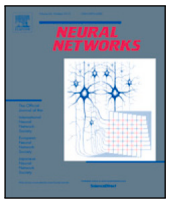
Neural Networks 185 (2025) 107208

Contents lists available at ScienceDirect

Neural Networks

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



Full Length Article

Deterministic Autoencoder using Wasserstein loss 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:
Deep neural networks
Tabular data synthesis
Latent space interpolation
Generative AI
Wasserstein Autoencoder

ABSTRACT

Tabular data generation is a complex task due to its distinctive characteristics and inherent complexities. While Variational Autoencoders have been adapted from the computer vision domain for tabular data synthesis, their reliance on non-deterministic latent space regularization introduces limitations. The stochastic nature of Variational Autoencoders can contribute to collapsed posteriors, yielding suboptimal outcomes and limiting control over the latent space. This characteristic also constrains the exploration of latent space interpolation. To address these challenges, we present the Tabular Wasserstein Autoencoder (TWAE), leveraging the deterministic encoding mechanism of Wasserstein Autoencoders. This characteristic facilitates a deterministic mapping of inputs to latent codes, enhancing the stability and expressiveness of our model's latent space. This, in turn, enables seamless integration with shallow interpolation mechanisms like the synthetic minority over-sampling technique (SMOTE) within the data generation process via deep learning. Specifically, TWAE is trained once to establish a low-dimensional representation of real data, and various latent interpolation methods efficiently generate synthetic latent points, achieving a balance between accuracy and efficiency. Extensive experiments consistently demonstrate TWAE's superiority, showcasing its versatility across diverse feature types and dataset sizes. This innovative approach, combining WAE principles with shallow interpolation, effectively leverages SMOTE's advantages, establishing TWAE as a robust solution for complex tabular data synthesis.

1. Introduction

The demand for high-quality tabular data is substantial, given its standing as one of the most prevalent data modalities across diverse fields (Wang, Chukova, Sporle et al., 2024). With its structured format, comprising rows and columns, tabular data serves as a fundamental choice for representing information in various data storage systems encountered in daily life. Despite its popularity, the challenges of data scarcity persist from multiple perspectives, including the lack of big data, the barrier of collecting specific classes of data, and privacy concerns that limit data accessibility (Borisov et al., 2022). Synthetic data emerges as a valuable solution to overcome these challenges, particularly when additional data is required, specific classes are needed, or privacy concerns restrict access to real datasets (James, Harbron, Branson, & Sundler, 2021).

Training generative models for tabular data is challenging compared to other domains, like computer vision. Recent literature highlights the dominance of traditional machine learning (ML) methods over deep learning (DL) techniques in this context, emphasizing the complexities inherent in tabular data, such as heterogeneous data types and a lack of inherent structure (Shwartz-Ziv & Armon, 2022). Thus, creating an effective DL framework for tabular data remains a significant challenge (Rabbani & Samad, 2023). Models like Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs) typically use stochastically generated latent variables, which limits control over the latent space and, in turn, affects the quality of the generated data (Gai & Zhang, 2023). Despite limited comparisons with simple interpolation models like the synthetic minority over-sampling technique (SMOTE) (Chawla, Bowyer, Hall, & Kegelmeyer, 2002), available evidence suggests that SMOTE is often challenging to outperform (Kim et al., 2022). This highlights the need for innovative DL approaches to address the unique challenges of tabular data synthesis (TDS).

Our study explores the use of the Wasserstein Autoencoder (WAE) for TDS, addressing key challenges in this domain. WAE offers two distinct advantages. First, it provides a deterministic latent space, unlike traditional GANs and VAEs, which improves stability and control (Tolstikhin, Bousquet, Gelly, & Schoelkopf, 2018). Second, this structured latent space enables *latent mixup* (Beckham et al., 2019), allowing interpolation directly within the latent space instead of the input space. This leads to the generation of more realistic and coherent synthetic data by preserving important features and creating

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

https://doi.org/10.1016/j.neunet.2025.107208
Received 26 July 2024; Received in revised form 19 December 2024; Accepted 22 January 2025
Available online 29 January 2025

smoother transitions between samples (Struski, Sadowski, Danel, Tabor, & Podolak, 2023). In this study, we employ SMOTE for latent space interpolation (Dablain, Krawczyk, & Chawla, 2022) due to its ability to generate high-quality synthetic data with minimal computational cost. This method is effective only after the heterogeneous data in tabular format has been transformed into numerical values through the deterministic and unified latent space established by the WAE. These combined mechanisms form the core of our proposed model, the Tabular Wasserstein Autoencoder (TWAE), which is specifically designed for the challenges of TDS. Our experiments demonstrate that TWAE consistently outperforms existing methods across various benchmark datasets. The main contributions of our study are: (1) a comprehensive evaluation of leading tabular data generation models, assessing their performance and computational efficiency across different tasks and datasets; (2) the introduction of TWAE, a model optimized for handling diverse tabular data types; (3) demonstrating TWAE's superior performance compared to traditional statistical methods, ML algorithms, and DL-based models; and (4) highlighting the effectiveness of shallow interpolation techniques like SMOTE, which generate high-quality synthetic data with minimal computational resources, with TWAE showing particularly strong results on large datasets.

2. Related work

2.1. Tabular data synthesis

Data generation involves creating synthetic data points that closely resemble real data (Raghunathan, 2021). This serves several purposes, including augmenting datasets, addressing data scarcity, enhancing privacy, and training/testing ML models (Wang, Chukova, Sporle et al., 2024). Generative algorithms learn patterns from a dataset to generate new samples with similar characteristics. Mathematically, consider a dataset of examples, $\{x_i | x_i \in \mathbb{R}, i = 1, \ldots, N\}$, sampled from a true data distribution, $\mathbb{P}$. The objective of a Deep Generative Model (DGM) is to construct deep neural networks (DNN) with parameters θ to define a distribution $p_\theta(x)$. The parameters θ are trained to ensure that the distribution $p_\theta(x)$ closely aligns with the true data distribution $\mathbb{P}$, allowing $p_\theta(x)$ to generate synthetic data.

Generating tabular data has become crucial due to its increasing use in solving business problems, but it poses several challenges. First, tabular data collection is often diverse and manual, coming from various systems, leading to arbitrary dataset lengths, inconsistent features, and missing values. Second, the mix of numerical and categorical data complicates ML model training and evaluation. Encoding categorical variables into numeric formats adds complexity, especially when features come from unrelated sources with different units. Additionally, conventional metrics like the "Inception Score", designed for 2D image data, may not effectively assess synthetic tabular data quality (Chong & Forsyth, 2020). Lastly, generating synthetic tabular data often requires specialized preprocessing and postprocessing to address specific business questions. It is important to distinguish between synthetic data created to enhance ML models and data created for privacy preservation, as each involves different trade-offs between data accuracy and privacy (Lampis, Lomurno, & Matteucci, 2023). Addressing these challenges in generative modeling for tabular data remains an open research area.

Traditional TDS models, like perturbation-based techniques and statistical methods, often struggle with modern datasets and scalability, leading to greater interest in DGM (Wang, Chukova, Simpson and Nguyen, 2024). CTGAN (Xu, Skoularidou, Cuesta-Infante, & Veeramachaneni, 2019) improves upon standard GANs to handle complex distributions through Mode-specific Normalization and conditional vectors. Variants such as CTAB-GAN (Zhao, Kunar, Birke, & Chen, 2021) and CopulaGAN (Gupta, Bhatt, & Pandey, 2021) further enhance these methods. TVAE, built on VAE, learns latent representations for effective sample generation. Diffusion models, including STaSy (Kim, Lee, & Park, 2023) and TabDDPM (Kotelnikov, Baranchuk, Rubachev, & Babenko, 2023), use Stochastic Differential Equations and denoising techniques for better sampling and efficiency, while TABSYN (Zhang et al., 2024) integrates VAE and Transformer architectures to boost data quality and synthesis speed. Large language models (LLMs) such as GReaT (Borisov, Seßler, Leemann, Pawelczyk, & Kasneci, 2023), REaLTabFormer (Solatorio & Dupriez, 2023), and Tabula (Zhao, Birke, & Chen, 2023b) represent state-of-the-art synthesizers. GReaT samples data by permuting feature orders, while REaLTabFormer specializes in synthesizing relational tables. Tabula aims to reduce the prohibitive computational requirements of the other two models, improving training times and overall performance. While GANs and VAEs are faster and require less tuning, they often struggle with data quality. Advanced models such as diffusion methods and LLMs require significant computational resources, and quality may suffer if power is constrained. To address these issues, our study aims to use WAE for better data and incorporate *latent mixup* techniques to enhance synthesis. Together, these approaches aim to maximize data quality while minimizing computational demands.

2.2. VAE

A VAE is a probabilistic generative model designed for learning latent representations of data (Kingma & Welling, 2014). It comprises two main elements: an encoder network and a decoder network. Let x represent the observed data and z the latent variable. While the encoder function $q_\phi(z|x)$ estimates the posterior distribution of the latent variable given the data, the decoder function $p_\theta(x|z)$ approximates 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 VAE is to maximize the Evidence Lower Bound (ELBO) on the log-likelihood of the observed data:

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

The VAE loss function is typically expressed as the negative ELBO:

$$\mathcal{L}_{\mathrm{VAE}} = -\mathbb{E}_{q(z|x)}[\log p(x|z)] + \mathrm{KL}[q(z|x) \parallel p(z)], \tag{2}$$

where there are two main components: $\mathbb{E}_{q(z|x)}[\log p(x|z)]$ is the reconstruction term, and $\mathrm{KL}[q(z|x) \parallel p(z)]$ is the Kullback–Leibler (KL) divergence as a regularization term. This loss function is designed to balance the trade-off between accurately reconstructing the input data and regularizing the latent space. However, using Kullback–Leibler (KL) divergence as a regularization term in generative models, especially VAEs, has several issues. Firstly, KL divergence can lead to over-regularization and mode collapse, where the model overly restricts the latent space (Tolstikhin et al., 2018). It is also sensitive to the choice of the prior distribution and assumes a specific form, limiting its ability to capture complex latent structures (Zhao, Kim, Zhang, Rush, & LeCun, 2018). Finally, KL divergence may result in loss reconstructions, face challenges with gradient vanishing during training, and introduce a trade-off between generation quality and regularization strength (Lai, Zou, & Lerman, 2023). To address these issues, Tolstikhin et al. (2018) generalized generative modeling to an optimal transport problem (Bousquet, Gelly, Tolstikhin, Simon-Gabriel, & Schoelkopf, 2017) and proposed the Wasserstein Autoencoder (WAE), a new family of regularized autoencoders.

2.3. WAE

Similar to VAE, WAE aims to minimize both the reconstruction cost and a regularization term, denoted as $D_z(P_z, Q_z)$, representing any arbitrary divergence metrics between Q_z and P_z. The general form of the loss function is expressed as:

$$\mathcal{L}_{\mathrm{WAE}} = \mathbb{E}_{q(z|x)}[\log p(x|z)] + \lambda \cdot D_z(q(z|x), p(z)), \tag{3}$$

where $\mathbb{E}_{q(z|x)}[\log p(x|z)]$ serves as the reconstruction term, $D_z(q(z|x), p(z))$ represents any arbitrary divergence metrics, and $\lambda > 0$ is a hyperparameter. Tolstikhin et al. (2018) proposed two specific regularizers: the Maximum Mean Discrepancy (MMD) and adversarial training, to address issues and provide more flexibility in latent space modeling. It is noteworthy that when adversarial training is employed, it aligns with the Adversarial Autoencoder proposed by Zhao et al. (2018). Kolouri, Pope, Martin, and Rohde (2019) proposed an alternative method that uses the Sliced-Wasserstein distance to transform the latent distribution of an autoencoder into a prior distribution that may be sampled. Following this innovative approach, recent state-of-the-art (SOTA) models have seen substantial advancements, as evidenced by studies like that of Yi and Liu (2023).The WAE leverages the Wasserstein distance to establish a direct and deterministic mapping from input data to the latent space, offering a solution to the inherent stochasticity of latent variables in both GANs and VAEs (Tolstikhin et al., 2018). This deterministic mapping not only addresses the issue of randomness but also allows for the inclusion of latent space interpolation, a well-recognized technique in computer vision and natural language processing. This aligns with the proven effectiveness of shallow interpolation methods such as SMOTE for tabular data. This versatility allows WAE to extend its applicability beyond traditional image and language data to diverse domains, including tabular datasets.

3. TWAE: Tabular Wasserstein autoencoder

Algorithm 1 Tabular Wasserstein Autoencoder

Input: Real data samples $\{x(i)\}_{i=1}^m$, learning rates $\alpha_{\text{enc}}, \alpha_{\text{dec}}, \alpha_{\text{dis}}$, GAN-based regularization coefficient λ_1, MMD-based regularization coefficient λ_2
Initialize: Autoencoder parameters ϕ, θ, Discrimiator parameters ψ
for each training iteration **do**
  Sample $x(i)$, $z'(i)$ and z from the training set, true prior p_z and $q_\phi(z|x)$
  Update the Critic (ψ) **:**
  Compute the critic loss:
  $L_{\text{cri}} \leftarrow -\frac{1}{m}\sum_{i=1}^m f_w(z(i)) + \frac{1}{m}\sum_{i=1}^m f_w(z'(i))$
  Update critic parameters:
    $\psi \leftarrow \psi - \alpha_{\text{cri}} \nabla_\psi (L_{\text{cri}})$
  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(es):
  $L_{\text{reg}} \leftarrow D_z(q(z|x), p(z))$
  Update encoder and decoder parameters:
    $\phi \leftarrow \phi - \alpha_{\text{enc}} \nabla_\phi (L_{\text{rec}} + \lambda L_{\text{reg}}), \quad \theta \leftarrow \theta - \alpha_{\text{dec}} \nabla_\theta (L_{\text{rec}} + \lambda L_{\text{reg}})$
end for

3.1. Description

TWAE employs an encoder/decoder/critic framework, incorporating latent space interpolation and a loss function with reconstruction and regularization terms. The process, illustrated in Fig. 1 and outlined in Algorithm 1, comprises two stages: (1) fine-tuning a WAE to create a smooth latent space, and (2) generating synthetic data by sampling from the interpolated latent space. Detailed procedures for each component are provided in subsequent sections, followed by a summary.

3.2. Preprocessing

Following the previous study by Xu et al. (2019), a reversible preprocessing step is used before model training and after data generation (transforming synthetic data back into the original format). To handle continuous variables, a variational Gaussian mixture model (VGM) is utilized to address non-Gaussian and multimodal distributions. Initially, the model estimates the number of modes, denoted as m, and subsequently fits a Gaussian mixture. The values are normalized within each mode, and a mode is represented as a one-hot vector $\beta_{i,j} = [0, 0, \ldots, 1]$, while $\alpha_{i,j}$ denotes the normalized value. Here, i represents the ith column, and j represents the jth row. Consequently, the original continuous variables for row j can be expressed as $\alpha_{1,j} \oplus \beta_{1,j} \oplus \cdots \oplus \alpha_{n_c,j} \oplus \beta_{n_c,j}$, where $\oplus$ signifies concatenation. For categorical features, one-hot encoding is employed to convert each unique categorical element into its binary vector column. This transformation leads to discrete columns $D_1, \ldots, D_{n_d}$ being converted into one-hot vectors $d_1, \ldots, d_{n_d}$, where the ith one-hot vector is denoted as $d_i = [d_i^{(k)}]$ for $k = 1, \ldots, C_i$ (representing categories in the ith column). Hence, the original categorical variables for row j can be represented as $d_{1,j} \oplus \cdots \oplus d_{n_d,j}$. As a result, the initial jth row, denoted as r_j, can be effectively represented as the concatenation of the processed numerical and categorical variables. This representation is ready to be fed into the subsequent DGMs:

$$r_j = \alpha_{1,j} \oplus \beta_{1,j} \oplus \cdots \oplus \alpha_{n_c,j} \oplus \beta_{n_c,j} + d_{1,j} \oplus \cdots \oplus d_{n_d,j}. \tag{4}$$

3.3. Enhanced loss function

Tolstikhin et al. (2018) proposed two loss functions: $\mathcal{L}_{\text{WAE-MMD}}$ and $\mathcal{L}_{\text{WAE-GAN}}$, each exhibiting distinct advantages and drawbacks. WAE-MMD effectively addresses mode collapse, enhancing diversity in generated samples, while WAE-GAN introduces a smoother latent space and improves image quality. To capitalize on the benefits of both approaches, our study introduces a novel hybrid loss function. By combining the strengths of WAE-MMD and WAE-GAN, we anticipate achieving a more stable training process, effectively mitigating mode collapse and training instability. This hybrid approach aims to strike a balance between stability, diversity, and quality in generative modeling. The loss function in TWAE comprises three key terms: a reconstruction loss L_{rec}, a regularization term from MMD L_{reg_m}, and an adversarial regularization term from GAN L_{reg_g}. The impact of each component can be adjusted using corresponding hyperparameters λ_m and λ_g ($\lambda > 0$). It is essential to highlight that there are numerous possibilities for this regularization term, enhancing the adaptability of our algorithm. The proposed loss function is articulated as follows:

$$\mathcal{L}_{\text{TWAE}} = L_{\text{rec}} + \lambda_m \cdot L_{reg_m} + \lambda_g \cdot L_{reg_g}, \tag{5}$$

where:

$$\begin{aligned} L_{\text{rec}} &\leftarrow -\mathbb{E}_{q(z|x)}[\log p(x|z)], \\ L_{reg_m} &\leftarrow \text{MMD}(q(z), p(z)), \\ L_{reg_g} &\leftarrow \text{W}_c(q(z|x), p(z)). \end{aligned} \tag{6}$$

When λ_{reg_g} is set to 0, $\mathcal{L}_{\text{TWAE}}$ coincides with $\mathcal{L}_{\text{WAE-MMD}}$, which is given by:

$$\mathcal{L}_{\text{WAE-MMD}} = -\mathbb{E}_{q(z|x)}[\log p(x|z)] \quad + \lambda \cdot \text{MMD}(q(z), p(z)). \tag{7}$$

Here, $\text{MMD}(q(z), p(z))$ represents the Maximum Mean Discrepancy between the approximate posterior $q(z)$ and the prior $p(z)$, as described below Arbel, Korba, Salim, and Gretton (2019). The parameter λ serves as a hyperparameter governing the intensity of the MMD regularization.

$$\begin{aligned} &\text{MMD}^2(q(z), p(z)) = \|\mu_q - \mu_p\|_F^2 \\ &= \frac{1}{n(n-1)}\sum_{i \neq j} k(z_i, z_j) + \frac{1}{n(n-1)}\sum_{i \neq j} k(z'_i, z'_j) - \frac{2}{nn}\sum_{i,j} k(z_i, z'_j), \end{aligned} \tag{8}$$

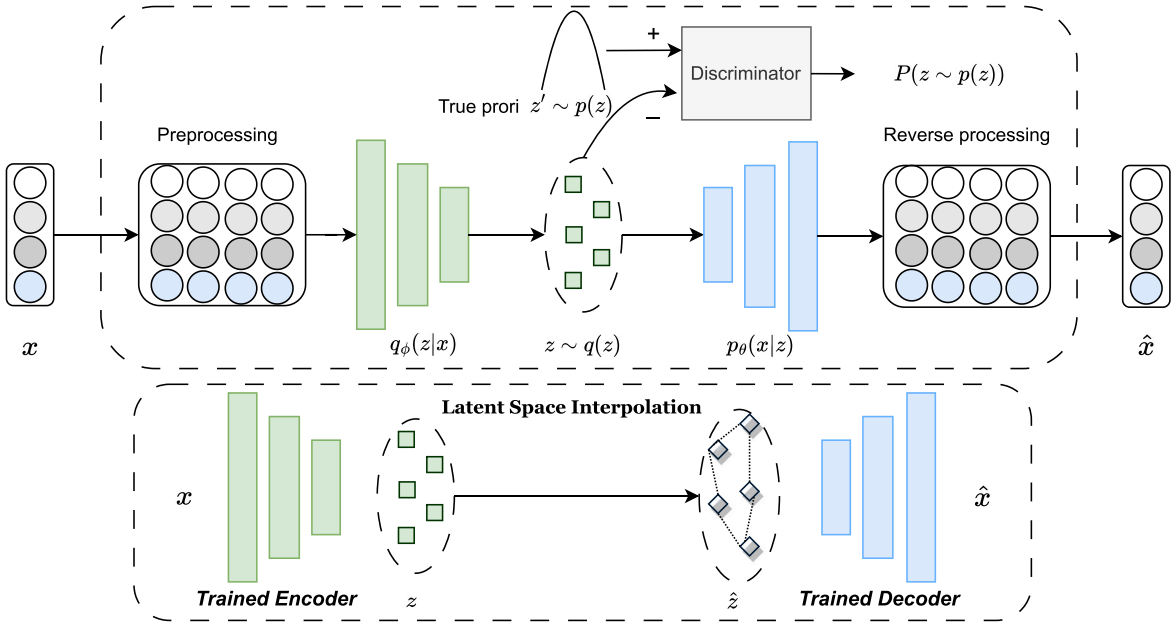


Fig. 1. Illustrating the TWAE with latent space interpolation: During the training phase, each data sample x undergoes preprocessing and is subsequently transformed by the encoder, resulting in a mapping to a latent space vector z. Subsequently, the encoded latent space vector z is utilized as an input for the decoder, which engages in a reconstruction process incorporating both reconstruction loss and regularization loss. During the generation phase, a synthetic latent space vector $\hat{z}$ is generated through latent space interpolation on the encoded latent space vector z. This synthetic vector is then fed into the trained decoder to generate synthetic data, followed by reverse processing to return the data to its original format, $\hat{x}$.

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

$$k(q,p) = \exp\left(-\frac{\|z - z'\|^2}{2\sigma^2}\right). \tag{9}$$

When λ_{reg_m} is set to 0, $\mathcal{L}_{\text{TWAE}}$ coincides with $\mathcal{L}_{\text{WAE-GAN}}$, which is given by:

$$\mathcal{L}_{\text{WAE-GAN}} = -\mathbb{E}_{q(z|x)}[\log p(x|z)] \quad + \lambda \cdot \mathrm{W}_c(q(z|x), p(z)), \tag{10}$$

where $\mathrm{W}_c(q(z|x), p(z))$ is the Wasserstein distance with regularization and λ is a hyperparameter controlling the strength of the Wasserstein regularization.

3.4. Latent space interpolation during data generation

Latent space interpolation is widely studied in generative modeling, especially in computer vision (Struski et al., 2023). Within DGMs, it brings several advantages to data generation (Mukherjee, Asnani, Lin, & Kannan, 2019). By enabling smooth transitions between data samples, it enhances control over data generation and deepens our understanding of the data structure. Recent research emphasizes its role in producing more meaningful and realistic data (Dablain et al., 2022). While latent space interpolation is well-established in DL, simpler methods like SMOTE are effective in tabular data. Leveraging our enhanced loss function and model architecture, we establish a deterministic latent space, allowing integration of this technique into the tabular data generation process. We employ SMOTE to create synthetic latent variables between a randomly chosen variable and one of its neighbors, then utilize the generated latent space for data generation.

4. Experiments

We conduct experiments to evaluate TWAE's performance, using various data quality metrics. We aim to provide a reliable evaluation through a unified framework for assessing TDS techniques. To support further research and ensure reproducibility, we have made our code publicly accessible on Github[1].

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

Table 1
Statistics of datasets used in this study: #S = number of samples, #N = number of numerical columns, #C = number of categorical columns, IR = Imbalance Ratio.

Abbr	Name	#S	#N	#C	IR
CR	credit-g	1000	3	18	2.33
SI	sick	3000	5	18	13.43
JS	jasmine	3000	8	137	1.00
NS	national-longitudinal-survey	4908	5	12	1.65
KD	KDDCup09_upselling	5128	27	21	1.00
EY	eye_movements	7608	17	6	1.00
DE	default-of-credit-card	13 272	14	8	1.00
NC	NewspaperChurn	15 855	3	14	4.22
CO	compass	16 644	5	13	1.00
LA	law-school-admission	20 800	2	9	2.11
NO	nomao	34 465	56	63	2.50
MV	mv	40 768	6	4	1.48
BM	Bank-Marketing	45 211	7	10	7.55
AD	adult	48 842	5	10	6.43
DB	diabetes	70 692	3	19	1.00
CI	Census-Income	299 285	9	33	15.12

4.1. Datasets

To evaluate our algorithm thoroughly, we collected 16 real-world public datasets. These datasets vary in size, characteristics, features, and distributions and have been used in previous studies to assess tabular models (Kotelnikov et al., 2023; Xu et al., 2019; Zhao, Kunar, Birke, & Chen, 2022). We provide a detailed list of datasets and their properties in Table 1. To prevent data leakage, we divided all datasets into 80% for training and 20% for testing. All models were trained using the same training data and default hyperparameters. To ensure reliable results, we repeated the experiments 5 times with 5 different random seeds for data split and reported the average (Borisov et al., 2023).

4.2. Reference tabular data synthesis models

Various methods are available for generating synthetic tabular data, each with its own strengths and weaknesses (Fonseca & Bacao, 2023).

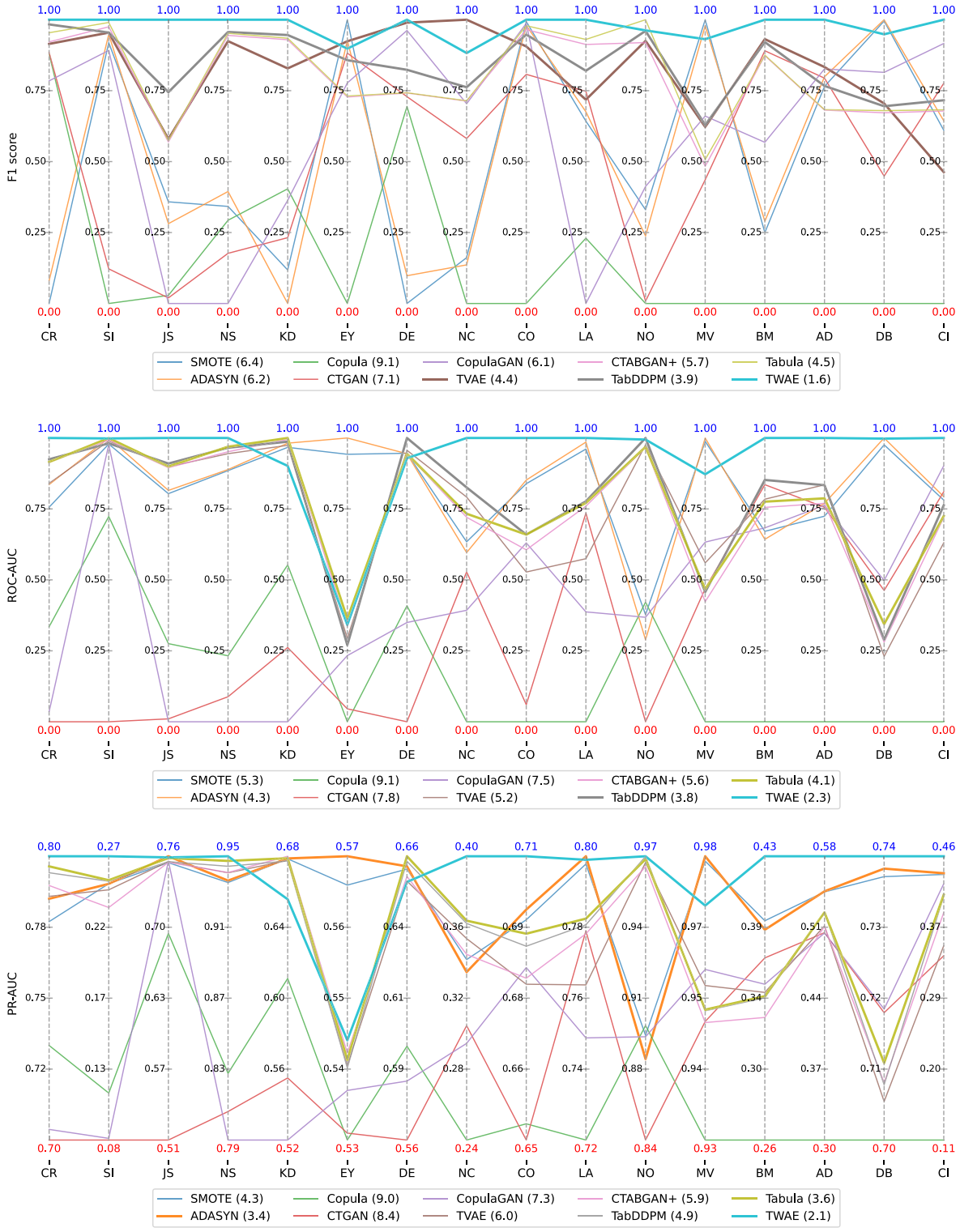


Fig. 2. F1 score, ROC-AUC, and PR-AUC computed w.r.t. 16 datasets, where higher indicates better. The best three methods correspond to thicker lines.

Perturbation-based models, such as linear or non-linear transformations, are simple and suitable for small datasets with continuous features. However, they may not provide data privacy guarantees and can struggle with categorical features and larger datasets. Statistical distribution-based approaches simulate data using known distributions, capturing statistical properties, but they often assume feature independence. Distance-based methods like SMOTE are effective for oversampling minority classes, with variants like ADASYN and SMOTENC handling categorical data (Wang, Chukova, & Nguyen, 2023b). DGMs are versatile in generating diverse data types but may face challenges with categorical data. Recent research has introduced a variety of models for tabular synthetic data generation, including tabular VAEs, GAN-based models, diffusion-based models, and LLM-based models such as TVAE, CTGAN (Xu et al., 2019), TabDDPM (Kotelnikov et al., 2023), and Tabula (Zhao et al., 2023b). In this study, we conducted a comprehensive comparison against SOTA tabular synthetic data models

Table 2
F1 score, ROC-AUC, and PR-AUC computed w.r.t. six classification models. **Bold** represents the best result on each classification model for each metric.

Synthesizers	F1						ROC-AUC						PR-AUC					
	MLP	KNN	LR	DT	RF	LGBM	MLP	KNN	LR	DT	RF	LGBM	MLP	KNN	LR	DT	RF	LGBM
SMOTE	0.5164	0.5076	0.4749	0.6018	0.6110	0.6288	0.7216	0.6486	0.7569	0.6737	0.8243	0.8376	0.6448	0.5759	0.6707	0.5797	0.7493	0.7640
ADASYN	0.5188	0.5073	0.4750	0.5986	0.6128	0.6292	0.7150	0.6508	0.7612	0.6708	0.8301	0.8403	0.6357	0.5781	0.6728	0.5783	0.7542	0.7653
Copula	0.4430	0.4098	0.4025	0.4871	0.4483	0.4563	0.6376	0.5905	0.7132	0.5674	0.7210	0.7346	0.5669	0.5171	0.6220	0.4956	0.6292	0.6467
CTGAN	0.5088	0.5371	0.5663	0.5551	0.5937	0.6035	0.6396	0.6065	0.6923	0.5929	0.7105	0.7071	0.5776	0.5367	0.6194	0.5113	0.6358	0.6410
CopulaGAN	0.5520	0.5597	0.5520	0.5812	0.6145	0.6204	0.6730	0.6400	0.6930	0.6201	0.7279	0.7337	0.6019	0.5599	0.6243	0.5325	0.6515	0.6632
TVAE	**0.6139**[a]	0.5760	0.6235	0.6444	0.6687	0.6778	0.7319	0.6702	0.7692	0.6913	0.8144	0.8152	0.6415	0.5793	0.6664	0.5863	0.7259	0.7314
CTABGAN+	0.5742	0.5559	0.6122	0.6413	0.6608	0.6696	0.7339	0.6698	0.7685	0.6908	0.8168	0.8188	0.6401	0.5751	0.6660	0.5851	0.7288	0.7330
TabDDPM	0.5934	0.5758	0.6273	0.6436	0.6731	0.6799	0.7416	0.6744	0.7723	0.6936	0.8203	0.8222	0.6525	0.5822	0.6707	0.5865	0.7331	0.7383
Tabula	0.5953	0.5767	0.6301	0.6460	0.6751	0.6832	0.7451	0.6754	0.7760	0.6964	0.8231	0.8232	0.6533	0.5823	0.6718	0.5876	0.7341	0.7388
TWAE	0.6089	**0.6090**	**0.6326**[a]	**0.6627**	**0.7032**	**0.7141**	**0.7563**[a]	**0.6964**	**0.7791**	**0.7012**	**0.8358**	**0.8416**	**0.6662**	**0.6023**	**0.6838**	**0.5951**	**0.7583**	**0.7678**
Real	0.5780	0.6128	0.6285	0.6847	0.7348	0.7509	0.7535	0.7134	0.7838	0.7279	0.8662	0.8729	0.6682	0.6144	0.6896	0.6184	0.7936	0.8058

[a] Denotes that the synthetic result is superior to that of real data.

from statistical, ML, and DL models, including Copula (Sun, Cuesta-Infante, & Veeramachaneni, 2019), SMOTE (Chawla et al., 2002), and its variants: ADASYN, along with TVAE, CTGAN, CopulaGAN (Gupta et al., 2021), CTABGAN+ (Zhao et al., 2022), TabDDPM and Tabula.

4.3. Evaluation metrics

Following prior research (Zhao, Birke, & Chen, 2023a), we assessed synthetic data quality across three dimensions: ML utility, statistical similarity, and privacy, each supported by visual aids. For ML utility assessment, we primarily used the F1 score, a common metric in prior studies (Kotelnikov et al., 2023; Xu et al., 2019). We also considered the area under the precision–recall curve (PR-AUC) and the area under the receiver operating characteristic curve (ROC-AUC) for a comprehensive evaluation. While ROC-AUC provides insights into overall model performance across various thresholds, F1 score and PR-AUC focus on the precision–recall trade-off, particularly useful for imbalanced data scenarios (Wang, Chukova, & Nguyen, 2023a). We trained classification models and tested them on a separate holdout dataset to prevent information leakage. We utilized six widely used classification methods: Multi-Layer Perceptron (MLP), k-Nearest Neighbors (k-NN), Logistic Regression (LR), Decision Tree (DT), Random Forest (RF), and Light Gradient Boosting Machine (LightGBM). These models were selected to cover different approaches and sensitivities to class imbalance, with fairness ensured by using default hyperparameters (Borisov et al., 2023).

For statistical similarity, we used three groups of metrics: column-wise (univariate), pair-wise column correlation (bivariate), and table-level (multivariate) (Zhao et al., 2021). For column-wise metrics, we employed Kullback–Leibler divergence (KLD), Jensen–Shannon divergence (JSD), and Wasserstein distance (WD) to evaluate synthetic data from univariate distribution (Zhao et al., 2022). Correlation analysis is often used to assess the similarity between columns in real and synthetic datasets, with the choice of correlation coefficients based on the variable types (Wang, Simpson, & Nguyen, 2025). We used Pearson correlation for continuous-to-continuous variables, correlation ratios for continuous-to-nominal, point-biserial correlation for continuous and dichotomous, and Cramer's V for nominal-to-nominal, including dichotomous variables. For dichotomous-to-dichotomous variables, we used the phi coefficient or Matthews Correlation Coefficient (MCC). Correlation coefficients range from 0 (no correlation) to 1 (perfect correlation). After calculating these coefficients for both real and synthetic data, we applied the Kolmogorov–Smirnov (KS) test and reported the p-value, with higher values indicating better performance. Finally, we measured the overall fidelity and diversity of synthetic data using density and coverage scores proposed in Naeem, Oh, Uh, Choi, and Yoo (2020). Coverage ranges from 0 to 1, while density has no upper limit; however, higher values indicate better performance for both metrics.

Privacy risks stem from the potential disclosure of sensitive information, threatening the confidentiality of individuals represented in the data. Evaluating privacy risks in synthetic data involves examining vulnerabilities when attackers access the dataset without knowledge of the underlying generative models (Yan et al., 2022). A common method is to identify records from the training dataset that are also present in the synthetic dataset (Platzer & Reutterer, 2021). This can be extended to measure the distance to the closest records (DCR), evaluating how closely each synthetic sample resembles its nearest original record (Torfi, Fox, & Reddy, 2022). A lower DCR suggests greater similarity to real data. However, this metric alone is insufficient for assessing privacy risks, as a high DCR may indicate lower risk but poor data quality. Therefore, we included ML detectability as a privacy measure, utilizing the propensity score (Woo, Reiter, Oganian, & Karr, 2009). This involves shuffling real and synthetic data together with flags indicating their origin. An ML model is cross-validated to predict these flags, and the discriminative ability is converted into a score, termed propensityMSE. This score is calculated as the mean squared difference of the probability and 0.5, normalized to a range of [0,1]. A score of 0 indicates that synthetic and real data are indistinguishable, while a score of 1 indicates complete distinguishability. Ideally, high-quality synthetic data should show a low DCR and a high propensityMSE, indicating a close resemblance to real data while minimizing privacy risks without simple replication. We used LR and DT to calculate the propensity score.

5. Results and discussion

5.1. ML utility

In this study, we evaluate our proposed algorithm's performance against other SOTA algorithms, focusing on its ML utility from two angles. Fig. 2 displays the average F1-score, ROC-AUC, and PR-AUC results for six classification models across 16 datasets, arranged by increasing complexity (from smallest CR to largest CI). Higher values indicate better performance, with competing algorithms and their rankings shown in the legend. The top three models are highlighted using bold lines for clarity. To complement this visual comparison, detailed results are provided in Table A.1 in the Appendix. Additionally, Table 2 presents the performance results of all datasets for the six classification models, organized by ascending algorithm complexity. This dual approach allows us to evaluate TDS models and their effectiveness across different data sizes and classification complexities (Shwartz-Ziv & Armon, 2022).

Based on the result presented in Fig. 2, TWAE consistently outperforms all other algorithms across various metrics, including average F1 score, ROC-AUC, and PR-AUC. Specifically, TWAE secures the top ranking in 11 datasets with an average F1 score ranking of 1.6. Additionally, in terms of ROC-AUC, TWAE leads in 9 out of 16 datasets with an average ranking of 2.3. For PR-AUC, TWAE tops the ranking in 9 out of 12 datasets. Notably, in datasets with a more severe class imbalance (IR > 4.00), including SI, NC, BM, AD, and CI, TWAE consistently achieves

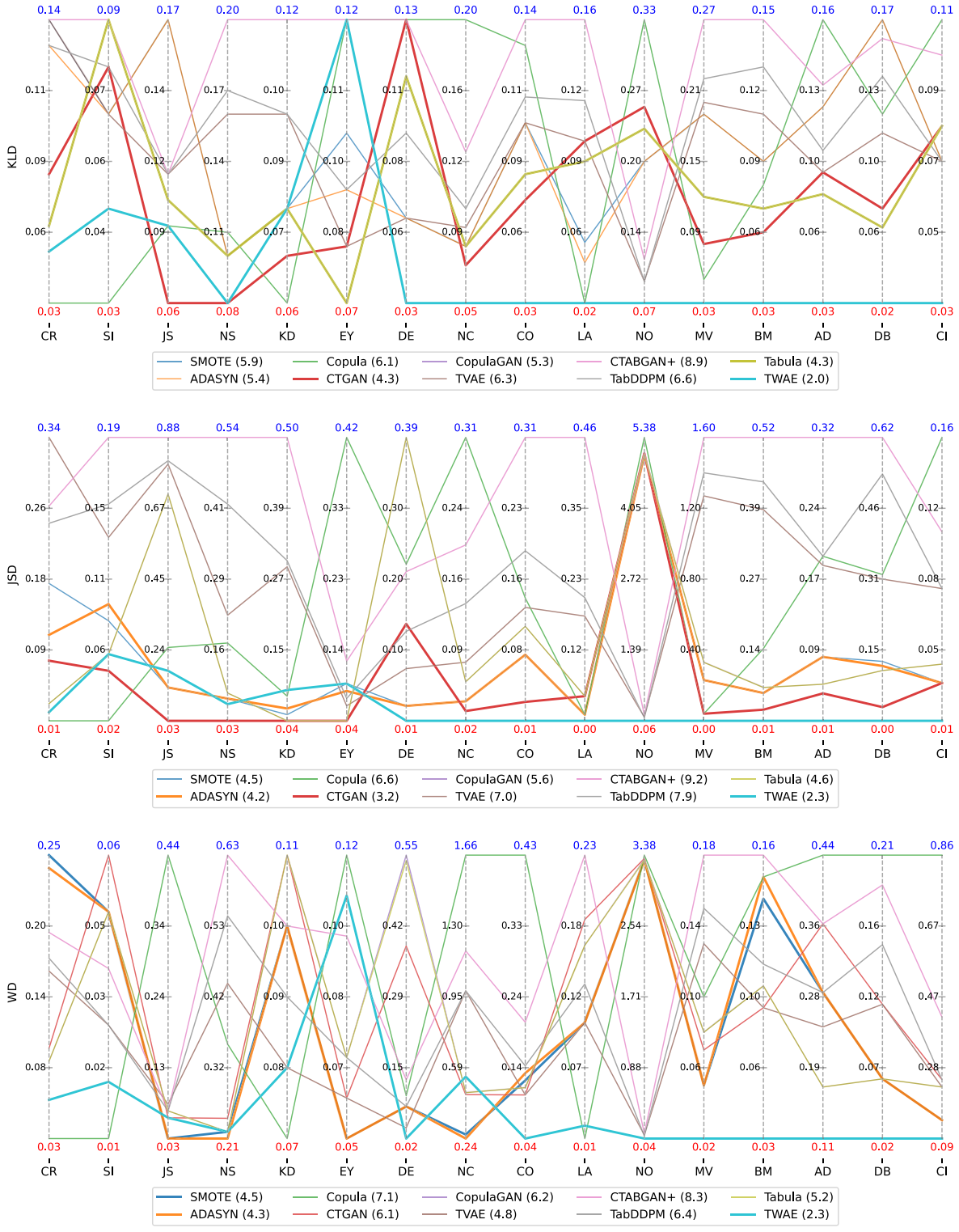


Fig. 3. Column-wise KLD, JSD, and WD computed w.r.t. 16 datasets, where lower indicates better. The best three methods correspond to thicker lines.

the highest rankings. Even in cases where TWAE does not secure the top rank, the difference in performance is minimal when compared to the best-performing algorithms. Moreover, TWAE outperforms all other algorithms on datasets with high dimensionality (>40), including JS, KD, and CI. These results collectively demonstrate the superiority and robustness of TWAE against different data sizes, imbalance levels, and high-dimensionality scenarios. The enhanced robustness of TWAE can be attributed to its capability to facilitate a more meaningful and smoother interpolation between points in the latent space, which in turn results in a better-structured latent representation.

In addition to the comprehensive performance breakdown across datasets, we conducted an analysis against various classification algorithms, and the results are presented in Table 2. Overall, the proposed TWAE demonstrates superior performance over all competing data generation algorithms across all three performance metrics, with the exception of MLP when assessed using the F1 score, where it falls short

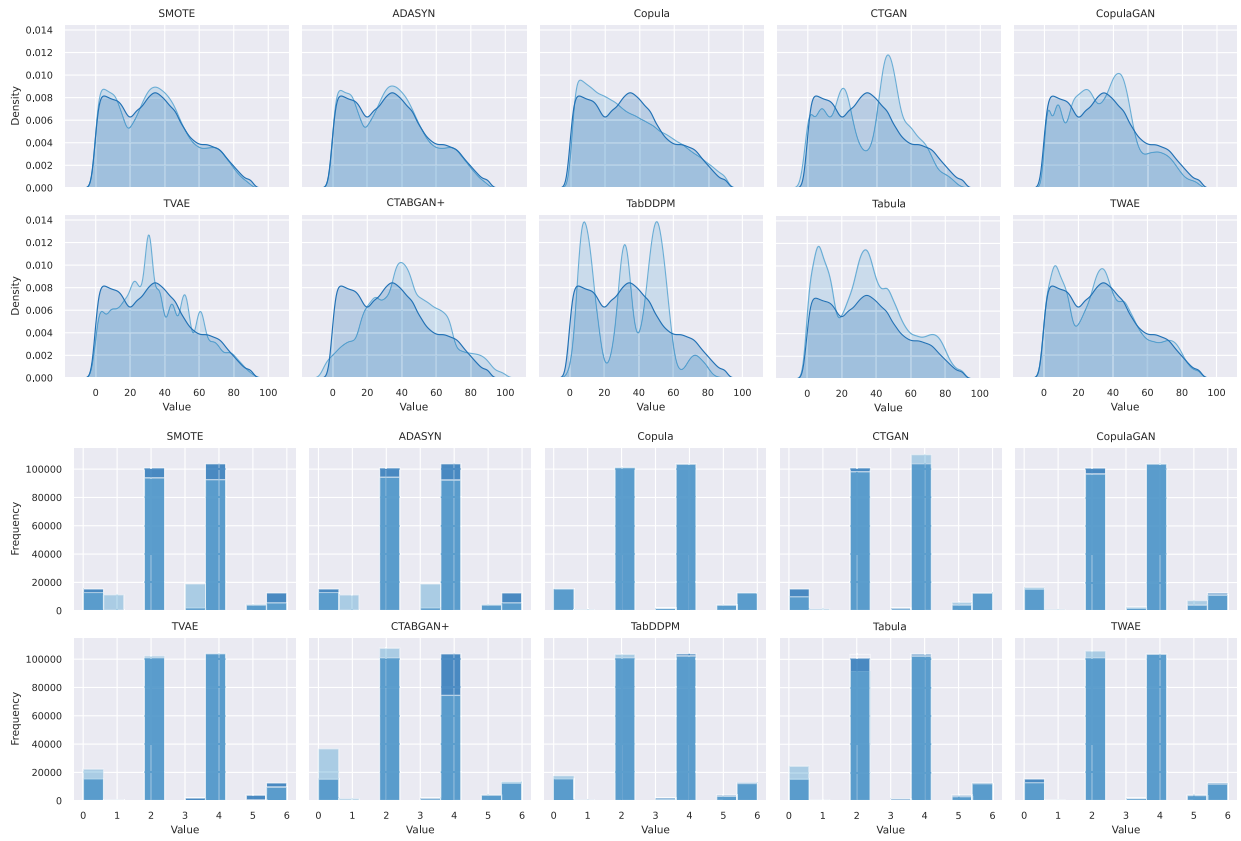


Fig. 4. Comparison of synthetic data and real data distributions for a single column in the CI dataset. Top: numerical column (*age*); Bottom: categorical column (*education*).

Table 3
Pair-wise column correlation score (p-value), where - represents values ≤ 0.001. **Bold** represents the best score on each dataset, where higher indicates better.

Synthesizers	CR	SI	JS	NS	KD	EY	DE	NC	CO	LA	NO	MV	BM	AD	DB	CI	Average	Ranking
SMOTE	**0.34**	–	–	**0.89**	–	–	**0.31**	**0.89**	–	0.59	–	–	0.13	–	–	0.15	0.21	4.9
ADASYN	0.02	–	–	**0.89**	–	–	0.14	0.77	–	**0.81**	–	–	0.19	–	–	0.20	0.19	4.8
Copula	–	–	–	0.01	–	–	–	–	–	–	–	0.28	–	–	–	–	0.02	6.3
CTGAN	–	0.03	–	–	–	0.01	–	0.01	0.25	–	–	0.04	–	0.06	0.23	–	0.04	5.6
CopulaGAN	–	0.04	–	0.19	–	0.16	–	0.01	0.02	0.02	–	0.09	–	0.34	0.14	–	0.06	5.4
TVAE	–	–	–	–	–	0.01	0.01	0.13	0.02	0.59	–	0.04	–	0.62	–	–	0.09	5.1
CTABGAN+	–	–	–	–	–	–	0.04	0.13	0.34	**0.81**	–	0.02	0.01	0.34	–	–	0.11	5.0
TabDDPM	–	–	–	–	–	–	–	0.09	0.25	0.59	–	0.04	0.02	0.62	–	–	0.10	5.1
Tabula	–	**0.04**	–	0.19	–	0.16	–	0.01	0.02	0.02	–	0.09	–	0.34	0.14	–	0.06	4.9
TWAE	–	–	–	–	–	**0.27**	0.01	0.77	**0.70**	**0.81**	–	**0.56**	**0.77**	**0.67**	**0.31**	**0.23**	**0.32**	**3.2**

compared to TVAE. Notably, MLP (0.5780 vs. 0.6089) and LR (0.6285 vs. 0.6326) trained on synthetic data generated by TWAE exhibit better performance compared to that on real data, as measured by the F1 score. This finding suggests that shallow models, such as MLP, may derive greater benefits from additional data supplementation through synthetic data.

Our extensive evaluations across diverse public benchmark datasets unveil the remarkable performance of TWAE, surpassing baseline models, including the widely recognized and challenging-to-beat shallow SMOTE model (Shwartz-Ziv & Armon, 2022). Specifically, TWAE demonstrates superiority over other SOTA methods across a majority of datasets, particularly for larger datasets. Additionally, TWAE exhibits resilience to different classification algorithms; for simpler algorithms such as MLP and LR, it even yields better results than real data. The observed improvement compared to TVAE underscores the positive impact of WAE-based architecture, the introduction of a new loss function, and the incorporation of latent space interpolation. Collectively, these elements contribute to the overall enhancement of VAE.

5.2. Statistical similarity

In this section, we provide a comprehensive comparison of univariate, bivariate, and multivariate aspects. Our main goal is to assess how well each data synthesis algorithm captures key statistical properties by comparing distributions at various levels between real and synthetic data. Along with quantitative results presented in tables, we also use visualizations for further insights.

At the univariate level, synthetic data generated by TWAE demonstrates the closest resemblance based on various distance metrics, including Kullback–Leibler divergence (KLD = 2.0), Jensen–Shannon divergence (JSD = 2.3), and Wasserstein Distance (WD = 2.3), as shown in Fig. 3. Detailed information is also provided in Table A.2 in Appendix. These distance metrics quantify the disparities between individual columns of real and synthetic data, providing insights into the similarity or dissimilarity of their distributions. To enhance readers' understanding of the advantages of each synthesis algorithm, paired histograms illustrating univariate statistical similarity are included in Fig. 4, which reinforces these findings by demonstrating that TWAE exhibits superior proficiency in reproducing univariate distributions, as

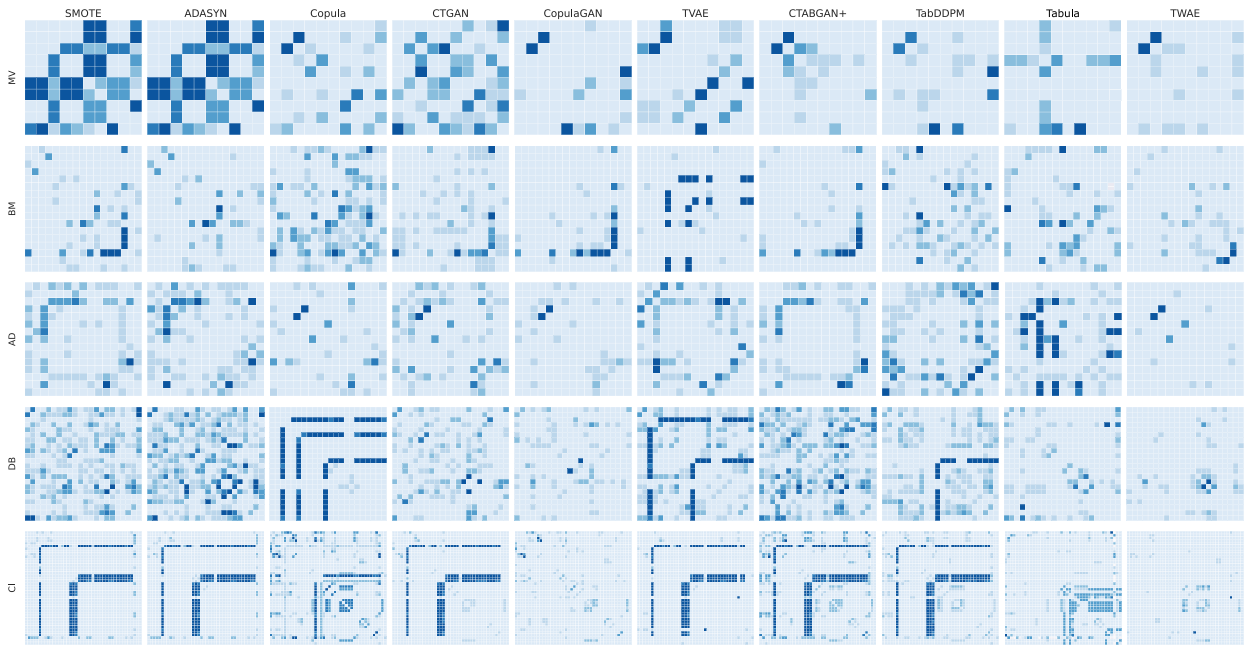


Fig. 5. Heatmaps of the pair-wise column correlation of the synthetic data v.s. real data for the top 5 largest datasets. The value represents the absolute divergence between the real and estimated correlations (the lighter, the better).

Table 4
Comparison of density and coverage scores, where - represents values ≤ 0.001. **Bold** represents the best score on each dataset. Higher values indicate better results.

Synthesizers	Density																Average	Ranking
	CR	SI	JS	NS	KD	EY	DE	NC	CO	LA	NO	MV	BM	AD	DB	CI		
SMOTE	0.49	**1.16**	4.17	0.84	0.37	0.87	0.84	0.75	0.48	1.09	–	0.27	0.68	0.16	1.30	0.40	0.87	6.8
ADASYN	0.51	1.15	4.11	0.85	0.36	0.87	0.84	0.75	0.48	1.08	–	0.27	0.68	0.16	1.29	0.39	0.86	7.0
Copula	0.53	0.70	–	0.38	0.02	0.66	0.10	0.01	0.02	0.50	–	0.26	0.31	–	0.20	–	0.23	9.3
CTGAN	0.41	0.29	0.01	0.66	0.08	0.60	0.14	0.54	0.52	0.79	–	0.74	0.75	0.61	0.84	0.58	0.47	7.9
CopulaGAN	0.46	0.27	6.51	0.49	0.05	0.92	0.21	0.38	0.99	0.81	–	0.88	0.70	0.54	1.00	0.44	0.92	7.4
TVAE	2.65	0.98	7.29	2.36	1.49	1.09	0.90	1.47	0.93	1.08	0.44	1.05	1.09	1.19	1.94	0.98	1.68	5.2
CTABGAN+	2.82	1.09	7.93	**2.70**	**1.56**	**1.22**	1.02	1.83	1.03	1.21	0.72	**1.15**	1.14	1.27	2.08	1.33	1.88	2.1
TabDDPM	2.67	1.04	8.29	2.56	1.54	1.17	**1.03**	1.72	1.00	1.18	0.77	1.11	**1.15**	1.31	2.07	1.29	1.87	2.5
Tabula	2.66	0.98	7.30	2.36	1.49	1.09	0.90	1.48	0.93	1.08	0.44	1.06	1.10	1.20	1.94	0.98	1.69	4.1
TWAE	**3.22**	1.09	**9.00**	2.69	1.37	1.14	1.01	**2.81**	**1.11**	**1.41**	**0.98**	1.04	1.13	**1.53**	**2.22**	**1.45**	**2.08**	**2.1**
	Coverage																Average	Ranking
	CR	SI	JS	NS	KD	EY	DE	NC	CO	LA	NO	MV	BM	AD	DB	CI		
SMOTE	0.73	0.73	**1.00**	0.88	0.58	0.91	0.83	0.89	0.69	0.96	–	0.66	0.60	0.36	0.97	0.65	0.71	4.4
ADASYN	0.76	0.75	**1.00**	0.91	0.58	**0.93**	0.85	0.91	0.70	0.97	–	0.67	0.61	0.37	0.97	0.66	0.73	3.6
Copula	0.79	0.81	–	0.59	0.06	0.48	0.12	0.02	0.01	0.71	–	0.51	0.35	0.01	0.25	–	0.30	8.3
CTGAN	0.69	0.37	0.02	0.79	0.20	0.71	0.10	0.74	0.67	0.80	–	0.76	0.77	0.71	0.80	0.71	0.55	6.4
CopulaGAN	0.74	0.38	0.92	0.72	0.09	0.76	0.09	0.58	0.65	0.80	–	0.48	0.72	0.61	0.83	0.66	0.56	6.6
TVAE	0.96	0.69	0.86	0.80	0.60	0.78	0.55	0.79	0.61	0.69	0.53	0.24	0.50	0.81	0.53	0.84	0.67	5.5
CTABGAN+	0.96	0.65	0.87	0.69	0.55	0.73	0.43	0.75	0.49	0.60	0.62	0.14	0.38	0.70	0.46	0.82	0.62	7.1
TabDDPM	0.95	0.67	0.86	0.78	0.59	0.77	0.50	0.80	0.56	0.67	0.71	0.19	0.45	0.78	0.50	0.89	0.67	5.9
Tabula	0.96	0.69	0.86	0.80	**0.60**	0.78	0.55	0.79	0.61	0.69	0.53	0.24	0.50	0.81	0.53	0.84	0.67	4.4
TWAE	**1.00**	**0.84**	0.95	**0.96**	0.57	0.50	**0.93**	**0.96**	**0.98**	**0.99**	**0.77**	**0.99**	**0.97**	**0.96**	**0.98**	**0.97**	**0.89**	**1.9**

evidenced by lower KLD, JSD, and WD, especially for larger datasets. Notably, CTGAN emerges as the second-best model when measured by KLD and JSD, while SMOTE and ADASYN rank second if measured by WD. Considering the computational overhead, SMOTE appears to be a suitable choice when both accuracy and computational efficiency are crucial.

We utilize a range of correlation measures to assess bivariate relationships, as recommended in a previous study (Wang et al., 2025). The results, as presented in Table 3, demonstrate that TWAE outperforms all other baseline methods in generating synthetic datasets with more realistic pairwise correlations (Ranking = 3.2). To provide further insights into this evaluation, we present the results for the top five largest datasets in Fig. 5, highlighting TWAE's consistently superior performance. In line with the findings at the univariate distribution level, TWAE demonstrates exceptional performance at the bivariate level, especially when dealing with larger datasets. However, for smaller datasets, traditional ML algorithms such as SMOTE and ADASYN demonstrate better performance. It is noteworthy that none of the algorithms successfully captured the correlations for the JS, KD, and NO datasets, which are ranked among the top three datasets in terms of dimensionality. This underscores a research gap in addressing bivariate relationships for high-dimensional tabular data.

At the multivariate level, we utilize the density score and coverage score to assess overall data fidelity and diversity, as outlined in Table 4.

Table 5
Comparison of distance to closest record (DCR). **Bold** represents the best score on each dataset, where lower indicates better.

Synthesizers	CR	SI	JS	NS	KD	EY	DE	NC	CO	LA	NO	MV	BM	AD	DB	CI	Average	Ranking
SMOTE	2.63	0.25	1.65	1.92	1.69	0.36	0.30	2.18	0.69	0.36	9.57	0.18	0.50	0.87	1.23	1.69	1.63	4.4
ADASYN	2.60	0.24	**1.41**	1.90	1.69	**0.35**	**0.29**	2.16	0.68	0.33	9.58	0.18	0.49	0.86	1.22	1.69	1.60	3.6
Copula	2.64	**0.15**	8.14	2.16	2.49	0.84	0.81	3.20	2.81	0.53	9.28	0.20	0.58	1.80	1.94	4.06	2.60	8.1
CTGAN	2.72	0.23	4.33	2.02	2.10	0.45	1.06	2.40	0.70	0.51	9.63	0.16	0.35	0.54	1.48	1.58	1.89	5.9
CopulaGAN	2.68	0.22	3.28	2.08	2.39	0.50	1.34	2.50	0.83	0.49	9.58	0.49	0.40	0.67	1.44	1.67	1.91	6.4
TVAE	2.36	0.33	3.33	2.00	1.71	0.49	0.38	2.33	0.83	0.70	1.34	0.75	0.77	0.52	1.70	1.48	1.31	5.9
CTABGAN+	2.37	0.35	3.32	2.08	1.75	0.54	0.46	2.36	0.97	0.82	1.23	0.79	0.87	0.62	1.75	1.51	1.36	7.6
TabDDPM	2.34	0.34	3.32	2.02	1.71	0.51	0.43	2.32	0.89	0.73	**1.12**	0.76	0.81	0.56	1.72	1.42	1.31	6.1
Tabula	2.35	0.33	3.32	2.00	1.70	0.49	0.38	2.33	0.83	0.70	1.34	0.74	0.76	0.52	1.70	1.48	1.31	4.9
TWAE	**2.11**	0.23	3.27	**1.79**	**1.67**	0.50	0.38	**1.77**	**0.42**	**0.28**	1.14	**0.09**	**0.27**	**0.32**	**1.19**	**1.28**	**1.05**	**1.9**

Table 6
Comparison of $propensityMSE$. **Bold** represents the best score on each dataset, where higher indicates better.

Synthesizers	CR	SI	JS	NS	KD	EY	DE	NC	CO	LA	NO	MV	BM	AD	DB	CI	Average	Ranking
SMOTE	0.64	0.57	0.69	0.63	0.69	0.70	0.57	0.62	0.75	0.39	**1.00**	0.82	0.70	0.83	0.76	0.68	0.69	7.4
ADASYN	0.61	0.55	0.68	0.61	0.69	0.70	0.55	0.62	0.74	0.34	**1.00**	0.82	0.69	0.82	0.68	0.68	0.67	8.4
Copula	0.49	0.54	**1.00**	0.83	0.94	0.91	**0.95**	0.94	**0.99**	0.62	**1.00**	0.85	0.96	**0.97**	**0.97**	**0.99**	0.87	3.3
CTGAN	0.65	**0.83**	0.80	0.61	**0.95**	0.88	0.95	0.60	0.72	0.59	**1.00**	0.62	0.62	0.89	0.61	0.98	0.77	6.0
CopulaGAN	0.57	0.77	0.95	0.67	0.93	0.88	0.96	0.75	0.73	0.59	**1.00**	0.72	0.72	0.67	0.63	0.91	0.78	6.1
TVAE	0.68	0.61	0.97	0.76	0.91	0.86	0.84	0.69	0.74	0.67	0.99	0.89	0.89	0.69	0.93	0.75	0.81	6.8
CTABGAN+	0.70	0.65	0.98	0.83	0.93	0.89	0.87	0.75	0.81	0.73	**1.00**	0.94	0.93	0.75	0.95	0.71	0.84	3.4
TabDDPM	0.68	0.62	0.98	0.79	0.92	0.86	0.86	0.72	0.77	0.69	**1.00**	0.92	0.92	0.71	0.94	0.70	0.82	4.7
Tabula	0.68	0.61	0.98	0.76	0.92	0.86	0.85	0.69	0.75	0.68	0.99	0.89	0.89	0.69	0.93	0.75	0.81	5.8
TWAE	**0.90**	0.82	0.99	**0.96**	0.94	**0.98**	**0.96**	**0.97**	0.94	**0.85**	**1.00**	**0.98**	**0.98**	0.96	0.97	0.97	**0.95**	**1.6**

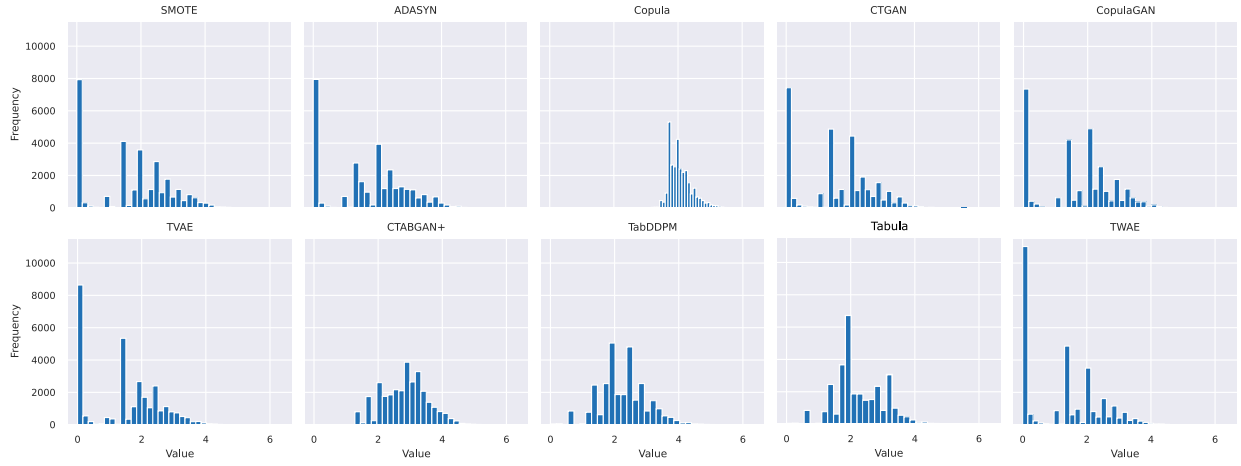


Fig. 6. Distance to closest record (DCR) distributions for the CI dataset with respect to the original train set. This experiment shows that the proposed method does not "copy" samples from the training set but rather generates new synthetic samples that are close to the original samples.

Table 7
Ablation analysis for TWAE (F1.diff.).

Synthesizers	CR	SI	JS	NS	KD	EY	DE	NC	CO	LA	NO	MV	BM	AD	DB	CI
w/o L_{reg_m}	−0.0181	−0.0961	−0.0163	−0.0163	−0.0322	+0.0113	−0.0023	+0.0323	−0.0387	−0.0365	−0.0044	−0.0235	−0.0264	−0.0736	−0.0474	−0.2330
w/o L_{reg_g}	−0.0108	−0.0081	−0.0057	−0.0274	−0.0018	−0.0160	−0.0139	+0.0032	−0.0092	−0.0112	−0.0070	−0.0315	−0.0109	−0.0055	−0.0095	−0.0116
w/o SMOTE	+0.0023	+0.0002	+0.0053	−0.0123	−0.0171	−0.0094	−0.0188	+0.0047	+0.0023	+0.0029	−0.0228	−0.0212	−0.0119	−0.0139	−0.0165	−0.0130
TWAE	0.7930	0.2390	0.7943	0.9405	0.6628	0.5149	0.6504	0.3049	0.6809	0.7862	0.9451	0.9501	0.4575	0.5372	0.7248	0.4535

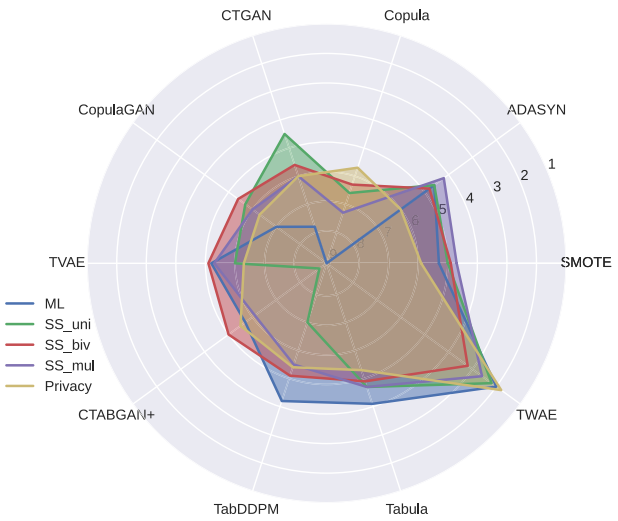


Fig. 7. Ranking analysis of data synthesis algorithms across various evaluation metrics, where lower rankings are closer to the center.

Once again, TWAE attains the top ranking based on the average density score, demonstrating its proficiency in capturing local neighborhood relationships within the generated data. The concurrent commendable performance of the other two DL-based algorithms, CTABGAN+ and TabDDPM, echoes the expected capabilities of DL algorithms in producing more realistic samples. Moreover, when considering the coverage score, TWAE maintains its position as the best-performing method, followed by SMOTE and ADASYN. This outcome aligns with expectations, given the distance metric foundations of both SMOTE and ADASYN, ensuring that the synthetic data generated remains within the bounds of the real data space. Concurrently, the lower rankings of CTGAN, CopulaGAN, and CTABGAN+ highlight the widely recognized problem of mode collapse that is inherent in GAN-based models (Xu et al., 2019). Overall, TWAE not only excels in generating high-quality synthetic data but also demonstrates comprehensive coverage of data diversity within the real data space.

5.3. Privacy

From a privacy perspective, TWAE takes the lead in the DCR (1.8), according to Table 5, indicating that the synthetic data generated by TWAE closely resembles the real dataset. Fig. 6 visually reinforces this result, illustrating that synthetic data generated by TAWE achieves the shortest average DCR. On the other hand, as indicated in Table 6, TWAE ranks at the top in $propensityMSE$ (1.9), suggesting that simple ML algorithms cannot 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 evaluations demonstrate that TWAE is a versatile and effective solution compared to other baselines, producing synthetic data of superior quality and demonstrating resilience against privacy attacks.

To enhance readability and interpretation, we have combined the algorithm rankings across all metrics into a radar plot, shown in Fig. 7. This plot averages the rankings of F1-score, AUC-ROC, and PR-AUC from Table A.1 for ML utility. For univariate testing, we used the average rankings of KLD, JSD, and WD from Table A.2. Bivariate testing rankings are based on correlation values from Table 3, while multivariate testing rankings come from density and coverage scores in Table 4. For privacy, we averaged the rankings of DCR and $propensityMSE$ from Tables 5 and 6. In this visualization, algorithms closer to the center exhibit weaker performance, while those farther away showcase stronger performance. This concise graphical approach facilitates a quick assessment of each algorithm's strengths across multiple metrics, aiding tailored decision-making for specific business needs. It is apparent from the analysis that TWAE outperforms all other algorithms across all metrics. Additionally, DL algorithms like TabDDPM, CTABGAN+, and TVAE perform better in ML utility, whereas SMOTE and ADASYN perform better in statistical similarity metrics. Tabula shows average performance across metrics but comes with high computational costs, even without hyperparameter tuning. This suggests LLMs may not be ideal for tabular data, unlike TWAE, which also has a computational advantage. Similarly, other DL algorithms, while effective, require significantly more time and resources to train compared to simpler methods like SMOTE and ADASYN. Therefore, in resource-constrained environments, ADASYN strikes a good balance, delivering solid performance across metrics with relatively low computational demand.

5.4. Ablation study

To evaluate the impact of each component in TWAE, we conducted three ablation studies aimed at understanding how the removal of specific elements affects the model's overall performance. The experiments included: (1) **w/o SMOTE**, where latent space interpolation is replaced with random latent space; (2) **w/o** L_{reg_g}, where the adversarial regularization term is removed, leaving only MMD; and (3) **w/o** L_{reg_m}, where MMD regulation is replaced with the standard ELBO, making the model similar to TVAE.

The results in Table 7 show the F1-score differences between the ablated models and the default TWAE across 16 classification datasets. Each component affects different datasets in various ways. For instance, removing SMOTE has little impact on smaller datasets like JS and NC (+0.0053 and +0.0047), but leads to drops in larger datasets like CI and NO (−0.0165 and −0.0228). This indicates that the *latent mixup* technique is more effective in larger datasets, where it plays a crucial role in capturing complex patterns and enhancing the model's ability to generalize. Removing the adversarial regularization term (L_{reg_g}) leads to small declines across all datasets, with the biggest drop observed in KD (−0.0274). In contrast, removing the MMD regulation term (L_{reg_m}) shows mixed effects, with the most significant negative impact on CI (−0.2330), highlighting its importance for larger datasets. Some datasets, like EY and NC, show slight improvements (+0.0113 and +0.0323).

Overall, TWAE performs best when all components are included, particularly in larger datasets such as CI (0.4535) and NO (0.9451). This highlights the significance of maintaining the complete model structure in order to attain optimal results. Each component contributes uniquely to enhancing the model's overall effectiveness, providing essential features that improve performance across various data contexts.

6. Conclusion

In this study, we explore the potential of WAE models for TDS, introducing TWAE, a design adept at handling mixed data types, including both numerical and categorical features. The integration of latent space interpolation enhances the controllability and interpretability of synthetic data generation. Across various benchmark datasets, TWAE

Table A.1
F1 score, ROC-AUC and PR-AUC computed w.r.t. 16 datasets. **Bold** represents the best score on each dataset for each metric.

Synthesizers	F1-Score																Ranking
	CR	SI	JS	NS	KD	EY	DE	NC	CO	LA	NO	MV	BM	AD	DB	CI	
SMOTE	0.5819	0.7653	0.0923	0.7976	0.4978	**0.5592**	0.4266	0.1067	0.6782	0.7399	0.8697	**0.9553**	0.1682	0.4333	0.7339	0.2844	6.4
ADASYN	0.5988	0.7758	0.0747	0.8090	0.4757	0.5276	0.4485	0.0997	0.6726	0.7442	0.8587	0.9537	0.1836	0.4484	**0.7348**	0.2989	6.2
Copula	0.7677	0.4360	0.0169	0.7870	0.5513	0.1249	0.5820	0.0623	0.2696	0.6865	0.8303	0.8796	0.0719	0.0957	0.5403	0.0195	9.1
CTGAN	0.7686	0.4797	0.0149	0.7617	0.5191	0.5081	0.5897	0.2221	0.6017	0.7535	0.8313	0.9126	0.4153	0.4463	0.6276	0.3563	7.1
CopulaGAN	0.7475	0.7555	0.0104	0.7233	0.5437	0.4624	0.6419	0.2558	0.6755	0.6567	0.8795	0.9296	0.2909	0.4606	0.6987	0.4171	6.1
TVAE	0.7750	0.7780	0.1429	0.9241	0.6307	0.5262	0.6481	**0.3372**	0.6422	0.7497	0.9407	0.9267	0.4311	0.4636	0.6774	0.2204	4.4
CTABGAN+	0.7765	0.7853	0.1405	0.9285	0.6496	0.4409	0.5926	0.2583	0.6665	0.7749	0.9400	0.9162	0.4086	0.3966	0.6710	0.3139	5.7
TabDDPM	0.7895	0.7781	0.1806	0.9311	0.6527	0.4972	0.6109	0.2718	0.6594	0.7629	0.9449	0.9272	0.4271	0.4342	0.6755	0.3303	3.9
Tabula	0.7833	0.7908	0.1416	0.9301	0.6506	0.4424	0.5928	0.2587	0.6722	0.7772	**0.9496**	0.9180	0.4094	0.3973	0.6725	0.3157	4.5
TWAE	**0.7930**	**0.7943**	**0.2390**	**0.9405**	**0.6628**	0.5149	**0.6504**	0.3049	**0.6809**	**0.7862**	0.9451	0.9501	**0.4575**	**0.5372**	0.7248	**0.4535**	**1.6**
Real	0.7987	0.7909	0.2539	0.9540	0.6399	0.5392	0.6153	0.3157	0.7032	0.7857	0.9685	0.9683	0.4254	0.5228	0.7319	0.4679	
	ROC-AUC																Ranking
	CR	SI	JS	NS	KD	EY	DE	NC	CO	LA	NO	MV	BM	AD	DB	CI	
SMOTE	0.6215	0.8132	0.7105	0.9159	0.6886	0.5842	0.6815	0.6526	0.7215	0.6811	0.8169	0.9719	0.7809	0.7186	0.7812	0.8283	5.3
ADASYN	0.6349	**0.8201**	0.7134	0.9167	0.6911	**0.5876**	0.6813	0.6471	0.7223	0.6843	0.7977	**0.9726**	0.7770	0.7280	**0.7823**	0.8312	4.3
Copula	0.5509	0.7310	0.5763	0.7694	0.6183	0.5284	0.6196	0.5609	0.6700	0.5555	0.8265	0.9140	0.6879	0.5688	0.7367	0.6438	9.1
CTGAN	0.4948	0.4981	0.5090	0.7370	0.5689	0.5311	0.5725	0.6373	0.6737	0.6516	0.7350	0.9415	0.8037	0.7251	0.7578	0.8353	7.8
CopulaGAN	0.5007	0.8126	0.5064	0.7172	0.5245	0.5422	0.6128	0.6177	0.7087	0.6061	0.8149	0.9511	0.7826	0.7266	0.7594	0.8569	7.5
TVAE	0.6356	0.8167	0.7345	0.9293	0.6898	0.5458	0.6827	0.6754	0.7024	0.6306	0.9468	0.9468	0.7966	0.7415	0.7472	0.7928	5.2
CTABGAN+	0.6480	0.8178	0.7337	0.9308	0.6934	0.5483	0.6812	0.6651	0.7072	0.6548	0.9447	0.9388	0.7926	0.7280	0.7495	0.8139	5.6
TabDDPM	0.6498	0.8140	0.7374	0.9340	0.6920	0.5444	**0.6877**	0.6803	0.7105	0.6570	**0.9520**	0.9408	0.8060	0.7412	0.7499	0.8240	3.8
Tabula	0.6482	0.8200	0.7348	0.9348	**0.6942**	0.5499	0.6813	0.6669	0.7105	0.6563	0.9452	0.9414	0.7954	0.7316	0.7524	0.8151	4.1
TWAE	**0.6624**	0.8196	**0.7603**	**0.9417**	0.6775	0.5486	0.6795	**0.7055**	**0.7314**	**0.6862**	0.9508	0.9651	**0.8265**	**0.7756**	0.7822	**0.8800**	**2.3**
Real	0.6706	0.8328	0.7902	0.9532	0.7102	0.6221	0.6750	0.7238	0.7579	0.6895	0.9786	0.9781	0.8357	0.7719	0.7855	0.8798	
	PR-AUC																Ranking
	CR	SI	JS	NS	KD	EY	DE	NC	CO	LA	NO	MV	BM	AD	DB	CI	
SMOTE	0.7798	0.2480	0.7529	0.9358	0.6767	0.5692	0.6598	0.3439	0.6928	0.8023	0.8914	0.9780	0.3896	0.5443	0.7412	0.4410	4.3
ADASYN	0.7884	0.2479	**0.7586**	0.9369	0.6769	**0.5742**	0.6608	0.3370	0.6950	**0.8047**	0.8809	**0.9789**	0.3841	0.5449	0.7423	0.4425	3.4
Copula	0.7332	0.1107	0.6902	0.8253	0.6108	0.5250	0.5964	0.2442	0.6488	0.7169	0.8958	0.9260	0.2556	0.3044	0.7050	0.1082	9.0
CTGAN	0.6976	0.0799	0.5070	0.8032	0.5560	0.5262	0.5628	0.3074	0.6453	0.7816	0.8449	0.9481	0.3669	0.5049	0.7225	0.3391	8.4
CopulaGAN	0.7016	0.0810	0.7543	0.7867	0.5218	0.5336	0.5839	0.2976	0.6825	0.7485	0.8908	0.9578	0.3508	0.5046	0.7230	0.4294	7.3
TVAE	0.7893	0.2440	0.7539	0.9414	0.6758	0.5379	0.6555	0.3555	0.6789	0.7649	0.9682	0.9548	0.3458	0.5115	0.7103	0.3518	6.0
CTABGAN+	0.7934	0.2324	0.7532	0.9416	**0.6781**	0.5401	0.6572	0.3468	0.6802	0.7807	0.9664	0.9479	0.3305	0.5110	0.7129	0.3934	5.9
TabDDPM	0.7982	0.2496	0.7538	0.9452	0.6755	0.5375	0.6626	0.3639	0.6871	0.7835	0.9709	0.9501	0.3425	0.5245	0.7126	0.4138	4.9
Tabula	0.8006	0.2503	0.7570	0.9483	0.6770	0.5389	**0.6644**	0.3654	0.6898	0.7854	0.9697	0.9503	0.3435	0.5245	0.7156	0.4156	3.6
TWAE	**0.8044**	**0.2660**	0.7576	**0.9510**	0.6544	0.5423	0.6552	**0.4010**	**0.7065**	0.8036	**0.9710**	0.9697	**0.4290**	**0.5788**	**0.7440**	**0.4638**	**2.1**
Real	0.8055	0.3096	0.7729	0.9628	0.6919	0.6022	0.6524	0.4250	0.7322	0.8065	0.9866	0.9846	0.4447	0.5908	0.7470	0.4989	

consistently outperforms existing state-of-the-art models, including statistical, ML, and DL approaches, in terms of ML utility, statistical similarity, and privacy preservation. The superior performance of TWAE is attributed to its novel architecture design and enhanced loss function. In conclusion, TWAE emerges as a versatile and effective solution for TDS, positioning it as a promising choice for diverse applications across different domains. Looking ahead, our future aims include developing a novel data representation learning algorithm that leverages graph neural networks to model intricate relationships between samples, thereby advancing beyond simple latent space interpolation.

CRediT authorship contribution statement

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

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

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

Appendix. Additional information and results

See Tables A.1 and A.2.

Data availability

Data and code will be made available to public via Github upon the acceptance of the manuscript.

Table A.2
Column-wise KLD, JSD and WD, where - represents values ≤ 0.001. **Bold** represents the best score on each dataset, where lower indicates better.

Synthesizers	KLD																Ranking
	CR	SI	JS	NS	KD	EY	DE	NC	CO	LA	NO	MV	BM	AD	DB	CI	
SMOTE	0.14	0.07	0.17	0.10	0.08	0.10	0.06	0.08	0.10	0.05	0.20	0.19	0.09	0.12	0.17	0.07	5.9
ADASYN	0.13	0.07	0.17	0.10	0.08	0.09	0.06	0.08	0.10	0.04	0.20	0.19	0.09	0.12	0.17	0.07	5.4
Copula	**0.03**	**0.03**	0.09	0.11	**0.06**	0.12	0.13	0.20	0.13	0.02	0.33	0.05	0.08	0.16	0.12	0.11	6.1
CTGAN	0.08	0.08	**0.06**	0.08	0.07	0.08	0.13	0.07	0.07	0.10	0.25	0.08	0.06	0.09	0.07	0.08	4.3
CopulaGAN	0.06	0.09	0.10	0.10	0.08	0.07	0.11	0.08	0.08	0.09	0.23	0.12	0.07	0.08	0.06	0.08	5.3
TVAE	0.14	0.07	0.11	0.16	0.10	0.08	0.06	0.09	0.10	0.10	0.09	0.20	0.11	0.09	0.11	0.07	6.3
CTABGAN+	0.14	0.09	0.11	0.20	0.12	0.12	0.13	0.13	0.14	0.16	0.11	0.27	0.15	0.13	0.16	0.10	8.9
TabDDPM	0.13	0.08	0.11	0.17	0.10	0.09	0.09	0.10	0.11	0.12	0.09	0.22	0.13	0.10	0.14	0.07	6.6
Tabula	0.06	0.09	0.10	0.10	0.08	**0.07**	0.11	0.08	0.08	0.09	0.23	0.12	0.07	0.08	0.06	0.08	4.3
TWAE	0.05	0.05	0.09	**0.08**	0.08	0.12	**0.03**	**0.05**	**0.03**	**0.02**	**0.07**	**0.03**	**0.03**	**0.03**	**0.02**	**0.03**	**2.0**
	JSD																Ranking
	CR	SI	JS	NS	KD	EY	DE	NC	CO	LA	NO	MV	BM	AD	DB	CI	
SMOTE	0.17	0.08	0.13	0.07	0.05	0.09	0.03	0.04	0.08	0.01	5.05	0.23	0.06	0.08	0.13	0.03	4.5
ADASYN	0.11	0.09	0.13	0.07	0.06	0.08	0.03	0.04	0.08	0.01	5.05	0.23	0.06	0.08	0.12	0.03	4.2
Copula	**0.01**	**0.02**	0.25	0.17	0.08	0.42	0.22	0.31	0.14	0.01	5.38	0.04	0.14	0.19	0.32	0.16	6.6
CTGAN	0.08	0.05	**0.03**	**0.03**	**0.04**	0.04	0.14	0.03	0.03	0.04	5.08	0.04	0.03	0.04	0.03	0.03	3.2
CopulaGAN	0.03	0.06	0.71	0.08	0.04	0.04	0.39	0.06	0.11	0.04	5.08	0.33	0.07	0.05	0.11	0.04	5.6
TVAE	0.34	0.13	0.80	0.22	0.29	0.06	0.08	0.08	0.13	0.17	0.13	1.27	0.39	0.18	0.31	0.08	7.0
CTABGAN+	0.26	0.19	0.88	0.54	0.50	0.12	0.21	0.20	0.31	0.46	0.17	1.60	0.52	0.32	0.62	0.11	9.2
TabDDPM	0.24	0.15	0.81	0.42	0.30	0.07	0.13	0.14	0.19	0.20	0.12	1.40	0.44	0.19	0.54	0.08	7.9
Tabula	0.03	0.06	0.71	0.08	0.04	**0.04**	0.39	0.06	0.11	0.04	5.06	0.33	0.07	0.05	0.11	0.04	4.6
TWAE	0.02	0.06	0.18	0.06	0.09	0.09	**0.01**	**0.02**	**0.01**	–	**0.06**	–	**0.01**	**0.01**	–	**0.01**	**2.3**
	WD																Ranking
	CR	SI	JS	NS	KD	EY	DE	NC	CO	LA	NO	MV	BM	AD	DB	CI	
SMOTE	0.25	0.05	0.03	0.22	0.10	0.05	0.08	0.26	0.12	0.10	3.32	0.05	0.14	0.28	0.06	0.14	4.5
ADASYN	0.24	0.05	**0.03**	**0.21**	0.10	**0.05**	0.08	**0.24**	0.13	0.10	3.32	0.05	0.15	0.28	0.06	0.14	4.3
Copula	**0.03**	**0.01**	0.44	0.35	**0.07**	0.12	0.10	1.66	0.43	**0.01**	3.38	0.10	0.15	0.44	0.21	0.86	7.1
CTGAN	0.10	0.06	0.06	0.24	0.11	0.06	0.38	0.46	0.10	0.18	3.34	0.07	0.09	0.36	0.11	0.25	6.1
CopulaGAN	0.09	0.05	0.07	0.22	0.11	0.07	0.55	0.47	0.11	0.16	3.33	0.08	0.10	0.17	0.06	0.23	6.2
TVAE	0.16	0.03	0.08	0.44	0.08	0.06	0.04	0.98	0.10	0.10	0.08	0.13	0.09	0.24	0.11	0.23	4.8
CTABGAN+	0.19	0.04	0.07	0.63	0.10	0.10	0.13	1.18	0.20	0.23	0.10	0.18	0.16	0.36	0.19	0.42	8.3
TabDDPM	0.17	0.03	0.07	0.54	0.09	0.07	0.08	0.98	0.14	0.13	0.07	0.15	0.11	0.28	0.15	0.25	6.4
Tabula	0.09	0.05	0.07	0.22	0.11	0.07	0.54	0.47	0.11	0.16	3.32	0.08	0.10	0.17	0.06	0.23	5.2
TWAE	0.06	0.02	0.06	0.22	0.08	0.11	**0.02**	0.55	**0.04**	0.02	**0.04**	**0.02**	**0.03**	**0.11**	**0.02**	**0.09**	**2.3**

References

Arbel, M., Korba, A., Salim, A., & Gretton, A. (2019). Maximum mean discrepancy gradient flow. *Advances in Neural Information Processing Systems*, *32*.

Beckham, C., Honari, S., Verma, V., Lamb, A. M., Ghadiri, F., Hjelm, R. D., et al. (2019). On adversarial mixup resynthesis. *Advances in Neural Information Processing Systems*, *32*.

Borisov, V., Leemann, T., Seßler, K., Haug, J., Pawelczyk, M., & Kasneci, G. (2022). Deep neural networks and tabular data: A survey. *IEEE Transactions on Neural Networks and Learning Systems*.

Borisov, V., Seßler, K., Leemann, T., Pawelczyk, M., & Kasneci, G. (2023). Language models are realistic tabular data generators. In *International conference on learning representations* (pp. 1–18).

Bousquet, O., Gelly, S., Tolstikhin, I., Simon-Gabriel, C.-J., & Schoelkopf, B. (2017). From optimal transport to generative modeling: the VEGAN cookbook. arXiv preprint arXiv:1705.07642.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, *16*, 321–357.

Chong, M. J., & Forsyth, D. (2020). Effectively unbiased fid and inception score and where to find them. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 6070–6079).

Dablain, D., Krawczyk, B., & Chawla, N. V. (2022). DeepSMOTE: Fusing deep learning and SMOTE for imbalanced data. *IEEE Transactions on Neural Networks and Learning Systems*, 1–15.

Fonseca, J., & Bacao, F. (2023). Tabular and latent space synthetic data generation: a literature review. *Journal of Big Data*, *10*(1), 115.

Gai, K., & Zhang, S. (2023). Tessellating the latent space for non-adversarial generative auto-encoders. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Gupta, A., Bhatt, D., & Pandey, A. (2021). Transitioning from real to synthetic data: Quantifying the bias in model. In *ICLR synthetic data generation workshop*.

James, S., Harbron, C., Branson, J., & Sundler, M. (2021). Synthetic data use: exploring use cases to optimise data utility. *Discover Artificial Intelligence*, *1*(1), 15.

Kim, J., Lee, C., & Park, N. (2023). Stasy: Score-based tabular data synthesis. In *International conference on learning representations*. ICLR 2023.

Kim, J., Lee, C., Shin, Y., Park, S., Kim, M., Park, N., et al. (2022). SOS: Score-based oversampling for tabular data. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining* (pp. 762–772).

Kingma, D. P., & Welling, M. (2014). Auto-encoding variational Bayes. In *International conference on learning representations*.

Kolouri, S., Pope, P. E., Martin, C. E., & Rohde, G. K. (2019). Sliced Wasserstein auto-encoders. In *International conference on learning representations*.

Kotelnikov, A., Baranchuk, D., Rubachev, I., & Babenko, A. (2023). TABDDPM: Modelling tabular data with diffusion models. In *International conference on machine learning* (pp. 17564–17579).

Lai, C.-H., Zou, D., & Lerman, G. (2023). Robust variational autoencoding with Wasserstein penalty for novelty detection. In *International conference on artificial intelligence and statistics* (pp. 3538–3567).

Lampis, A., Lomurno, E., & Matteucci, M. (2023). Bridging the gap: Enhancing the utility of synthetic data via post-processing techniques. arXiv preprint arXiv:2305.10118.

Mukherjee, S., Asnani, H., Lin, E., & Kannan, S. (2019). ClusterGAN: Latent space clustering in generative adversarial networks. *Vol. 33*, In *Proceedings of the AAAI conference on artificial intelligence* (pp. 4610–4617).

Naeem, M. F., Oh, S. J., Uh, Y., Choi, Y., & Yoo, J. (2020). Reliable fidelity and diversity metrics for generative models. In *International conference on machine learning* (pp. 7176–7185).

Platzer, M., & Reutterer, T. (2021). Holdout-based fidelity and privacy assessment of mixed-type synthetic data. arXiv preprint arXiv:2104.00635.

Rabbani, S. B., & Samad, M. D. (2023). Between-sample relationship in learning tabular data using graph and attention networks. arXiv preprint arXiv:2306.06772.

Raghunathan, T. E. (2021). Synthetic data. *Annual Review of Statistics and its Application*, *8*.

Shwartz-Ziv, R., & Armon, A. (2022). Tabular data: Deep learning is not all you need. *Information Fusion*, *81*, 84–90.

Solatorio, A. V., & Dupriez, O. (2023). Realtabformer: Generating realistic relational and tabular data using transformers. arXiv preprint arXiv:2302.02041.

Struski, Ł., Sadowski, M., Danel, T., Tabor, J., & Podolak, I. T. (2023). Feature-based interpolation and geodesics in the latent spaces of generative models. *IEEE Transactions on Neural Networks and Learning Systems*.

Sun, Y., Cuesta-Infante, A., & Veeramachaneni, K. (2019). Learning vine copula models for synthetic data generation. *Vol. 33*, In *Proceedings of the AAAI conference on artificial intelligence* (pp. 5049–5057).

Tolstikhin, I., Bousquet, O., Gelly, S., & Schoelkopf, B. (2018). A note on the evaluation of generative models. In *International conference on learning representations* (pp. 1–16).

Torfi, A., Fox, E. A., & Reddy, C. K. (2022). Differentially private synthetic medical data generation using convolutional GANs. *Information Sciences*, *586*, 485–500.

Wang, A. X., Chukova, S. S., & Nguyen, B. P. (2023a). Ensemble k-nearest neighbors based on centroid displacement. *Information Sciences*, *629*, 313–323.

Wang, A. X., Chukova, S. S., & Nguyen, B. P. (2023b). Synthetic minority oversampling using edited displacement-based k-nearest neighbors. *Applied Soft Computing*, *148*, Article 110895.

Wang, A. X., Chukova, S. S., Simpson, C. R., & Nguyen, B. P. (2024). Challenges and opportunities of generative models on tabular data. *Applied Soft Computing*, *166*, Article 112223.

Wang, A. X., Chukova, S. S., Sporle, A., Milne, B. J., Simpson, C. R., & Nguyen, B. P. (2024). Enhancing public research on citizen data: An empirical investigation of data synthesis using Statistics New Zealand's Integrated Data Infrastructure. *Information Processing & Management*, *61*(1), Article 103558.

Wang, A. X., Simpson, C. R., & Nguyen, B. P. (2025). Blending is all you need: Data-centric ensemble synthetic data. *Information Sciences*, *691*, Article 121610.

Woo, M.-J., Reiter, J. P., Oganian, A., & Karr, A. F. (2009). Global measures of data utility for microdata masked for disclosure limitation. *Journal of Privacy and Confidentiality*, *1*(1), 111–124.

Xu, L., Skoularidou, M., Cuesta-Infante, A., & Veeramachaneni, K. (2019). Modeling tabular data using conditional GAN. In *Advances in neural information processing systems* (pp. 7335–7345).

Yan, C., Yan, Y., Wan, Z., Zhang, Z., Omberg, L., Guinney, J., et al. (2022). A multifaceted benchmarking of synthetic electronic health record generation models. *Nature Communications*, *13*(1), 7609.

Yi, M., & Liu, S. (2023). Sliced Wasserstein variational inference. In *Asian conference on machine learning* (pp. 1213–1228).

Zhang, H., Zhang, J., Srinivasan, B., Shen, Z., Qin, X., Faloutsos, C., et al. (2024). Mixed-type tabular data synthesis with score-based diffusion in latent space. In *International conference on learning representations*. ICLR 2024.

Zhao, Z., Birke, R., & Chen, L. Y. (2023a). FCT-GAN: Enhancing global correlation of table synthesis via Fourier transform. In *Proceedings of the 32nd ACM international conference on information and knowledge management* (pp. 4450–4454).

Zhao, Z., Birke, R., & Chen, L. (2023b). Tabula: Harnessing language models for tabular data synthesis. arXiv preprint arXiv:2310.12746.

Zhao, J., Kim, Y., Zhang, K., Rush, A., & LeCun, Y. (2018). Adversarially regularized autoencoders. In *International conference on machine learning* (pp. 5902–5911).

Zhao, Z., Kunar, A., Birke, R., & Chen, L. Y. (2021). CTAB-GAN: Effective table data synthesizing. In *Asian conference on machine learning* (pp. 97–112).

Zhao, Z., Kunar, A., Birke, R., & Chen, L. Y. (2022). CTAB-GAN+: Enhancing tabular data synthesis. arXiv preprint arXiv:2204.00401.