further aid privacy assessment, we employed ML detectability as a propensity score for each record to measure the risk of a black-box privacy attack, which aims to infer whether a record belongs to the real training data [47,9]. This involves training an ML classification model to assess the difficulty of distinguishing synthetic data from real data. The process includes shuffling real and synthetic data, assigning flags indicating their origin, and cross-validating a binary classifier predicting these flags [48]. As shown in the equation below, we use the `propsensityMSE` (pMSE) score, which is the mean squared difference of the probability (propensity score) from 0.5. We normalize it to a range of [0,1] by dividing by 0.25. A score of 0 implies indistinguishability between synthetic and real data, while a score of 1 indicates complete distinguishability. Ideally, high-quality synthetic data should show a low DCR and a high pMSD, reflecting proximity to real data without mere replication, and thus minimizing privacy risks.

$$pMSE = \frac{1}{N}\sum_{i=1}^{N}\frac{(p_i - 0.5)^2}{0.25}, \tag{7}$$

where N is the sample size, and p_i represents predicted probabilities (propensity scores) from chosen classification algorithms, such as logistic regression and decision trees.

5. Results and discussion

In this section, we present our experimental results regarding ML utility, Statistical Similarity, and Privacy. Additionally, we offer insights into computational efficiency. An effective data synthesis algorithm should prioritize both accuracy and speed.

5.1. ML utility

The results of the ML utility evaluation are presented in Table 2, aiming to assess the performance of synthetic datasets in predictive tasks compared to real data. It is worth noting that SMOTENC struggles with large datasets, particularly those with either a high number of features or large sample sizes. As a result, it failed to generate data for the *CI* and *NO* datasets. A superior synthetic dataset should demonstrate performance in ML utility that is comparable to real data. Our extensive evaluations across diverse public benchmark datasets reveal the remarkable performance of TTVAE, surpassing baseline models and even outperforming the widely recognized and challenging-to-beat shallow SMOTE model [5,9]. Specifically, TTVAE exhibits superiority over other state-of-the-art (SOTA) methods across a majority of datasets, particularly for larger datasets. However, for smaller datasets, SMOTE-based algorithms remain competitive. This is expected given that DL typically benefits from larger datasets for optimal performance. This robust performance of TTVAE underscores its effectiveness in generating synthetic tabular data that retains utility comparable to real-world datasets across various scales. The enhancement observed in comparison to TVAE highlights the positive impact of the Transformer-based architecture, the introduction of a new loss function, and the incorporation of latent space interpolation. These elements collectively contribute to the improved learning of inter-feature correlations and the overall training effectiveness of VAE.

5.2. Statistical similarity

We qualitatively evaluate TTVAE's ability to model individual and joint feature distributions compared to baseline methods. Synthetic datasets are generated for each dataset using various data synthesis algorithms, matching the size of the real training set and employing stratified sampling for classification datasets. Statistical similarity metrics are then calculated. The results are summarized in Table 3, which consistently highlights TTVAE's superior performance over baselines across all selected metrics, with notable excellence in accuracy and privacy. At the univariate level, synthetic data generated by TTVAE exhibits the closest resemblance based on different distance metrics, including Kullback-Leibler divergence (KLD = 2.3), Jensen-Shannon divergence (JSD = 2.1), and Wasserstein Distance (WD = 2.6). To visually compare the univariate distributions of real and synthetic data, Fig. 3 present paired histogram visualizations of each competing algorithm on numerical and categorical variables selected from the *AD* dataset.

Table 2
The values of ML utility computed w.r.t. classification/regression models. − denotes that SMOTENC cannot handle large datasets. **Bold** represents the best score on each dataset. For classification, higher F1 ↑ means better while for regression, lower RMSE ↓ means better.

	CR (F1↑)	SI (F1↑)	AB (RMSE↓)	NA (RMSE↓)	PA (RMSE↓)	NC (F1↑)	CO (F1↑)	LA (F1↑)
SMOTE	0.5819±0.17	0.0923±0.07	**0.3138±0.11**	0.4261±0.14	0.5264±0.20	0.1067±0.09	0.6782±0.04	0.7399±0.03
ADASYN	0.5988±0.13	0.0747±0.08	0.3172±0.11	**0.4238**±0.14	0.5262±0.20	0.0997±0.08	0.6726±0.05	0.7442±0.03
SMOTENC	**0.7998±0.03**	**0.2599±0.07**	0.3301±0.12	0.5038±0.19	0.5295±0.20	0.2572±0.16	**0.6894±0.03**	0.7784±0.03
TVAE	0.7750±0.06	0.1429±0.10	0.4314±0.17	0.6004±0.23	0.6233±0.18	0.3172±0.13	0.6422±0.03	0.7497±0.04
CTGAN	0.7686±0.06	0.0149±0.03	0.3175±0.11	0.4465±0.14	0.5020±0.15	0.2221±0.13	0.6017±0.07	0.7535±0.04
CTABGAN+	0.7475±0.10	0.0104±0.03	0.3874±0.13	0.4797±0.14	0.5583±0.15	0.2558±0.10	0.6755±0.03	0.6567±0.27
TabDDPM	0.7538±0.12	0.1928±0.10	0.4336±0.15	0.6254±0.23	0.5200±0.18	0.3129±0.17	0.6758±0.04	0.7852±0.03
TabSyn	0.7965±0.12	0.2456±0.03	0.4125±0.12	0.4685±0.13	0.5087±0.16	0.3156±0.09	0.6892±0.16	**0.7865±0.24**
Tabula	0.5785±0.36	0.2308±0.16	0.3823±0.15	0.4594±0.12	0.5523±0.25	0.2326±0.11	0.5611±0.13	0.7213±0.12
TTVAE	0.7669±0.09	0.2284±0.11	0.4069±0.14	0.4924±0.23	**0.4981±0.17**	**0.3372±0.17**	0.6872±0.04	0.7857±0.03
Real	0.7987±0.05	0.2539±0.11	0.3001±0.11	0.3912±0.14	0.5251±0.21	0.3109±0.21	0.7032±0.05	0.7857±0.03
	KI (RMSE↓)	NO (F1↑)	NW (RMSE↓)	MV (F1↑)	BE (RMSE↓)	BM (F1↑)	AD (F1↑)	CI (F1↑)
SMOTE	0.0841±0.03	0.8697±0.16	0.0098±0.01	0.9553±0.07	0.5862±0.01	0.1682±0.09	0.4333±0.12	0.2844±0.08
ADASYN	0.0825±0.03	0.8587±0.14	0.0075±0.01	0.9537±0.07	0.5765±0.01	0.1836±0.09	0.4484±0.09	0.2989±0.08
SMOTENC	0.0840±0.03	-	0.0101±0.01	0.9412±0.06	0.5844±0.01	0.3419±0.07	0.4970±0.10	-
TVAE	0.1505±0.05	0.9382±0.02	0.1341±0.14	0.9267±0.10	0.6253±0.01	0.4125±0.05	0.4636±0.12	0.2204±0.06
CTGAN	0.1319±0.04	0.8313±0.14	0.0663±0.03	0.9126±0.06	0.6253±0.01	0.3955±0.05	0.4463±0.18	0.3563±0.09
CTABGAN+	0.1200±0.04	0.8795±0.04	0.0643±0.03	0.9296±0.07	0.5601±0.01	0.2909±0.14	0.4606±0.17	0.3955±0.06
TabDDPM	0.1085±0.04	0.9366±0.01	0.0112±0.54	0.9576±0.06	0.5688±0.01	0.3711±0.11	0.5128±0.16	0.3793±0.06
TabSyn	0.0866±0.12	0.9389±0.03	0.0098±0.12	0.9601±0.13	0.5648±0.16	0.4265±0.09	0.5198±0.16	0.4156±0.24
Tabula	0.1034±0.08	0.8623±0.23	0.0091±0.02	0.9234±0.11	0.5522±0.05	0.4212±0.03	0.5023±0.10	0.4024±0.09
TTVAE	**0.0811±0.03**	**0.9407±0.01**	**0.0073±0.01**	**0.9605±0.06**	**0.5265±0.01**	**0.4311±0.08**	**0.5206±0.14**	**0.4170±0.08**
Real	0.0711±0.03	0.9685±0.01	0.0068±0.01	0.9683±0.06	0.4564±0.01	0.4264±0.10	0.5228±0.17	0.4679±0.09

Table 3
Average ranks (smaller is better), are computed across all datasets for eight metrics, each representing distinct measurement aspects, to evaluate the resemblance between real and synthetic data.

	KLD	JSD	WD	Correlation	Density	Coverage	MMD	DCR	CR	pMSE
SMOTE	5.2	5.4	5.9	4.3	5.4	5.9	6.0	5.4	6.1	5.1
ADASYN	4.8	5.1	5.6	4.1	5.5	4.5	5.8	4.9	6.0	4.0
SMOTENC	4.1	4.3	5.0	4.3	3.6	3.6	5.6	4.4	6.4	3.4
TVAE	8.8	7.3	6.3	4.9	5.1	6.8	7.1	7.1	6.0	7.4
CTGAN	5.0	6.2	7.1	5.9	7.5	6.8	7.0	7.5	4.6	7.6
CTABGAN+	7.1	6.7	6.8	5.1	7.6	7.9	6.1	8.1	**2.2**	7.7
TabDDPM	4.7	4.2	4.2	3.8	4.5	4.6	4.5	4.1	6.2	4.9
TabSyn	3.1	2.9	3.1	2.9	3.2	3.8	2.9	2.8	5.9	2.6
Tabula	6.1	7.0	5.3	5.2	7.6	6.6	6.9	6.9	5.8	7.4
TTVAE	**2.3**	**2.1**	**2.6**	**2.3**	**2.7**	**2.4**	**2.1**	**2.1**	5.4	**2.4**

At the bivariate level, we employ diverse correlation measures, following a previous study [16]. We then compute differences between correlation matrices on real and synthetic data for each dataset. We use Pearson correlation coefficients when both are continuous variables, Eta coefficients for one nominal and one continuous variable, and point biserial correlation coefficients for one dichotomous and one continuous variable. When both variables are nominal (including dichotomous), Cramer's V is applied. Based on correlation metrics, TTVAE outperforms all other baselines in generating synthetic datasets with more realistic pairwise correlations (Correlation = 2.3). The results for the largest four datasets where all competing algorithms can handle (*AD, BM, BE, MV*) are presented in Fig. 4, demonstrating TTVAE's consistently superior performance in supporting this evaluation.

At the table level, TTVAE maintains its top ranking, showcasing superior performance in Density (2.7), Coverage (2.4), and MMD (2.1). To visually compare the multivariate relationships between real and synthetic data, Fig. 5 presents a paired scatter plot with density plots for the largest four synthetic datasets (*AD, BM, BE, MV*). In this visualization, we employ a dimensionality reduction algorithm trained on real data to transform both real and synthetic data into two dimensions with two components. We utilized t-distributed Stochastic Neighbor Embedding (t-SNE) in this study [49]. The coordinates of these two components are then used to generate a paired scatter plot accompanied by density plots. This visualization facilitates rapid visual comparison of different data synthesis algorithms at the multivariate level in a single view. We observe that the synthetic data generated by TTVAE exhibits significantly better overlap with the original dataset than other benchmarks when using t-SNE for visualization on these four large datasets covering both classification and regression problems, followed closely by TabSyn. In contrast, CTABGAN+ exhibits clustering of synthetic data in a specific area, leaving a wide range of the real data uncovered, as indicated by its low coverage score.

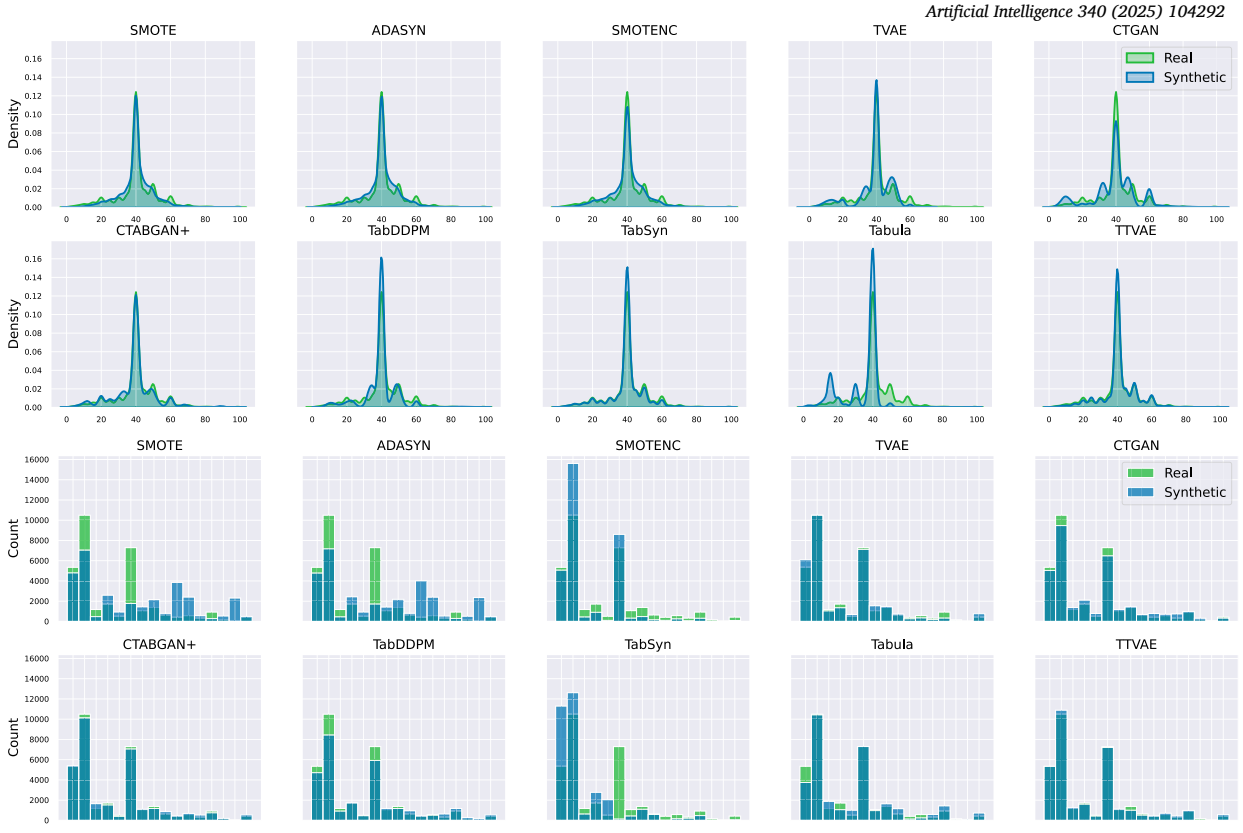

Fig. 3. Visualization of synthetic data's single column distribution v.s. the real data on the *AD* dataset. Upper: numerical column (*hours-in-week)*; Lower: Categorical column (*education*). TTVAE produces the most realistic feature distributions.

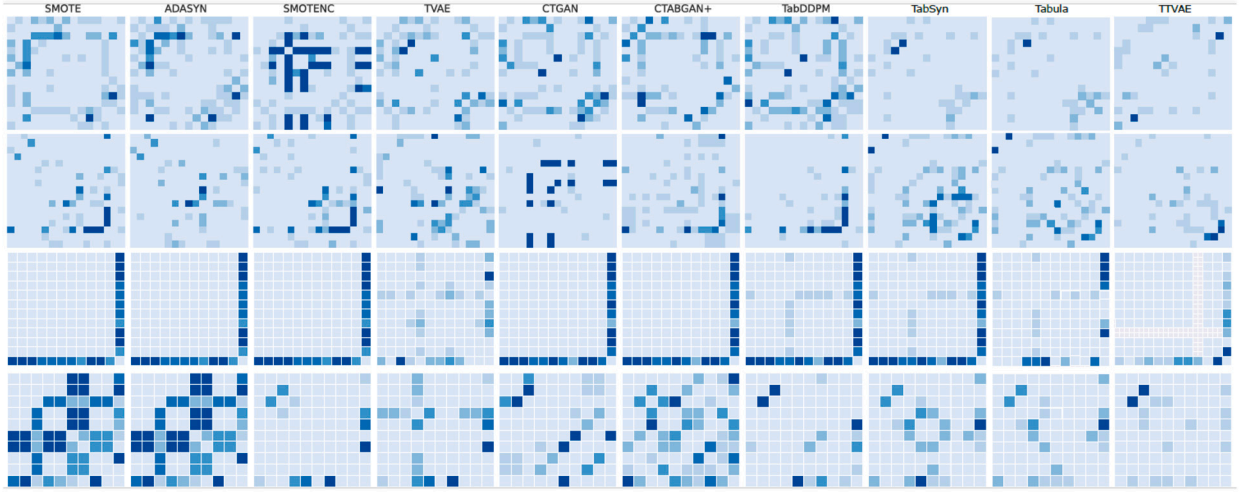

Fig. 4. Heatmaps of the pair-wise column correlation of synthetic data v.s. the real data. The value represents the absolute divergence between the real and estimated correlations (the lighter, the better). TTVAE gives the most accurate column correlation estimation on the *AD, BE, BM* and *MV* dataset.

5.3. Privacy

From a privacy perspective, TTVAE leads in terms of DCR (2.1), indicating that the synthetic data it generates closely resembles the real dataset. This similarity contributes to its exceptional ML utility, as evidenced by the low DCR shown in Fig. 6. Importantly, this low DCR does not come at the expense of duplicating real data, as TTVAE ranks third in the CR (5.4), behind CTABGAN+ and CTGAN (2.2 and 4.6, respectively). However, when considering the DCR rankings of CTABGAN+ and CTGAN (8.1 and 7.5), both models exhibit very high DCR values, indicating that their low duplication rates are achieved at the cost of poor data resemblance.

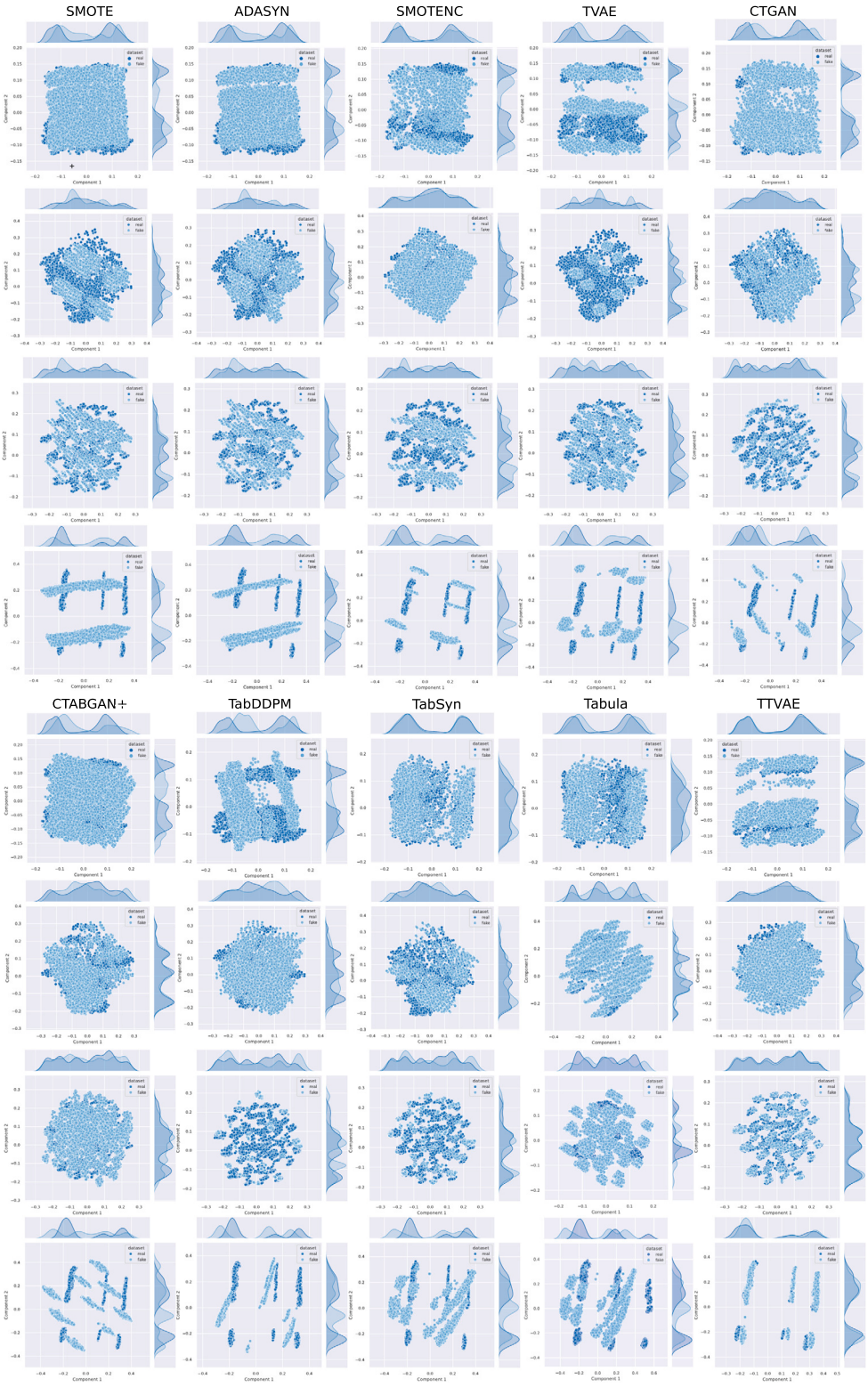

Fig. 5. Two-dimensional t-SNE scatter plot illustrating the distribution of data points, complemented by density plots for the first and second components. Close overlapping in scatter and density plots indicates higher quality synthetic data closely resembling the real data distribution. TTVAE effectively captures the underlying distribution from real data and performs the best for the *AD*, *BE*, *BM*, and *MV* datasets.

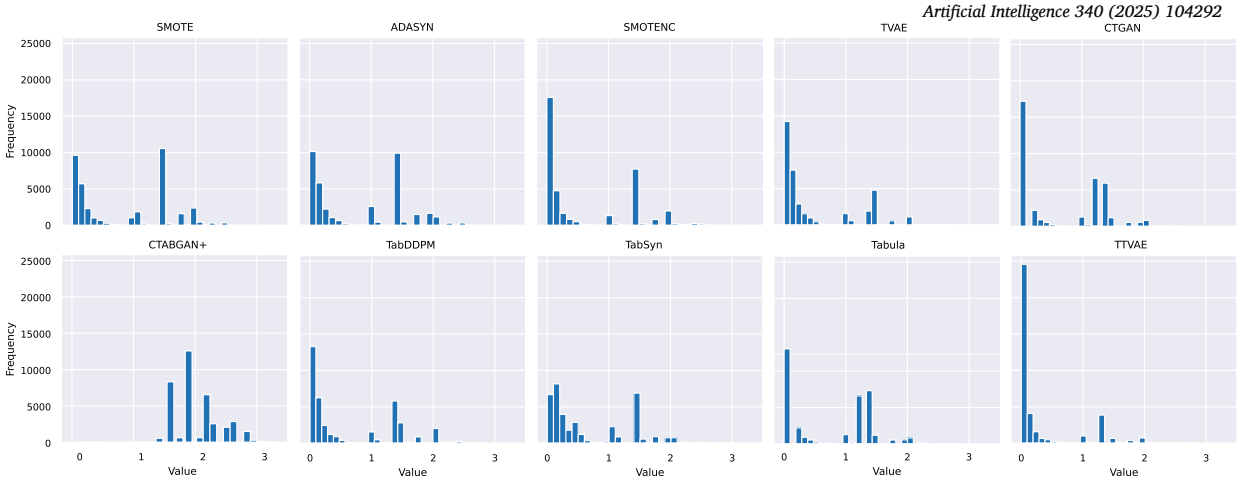

Fig. 6. Distance to closest record (DCR) distributions for the *AD* dataset with respect to the real data. This experiment shows that the proposed method does not "copy" samples from the training set but rather generates new synthetic samples that are close to the original samples.

Table 4
Average Training and Sampling time on *AD* dataset.

Method	Training	Sampling	Method	Training	Sampling
SMOTE	0.94 s	-	ADASYN	2.39 s	-
SMOTENC	82.37 s	-	TVAE	408.81 s	0.52 s
CTGAN	734.68 s	0.92 s	CTABGAN+	789.56 s	0.98 s
TabDDPM	1056.25 s	32.56 s	TabSyn	1958.25 s + 758 s	32.56 s
Tabula	3 h 21 m	3 m 37 s	TTVAE	855.36 s	0.89 s

In contrast, TTVAE ranks in the top position in pMSE (2.4), suggesting that simple ML algorithms can effectively distinguish between real and synthetic data. This finding implies that our synthetic data exhibits sufficient dissimilarity to prevent a straightforward replication of the real data. Overall, these results demonstrate that TTVAE offers a more balanced and effective approach compared to other models. It produces high-quality synthetic data that closely resembles real data without merely replicating it, while also maintaining resilience to privacy attacks.

5.4. Computational efficiency

Table 4 presents the average training and sampling times for the *AD* dataset. As expected, SMOTE, being a non-deep learning model, is much faster and generates data during training. Among the deep generative models, despite producing higher-quality synthetic data, TTVAE shows training times comparable to CTGAN, CTABGAN+, and TabDDPM. It is also worth noting that Tabula, a tabular data synthesizer based on a large language model (LLM) structure, is notably slower and demands significant computational power. While TVAE is faster than TTVAE due to its lower computational complexity, this speed advantage comes at the expense of poorer synthetic data quality. TabSyn, a diffusion-based model and the second-best synthesis algorithm, is also considerably slower than TTVAE. For example, to generate synthetic data of the same size as the training data for *AD*, TabSyn took 32.56 seconds, while TTVAE required only 0.89 seconds. Therefore, we conclude that our proposed TTVAE algorithm offers two key advantages: superior synthetic data generation and greater computational efficiency.

6. Conclusions

In this study, we investigate the potential of Transformer architectures in TDS by introducing the TTVAE design, which effectively manages mixed data types, including numerical and categorical features. By incorporating latent space interpolation, we improve the controllability and interpretability of synthetic data generation. Throughout various benchmark datasets, TTVAE consistently outperforms existing models, including those based on statistical methods, ML, DL, and LLM. We observed that LLM-based tabular synthesis algorithms often require significantly more time and computational power compared to other methods, but their performance does not always justify the additional resources. This issue may stem from the need for specific hyperparameter tuning in LLM-based TDS algorithms, leading to further computational challenges. To address this, exploring ways to enhance computational efficiency by using pre-trained models could be a valuable research direction. Notably, our proposed model outperforms SMOTE and its variants, which are traditionally considered effective baselines. Unlike the shallow interpolation models of SMOTE, TTVAE combines latent space interpolation with the Transformer architecture, enabling it to capture complex feature relationships and greatly improve the quality of synthetic data generation.

CRediT authorship contribution statement

Alex X. Wang: Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Binh P. Nguyen:** Writing – review & editing, Validation, Supervision, Methodology, Funding acquisition, Conceptualization.

Declaration of competing interest

All authors declare that they have no conflicts of interest regarding the publication of this paper.

Acknowledgements

The work of AXW was supported in part by the MBIE MedTech Research Acceleration Programme CoRE RAP1 [3725142 / AA7M].

Appendix A. Additional information and results

A.1. Implementation of TTVAE

We used the same set of parameters for different datasets. The detailed architecture of TTVAE is presented in Section 3 of the main text. Below are the detailed hyperparameter settings.

- Number of layers of Encoder/Decoder: 3,
- Latent space dimension: 32,
- Token dimension: $d = 8$,
- Transformer embedding dimension: 128,
- Number of layers of Transformers: 2,
- Hidden dimension of Transformer's Feed Forwards Network: 1028,
- Dropout rate: 0.1.

A.2. Implementation of baseline synthesis algorithms

We selected nine baseline data synthesis models, each offering a unique approach. To ensure a fair comparison, we adhered closely to the recommendations and original implementations outlined in their respective papers. For SMOTE and its variants, we utilized the imbalanced-learn library.[2] The deep learning models were recreated using their official code, available in public GitHub repositories. The detailed implementations are as follows: CTGAN and TVAE [7] were configured according to their original code[3]; CTABGAN+ [8] was implemented using the original settings[4]; TabDDPM [9] was set up using the implementation from the original paper[5]; TabSyn [40] was built with the official code[6]; and Tabula [41] was constructed based on the provided code.[7]

A.3. Datasets

We used 16 datasets from the UCI Machine Learning Repository[8] and the OpenML Repository.[9] Ten of them are associated with classification tasks, and the remaining six were designed for regression.

- credit-g [50] (OpenML)
- sick [51] (OpenML)
- abalone [52] (UCI)
- wine-quality [53] (UCI)
- parkinsons-telemonitoring [54] (UCI)
- NewspaperChurn [55] (OpenML)
- compass [56] (OpenML)
- law-school-admission [57] (OpenML)
- King [56] (OpenML)

[2] https://github.com/scikit-learn-contrib/imbalanced-learn.
[3] https://github.com/sdv-dev/CTGAN.
[4] https://github.com/Team-TUD/CTAB-GAN-Plus.
[5] https://github.com/yandex-research/tab-ddpm.
[6] https://github.com/amazon-science/tabsyn.
[7] https://github.com/zhao-zilong/Tabula.
[8] https://archive.ics.uci.edu/datasets.
[9] https://www.openml.org.

- nomao [58] (OpenML)
- news [59] (UCI)
- mv [60] (OpenML)
- Beijing [61] (UCI)
- Bank-Marketing [62] (OpenML)
- adult [63] (UCI)
- Census-Income [64] (UCI)

A.4. Additional statistical similarity results

See Tables A.1–A.10.

Table A.1
Total column-wise Kullback–Leibler divergence. **Bold** indicates the best score on each dataset, where lower values are considered better. TTVAE overall ranks at the top and produces consistently low values, indicating that the distribution of synthetic data generated by TTVAE closely aligns with the real distribution.

	CR	SI	AB	WQ	PT	NC	CO	LA	KI	NO	NE	MV	BE	BM	AD	CI	Average	Ranking
SMOTE	0.17	0.08	**0.00**	0.03	0.03	0.04	0.08	0.01	0.03	5.05	0.06	0.23	0.23	0.06	0.08	0.03	0.39	5.2
ADASYN	0.11	0.09	**0.00**	0.02	0.02	0.04	0.08	0.01	0.03	5.05	0.06	0.23	0.23	0.06	0.08	**0.02**	0.38	4.8
SMOTENC	0.21	0.03	0.01	0.03	0.03	0.03	0.02	0.01	0.03	-	**0.05**	0.04	0.04	0.04	0.08	-	0.05	4.1
TVAE	0.34	0.13	0.23	0.09	0.31	0.08	0.13	0.17	0.06	0.45	0.74	0.18	1.27	0.39	0.18	0.08	0.30	8.8
CTGAN	0.08	0.05	0.11	0.07	0.13	0.03	0.03	0.04	0.04	4.23	0.06	0.03	0.04	0.03	0.04	0.04	0.32	5.0
CTABGAN+	**0.03**	0.06	0.05	2.03	0.05	0.06	0.11	0.04	1.35	5.08	0.86	0.07	0.33	0.07	0.05	0.04	0.64	7.1
TabDDPM	0.11	0.03	0.03	0.04	0.02	0.03	0.11	0.01	0.02	0.65	1.03	0.07	0.01	0.03	0.04	0.11	0.15	4.7
TabSyn	0.06	0.04	0.03	0.02	0.02	0.02	0.02	0.01	0.02	0.24	0.10	0.04	0.01	0.02	0.02	0.06	0.05	3.1
Tabula	0.06	**0.02**	0.14	**0.00**	0.60	0.55	0.67	0.11	0.59	6.26	0.06	0.03	**0.00**	0.07	0.07	0.08	0.58	6.1
TTVAE	0.08	0.03	0.02	0.03	**0.01**	**0.01**	**0.01**	**0.00**	**0.01**	**0.23**	0.13	**0.01**	**0.00**	**0.01**	**0.01**	0.06	**0.04**	**2.3**

Table A.2
Total column-wise Jensen-Shannon divergence. **Bold** indicates the best score on each dataset, where lower values are considered better. TTVAE overall ranks at the top and produces consistently low values, indicating that the distribution of synthetic data generated by TTVAE closely aligns with the real distribution.

	CR	SI	AB	WQ	PT	NC	CO	LA	KI	NO	NE	MV	BE	BM	AD	CI	Average	Ranking
SMOTE	0.14	0.07	**0.02**	0.06	**0.03**	0.08	0.10	0.05	0.07	0.20	0.09	0.19	0.09	0.09	0.12	0.07	0.09	5.4
ADASYN	0.13	0.07	**0.02**	0.05	**0.03**	0.08	0.10	0.04	0.07	0.20	0.09	0.19	0.09	0.09	0.12	0.07	0.09	5.1
SMOTENC	0.12	0.05	0.03	0.06	**0.03**	0.05	0.03	0.03	0.07	-	0.10	0.08	0.08	0.07	0.10	-	0.06	4.3
TVAE	0.14	0.07	0.10	0.10	0.13	0.09	0.10	0.10	0.07	0.11	0.16	0.20	0.10	0.11	0.09	0.07	0.11	7.3
CTGAN	0.08	0.08	0.15	0.11	0.12	0.07	0.07	0.10	0.09	0.25	0.08	0.08	0.08	0.06	0.09	0.08	0.10	6.2
CTABGAN+	0.06	0.09	0.10	0.17	0.11	0.08	0.08	0.09	0.16	0.23	0.10	0.12	0.09	0.07	0.08	0.08	0.11	6.7
TabDDPM	0.08	0.09	0.04	0.05	0.05	0.04	0.04	0.03	0.04	0.10	0.18	0.02	0.08	0.03	**0.04**	0.09	0.06	4.2
TabSyn	**0.05**	0.05	0.05	**0.01**	0.05	0.04	**0.01**	0.05	0.06	**0.08**	0.08	0.07	0.06	0.05	0.09	**0.05**	0.05	2.9
Tabula	0.11	0.08	0.09	0.05	0.21	0.25	0.16	0.16	0.09	0.27	**0.06**	**0.01**	0.10	0.07	0.05	0.10	0.12	7.0
TTVAE	0.07	**0.04**	0.04	0.02	0.04	**0.03**	0.02	**0.02**	**0.03**	0.09	0.09	0.02	**0.05**	**0.02**	**0.04**	0.06	**0.04**	**2.1**

Table A.3
Total column-wise Wasserstein distances. **Bold** indicates the best score on each dataset, where lower values are considered better. TTVAE overall ranks at the top and produces consistently low values, indicating that the distribution of synthetic data generated by TTVAE closely aligns with the real distribution.

	CR	SI	AB	WQ	PT	NC	CO	LA	KI	NO	NE	MV	BE	BM	AD	CI	Average	Ranking
SMOTE	0.25	0.05	**0.02**	0.03	**0.02**	0.26	0.12	0.10	0.07	3.32	0.06	0.05	0.14	0.14	0.28	0.15	0.32	5.9
ADASYN	0.24	0.05	**0.00**	0.03	**0.01**	0.24	0.13	0.10	0.07	3.32	0.06	0.05	0.15	0.15	0.28	0.15	0.31	5.6
SMOTENC	0.17	**0.01**	0.05	0.02	**0.01**	0.48	0.05	0.04	0.06	-	0.04	0.07	0.17	0.10	0.37	-	0.12	5.0
TVAE	0.16	0.03	0.17	0.03	0.04	0.98	0.10	0.10	0.04	1.23	0.07	0.13	0.13	0.09	0.24	0.24	0.24	6.3
CTGAN	0.10	0.06	0.23	0.10	0.14	0.46	0.10	0.18	0.07	3.34	0.03	0.07	0.09	0.09	0.36	0.26	0.36	7.1
CTABGAN+	0.09	0.05	0.06	0.14	0.12	0.47	0.11	0.16	0.17	3.33	0.06	0.08	0.08	0.10	0.17	0.23	0.34	6.8
TabDDPM	0.10	0.02	0.07	0.02	0.02	0.24	0.04	0.04	0.04	0.90	0.09	0.02	0.09	0.03	**0.16**	0.42	0.14	4.2
TabSyn	**0.09**	0.04	0.05	**0.01**	0.02	**0.14**	**0.02**	0.09	0.06	**0.89**	0.06	0.02	0.09	0.04	0.17	**0.13**	0.12	3.1
Tabula	0.18	0.04	0.03	**0.00**	0.06	0.64	0.14	0.13	0.13	2.55	**0.02**	**0.00**	**0.06**	**0.02**	0.37	0.23	0.29	5.3
TTVAE	**0.08**	**0.02**	0.07	0.02	0.02	**0.22**	0.03	**0.03**	**0.02**	**0.76**	0.06	0.01	**0.07**	**0.02**	**0.14**	0.16	**0.11**	**2.6**

Table A.4
Pair-wise column correlation score. **Bold** indicates the best score on each dataset, where higher values are considered better. TTVAE consistently ranks at the top and yields consistently high scores, particularly for the large dataset.

	CR	SI	AB	WQ	PT	NC	CO	LA	KI	NO	NE	MV	BE	BM	AD	CI	Average	Ranking
SMOTE	**0.34**	0.00	0.00	0.02	0.14	**0.89**	0.00	0.59	0.40	0.00	**0.01**	0.00	0.13	0.13	0.00	0.09	0.17	4.3
ADASYN	0.02	0.00	**0.01**	0.01	0.08	0.77	0.00	**0.81**	0.31	0.00	**0.01**	0.00	0.19	0.19	0.00	0.11	0.16	4.1
SMOTENC	0.00	0.00	0.00	0.02	0.00	0.01	0.57	0.59	0.00	-	0.00	0.70	0.64	**0.89**	0.23	-	0.26	4.3
TVAE	0.00	0.00	0.00	0.00	0.00	0.13	0.02	0.59	0.31	0.00	0.00	0.08	0.00	0.00	0.62	0.00	0.11	4.9
CTGAN	0.00	0.03	0.00	0.00	0.00	0.01	0.25	0.00	0.02	0.00	0.00	0.04	0.00	0.00	0.04	0.00	0.02	5.9
CTABGAN+	0.00	**0.04**	**0.01**	0.00	0.00	0.01	0.02	0.02	0.00	0.00	0.00	0.70	0.00	0.00	0.47	0.00	0.08	5.1
TabDDPM	0.00	0.00	0.00	0.00	0.05	0.09	0.57	0.59	0.40	0.00	0.00	0.70	0.63	0.63	0.47	0.00	0.26	3.8
TabSyn	0.00	0.00	0.00	0.00	0.09	0.08	0.66	0.76	**0.76**	0.00	0.00	0.89	0.78	0.56	0.65	**0.18**	0.34	2.9
Tabula	0.19	0.00	0.00	**0.16**	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.19	0.03	0.52	0.26	0.10	0.09	5.2
TTVAE	0.00	0.00	0.00	0.00	**0.15**	0.13	**0.70**	**0.81**	0.75	**0.01**	0.00	**0.91**	**0.88**	0.63	**0.78**	0.16	**0.37**	**2.3**

Table A.5
Comparison of density scores. **Bold** represents the best score on each dataset. Higher values indicate better results. TTVAE consistently ranks at the top and produces consistently high scores, suggesting high data fidelity of synthetic data generated by TTVAE.

	CR	SI	AB	WQ	PT	NC	CO	LA	KI	NO	NE	MV	BE	BM	AD	CI	Average	Ranking
SMOTE	0.68	0.40	0.87	0.75	**1.38**	0.16	0.48	0.49	1.09	0.27	0.73	0.00	0.29	1.08	1.16	1.16	0.69	5.4
ADASYN	0.68	0.39	0.87	0.75	1.37	0.16	0.48	0.51	1.08	0.27	0.73	0.00	0.29	**1.09**	1.15	1.20	0.69	5.5
SMOTENC	0.89	0.40	0.73	**1.96**	1.36	0.94	**1.29**	1.93	**1.25**	-	**0.83**	0.00	0.54	0.95	**1.26**	-	1.02	3.6
TVAE	1.09	0.98	0.72	1.47	0.42	1.19	0.93	2.65	1.08	1.05	0.45	0.44	1.11	0.27	0.98	1.03	0.99	5.1
CTGAN	0.75	0.58	0.46	0.54	0.06	0.61	0.52	0.41	0.79	0.74	0.47	0.00	0.78	0.00	0.29	0.54	0.47	7.5
CTABGAN+	0.70	0.44	0.43	0.38	0.07	0.54	0.99	0.46	0.81	0.88	0.10	0.00	0.93	0.01	0.27	0.19	0.45	7.6
TabDDPM	1.10	0.29	0.89	1.41	0.79	1.17	1.18	1.91	1.17	1.08	0.36	0.72	1.14	0.29	1.13	1.06	0.98	4.5
TabSyn	1.12	0.81	0.96	1.15	0.97	1.22	1.11	2.37	1.19	1.12	0.75	0.75	1.18	0.37	1.11	1.14	1.08	3.2
Tabula	0.54	0.00	0.49	0.06	1.09	0.01	0.69	0.29	0.45	1.12	0.00	0.00	1.01	0.04	1.02	1.05	0.49	7.6
TTVAE	**1.14**	**1.33**	**1.02**	1.27	0.79	**1.27**	1.03	**2.82**	1.21	**1.15**	0.38	**0.77**	**1.22**	0.45	1.09	**1.21**	**1.13**	**2.7**

Table A.6
Comparison of coverage scores. **Bold** represents the best score on each dataset. Higher values indicate better results. TTVAE consistently ranks at the top and yields consistently high scores, suggesting a high data diversity in synthetic data generated by TTVAE.

	CR	SI	AB	WQ	PT	NC	CO	LA	KI	NO	NE	MV	BE	BM	AD	CI	Average	Ranking
SMOTE	0.60	0.65	0.78	0.89	0.99	0.36	0.69	0.72	0.96	0.66	0.67	0.00	0.66	0.93	0.73	0.89	0.70	5.9
ADASYN	0.61	0.66	0.79	0.91	**1.00**	0.37	0.70	0.76	0.97	0.67	0.67	0.00	0.67	**0.99**	0.75	**0.94**	0.72	4.5
SMOTENC	**0.97**	0.65	**0.87**	**0.99**	0.98	0.74	0.97	0.95	**0.99**	-	**0.73**	0.00	0.94	0.92	0.94	-	0.83	3.6
TVAE	0.50	**0.84**	0.51	0.79	0.42	0.81	0.61	0.96	0.69	0.24	0.25	0.53	0.61	0.25	0.69	0.68	0.59	7.1
CTGAN	0.77	0.71	0.47	0.74	0.10	0.71	0.67	0.69	0.80	0.76	0.66	0.52	0.67	0.55	0.37	0.35	0.60	7.0
CTABGAN+	0.43	0.66	0.09	0.13	0.13	0.87	0.82	0.74	0.80	0.48	0.22	0.45	0.38	0.53	0.38	0.56	0.48	7.9
TabDDPM	0.75	0.26	0.73	0.92	0.86	0.93	0.98	0.98	0.96	0.97	0.11	0.54	0.98	0.69	0.95	0.90	0.78	4.6
TabSyn	0.63	0.33	0.81	0.96	0.98	0.96	0.98	**0.99**	0.98	0.98	0.25	0.56	0.97	0.69	0.94	0.85	0.80	3.8
Tabula	0.43	0.23	0.30	0.25	0.89	0.35	0.57	0.76	0.85	0.99	0.22	0.55	0.97	0.67	0.95	0.91	0.62	6.6
TTVAE	0.89	0.42	0.82	0.82	0.92	**0.99**	**0.99**	**0.99**	**0.99**	**0.99**	0.61	**0.58**	**0.99**	0.94	**0.97**	0.92	**0.86**	**2.4**

Table A.7
Comparison of maximum mean discrepancy. **Bold** indicates the best score on each dataset, where lower values are considered better. TTVAE consistently ranks at the top and produces consistently low values, indicating that the distribution of synthetic data generated by TTVAE closely aligns with the real distribution.

	CR	SI	AB	WQ	PT	NC	CO	LA	KI	NO	NE	MV	BE	BM	AD	CI	Average	Ranking
SMOTE	0.0027	0.0587	0.0003	0.0181	0.0014	0.0002	0.0012	0.0004	0.0139	0.0046	0.0050	0.0100	0.0046	0.0037	0.0015	0.0006	0.0079	6.0
ADASYN	0.0026	0.0638	**0.0000**	0.0184	**0.0012**	0.0002	0.0011	0.0003	0.0142	0.0047	0.0050	0.0103	0.0047	0.0036	0.0015	0.0006	0.0083	5.8
SMOTENC	0.0034	**0.0053**	0.0050	0.0054	0.0046	0.0002	0.0006	0.0003	0.0142	-	0.0050	0.0276	0.0031	0.0031	0.0041	-	0.0059	5.6
TVAE	0.0046	0.0531	0.0324	0.0134	0.0094	0.0002	0.0031	0.0024	0.0138	0.0009	0.0136	0.1056	0.0024	0.0036	0.0005	0.0021	0.0163	6.9
CTGAN	0.0026	0.0339	0.0603	0.1539	0.0384	0.0002	0.0009	0.0013	0.0056	0.0019	**0.0003**	0.0164	0.0019	0.0014	0.0007	0.0022	0.0201	5.6
CTABGAN+	0.0025	0.0350	0.0110	0.2547	0.0165	0.0002	0.0030	0.0013	0.0416	0.0025	0.0007	0.0555	0.0012	0.0012	0.0010	0.0011	0.0268	6.1
TabDDPM	0.0028	0.0120	0.0026	0.0074	0.0016	0.0002	0.0015	0.0021	0.0026	0.0011	0.0022	**0.0009**	0.0012	0.0005	0.0008	0.0010	0.0025	4.5
TabSyn	0.0026	0.0129	0.0027	0.0065	0.0016	0.0002	0.0009	**0.0001**	0.0025	0.0010	0.0019	0.0011	0.0007	0.0004	0.0006	0.0006	0.0022	2.9
Tabula	0.0025	0.0532	0.0176	**0.0002**	0.0441	0.0052	0.0088	0.0062	0.0216	0.0039	0.0009	0.0011	0.0038	0.0004	0.0031	0.0040	0.0110	6.9
TTVAE	**0.0024**	0.0138	0.0027	0.0056	0.0015	**0.0001**	**0.0003**	0.0002	**0.0023**	**0.0008**	0.0015	0.0013	**0.0002**	**0.0003**	**0.0003**	**0.0005**	**0.0021**	**2.1**

Table A.8
Comparison of distance to the closest record (DCR). **Bold** represents the best score on each dataset, where lower values are considered better. TTVAE consistently ranks at the top and produces consistently low values, indicating that the synthetic data generated by TTVAE closely resemble the real data.

	CR	SI	AB	WQ	PT	NC	CO	LA	KI	NO	NE	MV	BE	BM	AD	CI	Average	Ranking
SMOTE	2.63	0.25	0.04	**0.09**	0.09	2.18	0.69	0.36	**0.24**	9.57	1.19	0.18	2.18	0.50	0.87	1.69	1.42	5.4
ADASYN	2.60	0.24	**0.03**	**0.09**	**0.08**	2.16	0.68	0.33	0.25	9.58	1.19	0.18	2.16	0.49	0.86	1.69	1.41	4.9
SMOTENC	2.40	0.18	0.06	0.11	0.10	1.93	0.45	0.35	0.28	-	**1.16**	0.14	1.93	0.44	0.63	-	0.73	4.4
TVAE	2.36	0.33	0.27	0.18	0.24	2.33	0.83	0.70	0.41	1.34	1.49	0.75	1.34	0.77	0.52	1.48	0.96	7.1
CTGAN	2.72	0.23	0.23	0.19	0.44	2.40	0.70	0.51	0.26	9.63	1.21	0.16	2.72	0.35	0.54	2.32	1.54	7.5
CTABGAN+	2.68	0.22	0.18	0.61	0.32	2.50	0.83	0.49	0.79	9.58	1.53	0.49	2.68	0.40	0.67	2.24	1.64	8.1
TabDDPM	1.81	0.17	0.10	0.13	0.16	1.48	0.28	**0.24**	0.26	1.38	1.70	0.09	1.38	0.26	0.43	1.45	0.71	4.1
TabSyn	1.75	0.16	0.09	0.13	0.16	1.42	0.27	**0.24**	0.25	1.36	1.24	0.09	**1.21**	0.25	**0.39**	1.37	0.65	2.8
Tabula	2.64	0.19	0.05	0.12	0.27	2.31	0.85	0.78	0.32	5.70	1.35	0.10	2.88	0.55	0.85	1.56	1.28	6.9
TTVAE	**1.69**	**0.14**	0.08	0.12	0.16	**1.36**	**0.26**	**0.24**	**0.24**	**1.33**	1.33	**0.08**	1.23	**0.23**	0.40	**1.35**	**0.64**	**2.1**

Table A.9
Comparison of Copy Ratio (CR). **Bold** represents the best score on each dataset, where lower values are considered better.

	CR	SI	AB	WQ	PT	NC	CO	LA	KI	NO	NE	MV	BE	BM	AD	CI	Average	Ranking
SMOTE	0.08	0.06	0.09	0.07	0.09	0.12	0.03	0.05	0.07	0.13	0.10	0.09	0.06	0.10	0.12	0.10	0.09	6.1
ADASYN	0.05	0.05	0.11	0.08	0.11	0.05	0.10	0.06	0.04	0.05	0.11	0.10	0.10	0.04	0.07	0.03	0.07	6.0
SMOTENC	0.09	0.08	0.05	0.08	0.06	0.08	0.09	0.08	0.09	-	0.08	0.04	0.10	0.02	0.09	-	0.07	6.4
TVAE	0.11	0.11	0.09	0.10	**0.01**	0.08	0.03	0.11	0.08	0.08	0.07	0.07	0.07	0.08	0.07	0.02	0.07	6.0
CTGAN	0.02	0.09	**0.02**	**0.04**	0.10	0.04	0.09	**0.01**	**0.01**	0.18	0.08	0.02	0.10	0.06	0.08	0.08	0.06	4.6
CTABGAN+	**0.02**	**0.04**	0.04	0.05	0.09	**0.03**	0.04	0.05	0.07	**0.01**	**0.02**	**0.01**	**0.02**	**0.01**	**0.00**	**0.00**	**0.03**	**2.2**
TabDDPM	0.12	0.08	0.10	0.10	0.11	0.05	**0.03**	0.05	0.07	0.19	0.09	0.12	0.03	0.01	0.02	0.09	0.08	6.2
TabSyn	0.10	0.09	0.10	0.07	0.02	0.05	0.04	0.06	0.03	0.05	0.12	0.02	0.12	0.11	0.06	0.12	0.07	5.9
Tabula	0.12	0.11	0.08	0.04	0.08	0.04	0.08	0.04	0.06	0.18	0.08	0.03	0.10	0.07	0.09	0.10	0.08	5.8
TTVAE	0.08	0.06	0.06	0.06	0.03	0.10	0.06	0.07	0.11	0.14	0.03	0.06	0.09	0.02	0.08	0.09	0.07	5.4

Table A.10
Comparison of propensityMSE. **Bold** represents the best score on each dataset, where lower values are considered better. TTVAE consistently ranks at the top and produces consistently low values, suggesting that the synthetic data generated by TTVAE is challenging to be distinguished by ML models.

	CR	SI	AB	WQ	PT	NC	CO	LA	KI	NO	NE	MV	BE	BM	AD	CI	Average	Ranking
SMOTE	0.64	0.57	0.43	0.50	0.40	0.62	0.75	0.39	0.64	1.00	0.80	0.82	0.88	0.70	0.83	0.69	0.67	5.1
ADASYN	0.61	0.55	**0.41**	**0.48**	**0.39**	0.62	0.74	**0.34**	0.62	1.00	**0.79**	0.82	0.79	0.69	0.82	0.68	0.65	4.0
SMOTENC	0.61	**0.44**	0.43	0.53	0.40	0.51	0.51	0.39	0.62	-	0.80	0.51	0.86	0.59	0.64	-	0.56	3.4
TVAE	0.68	0.61	0.71	0.70	0.83	0.69	0.74	0.67	0.80	0.99	0.99	0.89	0.92	0.89	0.69	0.75	0.78	7.4
CTGAN	0.65	0.83	0.87	0.77	0.96	0.60	0.72	0.59	0.87	1.00	0.99	0.62	0.99	0.62	0.89	0.98	0.81	7.6
CTABGAN+	0.57	0.77	0.82	0.99	0.88	0.75	0.73	0.59	0.97	1.00	1.00	0.72	1.00	0.72	0.67	0.91	0.82	7.7
TabDDPM	0.49	0.50	0.57	0.61	0.60	0.50	0.43	0.42	0.74	1.00	0.99	0.44	0.99	0.42	0.40	0.94	0.63	4.9
TabSyn	0.47	0.48	0.56	0.51	0.52	0.47	0.40	0.36	0.61	0.99	0.85	0.42	**0.65**	0.40	0.38	**0.45**	0.53	2.6
Tabula	0.76	0.54	0.49	0.65	0.90	0.97	0.89	0.86	0.90	0.99	0.99	0.41	0.99	0.79	0.97	0.98	0.82	7.4
TTVAE	**0.45**	0.46	0.54	0.59	0.60	**0.44**	**0.37**	0.42	**0.60**	**0.98**	0.90	**0.39**	0.69	**0.38**	**0.36**	0.51	**0.54**	**2.4**

Data availability

Data will be made available on request.

References

[1] V. Borisov, T. Leemann, K. Seßler, J. Haug, M. Pawelczyk, G. Kasneci, Deep neural networks and tabular data: a survey, IEEE Trans. Neural Netw. Learn. Syst. 35 (6) (2024) 7499–7519, https://doi.org/10.1109/TNNLS.2022.3229161.

[2] R. Venugopal, N. Shafqat, I. Venugopal, B.M.J. Tillbury, H.D. Stafford, A. Bourazeri, Privacy preserving generative adversarial networks to model electronic health records, Neural Netw. 153 (2022) 339–348, https://doi.org/10.1016/j.neunet.2022.06.022.

[3] M. Hernandez, G. Epelde, A. Alberdi, R. Cilla, D. Rankin, Synthetic data generation for tabular health records: a systematic review, Neurocomputing 493 (2022) 28–45, https://doi.org/10.1016/j.neucom.2022.04.053.

[4] S. James, C. Harbron, J. Branson, M. Sundler, Synthetic data use: exploring use cases to optimise data utility, Discov. Artif. Intell. 1 (1) (2021), https://doi.org/10.1007/s44163-021-00016-y.

[5] R. Shwartz-Ziv, A. Armon, Tabular data: deep learning is not all you need, Inf. Fusion 81 (2022) 84–90, https://doi.org/10.1016/j.inffus.2021.11.011.

[6] S.B. Rabbani, M.D. Samad, Between-sample relationship in learning tabular data using graph and attention networks, in: 2023 Congress in Computer Science, Computer Engineering & Applied Computing (CSCE), IEEE, 2023, pp. 1498–1504.

[7] L. Xu, M. Skoularidou, A. Cuesta-Infante, K. Veeramachaneni, Modeling tabular data using conditional GAN, in: Proceedings of the 33rd International Conference on Neural Information Processing Systems (NeurIPS 2019), Curran Associates Inc., Vancouver, Canada, 2019, pp. 7335–7345, https://proceedings.neurips.cc/paper_files/paper/2019/file/254ed7d2de3b23ab10936522dd547b78-Paper.pdf.

[8] Z. Zhao, A. Kunar, R. Birke, L.Y. Chen, CTAB-GAN+: enhancing tabular data synthesis, https://doi.org/10.48550/ARXIV.2204.00401, 2022.

[9] A. Kotelnikov, D. Baranchuk, I. Rubachev, A. Babenko, TabDDPM: modelling tabular data with diffusion models, in: A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, J. Scarlett (Eds.), Proceedings of the 40th International Conference on Machine Learning, in: Proceedings of Machine Learning Research, PMLR, vol. 202, 2023, pp. 17564–17579, https://proceedings.mlr.press/v202/kotelnikov23a.html.

[10] M. Villaizán-Vallelado, M. Salvatori, B. Carro, A.J. Sanchez-Esguevillas, Graph neural network contextual embedding for deep learning on tabular data, Neural Netw. 173 (2024) 106180, https://doi.org/10.1016/j.neunet.2024.106180.

[11] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, SMOTE: synthetic minority over-sampling technique, J. Artif. Intell. Res. 16 (2002) 321–357, https://doi.org/10.1613/jair.953.

[12] J. Kim, C. Lee, Y. Shin, S. Park, M. Kim, N. Park, J. Cho, SOS: score-based oversampling for tabular data, in: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, ACM, 2022, pp. 762–772.

[13] G. Badaro, M. Saeed, P. Papotti, Transformers for tabular data representation: a survey of models and applications, Trans. Assoc. Comput. Linguist. 11 (2023) 227–249, https://doi.org/10.1162/tacl_a_00544.

[14] D. Dablain, B. Krawczyk, N.V. Chawla, DeepSMOTE: fusing deep learning and SMOTE for imbalanced data, IEEE Trans. Neural Netw. Learn. Syst. (2022) 1–15, https://doi.org/10.1109/TNNLS.2021.3136503.

[15] T.E. Raghunathan, Synthetic data, Annu. Rev. Stat. Appl. 8 (1) (2021) 129–140, https://doi.org/10.1146/annurev-statistics-040720-031848.

[16] A.X. Wang, S.S. Chukova, A. Sporle, B.J. Milne, C.R. Simpson, B.P. Nguyen, Enhancing public research on citizen data: an empirical investigation of data synthesis using Statistics New Zealand's integrated data infrastructure, Inf. Process. Manag. 61 (1) (2024) 103558, https://doi.org/10.1016/j.ipm.2023.103558.

[17] S.A. Assefa, D. Dervovic, M. Mahfouz, R.E. Tillman, P. Reddy, M. Veloso, Generating synthetic data in finance: opportunities, challenges and pitfalls, in: Proceedings of the First ACM International Conference on AI in Finance, 2020, pp. 1–8.

[18] M.J. Chong, D. Forsyth, Effectively unbiased FID and inception score and where to find them, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2020), IEEE, 2020, pp. 6069–6078.

[19] A.X. Wang, S.S. Chukova, B.P. Nguyen, Synthetic minority oversampling using edited displacement-based k-nearest neighbors, Appl. Soft Comput. 148 (2023) 110895, https://doi.org/10.1016/j.asoc.2023.110895.

[20] A. Lampis, E. Lomurno, M. Matteucci, Bridging the gap: enhancing the utility of synthetic data via post-processing techniques, https://doi.org/10.48550/ARXIV.2305.10118, 2023.

[21] D.P. Kingma, M. Welling, Auto-encoding variational Bayes, in: International Conference on Learning Representations (ICLR), 2014, pp. 1–14.

[22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. u. Kaiser, I. Polosukhin, Attention is all you need, in: I. Guyon, U.V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems, vol. 30, Curran Associates, Inc., 2017, https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[23] D. Zhang, L. Wang, X. Dai, S. Jain, J. Wang, Y. Fan, C.-C.M. Yeh, Y. Zheng, Z. Zhuang, W. Zhang, FATA-Trans: field and time-aware transformer for sequential tabular data, in: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, 2023, pp. 3247–3256.

[24] J. Yan, J. Chen, Y. Wu, D.Z. Chen, J. Wu, T2G-FORMER: organizing tabular features into relation graphs promotes heterogeneous feature interaction, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, 2023, pp. 10720–10728.

[25] W. Song, C. Shi, Z. Xiao, Z. Duan, Y. Xu, M. Zhang, J. Tang, Autoint: automatic feature interaction learning via self-attentive neural networks, in: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, 2019, pp. 1161–1170.

[26] X. Huang, A. Khetan, M. Cvitkovic, Z. Karnin, TabTransformer: tabular data modeling using contextual embeddings, https://doi.org/10.48550/ARXIV.2012.06678, 2020.

[27] Y. Gorishniy, I. Rubachev, V. Khrulkov, A. Babenko, Revisiting deep learning models for tabular data, in: M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, J.W. Vaughan (Eds.), Advances in Neural Information Processing Systems, vol. 34, Curran Associates, Inc., 2021, pp. 18932–18943, https://proceedings.neurips.cc/paper_files/paper/2021/file/9d86d83f925f2149e9edb0ac3b49229c-Paper.pdf.

[28] A. Gretton, K.M. Borgwardt, M.J. Rasch, B. Schölkopf, A. Smola, A kernel two-sample test, J. Mach. Learn. Res. 13 (25) (2012) 723–773, http://jmlr.org/papers/v13/gretton12a.html.

[29] S. Zhao, J. Song, S. Ermon, InfoVAE: balancing learning and inference in variational autoencoders, in: Proceedings of the AAAI Conference on Artificial Intelligencem vol. 33, Association for the Advancement of Artificial Intelligence (AAAI), 2019, pp. 5885–5892.

[30] M. Arbel, A. Korba, A. Salim, A. Gretton, Maximum mean discrepancy gradient flow, in: H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, R. Garnett (Eds.), Advances in Neural Information Processing Systems, vol. 32, Curran Associates, Inc., 2019, https://proceedings.neurips.cc/paper_files/paper/2019/file/944a5ae3483ed5c1e10bbccb7942a279-Paper.pdf.

[31] L. Struski, M. Sadowski, T. Danel, J. Tabor, I.T. Podolak, Feature-based interpolation and geodesics in the latent spaces of generative models, IEEE Trans. Neural Netw. Learn. Syst. 35 (9) (2024) 12068–12082, https://doi.org/10.1109/tnnls.2023.3251848.

[32] J. Fonseca, F. Bacao, Tabular and latent space synthetic data generation: a literature review, J. Big Data 10 (1) (2023) 115, https://doi.org/10.1186/s40537-023-00792-7.

[33] H. Zhang, M. Cisse, Y.N. Dauphin, D. Lopez-Paz, Mixup: beyond empirical risk minimization, in: International Conference on Learning Representations (ICLR), 2018, pp. 1–13.

[34] C. Beckham, S. Honari, V. Verma, A.M. Lamb, F. Ghadiri, R.D. Hjelm, Y. Bengio, C. Pal, On adversarial mixup resynthesis, in: H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, R. Garnett (Eds.), Advances in Neural Information Processing Systems, vol. 32, Curran Associates, Inc., 2019, https://proceedings.neurips.cc/paper_files/paper/2019/file/f708f064faaf32a43e4d3c784e6af9ea-Paper.pdf.

[35] A.X. Wang, S.S. Chukova, C.R. Simpson, B.P. Nguyen, Challenges and opportunities of generative models on tabular data, Appl. Soft Comput. 166 (2024) 112223, https://doi.org/10.1016/j.asoc.2024.112223.

[36] B. Nowok, G.M. Raab, C. Dibben, Synthpop: bespoke creation of synthetic data in R, J. Stat. Softw. 74 (11) (2016) 1–26, https://doi.org/10.18637/jss.v074.i11.

[37] A. Zhang, H. Yu, Z. Huan, X. Yang, S. Zheng, S. Gao, SMOTE-RkNN: a hybrid re-sampling method based on SMOTE and reverse k-nearest neighbors, Inf. Sci. 595 (2022) 70–88, https://doi.org/10.1016/j.ins.2022.02.038.

[38] H. He, Y. Bai, E.A. Garcia, S. Li, ADASYN: adaptive synthetic sampling approach for imbalanced learning, in: IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), IEEE, 2008, pp. 1322–1328.

[39] M. Mukherjee, M. Khushi, SMOTE-ENC: a novel SMOTE-based method to generate synthetic data for nominal and continuous features, Appl. Syst. Innov. 4 (1) (2021) 18, https://doi.org/10.3390/asi4010018.

[40] H. Zhang, J. Zhang, B. Srinivasan, Z. Shen, X. Qin, C. Faloutsos, H. Rangwala, G. Karypis, Mixed-type tabular data synthesis with score-based diffusion in latent space, in: International Conference on Learning Representations (ICLR), 2023, pp. 1–29.

[41] Z. Zhao, R. Birke, L. Chen, TabuLa: harnessing language models for tabular data synthesis, https://doi.org/10.48550/ARXIV.2310.12746, 2023.

[42] A. Alaa, B. Van Breugel, E.S. Saveliev, M. van der Schaar, How faithful is your synthetic data? Sample-level metrics for evaluating and auditing generative models, in: K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, S. Sabato (Eds.), Proceedings of the 39th International Conference on Machine Learning, in: Proceedings of Machine Learning Research, PMLR, vol. 162, 2022, pp. 290–306, https://proceedings.mlr.press/v162/alaa22a.html.

[43] Z. Zhao, R. Birke, L.Y. Chen, FCT-GAN: enhancing global correlation of table synthesis via Fourier transform, in: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, 2023, pp. 4450–4454.
[44] V.S. Chundawat, A.K. Tarun, M. Mandal, M. Lahoti, P. Narang, A universal metric for robust evaluation of synthetic tabular data, IEEE Trans. Artif. Intell. 5 (1) (2024) 300–309, https://doi.org/10.1109/tai.2022.3229289.
[45] L. Theis, A. van den Oord, M. Bethge, A note on the evaluation of generative models, in: International Conference on Learning Representations (ICLR 2016), 2016, pp. 1–10.
[46] M.F. Naeem, S.J. Oh, Y. Uh, Y. Choi, J. Yoo, Reliable fidelity and diversity metrics for generative models, in: H.D. III, A. Singh (Eds.), Proceedings of the 37th International Conference on Machine Learning, in: Proceedings of Machine Learning Research, PMLR, vol. 119, 2020, pp. 7176–7185, https://proceedings.mlr.press/v119/naeem20a.html.
[47] J. Lee, J. Hyeong, J. Jeon, N. Park, J. Cho, Invertible tabular GANs: killing two birds with one stone for tabular data synthesis, in: M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, J.W. Vaughan (Eds.), Advances in Neural Information Processing Systems, vol. 34, Curran Associates, Inc., 2021, pp. 4263–4273, https://proceedings.neurips.cc/paper_files/paper/2021/file/22456f4b545572855c766df5eefc9832-Paper.pdf.
[48] S. Hediger, L. Michel, J. Näf, On the use of random forest for two-sample testing, Comput. Stat. Data Anal. 170 (2022) 107435, https://doi.org/10.1016/j.csda.2022.107435.
[49] T. Liu, Z. Qian, J. Berrevoets, M. van der Schaar, GOGGLE: generative modelling for tabular data by learning relational structure, in: International Conference on Learning Representations (ICLR 2023), 2023, pp. 1–22, https://openreview.net/forum?id=fPVRcJqspu.
[50] H. Hofmann, Statlog (German Credit Data), UCI Machine Learning Repository, https://doi.org/10.24432/C5NC77, 1994.
[51] R. Quinlan, Thyroid Disease, UCI Machine Learning Repository, https://doi.org/10.24432/C5D010, 1986.
[52] S. Nash Warwick, W. Ford, Abalone, UCI Machine Learning Repository, https://doi.org/10.24432/C55C7W, 1994.
[53] P. Cortez, J. Reis, Wine Quality, UCI Machine Learning Repository, https://doi.org/10.24432/C56S3T, 2009.
[54] A. Tsanas, M. Little, Parkinsons Telemonitoring, UCI Machine Learning Repository, https://doi.org/10.24432/C5ZS3N, 2009.
[55] J. Vanschoren, J.N. van Rijn, B. Bischl, L. Torgo, OpenML: networked science in machine learning, ACM SIGKDD Explor. Newsl. 15 (2) (2014) 49–60, https://doi.org/10.1145/2641190.2641198.
[56] L. Grinsztajn, E. Oyallon, G. Varoquaux, Why do tree-based models still outperform deep learning on typical tabular data? in: S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, A. Oh (Eds.), Advances in Neural Information Processing Systems, vol. 35, Curran Associates, Inc., 2022, pp. 507–520, https://proceedings.neurips.cc/paper_files/paper/2022/file/0378c7692da36807bdec87ab043cdadc-Paper-Datasets_and_Benchmarks.pdf.
[57] R.H. Sander, A systemic analysis of affirmative action in American law schools, Stanf. Law Rev. 57 (2) (2004) 367–483, http://www.jstor.org/stable/40040209.
[58] L. Candillier, V. Lemaire, Nomao, UCI Machine Learning Repository, https://doi.org/10.24432/C53G79, 2012.
[59] V. Fernandes, Kelwin, S. Pedro, Online News Popularity, UCI Machine Learning Repository, https://doi.org/10.24432/C5NS3V, 2015.
[60] L. Torgo, Functional models for regression tree leaves, in: Proceedings of the Fourteenth International Conference on Machine Learning, 1997, pp. 385–393.
[61] S. Chen, Beijing PM2.5, UCI Machine Learning Repository, https://doi.org/10.24432/C5JS49, 2015.
[62] S. Moro, P. Cortez, Bank Marketing, UCI Machine Learning Repository, https://doi.org/10.24432/C5K306, 2014.
[63] B. Becker, R. Kohavi, Adult, UCI Machine Learning Repository, https://doi.org/10.24432/C5XW20, 1996.
[64] Census-Income (KDD), UCI Machine Learning Repository, https://doi.org/10.24432/C5N30T, 2000.