CS6890: Fraud Analytics - Assignment 4: GAN Generation of Synthetic Data

Nitya Bhamidipaty - CS21BTECH11041
Akshay Santoshi - CS21BTECH11012

April 2025

1 Approach

- We employ a **Wasserstein Generative Adversarial Network (WGAN)** with Gradient Penalty to generate synthetic tabular data that mimics real-world financial features.
- The training data is read from an Excel file and contains 10 numerical features. A PyTorch `DataLoader` is used to iterate over the dataset in mini-batches.
- The **Generator** takes 10-dimensional noise vectors sampled from a standard normal distribution and outputs synthetic samples with 10 features. It consists of:
  - 5 fully connected layers with increasing then decreasing width.
  - Batch normalization and ReLU activation for stability and non-linearity.
- The **Discriminator** (or Critic) takes a 10-dimensional real or fake sample and outputs a scalar score. Its architecture includes:
  - 5 fully connected layers with Layer Normalization and Leaky ReLU activations.
  - No sigmoid activation at the output, as required by the WGAN framework.
- Training follows the WGAN-GP protocol:
  - For every generator update, the critic is updated $n = 3$ times.
  - A gradient penalty term ensures stable training.
- After training, the generator is used to produce synthetic data, which is evaluated using:

– **Kernel Density Estimates (KDE)** of each feature to compare real vs. synthetic distributions.

– **Pearson correlation heatmaps** to assess how well inter-feature relationships are preserved.

2 Results

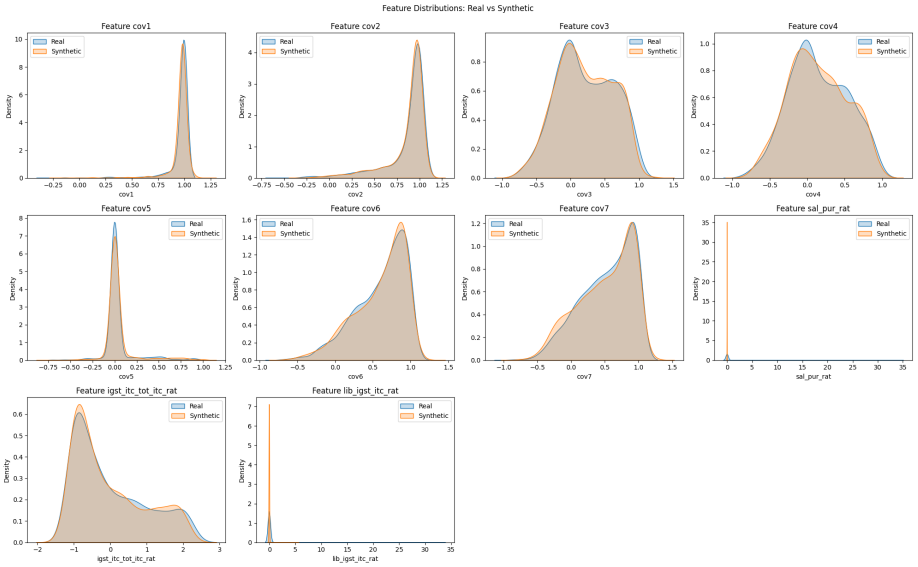

Figure 1: Feature Distributions of Real and Synthetic Data

- **cov1, cov2, cov5**: Synthetic distributions closely match real ones, indicating strong performance.
- **cov3, cov4, cov6, cov7**: Minor deviations in shape and tails, but overall alignment is good.
- **sal_pur_rat**: Large spike at 0 in synthetic data; poor fit suggests mode collapse or overfitting.
- **igst_itc_tot_itc_rat**: Overall shape captured, but synthetic data shows sharper peaks.
- **lib_igst_itc_rat**: Synthetic distribution is highly concentrated; fails to capture spread in real data.
- **Summary**: WGAN performs well on simpler distributions, but struggles with skewed or sparse features.

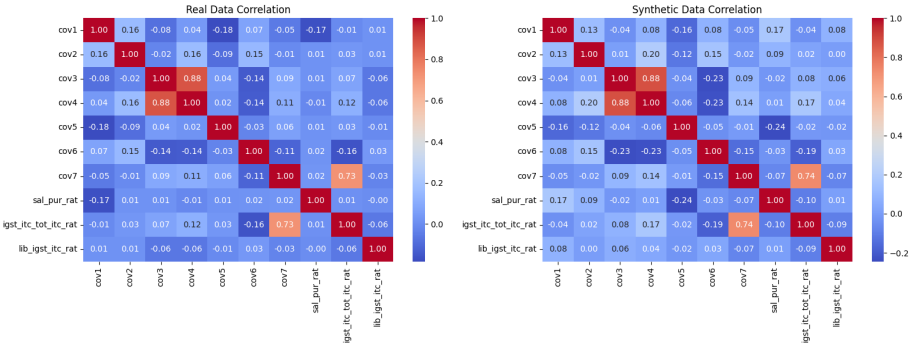

Figure 2: Pearson Correlation Heatmaps for Real and Synthetic Data

- **Overall**: The synthetic data captures the general correlation structure of the real data, with most strong and moderate correlations preserved.
- **Strong correlations** such as `cov3-cov4` and `cov6-igst_itc_tot_itc_rat` are well replicated in the synthetic set.
- **cov3-cov4**: High correlation in both real (0.88) and synthetic (0.88), indicating good structural learning.
- **sal_pur_rat correlations**: The real data shows weak correlations with most features, which is mostly retained, though `sal_pur_rat-cov7` is slightly inflated in synthetic (0.02 vs 0.07).
- **cov6 and cov7**: Correlation with `igst_itc_tot_itc_rat` is preserved fairly well (real: 0.73, synthetic: 0.74).
- **lib_igst_itc_rat**: Shows weak correlations in both datasets, indicating the model has preserved its low dependency nature.
- **Conclusion**: WGAN preserved key correlation structures, though minor differences exist in weaker correlations and noise sensitivity.