GAN Generation of Synthetic Data

Syed Imam Ali
AI24MTECH14005

Soumy Suwas
AI24MTECH14001

Code can be found here here.

1. Problem Statement

The growing need for high-quality synthetic data has become increasingly important in data-driven research and applications, especially in scenarios involving privacy constraints, limited data availability, or the need for data augmentation. Generative models such as Generative Adversarial Networks (GANs) have shown promise in synthesizing realistic data for domains like image and speech generation. However, generating high-fidelity synthetic data for **structured tabular datasets**, especially those with complex inter-feature dependencies, remains a challenging task.

Traditional GANs are often tailored to image data and tend to perform poorly on tabular datasets due to:

- The presence of **heterogeneous feature types** (continuous, categorical),
- The **lack of spatial locality** or inherent structure like that found in images,
- The **difficulty in capturing joint feature distributions** and correlations.

Moreover, while tabular GANs like CTGAN address some of these challenges, they still rely on treating features independently during sampling or embedding them through conditional vectors, which may not fully preserve complex, nonlinear dependencies.

To address this, we propose a method that:

- Utilizes a **Gaussian copula transformation** to map the original dataset into a space where each feature is marginally Gaussian, while preserving the original correlation structure.
- Trains a **Wasserstein GAN with Gradient Penalty (WGAN-GP)** on this copula-transformed space to generate new synthetic samples.
- Applies an **inverse copula transformation** to convert the generated samples back into the original data space.

The primary objective of this work is to **generate high-quality synthetic tabular data** that:

1. Closely matches the **marginal distributions** of real data,
2. Preserves the **inter-feature dependencies and correlations**,
3. Enables meaningful downstream analysis or model training.

This methodology is evaluated through visual (KDE, correlation heatmaps) and quantitative metrics to assess the fidelity and utility of the synthetic data.

2. Dataset Description

The dataset utilized in this study consists of 1199 rows (samples) and 10 columns (features), all of which are continuous real-valued variables of type `float64`. The dataset has been fully cleaned and standardized, with no missing values or categorical data present. As such, it serves as a well-suited foundation for advanced generative modeling techniques like copula-based transformations and generative adversarial networks.

The data appears to stem from a financial or tax-related domain, where the features are engineered metrics or ratios involving key financial attributes such as sales, purchases, and input tax credits (ITC). The inclusion of multiple normalized variables and standardized ratios suggests that the data underwent preprocessing steps such as scaling, transformation, and possibly feature engineering from raw transactional records.

2.1. Dataset Structure and Format

- **Number of samples:** 1199 — each row represents an individual entity, case, or record, possibly corresponding to a business transaction or financial profile.
- **Number of features:** 10 — all features are continuous and numeric, supporting statistical modeling.

- **Data type:** All features are of type `float64`, ensuring compatibility with operations that require real-valued arithmetic and distributional assumptions.

- **Missing data:** None — the dataset is complete, making it directly suitable for copula transformation and GAN training without imputation.

2.2. Feature Overview

The 10 columns in the dataset can be grouped based on their semantic or statistical properties:

- **Normalized Covariates:** `cov1` through `cov7` appear to be bounded within $[0, 1]$, though a few values extend slightly into the negative range, likely due to numerical error or zero-centering. These may represent normalized financial metrics, such as proportions or scoring functions.

- **Standardized Ratios:** The remaining three features — `sal_pur_rat`, `igst_itc_tot_itc_rat`, and `lib_igst_itc_rat` — are zero-centered and standardized to unit variance. Their summary statistics confirm that they have a mean near zero and standard deviation of approximately 1, consistent with z-score normalization.

2.3. Feature Names

- `cov1`, `cov2`, `cov3`, `cov4`, `cov5`, `cov6`, `cov7`
- `sal_pur_rat`
- `igst_itc_tot_itc_rat`
- `lib_igst_itc_rat`

2.4. Data Types

Feature	Data Type
cov1	float64
cov2	float64
cov3	float64
cov4	float64
cov5	float64
cov6	float64
cov7	float64
sal_pur_rat	float64
igst_itc_tot_itc_rat	float64
lib_igst_itc_rat	float64

Table 1. Data types of all features in the dataset

Table 2. Summary statistics of the dataset features (side-by-side)

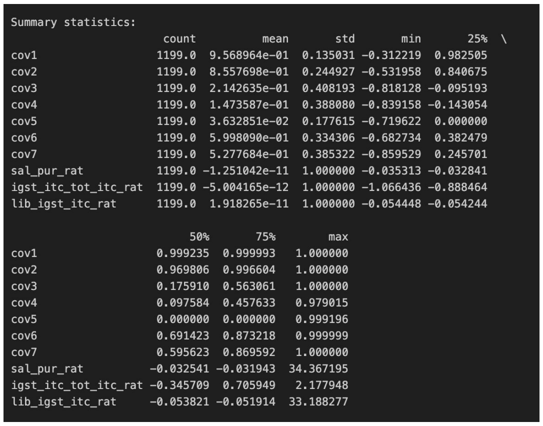
Feature	Mean	Std	Min	Max
cov1	0.9569	0.1350	-0.3122	1.0000
cov2	0.8558	0.2449	-0.5320	1.0000
cov3	0.2143	0.4082	-0.8181	1.0000
cov4	0.1474	0.3881	-0.8392	0.9790
cov5	0.0363	0.1776	-0.7196	0.9992
cov6	0.5998	0.3343	-0.6827	1.0000
cov7	0.5278	0.3853	-0.8595	1.0000
sal_pur_rat	0.0000	1.0000	-0.0353	34.3672
igst_itc_tot_itc_rat	0.0000	1.0000	-1.0664	2.1779
lib_igst_itc_rat	0.0000	1.0000	-0.0544	33.1883

Observations:

- `cov5` is a sparse variable with 75% of its values equal to zero. This may represent the presence/absence of a feature or a highly skewed attribute.
- `sal_pur_rat`, `igst_itc_tot_itc_rat`, and `lib_igst_itc_rat` are standardized and exhibit extreme values at the upper tail, e.g., max values of 34.37 and 33.19, suggesting outliers or rare cases.
- Despite some features exhibiting high peaks or long tails, the dataset is statistically well-behaved and amenable to copula transformation techniques.

2.5. Descriptive Statistics and Feature Analysis

The first stage of analysis involves understanding the nature of the real dataset. Summary statistics (mean, std, min, max) help us evaluate the feature distributions, spread, and outliers.



```
Summary statistics:
                      count          mean       std       min       25%  \
cov1                 1199.0  9.568964e-01  0.135031 -0.312219  0.982505
cov2                 1199.0  8.557698e-01  0.244927 -0.531958  0.840675
cov3                 1199.0  2.142635e-01  0.408193 -0.818128 -0.095193
cov4                 1199.0  1.473587e-01  0.388080 -0.839158 -0.143054
cov5                 1199.0  3.632851e-02  0.177615 -0.719622  0.000000
cov6                 1199.0  5.998090e-01  0.334306 -0.682734  0.382479
cov7                 1199.0  5.277684e-01  0.385322 -0.859529  0.245701
sal_pur_rat          1199.0 -1.251042e-11  1.000000 -0.035313 -0.032841
igst_itc_tot_itc_rat 1199.0 -5.004165e-12  1.000000 -1.066436 -0.888464
lib_igst_itc_rat     1199.0  1.918265e-11  1.000000 -0.054448 -0.054244

                           50%       75%        max
cov1                  0.999235  0.999993   1.000000
cov2                  0.969806  0.996604   1.000000
cov3                  0.175910  0.563061   1.000000
cov4                  0.097584  0.457633   0.979015
cov5                  0.000000  0.000000   0.999196
cov6                  0.691423  0.873218   0.999999
cov7                  0.595623  0.869592   1.000000
sal_pur_rat          -0.032541 -0.031943  34.367195
igst_itc_tot_itc_rat -0.345709  0.705949   2.177948
lib_igst_itc_rat     -0.053821 -0.051914  33.188277
```

Figure 1. Summary statistics of dataset (mean, std, range)

2.6. Distribution and Normality Analysis

To assess how features deviate from normality, we visualize histogram-KDE combinations alongside Q-Q plots for all features. This is essential before copula normalization.

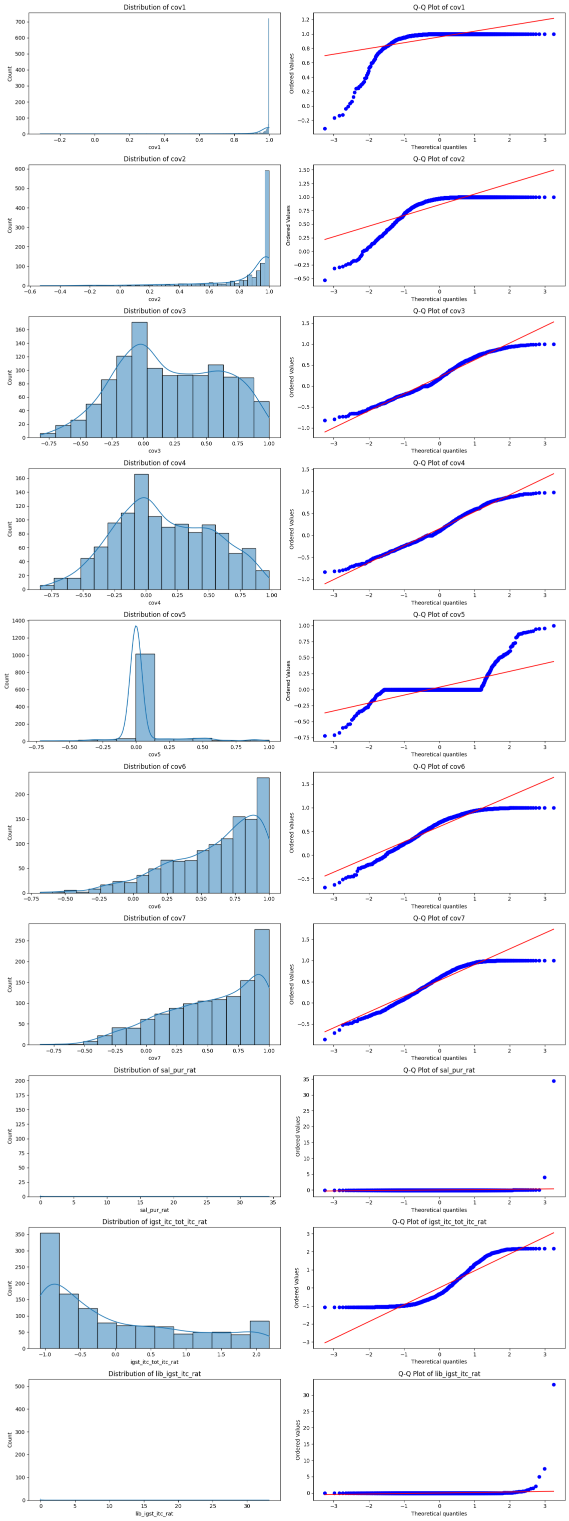


Figure 2. Histogram, KDE and Q-Q plots of real dataset features

2.7. Skewness and Transformation Insights

The skewness of features guides whether log transformation is needed. We visualize both original and log-transformed histograms for highly skewed variables, and boxplots for others.

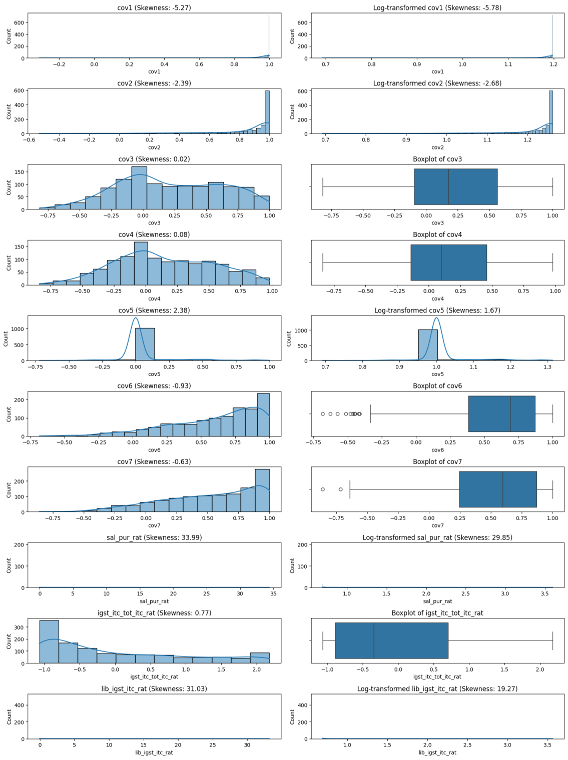


Figure 3. Skewness analysis with log-transformed histograms and boxplots

3. Methodology

This section presents the complete methodology employed to generate high-fidelity synthetic tabular data. Our approach integrates two powerful frameworks: the **Gaussian copula** for dependency-preserving transformation, and the **Wasserstein GAN with Gradient Penalty (WGAN-GP)** for stable generative modeling. Together, they provide a mechanism to synthesize samples that are statistically and structurally similar to the original data distribution.

3.1. Overview

Let $X \in \mathbb{R}^{n \times d}$ be the original dataset with n samples and d continuous features. Our pipeline involves the following steps:

1. **Transform** the original data into a Gaussian copula space.

2. **Train** a WGAN-GP model on this copula-transformed dataset.

3. **Generate** new samples from the generator in copula space.
4. **Invert** the copula transformation to obtain synthetic samples in the original data space.

3.2. Gaussian Copula Transformation

The copula framework enables the modeling of multivariate distributions by separating marginal distributions from dependency structure. Formally, by Sklar's theorem, any multivariate joint distribution F_X can be written as:

$$F_X(x_1, \ldots, x_d) = C(F_1(x_1), \ldots, F_d(x_d))$$

where:

- $F_j(x_j)$ is the marginal CDF of feature x_j,
- C is the copula function capturing dependency between the marginals.

To transform data into Gaussian copula space, we follow these steps:

1. Compute the **empirical CDF** (ECDF) for each feature x_j, mapping its values to the uniform interval $(0, 1)$:

$$U_j = \frac{\text{rank}(x_j)}{n+1}$$

2. Apply the inverse normal CDF (probit function) to each uniform value to obtain standard normal marginals:

$$Z_j = \Phi^{-1}(U_j)$$

This results in a transformed dataset $Z \in \mathbb{R}^{n \times d}$, where each feature is marginally standard normal, but the inter-feature dependencies (correlation structure) from the original data are preserved.

3.3. Wasserstein GAN with Gradient Penalty (WGAN-GP)

Traditional GANs suffer from training instabilities, mode collapse, and non-informative loss signals due to their use of the Jensen-Shannon divergence. WGAN overcomes this by minimizing the Wasserstein-1 distance (Earth Mover's Distance), which provides smoother gradients and better convergence behavior.

3.3.1 Wasserstein-1 Distance

The Wasserstein-1 distance between two distributions $\mathbb{P}_r$ and $\mathbb{P}_g$ is defined as:

$$W(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\gamma \in \Pi(\mathbb{P}_r, \mathbb{P}_g)} \mathbb{E}_{(x,y)\sim\gamma}[\|x - y\|]$$

This intuitively represents the minimum cost of transporting mass from $\mathbb{P}_r$ to match $\mathbb{P}_g$.

Using the Kantorovich-Rubinstein duality, the distance can be approximated by:

$$W(\mathbb{P}_r, \mathbb{P}_g) = \sup_{\|f\|_L \leq 1} \mathbb{E}_{x\sim\mathbb{P}_r}[f(x)] - \mathbb{E}_{x\sim\mathbb{P}_g}[f(x)]$$

where f is 1-Lipschitz. In practice, f is parameterized as a neural network called the *critic* (not a discriminator).

3.3.2 Gradient Penalty

To enforce the 1-Lipschitz constraint, WGAN-GP introduces a gradient penalty on the critic's outputs with respect to inputs. The loss function for the critic becomes:

$$\mathcal{L}_D = \mathbb{E}_{\hat{x}\sim\mathbb{P}_g}[D(\hat{x})] - \mathbb{E}_{x\sim\mathbb{P}_r}[D(x)] + \lambda \cdot \mathbb{E}_{\tilde{x}\sim\mathbb{P}_{\text{interp}}}\left[(\|\nabla_{\tilde{x}} D(\tilde{x})\|_2 - 1)^2\right]$$

Here:

- $\hat{x}$ are generated samples from the generator,
- $\tilde{x}$ are interpolated samples between real and generated data,
- λ is a hyperparameter controlling the gradient penalty (typically set to 10).

The generator is trained to minimize:

$$\mathcal{L}_G = -\mathbb{E}_{\hat{x}\sim\mathbb{P}_g}[D(\hat{x})]$$

3.4. Training Procedure

- The generator G takes a noise vector $z \sim \mathcal{N}(0, I) \in \mathbb{R}^{100}$ and outputs a synthetic sample in copula space.
- The critic D is trained for $n_{\text{critic}} = 5$ steps for every generator step.
- Optimization is done using the Adam optimizer with $\beta_1 = 0.5$, $\beta_2 = 0.9$, and learning rate 10^{-4}.
- Training is continued for 500 epochs with batch size 64.

3.5. Algorithm Description

The complete process of generating synthetic tabular data using WGAN-GP with Gaussian copula transformation is outlined in Algorithm 1. This includes preprocessing the data into copula space, training the WGAN-GP model, and applying inverse transformation to recover data in the original scale.

Algorithm 1 Synthetic Data Generation using WGAN-GP with Copula

```
Input: Real dataset $X \in \mathbb{R}^{n\times d}$, epochs $T$, batch size $B$,
    noise dimension $z$, gradient penalty coefficient $\lambda$
Output: Synthetic dataset $\hat{X} \in \mathbb{R}^{m\times d}$
Stage 1: Copula Transformation
for each feature $X_j \in X$ do
    Compute empirical CDF: $U_j = \frac{\text{rank}(X_j)}{n+1}$
    Transform to standard normal: $Z_j = \Phi^{-1}(U_j)$
end for
Form copula-transformed dataset $Z = [Z_1, Z_2, \dots, Z_d]$
Stage 2: WGAN-GP Training
Initialize Generator $G$ and Critic $D$
for epoch $= 1$ to $T$ do
    for each batch $Z_b \subset Z$ do
        for $k = 1$ to $n_{\text{critic}}$ do
            Sample noise $z \sim \mathcal{N}(0, I)$
            Generate fake batch: $\hat{Z} = G(z)$
            Compute critic outputs: $D(Z_b), D(\hat{Z})$
            Compute gradient penalty:
                $\text{GP} = \lambda \cdot \mathbb{E}_{\tilde{z}} \left[ (\|\nabla_{\tilde{z}} D(\tilde{z})\|_2 - 1)^2 \right]$
            Update critic:
                $\mathcal{L}_D = \mathbb{E}[D(\hat{Z})] - \mathbb{E}[D(Z_b)] + \text{GP}$
        end for
        Sample noise $z \sim \mathcal{N}(0, I)$
        Generate fake batch $\hat{Z} = G(z)$
        Update generator:
            $\mathcal{L}_G = -\mathbb{E}[D(\hat{Z})]$
    end for
end for
Stage 3: Inverse Copula Transformation
Generate synthetic samples $\hat{Z} = G(z)$
for each feature $\hat{Z}_j \in \hat{Z}$ do
    Apply inverse transform: $\hat{X}_j = F_j^{-1}(\Phi(\hat{Z}_j))$
end for
    return $\hat{X}$
```

3.6. Inverse Copula Mapping

After training, the generator produces samples $\hat{Z} \in \mathbb{R}^{m\times d}$ in the Gaussian copula space. These are mapped back to the original data space by applying the inverse copula transform:

$$\hat{X}_j = F_j^{-1}(\Phi(\hat{Z}_j))$$

where:

- Φ is the standard normal CDF,
- F_j^{-1} is the empirical inverse CDF of feature X_j,
- $\hat{X}$ is the final synthetic dataset.

3.7. Evaluation Metrics

We evaluate the quality of synthetic samples using:

- **Kernel Density Estimation (KDE)** plots: to visually compare the marginal distributions of real and synthetic data.
- **Correlation matrices**: to assess how well the generator captures inter-feature dependencies.
- **t-SNE or PCA visualizations** (optional): to compare real and synthetic samples in low-dimensional embedding spaces.

3.8. Summary

This methodology leverages the statistical rigor of copula theory and the generative power of WGAN-GP to synthesize continuous tabular data that closely mirrors the original dataset. The copula transformation ensures marginal normalization while preserving dependencies, and WGAN-GP provides a stable and interpretable optimization objective for generative modeling.

4. Results

In this section, we analyze the performance of the proposed Copula-WGAN model by comparing the statistical properties of the original and synthetic datasets. Specifically, we examine the preservation of inter-feature dependencies using correlation matrices and the fidelity of marginal distributions using Kernel Density Estimates (KDEs).

4.1. Correlation Structure Preservation

Figure **??** illustrates the Pearson correlation matrices for the original and synthetic datasets. These matrices offer insights into how well the synthetic data preserves the pairwise linear relationships among features.

- **Structural Similarity:** The synthetic correlation matrix demonstrates a high degree of structural similarity with the original. Key correlations, both positive and negative, are closely mirrored.
- **Notable Correlations:** For instance, the strong correlation between `cov3` and `cov4` (around 0.88) is clearly retained. Similarly, the relationship between `cov6` and `igst_itc_tot_itc_rat` (approximately 0.73) is also well-replicated.

- **Noise Resilience:** Minor discrepancies exist in low-correlation pairs, which is expected due to stochasticity in GAN training. Nonetheless, the model avoids introducing spurious correlations and maintains overall interpretability.

This strong alignment in correlation structure indicates that the Copula-WGAN model successfully captures the joint dependencies between features—crucial for applications where interaction between variables carries significant semantic meaning.

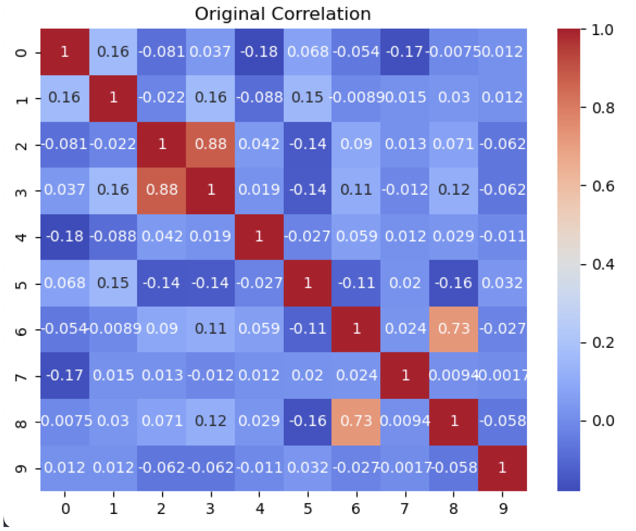


Figure 4. Pearson correlation heatmaps of the original dataset

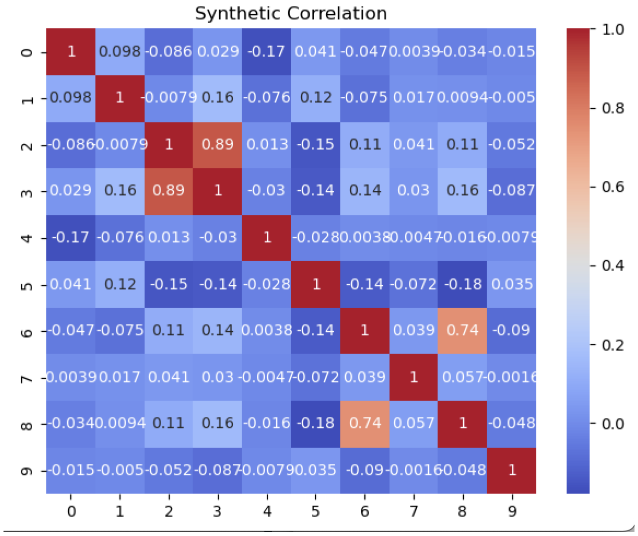


Figure 5. Pearson correlation heatmaps of the synthetic dataset

4.2. Univariate Distribution Fidelity

Figure 6 displays KDE plots for each feature, comparing the empirical probability density functions of the real and synthetic data. This analysis allows us to assess the generator's ability to reproduce the marginal distributions.

- **High-Fidelity Matching:** Most features show nearly perfect alignment in distribution shapes, including those with non-Gaussian, skewed, or multimodal characteristics.

- **Complex Distributions:** Features like `cov3`, `cov4`, and `igst_itc_tot_itc_rat` present asymmetrical and wider spread distributions. The synthetic KDEs closely match these, highlighting the model's expressiveness.

- **Sharp Peaks and Sparsity:** Features such as `cov5`, `sal_pur_rat`, and `lib_igst_itc_rat` exhibit extremely narrow peaks or high sparsity. The generator effectively mimics these fine-grained details, demonstrating sensitivity to subtle variations in the data.

- **Consistency Across Features:** The smoothness and coverage across all ten features suggest that the model does not overfit to any particular dimension but rather learns a holistic representation of the data space.

These KDE comparisons validate the ability of the Copula-WGAN to approximate the true underlying distributions of the features, ensuring that synthetic samples remain statistically representative.

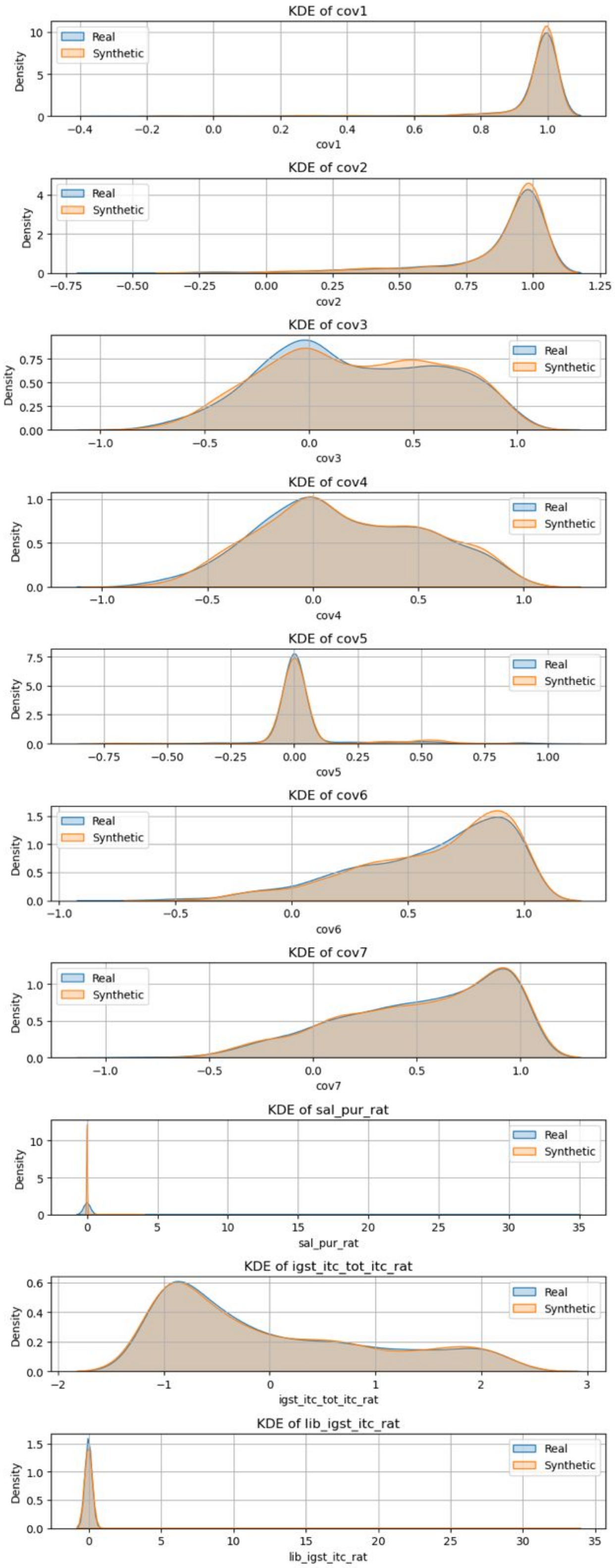


Figure 6. Kernel Density Estimates (KDEs) comparing feature-wise distributions of real and synthetic datasets

5. Conclusion

The experimental results affirm that the proposed Copula-WGAN framework effectively generates synthetic data that retains both the marginal and joint statistical properties of the original dataset. Key takeaways include:

- **Preservation of Correlation Structure:** The synthetic data maintains meaningful relationships between variables, a critical requirement in domains like finance, healthcare, and taxation, where feature dependencies encode domain semantics.
- **Accurate Distribution Modeling:** The generator learns complex, non-Gaussian, and sharp-peaked feature distributions without mode collapse or over-regularization.
- **Model Reliability:** The fidelity of the synthetic data positions the Copula-WGAN as a reliable candidate for privacy-preserving data generation, scenario simulation, or data augmentation.

These findings suggest that copula-based transformations combined with WGAN-GP provide a powerful and flexible approach for modeling continuous tabular data. In future work, the following directions can be explored:

- **Mixed-Type Extension:** Integrating mechanisms for modeling categorical or ordinal variables via hybrid copula structures or conditional generators.
- **Utility Evaluation:** Using synthetic data in downstream classification or regression tasks to quantitatively assess real-world applicability.
- **Robustness Analysis:** Testing on datasets with imbalanced, missing, or noisy features to evaluate generalization.

Overall, the Copula-WGAN offers a statistically faithful and practical approach to synthetic data generation, with broad implications for data privacy, reproducible research, and simulation-based modeling.

References

[1] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y. (2014). Generative Adversarial Nets. In Advances in Neural Information Processing Systems.

[2] Arjovsky, M., Chintala, S., and Bottou, L. (2017). Wasserstein GAN. arXiv preprint arXiv:1701.07875.

[3] Xu, L., Skoularidou, M., Cuesta-Infante, A., and Veeramachaneni, K. (2019). Modeling Tabular Data using Conditional GAN. In Advances in Neural Information Processing Systems.