Evaluating Generative Models for Synthetic Tabular Data: A Comparative Analysis of Fidelity, Diversity, and Generalization

Zoran Mahovac, Andrija Petrović, Sandro Radovanović, Boris Delibašić

Faculty of Organizational Sciences, Jove Ili´ca 154, Belgrade, 11000, Serbia
zm20200109@student.fon.bg.ac.rs, andrija.petrovic@fon.bg.ac.rs,
sandro.radovanovic@fon.bg.ac.rs, boris.delibasic@fon.bg.ac.rs

Abstract: Tabular data, such as relational tables, Web tables and CSV files, is among the most primitive and essential forms of data in machine learning, characterized by excellent structural properties, readability, and interpretability. However, acquiring substantial amounts of high-quality tabular data for ML model training remains a persistent challenge. This study evaluates the performance of six generative models — TVAE, RTVAE, CTGAN, ADSGAN, BNN, and Marginal Distributions on synthetic data generation. The evaluation is based on three key metrics: Fidelity, Diversity, and Generalization. Fidelity measures the quality of synthetic data, Diversity assesses how well the samples cover the variability of the real dataset, and Generalization quantifies the risk of overfitting. The research applies these metrics to four datasets: Abalone, Acute Inflammation, Census Income, and Pittsburgh Bridges. Results show that CTGAN consistently outperforms other models measured by IPα and IRβ metrics, while RTVAE excels in the Census Income dataset in terms of Generalization. Marginal Distributions stands out in preserving data authenticity. This study offers a refined method of evaluating generative models, emphasizing precision-recall analysis grounded in minimum volume sets, thus providing a deeper understanding of model performance across multiple dimensions.

Keywords: Generative models, synthetic data, Fidelity, Diversity, Generalization, tabular data generation

1. Introduction

Generative AI (GenAI) is a class of machine learning (ML) algorithms that can learn from content such as text, images, and audio in order to generate new content. In contrast to discriminative ML algorithms, which learn decision boundaries, GenAI models produce artifacts as output, which can have a wide range of variety and complexity. [1] A central problem in statistical inference is to calculate a posterior distribution of interest. Given a likelihood function, $p(y \mid \theta)$ or a forward model $y = f(\theta)$, and a prior distribution $\pi(\theta)$, the goal of an inverse probability calculation is to compute the posterior distribution $p(\theta|y)$.

This is notoriously hard for high-dimensional models. MCMC methods solve this by generating samples from the posterior using density evaluation. Generative AI techniques, on the other hand, directly learn the mapping from the a uniform to the distribution of the interest. The main advantage of generative AI is that it is model free and doesn't require the use of iterative density methods. [2]

Tabular data, such as relational tables, Web tables and CSV files, is among the most primitive and essential forms of data in machine learning (ML), characterized by excellent structural properties, readability, and interpretability. [3] However, acquiring substantial amounts of high-quality tabular data for ML model training remains a persistent challenge. [4] Additionally, in the industrial sector where tabular data is most commonly used, the availability of data is often limited due to privacy concerns. [5]

In this paper, we aimed to investigate to what extent generative models applied to tabular data can synthesize data that has high fidelity, meaning it belongs to the distribution of real data, is sufficiently divergent to describe the entire variability of the original data, and demonstrates the model's ability to generalize, i.e., not just produce copies of the training data. Generative models used for synthetic data generation include Bayesian Neural Network (BNN), Adversarial Deep Synthetic Generative Adversarial Network (ADSGAN), Conditional Generative Adversarial Network (CTGAN), Tabular

Variational Autoencoder (TVAE), Recurrent Temporal Variational Autoencoder (RTVAE), as well as a simpler model that learns the marginal distributions of all attributes individually. We used the implementation of these models from the synthcity [6] library.

To evaluate the performance of the generative models, we used metrics that rep-resent its performance as points in a 3-dimensional space, where each dimension corresponds to a different aspect of the model's quality. These qualities are: Fidelity, Diversity and Generalization. Fidelity measures the quality of a model's synthetic samples, and Diversity is the extent to which these samples cover the full variability of real samples, whereas Generalization quantifies the extent to which a model overfits (copies) training data. Since statistical comparisons of complex data types in the raw input space are difficult, the evaluation pipeline starts by embedding original training data and synthetic data into a meaningful feature space through an evaluation embedding function, and evaluating Fidelity, Diversity and Generalization on the embedded features [7] We verified the effectiveness of these models on four datasets: Abalone, Acute Inflammations, Pittsburgh Bridges, and Census Income.

2. Related work

Generative Models Based on Variational Autoencoder (VAE) VAEs are generative models that adopt variational inference and graphical models. VAE has two components: an encoder and a decoder. The encoder transforms high dimensional data to a low-dimensional latent space with an approximate tractable posterior distribution. The decoder samples from this distribution and transforms the sample back to the original dimension. [8] In variational methods, the functions used as prior and posterior distributions are restricted to those that lead to tractable solutions. For any choice of a tractable distribution q(Z), the distribution of the latent variable, the following decomposition holds:

$$\log \mathrm{p}_\theta\,(\mathrm{X}) = \mathcal{L}(\mathrm{q}(\mathrm{Z}), \theta) + \mathrm{D}_{\mathrm{KL}}\big(\mathrm{q}(\mathrm{Z}) \parallel \mathrm{p}_\theta(\mathrm{Z}|\mathrm{X})\big) \tag{1}$$

where DKL represents the Kullback-Leibler (KL) divergence. Instead of maximizing the loglikelihood $p_\theta(X)$, with respect to the model parameters θ, the variational inference approach maximizes its variational evidence lower bound ELBO [9]:

$$\mathcal{L}(\mathrm{q}, \theta) = \mathrm{E}_{\mathrm{q}(\mathrm{Z})}[\log \mathrm{p}_\theta\,(\mathrm{X}|\mathrm{Z})] - \mathrm{D}_{\mathrm{KL}}\big(\mathrm{q}(\mathrm{Z}) \parallel \mathrm{p}_\theta(\mathrm{Z})\big). \tag{2}$$

The Synthcity implementation of TVAE is based on a combination of a tabular encoder and a VAE to form a generative model for tabular data.

Notable approach in tabular data generation includes a novel approach based on Variational Autoencoders (VAEs) enhanced with a Bayesian Gaussian Mixture (BGM) model, which trains the VAE conventionally and then applies the BGM model to the learned latent space. This allows for a more accurate representation of the underlying data distribution during data generation. Moreover, this approach offers enhanced flexibility by accommodating various differentiable distributions for individual features, enabling the handling of continuous and discrete data types. [10]

An additional noteworthy proposition is a robust variational autoencoder with β divergence for tabular data (RTVAE) with mixed categorical and continuous features. In order to ensure robust variational inference, this approach modifies reconstruction term LREC in order to make it more robust to outliers for categorical data. In this approach, the empirical distribution $\hat{p}(X)$ is defined as a sum of Dirac delta functions representing discrete data points. Then, KL-divergence is used, which measures the difference between this empirical distribution and the generative distribution $p_\theta(X|z)$. In essence, KL-divergence quantifies the difference between the true data distribution and the model's distribution. KL-divergence is sensitive to outliers because for data with very low probability (e.g., outliers), the negative log-likelihood can become very large, disrupting the model's learning process. Instead of using KL-divergence, β-divergence is used, which is defined as [11]:

$$D_\beta\big(\hat{p}(X) \parallel p_\theta(X|z)\big) = \frac{1}{\beta}\int \hat{p}(X)^{\beta+1}dX - \frac{\beta}{\beta+1}\int \hat{p}(X)p_\theta(X|z)^{\beta}dX + \int p_\theta(X|z)^{\beta+1}dX \tag{3}$$

β-divergence is a generalization of KL-divergence, which converges to KL-divergence as β → 0. By adjusting the β parameter, this approach becomes more robust to outliers, providing greater flexibility in modeling the data distribution. β-divergence is related to β-cross-entropy, which measures the similarity between the real and modeled distributions. Therefore, the reconstruction loss is expressed as [11]:

$$\mathcal{L}_{\mathcal{REC}-\beta} = \mathrm{E}_{\mathrm{z} \sim \mathrm{q}_{\phi}(\mathrm{z}|\mathrm{x})} \left[\mathcal{H}_{\beta}(\mathrm{x_i}|\mathrm{z})\right] \tag{4}$$

Generative Adversarial Networks (GAN). In this framework two models are trained simultaneously: a generative model G that captures the data distribution, and a discriminative model D that estimates the probability that a sample came from the training data rather than G. The training procedure for G is to maximize the probability of D making a mistake. [12] To learn the generator's distribution p_g over data x, we define a prior on input noise variables $p_z(z)$, then represent a mapping to data space as $G(z; \theta_g)$, where G is a differen-tiable function represented by a multilayer perceptron with parameters θg. We also define a second multilayer perceptron $D(x; \theta_d)$ that outputs a single scalar. $D(x)$ represents the probability that x came from the data rather than pg. In other words, D and G play the following two-player minimax game with value function V(G, D): [12]

$$\min_{G} \max_{D} V\,(D, G) = E_{x \sim p_{\mathrm{dt}}(x)}[\log D\,(x)] + E_{z \sim p_z(z)} \left[\log\left(1 - D\big(G(z)\big)\right)\right] \tag{5}$$

Tabular data usually contains a mix of discrete and continuous columns. Continuous columns may have multiple modes whereas discrete columns are sometimes imbalanced making the modeling difficult. Existing statistical and deep neural network models fail to properly model this type of data. A conditional GAN (CTGAN) as a synthetic tabular data generator is proposed to address issues mentioned above. Xu et al. [13] invented mode-specific normalization to over-come the non-Gaussian and multimodal distribution. This method is applied for each continuous column in three steps; (1) for each continuous column Ci variational Gaussian mixture model (VGM) is being used to estimate the num-ber of modes mi and fit a Gaussian mixture. The learned Gaussian mixture is $P_{C_i}(c_i, j) = \sum_{k=1}^{3} \mu_k \, N\big(c_{i,j}; \eta_k, \phi_k\big)$, where μ_k and ϕ_k are the weight and standard deviation of a mode, respectively. (2) Then, for each value $c_{i,j}$ in C_i, compute the probability of $c_{i,j}$ coming from each mode. (3) Sample one mode from given the probability density and use the sampled mode to normalize the value. [13] The traditional GAN generator takes a random vector sampled from a standard multivariate normal distribution (MVN) as input. During training, the generator learns to map this distribution to the data distribution. However, this approach does not account for the potential imbalance in categorical variables. If the data is randomly sampled, underrepresented categories may not be sufficiently represented, leading to poor generator training. Alternatively, if the data is resampled to balance categories, the generator may learn an altered distribution that does not reflect the true data distribution. The solution to this issue is the use of a Conditional Generator in GANs. It generates data conditioned on specific categorical values, ensuring balanced representation during training. This approach learns the true conditional distribution of the data, maintaining the integrity of the original data distribution during both training and testing.

In the medical and machine learning communities, AI has the potential to trans-form personalized treatment and decision-making. However, legal and ethical concerns regarding unconsented patient data and privacy limit data sharing, restricting access to electronic health records (EHR). To address this, a novel framework, ADS-GAN, uses conditional generative adversarial networks (GANs) to generate synthetic data that closely approximates the original EHR dataset while minimizing patient identifiability. The framework introduces a quantifiable definition of ”identifiability” based on re-identification probabilities. ADS-GAN outperforms existing methods, providing a legally and ethically sound solution for open data sharing and AI development. [14]

Bayesian Neural Networks (BNN). Bayesian Neural Networks (BNNs) rep-resent a powerful generative approach that integrates uncertainty estimation into the modeling process. Unlike standard neural networks, which learn fixed weight values, BNNs assign probabilistic distributions to weights, enabling a principled way to quantify uncertainty. This characteristic makes them particularly useful for synthetic data generation, where capturing both the underlying data struc-ture and variability is essential. A natural extension of BNNs in this context involves their formulation within Directed Acyclic Graphs (DAGs), where prob-abilistic dependencies between variables are explicitly modeled. By structuring BNNs as DAGs, the generative process becomes interpretable, as each node rep-resents a latent or observed variable, and directed edges capture conditional dependencies. This setup is particularly advantageous in tabular data synthesis, where relationships between features are often complex and hierarchical. [16]

3. Experimental setup

The main goal of this research is to evaluate and compare the performance of different generative models for synthetic data in the context of their ability to represent the structure of the original data. Specifically, the models TVAE, RT-VAE, CTGAN, ADSGAN, BNN, and Marginal Distributions are analyzed using the metrics Fidelity, Divergence, and Generalization. Fidelity measures the quality of a model's synthetic samples, and Diversity is the extent to which these samples cover the full variability of real samples, whereas Generalization quantifies the extent to which a model overfits (copies) training data. [7] This is an alternative approach to evaluating generative models, where instead of assessing the generative distribution by looking at all synthetic samples collectively to compute likelihood or statistical divergence, we classify each sample individually as being of high or low quality. [7] The main contribution proposed by Alaa et al [7] is a refined precision-recall analysis of the Fidelity and Diversity performance of generative models that is grounded in the notion of minimum volume sets. Given a probability measure P and a reference measure μ, one is often interested in the minimum μ - measure set with P - measure at least α. Minimum volume sets of this type summarize the regions of greatest probability mass of P, and are useful for detecting anomalies and constructing confidence regions. [15] As mentioned, the approaches were evaluated on the following datasets: the Abalone, Acute Inflammation, Census Income, and Pittsburgh Bridges datasets.

We first preprocessed the data using two distinct pipelines: one for continuous variables, and another for categorical variables. The data were then split into training and test sets with an 80:20 ratio. Hyperparameter optimization was performed by sampling hyperparameter values over five trials, with the best parameters selected based on evaluation metrics. We trained the evaluation neural network on the training data, while synthetic data were passed through the trained network, and evaluation metrics were computed on both the original and synthetic embeddings.

4. Discussion

In this section, we will discuss the data preprocessing steps applied to handle continuous and categorical features using different strategies for imputation and transformation. We will also cover the neural network architecture imple-mentation used for embedding representations and how it was trained. Finally, we will describe results of the following approaches: TVAE, RTVAE, CTGAN, ADSGAN, BNN, and MD using evaluation metrics Fidelity, Diversity, and Generalization on a dataset that includes the Abalone, Acute Inflammation, Census Income, and Pittsburgh Bridges datasets.

Data preprocessing. For data preprocessing, two pipelines were created: (1) for continuous data, which involves filling missing values with *SimpleImputer* using the constant strategy and *StandardScaler* for data standardization, and (2) for categorical data, which uses *SimpleImputer* with the most frequent strategy, along with *OneHotEncoder* for transformation. For each dataset, the data were split with a training-to-test ratio of 80:20. Hyperparameter optimization was performed by sampling hyperparameter values for each model from their respective hyperparameter spaces in five trials, with the best hyperparameters selected based on the evaluation metric values when comparing the training dataset and synthesized examples. The network was trained on the training data, while the synthetic data were passed through the trained network, and evaluation metrics were computed on the original and synthetic embeddings.

Neural Architecture Evaluator. To evaluate the performance of all mod-els, both synthetic and original training data were transformed into embedding representations using a neural network. The architecture of the network was determined by modules composed of Dropout, Linear, and Nonlinear layers (activation functions). Our architecture consisted of 3 such modules with a dropout probability of 0.4. The hidden layer dimension of the linear transformations was set to 200, and the ReLU activation function was used for the nonlinear trans-formation. The embedding representations were centered around a point defined by a vector of value 1 in all dimensions, ensuring they belong to the interior of a hypersphere with a radius of 1. The learning rate parameter used when fitting the neural network was 0.0001. The neural network was trained for 100 epochs using the AdamW optimizer with a weight decay parameter value of 0.01.

Fidelity implementation. The implementation of evaluation metrics is based on [7]. IPα is implemented by integrating the absolute difference between α -precision and the alpha value describing the support, over all α - support values from 0 to 1 and a scalar in the range [0, 1/2] is obtained. α - precision is implemented by calculating the Euclidean distance of each real point from the center of the hypersphere using the norm function from the torch library, and the radius containing α * 100% of the training embedding distances from the center of the hypersphere is returned by the quantile function. Then, the Euclidean distances of the synthetic points' embeddings from the center of the hypersphere are calculated, and the total number of such distances less than the radius describing the α - quantile of the training points' embeddings is summed (the total number of reconstructions belonging to the training support). Finally, α precision is defined as the proportion of these points in the total number of embeddings of synthetic points.

Diversity implementation. For the implementation of the Diversity metric, the center of the synthetic data embeddings was calculated as the mean of all the vectors across all dimensions. A radius was determined that encompasses the beta quantile of all distances between the synthetic data embeddings and their center. All embeddings of synthetic data points belonging to the beta support, defined by the hypersphere with the center and radius described above, were filtered. Using these, a *NearestNeighbors* model was fitted, and for each training data embedding, the closest embedding from the synthetic data was found. Additionally, another *NearestNeighbors* model was fitted on the embeddings of real data, with the number of neighbors set to k = 5. Finally, if the distance from the closest point in the beta support to the real data embedding is smaller than the distance of that embedding to its k nearest neighbors, it can be concluded that the variability of the real data point is represented in the synthetic examples. The proportion of real data points whose variance is explained by the synthetic examples, relative to the total number of real data points for a given value of the beta support, represents the beta recall.

Generalization implementation. The Generalization metric was also calculated by comparing the distances between real and synthetic data point embed-dings using the *NearestNeighbors* algorithm. For each real data point, the closest real neighbor was identified, and the distance to its nearest real neighbor was recorded. Similarly, for each synthetic data point, the distance to its closest real data point was calculated. A mask was then applied to check if the synthetic data point's distance to the nearest real neighbor was greater than the real data point's distance to its nearest neighbor. The authenticity ratio was computed as the proportion of synthetic data points satisfying this condition, indicating how well the synthetic data approximates real data.

The implementation of generative models is based on the implementation of existing models from the Synthcity [6] library.

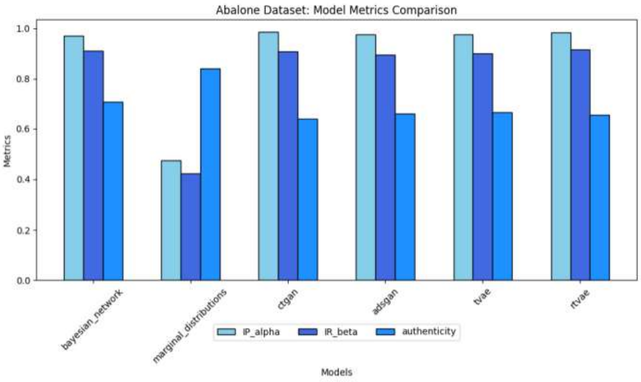

Figure 1. Model metrics comparison on Abalone Dataset

Results. For the Abalone dataset, the CTGAN model emerged as the best, achieving the highest values in IPα (0.9856) and IRβ (0.9076), with an Authenticity score of 0.6411. This model demonstrated exceptional effectiveness in pre-serving information and data representation. On the other hand, the Marginal Distributions model showed significantly lower values for IPα and IRβ, although it excelled in the Authenticity metric (0.8402), indicating its ability to preserve the authenticity of the data, even though it wasn't the most efficient in other metrics. The visual representation of the metric values for the Abalone dataset is shown in 1.

In the Census Income dataset, the RTVAE model showed outstanding performance with values of 0.8475 for IPα, 0.7578 for IRβ, and 0.75 for Authenticity, making it the most successful model in this dataset. Marginal Distributions once again showed weakness in the IPα (0.2899) and IRβ (0.5831) metrics, but recorded a very good result for Authenticity (0.8583), suggesting its ability to generate authentic data. The visual representation of the metric values for the Census Income dataset is shown in 2.

For the Acute Inflammations dataset, the CTGAN again achieved the best results with an IPα value of 0.9382, IRβ value of 0.7906, and Authenticity value of 0.5729. These results highlight its high effectiveness in preserving information and generating relevant data. In comparison, the Bayesian Network had a higher value for IRβ (0.8449), but did not achieve the best results in other metrics. The visual representation of the metric values for the Acute Inflammations dataset is shown in 3.

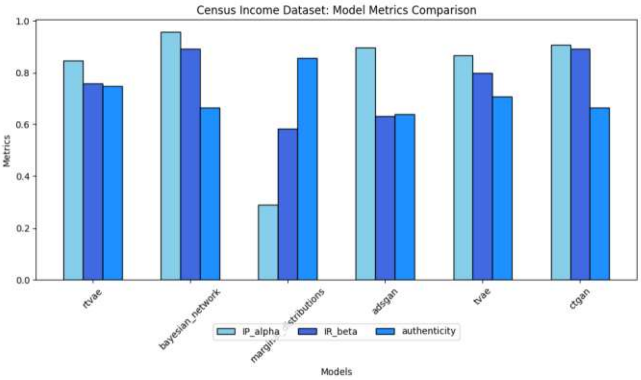

Figure 2. Model metrics comparison on Census Income Dataset

On the final dataset, Pittsburgh Bridges, CTGAN once again demonstrated exceptional results, with the highest values for IRβ (0.8733) and Authenticity (0.7143), while RTVAE delivered solid values across all three metrics, but did not reach the same heights as CTGAN. ADSGAN had the lowest values in all metrics, indicating that this model was not the most efficient on this dataset. The visual representation of the metric values for the Pittsburgh Bridges dataset is shown in 4.

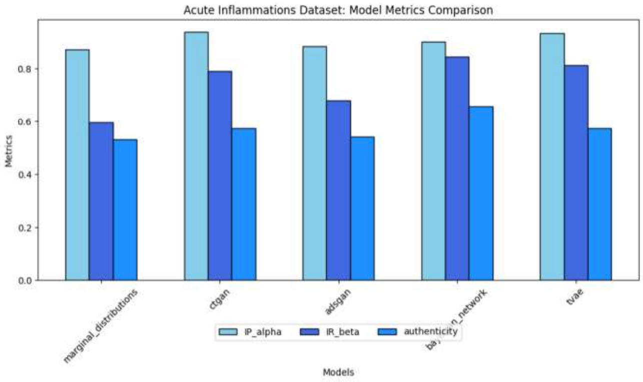

Figure 3. Model metrics comparison on Acute Inflammations Dataset

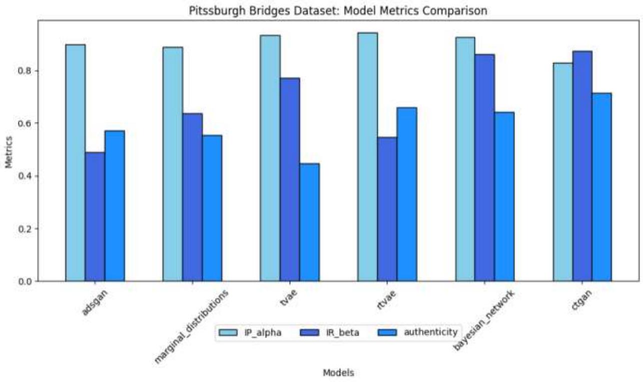

Figure 4. Model metrics comparison on Pittsburgh Bridges Dataset

In conclusion, CTGAN was generally the most effective model in this study, achieving the highest values in IPα and IRβ across most datasets, while RTVAE was the best on the Census Income dataset, particularly in terms of Generalization. Marginal Distributions was competitive in preserving the authenticity of the data, while Bayesian Network and ADSGAN had various strengths in certain metrics but did not reach the highest values in all cases. The detailed values of the metrics can be found in the tables below.

Table 1. Results for Abalone Dataset

Model	**IPα**	**IRβ**	**Authenticity**
Bayesian network	0.9706	0.9111	0.7085
Marginal distributions	0.4746	0.4225	**0.8402**
CTGAN	**0.9856**	0.9076	0.6411
ADSGAN	0.9772	0.8962	0.6618
TVAE	0.9764	0.9009	0.6675
RTVAE	0.9841	**0.9165**	0.6561

Table 2. Results for Census Income Dataset

Model	**IPα**	**IRβ**	**Authenticity**
RTVAE	0.8475	0.7578	0.7500
Bayesian network	**0.9572**	0.8928	0.6667
Marginal distributions	0.2900	0.5831	**0.8583**
ADSGAN	0.8987	0.6322	0.6417
TVAE	0.8665	0.8000	0.7083
CTGAN	0.9070	**0.8928**	0.6667

Table 3. Results for Acute Inflammations Dataset

Model	**IPα**	**IRβ**	**Authenticity**
Marginal distributions	**0.8722**	0.5950	0.5313
CTGAN	0.9382	0.7906	0.5729
ADSGAN	0.8845	0.6798	0.5417
Bayesian network	0.9026	**0.8450**	**0.6563**
TVAE	0.9346	0.8137	0.5729
RTVAE	0.8213	0.7511	0.5822

Table 4. Results for Pittsburgh Bridges Dataset

Model	IPα	IRβ	Authenticity
ADSGAN	0.8982	0.4890	0.5714
Marginal distributions	0.8884	0.6366	0.5536
TVAE	0.9322	0.7712	0.4464
RTVAE	**0.9419**	0.5463	0.6607
Bayesian network	0.9260	0.8615	0.6429
CTGAN	0.8278	**0.8733**	**0.7143**

5. Conclusion

In this research, CTGAN emerged as the most effective model overall, with superior results in the Fidelity and Diversity metrics across most datasets. RTVAE demonstrated excellent performance on the Census Income dataset, particularly in Generalization, making it a strong contender for real-world applications where minimizing overfitting is crucial. While the Marginal Distributions model was not the best in most metrics, it excelled at preserving the authenticity of the data, which may be valuable in specific scenarios where data authenticity is a priority. Other models, such as BNN and ADSGAN, showed competitive strengths but did not dominate in all evaluated metrics. Overall, this study highlights the im-portance of a comprehensive evaluation framework when comparing generative models, emphasizing the need to balance Fidelity, Diversity, and Generalization to effectively generate synthetic data that is both accurate and varied.

References

1. Sun, J., Liao, Q. V., Muller, M., Agarwal, M., Houde, S., Talamadupula, K., Weisz, J. D.: Investigating Explainability of Generative AI for Code through Scenario-based Design. arXiv preprint arXiv:2202.04903, 2022.

2. Polson, N., Sokolov, V.: Generative Modeling: A Review. arXiv preprint arXiv:2501.05458, 2024.

3. Borisov, V., Seßler, K., Leemann, T., Pawelczyk, M., Kasneci, G.: Language Models are Realistic Tabular Data Generators. arXiv preprint arXiv:2210.06280, 2023.

4. Liu, J., Chai, C., Luo, Y., Lou, Y., Feng, J., Tang, N.: Feature Augmentation with Reinforcement Learning. In: 2022 IEEE 38th International Conference on Data En-gineering (ICDE), pp. 3360–3372. IEEE Press, 2022. https://doi.org/10.1109/ ICDE53745.2022.00317

5. Li, Y., Li, Z., Zhang, K., Dan, R., Jiang, S., Zhang, Y.: ChatDoctor: A Medical Chat Model Fine-Tuned on a Large Language Model Meta-AI (LLaMA) Using Medical Domain Knowledge. Cureus 15 (n. d.), e40895. https://doi.org/10.7759/cureus. 40895

6. Qian, Z., Cebere, B.-C., van der Schaar, M.: Synthcity: facilitating innovative use cases of synthetic data in different data modalities. arXiv preprint arXiv:2301.07573, 2023.

7. Alaa, A. M., van Breugel, B., Saveliev, E., van der Schaar, M.: How Faithful is your Synthetic Data? Sample-level Metrics for Evaluating and Auditing Generative Mod-els. arXiv preprint arXiv:2102.08921, 2022. https://arxiv.org/abs/2102.08921

8. Akrami, H., Aydore, S., Leahy, R. M., Joshi, A. A.: Robust Variational Autoencoder for Tabular Data with Beta Divergence. arXiv preprint arXiv:2006.08204, 2020.

9. Akrami, H., Joshi, A. A., Li, J., Aydore, S., Leahy, R. M.: Robust Variational Autoencoder. arXiv preprint arXiv:1905.09961, 2019.

10. Apella´niz, P. A., Parras, J., Zazo, S.: An improved tabular data generator with VAE-GMM integration. arXiv preprint arXiv:2404.08434, 2024. https://arxiv.org/ abs/2404.08434

11. Akrami, H., Aydore, S., Leahy, R. M., Joshi, A. A.: Robust Variational Autoen-coder for Tabular Data with Beta Divergence. arXiv preprint arXiv:2006.08204, 2020. https://arxiv.org/abs/2006.08204

12. Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative Adversarial Networks. arXiv preprint arXiv:1406.2661, 2014.

13. Xu, L., Skoularidou, M., Cuesta-Infante, A., Veeramachaneni, K.: Modeling Tab-ular Data using Conditional GAN. arXiv preprint arXiv:1907.00503, 2019.

14. Yoon, J., Drumright, L. N., van der Schaar, M.: Anonymization Through Data Syn-thesis Using Generative Adversarial Networks (ADS-GAN). IEEE J Biomed Health Inform, 24(8):2378-2388, 2020. https://doi.org/10.1109/JBHI.2020.2980262.

15. Scott, C. D., Nowak, R. D.: Learning Minimum Volume Sets. Journal of Machine Learning Research, 7, 665–704, 2006.

16. Goan, E., Fookes, C.: Bayesian Neural Networks: An Introduction and Survey. In: Case Studies in Applied Bayesian Data Science, Springer International Publishing, 45–87, 2020. https://doi.org/10.1007/978-3-030-42553-1 3.