Hybrid Generative Models for High-Fidelity Synthetic Tabular Data: VAE-DBM and Transformer-Enhanced CTGAN

Tauseef Ahmed[§], Jonathon Bird[§], Noah Scharrenberg[§] and Aditya Shourya[§]
Department of Advanced Computing Sciences
Faculty of Science and Engineering
Maastricht University
Maastricht, The Netherlands

***Abstract*— The generation of synthetic tabular data that faithfully mirrors real-world datasets is a critical challenge in machine learning, lagging behind the advancements in generating other data types like images and text. This paper tackles this challenge by proposing two novel models: Transformer-based CTGAN and VAE-DBM. Transformer-based CTGAN leverages the power of attention mechanisms to learn complex relationships in tabular data, while VAE-DBM employs a hybrid approach, using Variational Autoencoders (VAE) for continuous variables and Deep Boltzmann Machines (DBM) for categorical variables. We evaluate these models on the Adult Census Income dataset, assessing their performance in terms of fidelity and utility. Our results demonstrate the potential of these approaches to advance the state-of-the-art in synthetic tabular data generation.**

I. Introduction

Generating synthetic data that accurately reflects the complexities and distributions of real-world data is crucial for various applications, including privacy preservation, dataset augmentation, and experiments simulation [1], [2]. Despite the ubiquitous nature of tabular data across various domains - such as financial records, medical datasets or retail inventories - the ability to generate synthetic tabular data lags significantly behind the advancements seen with other kinds of data, like image data. This disparity is striking considering that tabular data, arguably the most prevalent form of structured information, underpins critical decision-making processes across virtually every sector. Generative models have been adapted for tabular data, like CTGAN [3], which outperformed Bayesian methods, a feat other deep learning models failed to achieve. [4], [5]. Recent advancements in attention mechanisms provide a potential avenue to enhance the model's ability to interpret and replicate data [6]–[8].

A. Problem Statement

The discrepancy is unsurprising given the inherent challenges posed by tabular data. Namely its mix of categorical and continuous variables, as well as the subtle, complex and non-linear dependencies between variables. Both make it difficult for a model to capture the underlying statistical characteristics of the data, leading to limitations in two critical aspects; fidelity and utility. Fidelity refers to how well the synthetic data preserves the statistical properties of the original data, while utility pertains to the practical effectiveness of the synthetic data in performing similarly to the original data when utilized in downstream tasks such as classification.

B. Research questions

- RQ1: To what extent can a model improve the fidelity of generated tabular data compared to the state-of-the-art baseline?
- RQ2: To what extent can a model improve the utility of generated tabular data compared to the state-of-the-art baseline?

C. Proposed Approaches

This research proposes two novel adaptations:

1) **Transformer-based CTGAN**: Incorporating transformer architectures to enhance the learning of complex dependencies in tabular data. Transformers are well-suited for capturing intricate relationships due to their attention mechanisms, which can dynamically focus on different parts of the input data.
2) **VAE-DBM**: Recognizing the inherent difficulty of capturing both categorical and continuous variables in the same model, this approach utilizes Variational Auto-Encoders (VAE) for continuous and Deep Boltzmann Machine (DBM) for categorical variables.

The structure of this paper is as follows:

1) Section II: Presents a comprehensive review of relevant literature, highlighting existing gaps in research, and discussing the theoretical foundations and related works pertaining to tabular data generation.
2) Section III: Details the proposed approaches (Transformer-based CTGAN and VAE-DBM), describing the underlying techniques and architectures.

This paper was prepared in partial fulfilment of the requirements for the Degree of Master of Science in Data Science and/or Artificial Intelligence, Maastricht University. Supervisor(s): Chang Sun

[§]Equal contribution.

3) Section IV: Outlines the experimental setup, including the datasets used, evaluation metrics, and comparative baselines.
4) Section V: Presents and analyzes the experimental findings.
5) Section VI: Interprets the results, discusses implications, addresses limitations, and suggests future work.
6) Section VII: Summarizes the main findings, contributions, practical implications, and potential future work.

II. Background

A. Overview of synthetic data generation

As the need for data in society grows all-consuming so does the need to replicate existing data. Synthetic replication addresses many needs, most obvious of which being if additional data is needed or existing data is incomplete, to help with downstream machine learning tasks. The ability to fill in the gap with data that is indistinguishable to the ML model from the original data is vital [9]. An additional need is to be able to fine-tune the generated sample outputs, like to balance a class that's unbalanced in the original dataset, or to modify aspects of a simulated data for an experiment. The most pressing need in modern society however is privacy. As the value of quantifying aspects of people becomes more evident to corporations, the more private facets of ones self is stored across large datasets. This means that information can be linked back to the individual, violating their privacy. Synthetic data offers the chance to keep the necessary and significant information contained in the data, without the specific details about individuals. It's important to note that synthetic data doesn't inherently preserve privacy, and special architecture is needed to ensure it [10].

Generative AI has moved into prominence not from generating tabular data, but instead mainly through images and text. The ability to generate realistic and novel data on command has captured the public's attention, whether that be through tools like DALL E or chat bots like ChatGPT. Coupled with hardware advancements, and advancements in deep learning techniques have propelled the rapid development of generative AI in those domains. "Precedence Research has reported that the worldwide market for generative AI was worth USD 10.79 billion in 2022. It is projected to reach approximately USD 118.06 billion by 2032, with a compound annual growth rate (CAGR) of 27.02% during the period from 2023 to 2032" [11]. However, generative AI for tabular data has been largely left out of this excitement. Partly because fresh tabular data can be less engaging than an image or text to a mass audience, but mainly its inherent hurdles that models, including deep learning models, find difficult to surmount.

Deep learning models struggle on tabular data, despite "achieving great success across various domains, including images, audio, and text" [12]. Text and images inherently have an underlying structure and pattern that can be more easily learnt. The word that comes at the end of a sentence is much easier to predict if you know the preceding words for example, or the outer pixels of an image is more easily predicted if you know the pixels in the center. Tabular data offers no such guarantees, you can't make any specific assumptions about how the variables will relate to one other, or even if they will. Moreover, tabular data is usually a mix of categorical and continuous variables, while image data is only continuous or text is only categorical. Shwartz-Ziv & Armon (2021) demonstrated that the tree-based model XGBoost outperformed deep neural networks for classifying tabular data [12], and Borisov & et al. (2024) demonstrated the same for supervised tasks [13].

The state-of-the-art for generative tasks on tabular data is Conditional Tabular GAN (CTGAN). CTGAN addresses some unique challenges of tabular data, like the fact that you can't assume Gaussian prior for each continuous variable, like you can with spatial variables in image data. Instead it specially normalizes each continuous column separately utilizing a Bayesian Gaussian Mixture Model, addressing the possibility of multi-modal distribution. Additionally it addresses the issue that for minor classes in the database, randomly sampling could present a different distribution for that class compared to the original, which is what the conditional generator is meant to address [3]. DP-CGAN builds on this framework for additional privacy measures [4].

Aside from GANs, another leading deep generative model is the Variational Auto-Encoder (VAE). VAE consists of two components, an encoder and a decoder. The encoder tries to contain all of the information from the input into a 'latent space', values sampled from an assumed prior distribution (normally Gaussian), and the decoder is trained to reconstruct the inputted data only from the latent space. Due to the fact the latent space is regularized to fit a smooth continuous distribution, the network doesn't just train it to encode and reconstruct the original data, but also fill in the gaps between each data point in the latent space, meaning that fresh samples can be created by generating points in the latent space then utilizing the decoder. TVAE is adapted to tabular data by making the same prepossessing steps as CTGAN [3].

B. Research gap

CTGAN improves on other deep learning models mainly through specially normalizing each continuous variable, with the addition of the conditional generator for imbalanced classes. While it does handle continuous and categorical variable separately (applying softmax on the output instead of tanh), it doesn't treat them as inherently disparate forms of storing information. Additionally, the TVAE doesn't make any specific modifications to the architecture, which doesn't address the inherent problem of representing categorical variables in a continuous Gaussian latent space.

1) Deep Boltzmann Machine: Deep Boltzmann Machines (DBMs) are structurally unlike typical feed forward networks, whose architecture is based on supervised back-propagation. Instead, DBMs are based around training an energy function to return a low value when the input matches the data's underlying distribution, and a high value when the input doesn't. DBMs undergo 'wake-sleep' training, where it is fed

positive examples (from the real data) in the wake phase, and rewarded for returning a lower energy. Conversely, it is fed negative examples (randomly sampled) in the sleep phase, and rewarded for returning a higher energy. This approach at training is called 'Contrastive Divergence'. Critically, DBMs work with binary nodes, utilizing probabilities instead of continuous values for granular control. They are known as "a universal approximator of the probability mass function on discrete variables" [14].

The reason they are not more prevalent is their difficulty at training, especially when handling higher dimensional data. Markov chain Monte Carlo (MCMC) methods, chiefly Gibbs sampling, are used to approximate samples from the model that reflect the underlying probability distribution, which can then be compared in the wake-sleep cycle. These approximations have been shown to introduce biases, impacting the ability for the network to converge, mitigated by the iterative MCMC coupling technique [15] in the Unbiased Contrastive Divergence training algorithm. The speed of the UCD algorithm can be improved, especially with high dimensional data, with the use of the Metropolis–Hastings (MH) algorithm instead of Gibbs sampling in the coupling phase. The MH algorithm has a property that it can often converge in 1 step, no matter the dimensionality of the data, so long as the initial state is low energy, so the additional of a search algorithm that finds a local minimum for a state means that DBM machines can feasibly train on high dimensionality data.

C. Variational Autoencoder with Implicit Optimal Priors

A significant drawback with VAEs is the assumption of a prior distribution for the regularization of the latent space. With data as potentially complex and disparate as a collection of continuous tabular variables, you risk losing information by forcing the latent space into that shape. Instead, there exists the concept of an aggregated posterior prior, which essentially captures the overall behaviour of the entire dataset. This prior is considered optimal, however its calculation is intractable. In Takahashi et al. (2018) [16] they propose a VAE network that utilizes the aggregated posterior as its prior, but without having to model it explicitly. VAEIOP estimates the divergence from the aggregated posterior through the use of a discriminator neural network, allowing more flexibility in regularization.

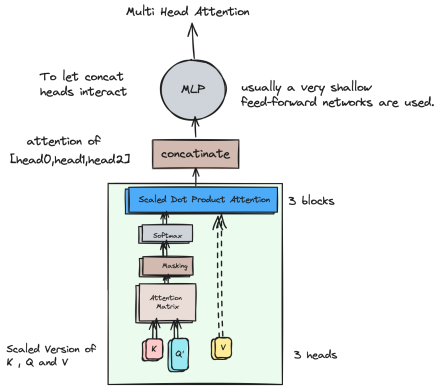
D. BSRBF-KAN

BSRBF-KAN: A combination of B-splines and Radial Basic Functions in Kolmogorov-Arnold Networks.

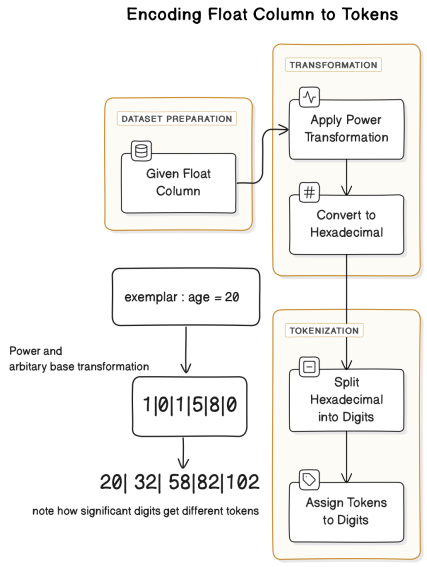
Instead of typical fixed activation functions like ReLU or Sigmoid, Kolmogorov-Arnold Networks (KAN) replaces them with trainable polynomial basis functions, namely B-splines. This introduces further potential for non-linear expressiveness. Upon this base BSRBF-KAN introduces Radial Basis Functions, which can themselves be building blocks for approximating non-linear functions. Together both techniques iron out each others blind spots, with the B-splines capturing fine grained detail, while the RBFs add flexibility and smoothness to the representation [17].

E. Transformers as a Seq-Seq Mode

The attention mechanism, initially crafted for transduction tasks to facilitate connections within autoregressive sequences, has proven versatile enough to apply to any directed graph structure, including tabular data. In this system, each node generates three trainable vectors: a Query (q_i) that identifies the type of node it seeks, a Key (k_i) that details the node's content, and a Value (v_i) that contributes based on the alignment between its query and the other nodes' keys. The self-contained and independent processing nature of self-attention makes it particularly suitable for parallelization in architectures like transformers. Furthermore, the embeddings created by attention mechanisms can be seamlessly integrated into various other models, such as Variational Autoencoders and Generative Adversarial Networks, enhancing their functionality and application scope. The concept of "Multihead"



(a) Illustrates the process of computing Scaled Dot Product Attention mechanism and a multi-head block, which is then concatenated and input into a feed forward network. [18]



(b) Multiple Tokens to Encode a Float column [19]

attention was first introduced by Vaswani et al. in their seminal paper [18]. The core concept of multihead attention involves using several independently initialized attention heads that compute scaled versions of the vectors k_i, q_i, and v_i separately. The outputs from these heads are combined and processed through a shallow feedforward network, facilitating interaction between different weights. This structure allows the model to process diverse information across multiple representational states simultaneously, a capability not possible with a single attention head.

Extending on Attention Mechanisms: An exemplary implementation of advanced attention mechanisms in handling tabular data is the TabTransformer architecture, introduced by Huang et al. [20]. TabTransformer addresses the representation of tabular data by quantizing both categorical and continuous fields. This approach converts each field into a finite vocabulary, akin to tokens in language modeling, allowing for the application of transformer-based models that were originally developed for natural language processing. Categorical fields are naturally suited for quantization as they can be directly converted into discrete tokens that represent different categories or classes within the data. Continuous data fields are quantized to transform them into a discrete set of values, each represented by a unique token. This process involves defining a range or scale that maps continuous values to these tokens. Once quantized, these tokens of continuous data are processed similarly to categorical data, allowing them to be embedded and manipulated within the transformer architecture. TabTransformer embeddings deliver a rich, context-aware representation of each feature within the dataset. By incorporating these embeddings, GANs could gain a deeper insight into the underlying patterns and relationships essential for producing realistic synthetic outputs. Additionally, because TabTransformer embeddings effectively manage both categorical and continuous data, they could enhance GANs' ability to process datasets with diverse data types, thereby improving the authenticity and usability of the synthetic data produced. These embeddings might also stabilize the GAN training process by providing more consistent and interpretable inputs for features. This stability could help address common issues in GAN training, such as mode collapse, where the model struggles to capture the dataset's diversity.

III. Methodology

A. Transformers

Attention mechanisms have been extensively validated across various domains, including text, images, and multimodal scenarios. Their adaptability and success in these areas suggest significant potential for enhancing GANs dedicated to producing synthetic tabular data. In this paper, we have mainly explored two kinds of techniques involving attention to improve the performance of GANs in generating synthetic tabular data.

Approach-1: Conditionally Augmenting the Transformer Embedding into GANs.

This approach integrates Transformer-based embeddings with GAN architecture, inspired by the successes observed in multimodal domains such as text-to-image synthesis. Specifically, Transformer embeddings are processed through a Conditioning Augmentation layer, which generates a multivariate normal distribution. Samples from this distribution, combined with random noise inputs, are fed into the GAN's generator.

This methodology draws on concepts from AttnGANs [21], which have shown promising results in generating images from textual descriptions. However, the adaptation to tabular data presented unique challenges. During trials, it was observed that generator loss could spike or vanish, leading to potential model collapse. This indicates that while the approach holds promise, it requires further refinement to stabilize training dynamics and ensure consistent model performance in the context of tabular data.

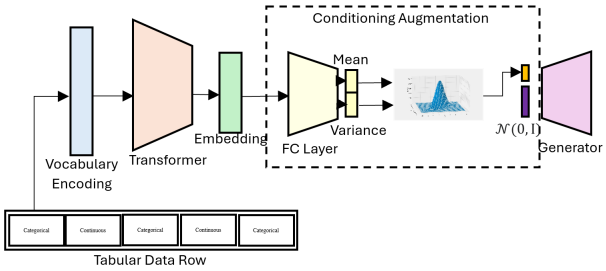


Fig. 2: Conditioning Augmentation with GANs

Approach-2: Integrating Attention between Generator hidden layer and Transformer Embeddings.

This approach modifies the conventional CTGAN framework by incorporating an attention mechanism between the generator's hidden layers and the Transformer embeddings. In this setup, the standard GAN input is augmented by attention-modulated signals at an intermediate layer. The Transformer embeddings provide a context-rich signal that is integrated with the GAN's hidden layer through attention before being upsampled to produce the final output distribution. The key function of the attention module is to dynamically adjust how much emphasis the generator's hidden layers should place on specific features presented by the Transformer embeddings. This modulation is crucial because it allows the model to focus selectively on more informative parts of the embedding, potentially enhancing the relevance and precision of the features that are generated. The attention mechanism can prioritize information from the embeddings that is most likely to improve the generation of realistic and relevant synthetic data points.

This integration aims to enable the model to better account for complex, non-linear relationships within the data, potentially enhancing the quality and authenticity of the generated synthetic data. By allowing the model to "focus" on relevant features dynamically, this approach hopes to improve the generalizability and usefulness of the synthetic data across various applications.

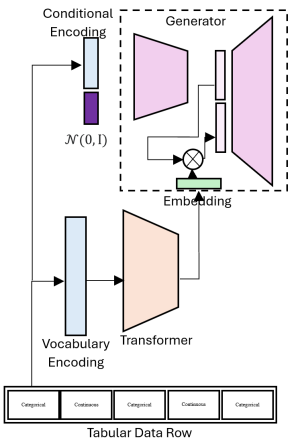


Fig. 3: Attention based GANs

B. Variational Autoencoder with Deep Boltzmann Machine

The VAE-DBM model was trained on the Adult Census dataset only due to time constraints, mainly due to the time needed to run the statistical tests for a new dataset.

1) Preprocessing: To preprocess the data, first the categorical and continuous variables were separated. The continuous variables were normalized using CTGAN's DataTransformer [3], allowing the use of Cluster-Based Normalization with Bayesian Gaussian Mixture Models to account for potential irregular (non-gaussian) distributions. Categorical data was one-hot encoded.

Continuous variables 'capital-gain' and 'capital-loss' were irregular, either a high value or 0, which would be hard for to model without any context as a distribution, so 2 additional categorical variables were added; 'capital-gain-binary' and 'capital-loss-binary', which are 0 if capital-gain or capital-loss is 0 respectively, else 1. These were also intended to help the VAE model understand the relationship between categorical and continuous variables easier, given the direct relationship.

2) DBM implementation: The DBM implementation was taken from the previously discussed paper's Github [14].

Initially the DBM would converge at a fairly high local minimum for its validation loss, despite adding hidden nodes and additional layers. Through experimentation it was found that smaller networks could more easily encode the underlying distribution of the data.

107 input categorical variables. 64 hidden nodes per layer. 3 layers. Batch size 500. trained for 400 epochs.

3) VAE implementation: The VAE implementation was from the implicit optimal prior paper's Github [16].

The base idea for the VAE was to feed it in the isolated continuous variables, and expect it to be able to encode their underlying relationships in a meaningful and effective manner. However with the typical lack of interpretable relationships in tabular data, further isolating the context by only feeding in the continuous variables would make it impossible for the model. The idea with both the DBM and VAE networks is to create a sort of implicit mapping of the subtle dependencies and relationships between variables, so to help the VAE understand the data better the input was modified with an explicit mapping of relationships between variables in the form of a custom convolutional layer.

The convolutional layer has two inputs; the first is for the entire row of input data (including the categorical variables), and the second is the pairwise variable attribute matrix. For each pair of variables statistical tests were recorded that aim to capture different quantifiable aspects of their relationships. Tests like mutual information, distance correlation and HSIC were recorded for each pair of variables. An additional 110 tests for categorical variable pairs from the 'pypair' library were calculated also. Additionally, a binary variable was added for whether the pair of variables were both continuous, both categorical or mixed. From those inputs, BSRBF-KAN layers were utilized for increased capacity for non-linearity, ending into an output of 512 nodes. BSRBF-KAN implementation from the paper's Github [17].

Those 512 nodes serve as the input into the encoder of the VAEIOP. This solves an additional issue that came with feeding in the isolated continuous variables, there were only 5 in the dataset, and to meaningful encode and compact information in the latent space it needs to have less nodes than the input, so the latent space would have had to have been 2 or 3 variables, which could be difficult to store meaningful information inside. Instead the encoder has to encode information from the entire input as well as the mapping of statistical tests. The decoder then is trained to reconstruct the continuous variables from the latent space.

Trained for 9 epochs, 32 sized latent space.

4) Converter: Once the DBM and VAE are fully trained both can be sampled from. However, since there is 2 disparate models, to sample a complete row would mean independently sampling from the categorical and the continuous, with no guarantee that they would match. Instead there needs to be way of connecting the models.

To do that train a converter (built from BSRBF-KAN layers) to convert from categorical variables to their equivalent point in the trained VAE's latent space. So for each row in the data, split the categorical and continuous variables, and encode the continuous variables into the latent space to create a target. Then input the categorical variables into the converter, which outputs a point in the latent space which can be compared against the target.

Once you have a trained converter, sample a row of continuous variables from the DBM, feed it into the converter to find it's equivalent point in the VAE's latent space, then decode that point to get its equivalent continuous sample.

128 hidden dim. trained for 25 epochs.

IV. Experiments

1) Dataset: The two primary datasets evaluated were the Adult Income Dataset [22] and the Bank Marketing Dataset [23]. Both contain a mixture of categorical and continuous variables. The Adult dataset contains significant class imbalances, and insufficient data for many single categories.

The Bank Marketing Dataset contains feature variability and correlation structure among numerical features. The VAE-DBM model was only run on the Adult Income Dataset. There were differences in how each model encoded the database.

A. Utility

A selection of machine learning models were trained on the real data to classify a target variable. Then validated against both the test split of the real data as well as on the synthetic generated data. The aim is for the machine learning model to performs similarly on the real data as with the generated data. This assesses the practicality of substituting real data with synthetic for use cases. The machine models selected were Logistic Regression (LR) and XGBoost (XGB) for their ubiquitous nature in the case of logistic regression providing a good baseline, as well as their natural proficiency with tabular data in the case of XGBoost [12].

For each evaluation, the F1 score, AUC, Cohen's Kappa and Matthews Correlation Coefficient were recorded.

Additionally, SynthEval also calculating the comparison for downstream tasks.

B. Fidelity

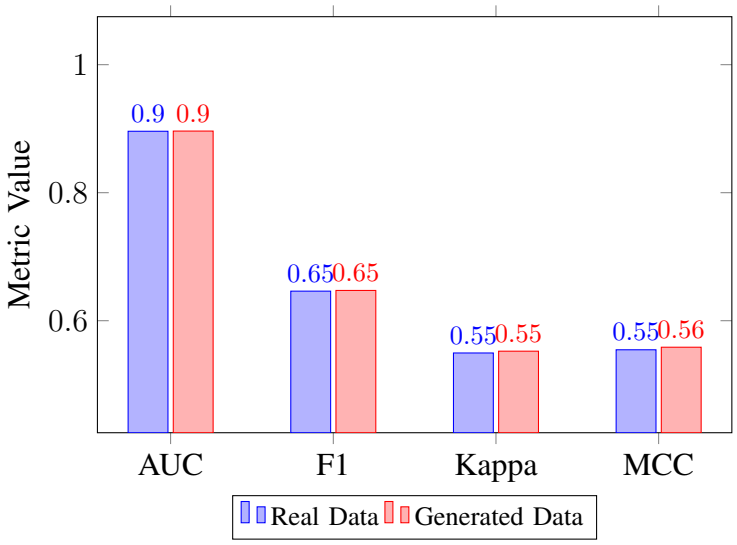
The statistical tests were doing using the SynthEval framework [24]. SynthEval automatically runs a variety of different synthetic data evaluations. Here are the metrics covered:

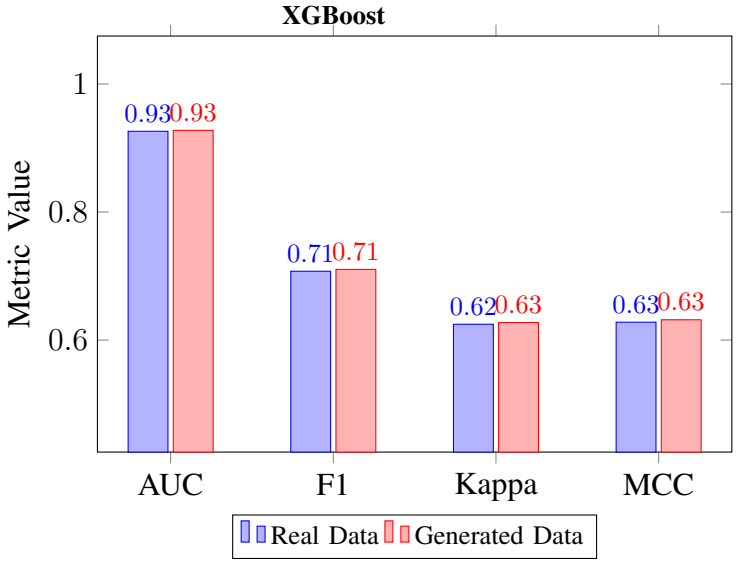
- Dimensionwise Means Difference - average difference between the real and synthetic datasets for numerical variables. Both as a value and a plot.
- Prinicipal Component Analysis, first two components - values for the difference in eigenvalues and angle between eigenvectors. Additionally a plot of both components.
- Average confidence interval overlap.
- Correlation Matrix Difference.
- Pairwise mutual information difference.
- Kolmogorov–Smirnov test
- Empirical Hellinger distance
- Propensity mean squared error (pMSE)
- Nearest neighbour adversarial accuracy

V. Results

A. VAE-DBM

1) Utility: **Logisitic Regression**



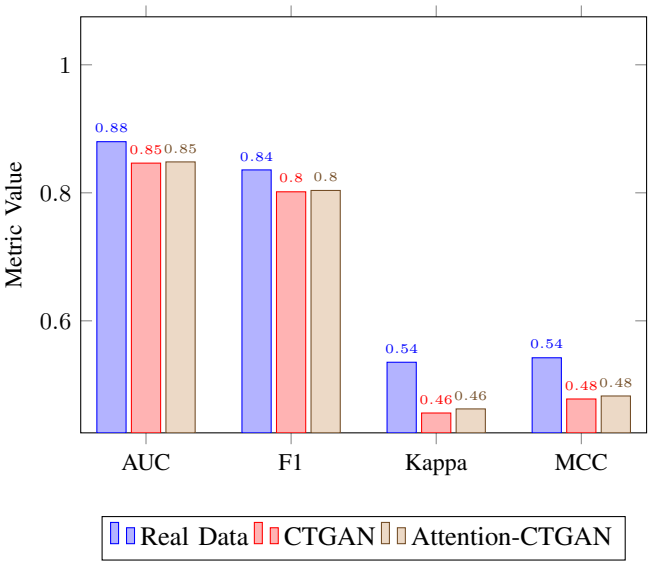


SynthEval's utility tests

Classifier Model	acc_r	acc_f	—diff—	error
DecisionTreeClassifier	0.8168	0.7542	0.0626	0.0157
AdaBoostClassifier	0.8214	0.7676	0.0538	0.0108
RandomForestClassifier	0.8288	0.7794	0.0494	0.0083
LogisticRegression	0.8372	0.7756	0.0616	0.0120
Average	**0.8260**	**0.7692**	**0.0568**	**0.0060**

2) Fidelity: —

Metric	Value	Error
Average difference between means (numerical variables)	0.0145	0.0007
PCA difference in eigenvalues (exp. var.) (numerical variables)	0.0155	
PCA angle between eigenvectors (radians) (numerical variables)	1.5253	
Average confidence interval overlap	0.0000	0.0000
# non-overlapping COIs at 95%	5	
fraction of non-overlapping CIs	1.0000	
Mixed correlation matrix difference	4.3504	
Pairwise mutual information difference	67.7433	
Kolmogorov–Smirnov / Total Variation Distance test		
Average combined statistic	0.0492	0.0095
avg. Kolmogorov–Smirnov dist.	0.4145	0.0966
avg. Total Variation Distance	0.0322	0.0047
average combined p-value	0.0429	0.0120
significant tests at a=0.05	92	
fraction of significant tests	0.8214	
Average empirical Hellinger distance	0.6152	0.0410
Propensity mean squared error (pMSE)	0.2217	0.0002
Average pMSE classifier accuracy	0.9291	0.0011
Nearest neighbour adversarial accuracy	0.8530	0.0007



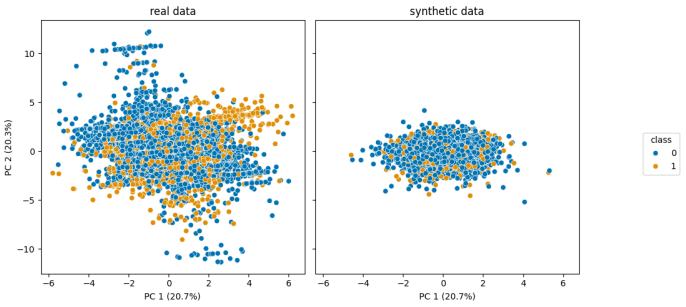
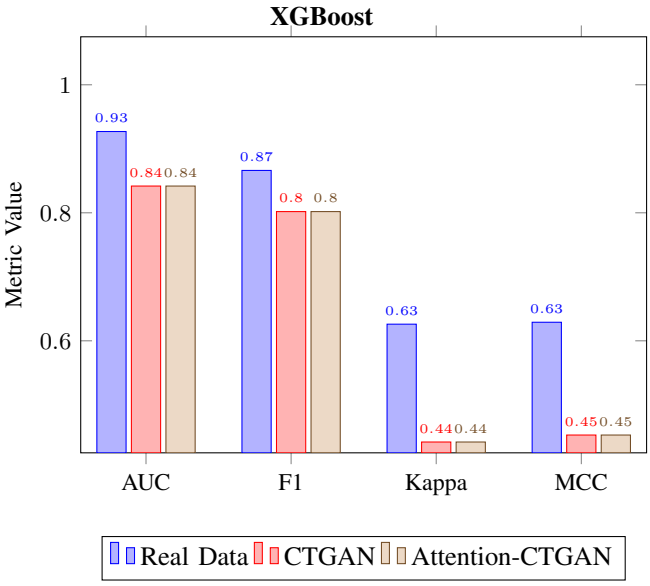


Fig. 4: DBM-VAE: First two principal components plotted (with target class "¿50k" income highlighted)



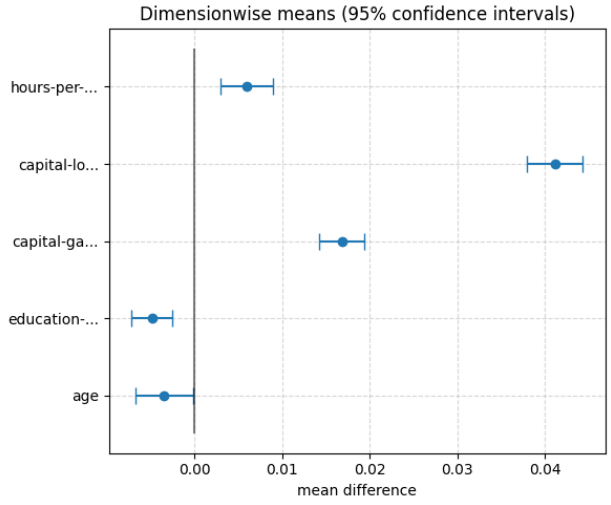


Fig. 5: DBM-VAE: Average difference in mean across continuous variables

SynthEval's utility tests

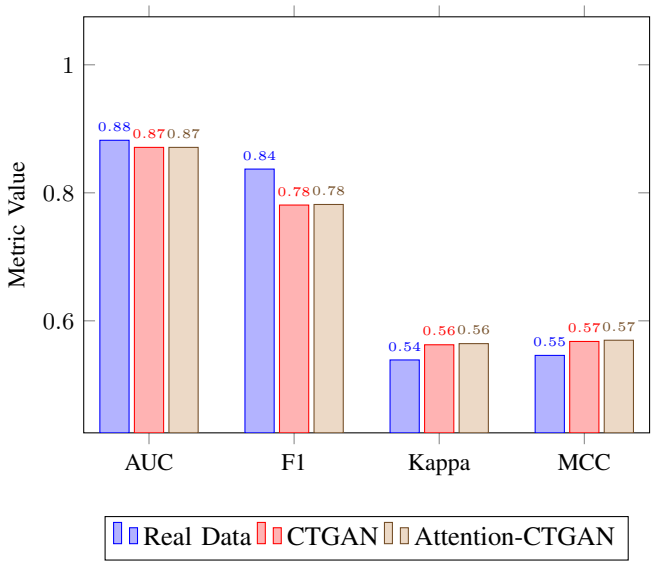
Classifier Model	acc_r	acc_f	—diff—	error
DecisionTreeClassifier	0.8147	0.7892	0.0256	0.0375
AdaBoostClassifier	0.8238	0.8248	0.0010	0.0289
RandomForestClassifier	0.8318	0.8154	0.0163	0.0319
LogisticRegression	0.7990	0.8088	0.0098	0.0384
Average	**0.8173**	**0.8096**	**0.0132**	**0.0342**

B. Attention Aided CT-GAN on Income Dataset

1) Utility: **Logisitic Regression**

2) Fidelity: —

Metric	Value	Error
Average dimensionwise means diff. (nums)	0.0097	0.0004
PCA difference in eigenvalues (exp. var.)	0.0350	
PCA angle between eigenvectors (radians)	0.2715	
Average confidence interval overlap	0.0000	0.0000
# non-overlapping COIs at 95%	6	
Fraction of non-overlapping CIs	1.0000	
Mixed correlation matrix difference	0.6017	
Pairwise mutual information difference	0.8294	
Kolmogorov–Smirnov / Total Variation Distance test		
Average combined statistic	0.1190	0.0329
Avg. Kolmogorov–Smirnov dist.	0.1731	0.0780
Avg. Total Variation Distance	0.0830	0.0147
Average combined p-value	0.0006	0.0001
# significant tests at $\alpha = 0.05$	15	
Fraction of significant tests	1.0000	
Average empirical Hellinger distance	0.0779	0.0169
Propensity mean squared error (pMSE)	0.0144	0.0003
Average pMSE classifier accuracy	0.5999	0.0016
Nearest neighbour adversarial accuracy	0.7193	0.0000



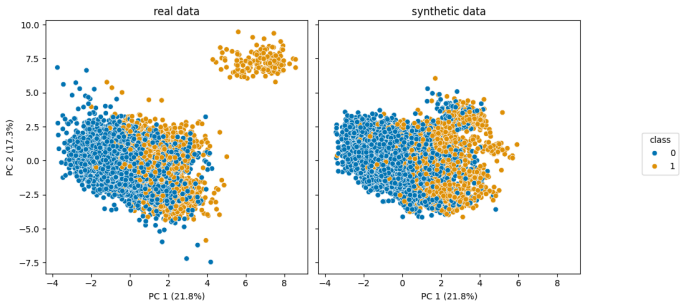
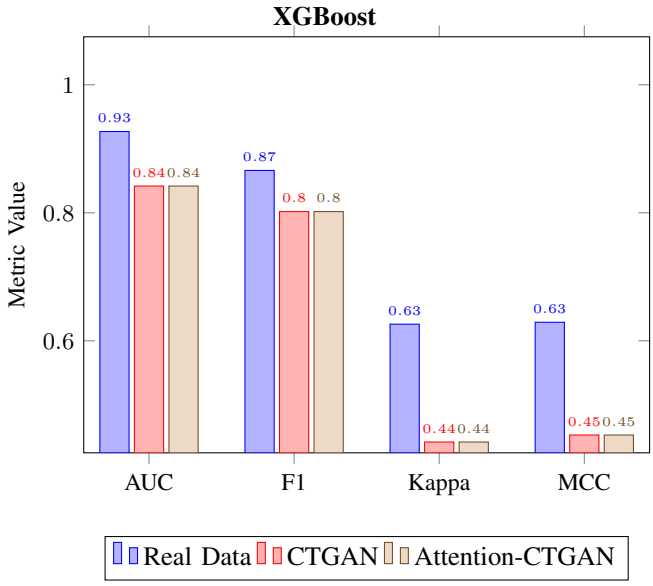


Fig. 6: First two principal components plotted



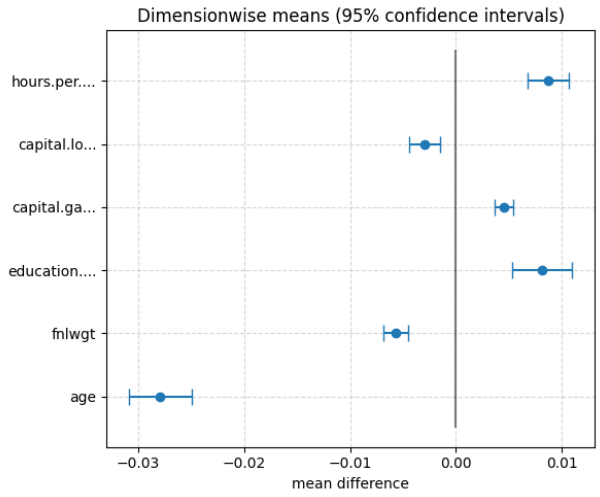


Fig. 7: Average difference in mean across continuous variables

SynthEval's utility tests

Classifier Model	acc_r	acc_f	—diff—	Error
DecisionTreeClassifier	0.6850	0.6472	0.0378	0.0839
AdaBoostClassifier	0.6007	0.7135	0.1128	0.1399
RandomForestClassifier	0.6850	0.6904	0.0054	0.1215
LogisticRegression	0.6049	0.6451	0.0402	0.1700
Average	**0.6439**	**0.6741**	**0.0490**	**0.1288**

C. Attention Aided CT-GAN on Bank Dataset

1) Utility: **Logisitic Regression**

2) Fidelity: —

Metric	Value	Error
Average dimensionwise means diff. (nums)	0.0080	0.0007
PCA difference in eigenvalues (exp. var.)	0.0144	
PCA angle between eigenvectors (radians)	0.1867	
Average confidence interval overlap	0.1499	0.1245
# non-overlapping COIs at 95%	5	
Fraction of non-overlapping CIs	0.7143	
Mixed correlation matrix difference	0.4541	
Pairwise mutual information difference	0.5090	
Average combined statistic	0.0708	0.0095
Avg. Kolmogorov–Smirnov dist.	0.0767	0.0175
Avg. Total Variation Distance	0.0666	0.0111
Average combined p-value	0.0365	0.0359
# significant tests at $\alpha = 0.05$	16	
Fraction of significant tests	0.9412	
Average empirical Hellinger distance	0.0821	0.0170
Propensity mean squared error (pMSE)	0.0168	0.0002
Average pMSE classifier accuracy	0.6138	0.0037
Nearest neighbour adversarial accuracy	0.6129	0.0000

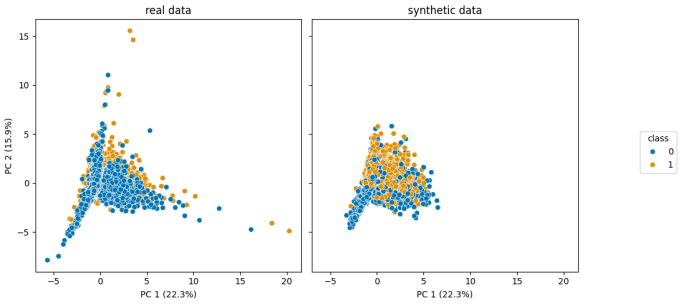


Fig. 8: First two principal components plotted

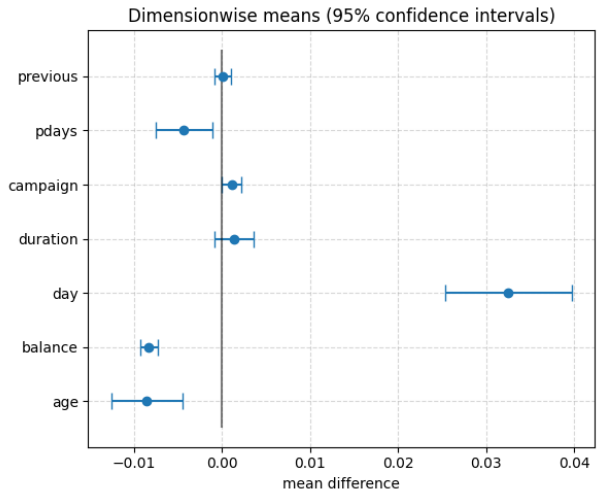


Fig. 9: Average difference in mean across continuous variables

VI. Discussion

A. DBM-VAE

1) Utility: For our utility tests the model proved to be almost indistinguishable from the real data for both classifier models. That is a promising sign that the data generated by the model is fit to augment or substitute for real data for machine learning based tasks. For Syntheval's tests the classifiers found it noticeably harder than with the real data, averaging a 0.05 difference in accuracy.

2) Fidelity: With the level of detail provided by SynthEval the underlying problems with the synthetic generation become clear. Firstly however, the numerical variables are exceeding expectations, an average difference of less than 0.02 for the mean is a good result, especially given that capital loss and capital gain were always going to be hard to model accurately, given its uneven and non-linear distribution. The PCA visualization shows the heart of the issue, the VAE could successfully model the normal data-points collected in the center, but it failed to capture any of the more unpredictable outer edges of data points. The model failed to capture the variance in the original data. No overlap in confidence intervals shows a significant difference between the datasets. The poor results for the correlation matrix and especially the pairwise matrix reflect badly on the models ability to intuit the dependencies between variables, perhaps the fact that the DBM is completely separated from the statistical input could be a factor. The K-S statistic shows a noticeable difference between populations, however the combined p-value is hovering around the significance threshold so while it does reject the null hypothesis that the distributions come from the same source it isn't confident about it. The Hellinger distance also displays significant dissimilarity. The 92.9% pMSE classifier accuracy and the 85.3% nearest neighbour adversarial accuracy show that it's fairly trivial to differentiate between the populations.

These results made me reflect on the choice to simplify the DBM model down to a small number of hidden nodes and layers, while it did lower the validation loss, maybe it ensured that the model would fail to capture the more complex and subtle aspects of the data, which was the whole point of this investigation.

3) Additional comments: In the Adult dataset there are a few categories with many different options, like native country or education. These are one hot encoded in the data which means out of all of those variables one of them must equal 1 and the rest 0, however there's nothing specifically instructing the models of that aside from the fact that it's a pattern inherent to the data. Looking through the generated data the DBM model would sometimes not put 1 for a single choice in a category, or put two 1s instead of one. Given how expansive these categories are it's logical that it would do that, but it does deviate from what we know to be true about the data.

It's unknown to what degree components like the statistical convolutional layer or the converter were beneficial or necessary to reach the final result. This is due to time constraints not allowing for comprehensive structured experimentation. Another big oversight is not being able to test in on other datasets aside from the Adult dataset, particularly ones more challenging for generative models to see how well it can adapt compared to other state of the art models.

These results are with a rushed training time, and lack of

proper investigation as to the optimal hyper parameters and architecture. In terms of improvements to the overall model, avenues for information sharing between the two components would be interesting, as currently interaction is a one way street coming from the DBM feeding into the converter. If there were some way of letting the DBM meaningfully learn from the base statistical mapping that could be impactful given the model's inherent power at capturing dependencies as it is. Additionally with the DBM, finding an architecture or environment that facilitates very wide and deep models to continue to meaningfully learn. The current model contained just 192 hidden nodes in total, but in terms of RAM usage could easily handle tens of thousands of nodes at least.

B. Attention Aided CT-GANs

1) Utility: Evaluating the results for both the logistic regression as well as the XGBoost reveals that there's always a slight discrepancy between the results for the classifier on the real data compared to the CTGAN generated data, with or without attention. Interestingly, there's a bigger discrepancy for the Kappa and MCC statistics over AUC and F1 statistics, AUC and F1 are more focused on overall accuracy, while Kappa and MCC are more sensitive to class imbalance. This suggests that the data CTGAN generated had a different class distribution than the original data.

The discrepancy mainly exists for our utility evaluation but not SynthEval's utility tests, where CTGAN with attention mechanism was fairly indistinguishable from the original data, with only a 0.01 average difference for the Income dataset, and only 0.05 for the Bank dataset, though with fairly significant error.

2) Fidelity: **Adult Income dataset** The synthetic data generated was closely aligned with the original data. The average difference in mean for continuous variables was only 0.01. The correlation matrix and pairwise mutual information results show it understood the underlying statistical relationships between variables to a relatively high degree. The K-S and Total Variance test however has the p-value near 0 at 0.006, showing it confidently rejects the assertion that the samples came from the same distribution. The direct measures of distribution similarity, like Hellinger distance or pMSE are all low. The classifiers still achieved a good accuracy in terms of differentiating between samples.

From the PCA visualization we can see there was a separate cluster of data-points that CTGAN merged into one cluster. CTGAN does seem to preserve the split in placement between the target class. For the continuous variables their means were close to the original data with the exception of age.

Bank dataset The generated data was even more closely aligned in the case of the Bank dataset. Every single statistic is either equal the Adult dataset or even lower. Still the KS test rejected the hypothesis that the samples came from the same distribution. By direct distribution distrance (pMSE and Hellinger distnace) the sampled data are quite close, and the classifiers only had a slight probability of identifying correctly.

We can see from the PCA visualization that the target class follows along the edge of the original data, but with the synthetic data it's been merged into the center.

The synthetic continuous variable's means were close to the original with the exception of day.

The only time base CTGAN was directly compared to Attention-CTGAN they displayed practically equivalent results so it's inconclusive to what extent the attention mechanisms benefited the model.

C. Comparison of models

CTGAN seems to uniformly outperform DBM-VAE outside of our utility experiments, and the K-S statistic p-value. In terms of the utility experiments, the results are interesting though one would place more weight on SynthEval's utility tests over our implementaion.

The p-value difference could suggest that while the absolute difference in distribution is larger for the DBM-VAE, the discrepancies are more consistent for CTGAN and are more unlikely to be because of random chance. This is even despite DBM's KS distance being 0.41, while on the Bank dataset CTGAN's was 0.08, yet DBM's p-value was still higher.

Looking at the Adult Census dataset, the capital-loss and gain variables stood out. If you look at the data the values in those columns are mostly 0, but occasionally are very high values in the thousands. This seems like a difficult distibution to generate from but for CTGAN those were the variables closest to the original data. For the DBM it was the opposite. This could be because CTGAN is better designed for handling unorthodox continuous distributions, aside from the initial multi modal clustering normalization, which the DBM model incorporated also.

The encoding was different for DBM-VAE's Adult Income dataset compared to CTGAN, but the difference in how the target class looks on the PCA visualization is quite stark. For CTGAN the class is evenly split through the middle, while for DBM-VAE it's spread across the visual. In both cases the synthetic data preserved the overall pattern of target class distribution. This could be because the first two components that were selected for the DBM happened to contain different information than the two selected for the CTGAN.

VII. Conclusion

This research explored novel approaches to enhancing the generation of synthetic tabular data. By identifying and focusing on the inherent hurdles with tabular data, namely mixed variables and intricate relationships between them, we aimed to open up potential avenues and perspectives on how to extract the underlying information from this type of data.

There isn't enough information to conclude that the transformer enhanced CTGAN outperforms the base model by itself, however it did prove to be able to replicate both the Adult and Bank databases to a high degree of sophistication.

The VAE-DBM model did outperform the state-of-the-art CTGAN baseline in our utility experiments, near exactly matching the original data, though it was unable to provide

the same results with SynthEval's tests. In terms of fidelity the VAE-DBM model under-performed compared to the CTGAN with the sole exception of the KS/Total Variance p-value.

Overall, the research highlighted the potential for capturing complex categorical and continuous distributions through Deep Boltzmann Machines and Variational Auto-Encoders. Given the time constraints and lack of experience with these specific models, the model performing as well as it did to the baseline shows promise. Undoubtedly, the joint architecture detailed here can be optimized and improved upon.

REFERENCES

[1] M. Hernadez, G. Epelde, A. Alberdi, R. Cilla, and D. Rankin, "Synthetic Tabular Data Evaluation in the Health Domain Covering Resemblance, Utility, and Privacy Dimensions," *Methods Inf Med*, vol. 62, pp. 19–38, 2023. [Online]. Available: https://doi.org/

[2] D. Manousakas, S. Serg¨, and S. Aydöre, "On the Usefulness of Synthetic Tabular Data Generation," 2023.

[3] L. Xu, M. Skoularidou, A. Cuesta-Infante, and K. Veeramachaneni, "Modeling Tabular Data using Conditional GAN." [Online]. Available: https://github.com/DAI-Lab/CTGAN

[4] C. Sun, J. van Soest, and M. Dumontier, "Improving Correlation Capture in Generating Imbalanced Data using Differentially Private Conditional GANs," 6 2022. [Online]. Available: https://arxiv.org/abs/2206.13787v1

[5] D. Shin, D. Han, and S. Kyeong, "Performance Enhancement of Malware Classifiers Using Generative Adversarial Networks," *Proceedings - 2022 IEEE International Conference on Big Data, Big Data 2022*, pp. 6528–6533, 2022.

[6] J. G. Choi, Y. Nah, I. Ko, and S. Han, "Deep Learning Approach to Generate a Synthetic Cognitive Psychology Behavioral Dataset," *IEEE Access*, vol. 9, pp. 142 489–142 505, 2021.

[7] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," in *36th International Conference on Machine Learning, ICML 2019*, vol. 2019-June, 2019.

[8] T. Liu, Z. Qian, J. Berrevoets, and M. Van Der Schaar, "GOGGLE: GENERATIVE MODELLING FOR TABULAR DATA BY LEARNING RELATIONAL STRUCTURE."

[9] J. Jordon, L. Szpruch, F. Houssiau, M. Bottarelli, G. Cherubin, C. Maple, S. N. Cohen, and A. Weller, "Synthetic Data – what, why and how?" 2022. [Online]. Available: https://arxiv.org/abs/2205.03257

[10] T. E. Raghunathan, "Synthetic data," *ANNUAL REVIEW OF STATISTICS AND ITS APPLICATION*, 2020. [Online]. Available: https://www.annualreviews.org/content/journals/10.1146/annurev-statistics-040720-031848

[11] A. Bandi, P. V. S. R. Adapa, and Y. E. V. P. K. Kuchi, "The power of generative ai: A review of requirements, models, input–output formats, evaluation metrics, and challenges," *State-of-the-Art Future Internet Technology in USA 2022–2023*, 2023. [Online]. Available: https://www.mdpi.com/1999-5903/15/8/260

[12] R. Shwartz-Ziv and A. Armon, "Tabular data: Deep learning is not all you need," 2021. [Online]. Available: https://arxiv.org/abs/2106.03253

[13] V. Borisov, T. Leemann, K. Seßler, and J. Haug, "Deep neural networks and tabular data: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, 2022. [Online]. Available: https://www.researchgate.net/publication/366552986_Deep_Neural_Networks_and_Tabular_Data_A_Survey

[14] S. Taniguchi, M. Suzuki, Y. Iwasawa, and Y. Matsuo, "End-to-end training of deep boltzmann machines by unbiased contrastive divergence with local mode initialization," 2023. [Online]. Available: https://arxiv.org/abs/2305.19684

[15] Y. Qiu, L. Zhang, and X. Wang, "Unbiased contrastive divergence algorithm for training energy-based latent variable models," 2019. [Online]. Available: https://openreview.net/forum?id=r1eyceSYPr

[16] H. Takahashi, T. Iwata, Y. Yamanaka, M. Yamada, and S. Yagi, "Variational autoencoder with implicit optimal priors," 2018. [Online]. Available: https://arxiv.org/abs/1809.05284

[17] H.-T. Ta, "Bsrbf-kan: A combination of b-splines and radial basic functions in kolmogorov-arnold networks," 2024. [Online]. Available: https://arxiv.org/abs/2406.11173

[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need," 6 2017.

[19] Y. Dong and E. Scoullos, "Generating synthetic data with transformers: A solution for enterprise data challenges," May 2022, accessed: 2024-06-26. [Online]. Available: https://developer.nvidia.com/blog/generating-synthetic-data-with-transformers-a-solution-for-enterprise-data-challenges/

[20] X. Huang, A. Khetan, M. Cvitkovic, and Z. Karnin, "TabTransformer: Tabular Data Modeling Using Contextual Embeddings," 12 2020.

[21] T. Xu, P. Zhang, Q. Huang, H. Zhang, Z. Gan, X. Huang, and X. He, "AttnGAN: Fine-Grained Text to Image Generation with Attentional Generative Adversarial Networks," 11 2017.

[22] B. Becker and R. Kohavi, "Adult," 1996. [Online]. Available: https://archive.ics.uci.edu/dataset/2/adult

[23] S. Moro, P. Rita, and P. Cortez, "Bank Marketing," 2012. [Online]. Available: https://archive.ics.uci.edu/dataset/222/bank+marketing

[24] A. D. Lautrup, T. Hyrup, A. Zimek, and P. Schneider-Kamp, “Systematic review of generative modelling tools and utility metrics for fully synthetic tabular data,” 2024.