ContraCTGAN: Enhancing CTGAN's Tabular Data Generation with Contrastive Loss Integration

Amit John
Department of Computer Science
University College London, UK
amit.john.24@ucl.ac.uk

Abstract

Driven by the high data collection costs and growing privacy concerns, there is increasing demand for high-quality synthetic data to support the development of fraud detection systems. Traditional synthetic data generation frameworks struggle to capture the subtle inter-feature dependencies that are essential for identifying fraudulent transactions. In this paper, I propose ContraCTGAN, an extension of CTGAN that integrates contrastive learning via the Normalised Temperature-Scaled Cross Entropy (NT-Xent) loss. Using augmented positive data pairs, the model trains the discriminator to learn robust representations of the real data, which guides the generator to synthesise data that closely captures the inter-feature dependencies. Evaluations performed on a well-established credit card fraud detection dataset demonstrate that ContraCTGAN significantly improves data fidelity and utility.

1 Introduction

1.1 Background and Motivation

Driven by the cost of collecting real data and growing concerns around privacy, there is an increasing demand for large high-quality datasets to support the development of effective machine learning models. One key area where such datasets are required in the financial domain is fraud detection. Datasets used for fraud detection and credit risk analysis are often sensitive due to the nature of the personal transactional data and imbalanced with fewer minority classes like fraud. Additionally access to high quality financial data is restricted by regulatory constraints, limiting its availability for model development. In recent years, synthetic data has emerged as a viable alternative, enabling robust model development while preserving data privacy and maintaining regulatory compliance. Its utility has been explored in academic and industry initiatives such as (Financial Conduct Authority, 2024).

A popular architecture used for synthetic data generation are Generative Adversarial Networks (GANS). Conditional Tabular Generative Adversarial Network (CTGAN) is a modified conditional GAN architecture designed to synthesise tabular data, a type of data which is present in most financial datasets. CTGAN effectively handles both continuous and discrete features through mode-specific normalization and conditional sampling, which are particularly beneficial for imbalanced data distributions common in fraud detection scenarios.

1.2 Problem Statement

Many of the current GAN-based approaches, including CTGAN, struggle to capture the inter-features dependencies in financial data. As a result, the synthetic data they produce may resemble real data yet fail to preserve the nuanced statistical properties that are essential for training effective models in downstream classification tasks like fraud detection.

This project investigates whether integrating contrastive learning, in particular the Normalized Temperature Cross Entropy (NT-XENT) loss as a regularization strategy into the CTGAN training process can enhance the fidelity of the generated synthetic data. My hypothesis is that by improving the discriminators ability to learn robust representations of the real data distribution, the improved discriminator will be able to provide better feedback to the generator enabling the generator to synthesise data that better captures the inter-feature dependencies in the true distribution, hence improving the downstream classification performance of a model trained on the synthetic data.

1.3 Literature Review

1.3.1 Generative Adversarial Networks

Generative Adversarial Networks (Goodfellow et al., 2014) introduced a novel framework for generative modelling through an adversarial training process. The GAN architecture is made up of two neural networks, a generator G and a discriminator D that are trained simultaneously in an adversarial environment. The generator aims to generate synthetic data samples from random noise that attempt to mimic the true data distribution. The discriminator must evaluate and distinguish between real data and synthetic data produced by the generator.

The adversarial training process is viewed as a zero-sum game and is guided by the a min-max objective function

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{\text{data}}(x)}\left[\log D(x)\right] + \mathbb{E}_{z\sim p_z(z)}\left[\log\left(1 - D\left(G(z)\right)\right)\right] \tag{1}$$

$$L_D = -\mathbb{E}_{x\sim p_{\text{data}}(x)}\left[\log D(x)\right] - \mathbb{E}_{z\sim p_z(z)}\left[\log\left(1 - D\left(G(z)\right)\right)\right] \tag{2}$$

$$L_G = -\mathbb{E}_{z\sim p_z(z)}\left[\log D\left(G(z)\right)\right] \tag{3}$$

where generator G must minimize the probability of the discriminator classifying a synthetic sample as fake, and discriminator D is trained to maximise the probability of correctly classifying the real and synthetic data.

The adversarial loss from the objective function is then passed back to the generator as feedback to encourage the generator to synthesise higher quality samples that better capture the underlying true data distribution, With GANs, the training process attempts to reach a Nash equilibrium where the generators distribution matches the true distribution and the synthetics are indistinguishable from the real samples by the discriminator often presented as a stabilisation of generator and discriminator loss.

While GANs excelled with image data, their application to tabular data presented further challenges. The limitations of the original architecture led to the development of specialised models such as CTGAN, which adapt the GAN framework to effectively handle tabular data.

1.3.2 Conditional Tabular Generative Adversarial Networks

Conditional Tabular Generative Adversarial Networks (Xu et al., 2019) build upon the original GAN framework to address the unique challenges of tabular data. In particular, tabular datasets consist of a mix of continuous and categorical columns. Continuous columns often exhibit non-Gaussian multimodal distributions, while discrete columns can be imbalanced with minority classes like fraud cases being significantly under-represented.

Mode-Specific Normalisation:

Instead of applying a standard min-max normalisation $\frac{x-min}{max(x)-min(x)}$, CTGAN fits a variational Gaussian mixture model (VGM) to each continuous column. For each value $c_{i,j}$ in continuous column C_i, this can be expressed as:

$$P_{C_i}(c_{i,j}) = \sum_{k=1}^{m_i} \mu_k \, \mathcal{N}(c_{i,j}; \eta_k, \phi_k) \tag{4}$$

where η_k is the kth node and μ_k and ϕ_k are its respective weight and standard deviation.

Based on the probability density, one mode is sampled and used to normalise the value $c_{i,j}$. The mode used normalise is represented as a one-hot vector $\beta_{i,j}$ and the value within the mode is represented as a scalar $\alpha_{ij} = \frac{c_{ij} - \eta_k}{4\phi_k}$.

The final representation of a row is the concatenation of continuous and discrete columns.

$$\mathbf{r}_j = \alpha_{1,j} \oplus \beta_{1,j} \oplus \cdots \oplus \alpha_{N_c,j} \oplus \beta_{N_c,j} \oplus \mathbf{d}_{1,j} \oplus \cdots \oplus \mathbf{d}_{N_d,j} \tag{5}$$

Conditional Generation and Training-by-sampling:

To address the challenge of class imbalance in discrete columns, CTGAN uses a conditional generator, which takes a conditional vector that indicates the desired categorical column in the discrete column as input. The discrete columns are represented as one-hot vectors and the conditional vector is defined as a concatenation of conditions $cond = m_1 \oplus ... \oplus m_{N_d}$. A condition can be expressed as a binary mask vector where

$$\mathbf{m}_i^{(k)} = \begin{cases} 1 & \text{if } i = i^* \text{ and } k = k^* \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

During training the conditioning guides the generator to produce synthetic data for specific under-represented categories, mitigating the class imbalance. The distribution learned by the conditional generator can be expressed as

$$P(\text{row}) = \sum_{k \in D_{i^*}} P_G(\text{row} \mid D_{i^*} = k^*)\, P(D_{i^*} = k) \tag{7}$$

where D_i^* is a randomly sampled discrete column, k is a categorical value within the column.

Training-by-sampling randomly selects discrete columns and their categorical values according to a probability distribution that reflects their frequency in the real data. This balanced sampling approach ensures the generator explores a wide range of categories including the under-represented classes, reducing mode collapse and improving the quality and diversity of the synthesised data.

Integration of WGAN-GP:

CTGAN also incorporates mechanisms from the Wasserstein GAN with Gradient Penalty (WGAN-GP) (Gulrajani et al., 2017) to stabilise training.

WGAN-GP improves the stability of GAN training by redefining the discriminator's loss function to compute the Wasserstein distance between real and synthetic data distributions. Instead of standard gradient clipping, WGAN-GP introduces a gradient penalty to enforce Lipschitz continuity. The gradient penalty penalises the discriminators output if it changes too rapidly in response to small changes in input, enforcing a smooth Lipschitz continuous discriminator function. The WGAN-GP discriminator loss is defined as

$$L_D^{WGAN-GP} = \underbrace{\mathbb{E}_{\tilde{x} \sim p_g(\tilde{x})}[D(\tilde{x})] - \mathbb{E}_{x \sim p_{\text{data}}(x)}[D(x)]}_{\text{Wasserstein loss}} + \underbrace{\lambda\, \mathbb{E}_{\hat{x} \sim p_{\hat{x}}(\hat{x})}\left[(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2\right]}_{\text{gradient penalty}} \tag{8}$$

Integration of PacGAN:

PacGAN's mechanism is integrated into the CTGAN architecture to address mode collapse, a common issue in standard GAN architecture where the generator produces repetitive, non-diverse synthetic samples that manage to "fool" the discriminator. PacGAN groups multiple samples into a single pack before sending it to the discriminator (Lin et al., 2018). By examining the samples in a pack, the discriminator is able to evaluate the overall diversity within the pack. If the samples are too similar to each other, the discriminator penalises the generator. Over multiple iterations this approach guides the generator to produce a more diverse range of synthetic samples.

1.3.3 SimCLR and NT-Xent Loss

SimCLR is a self-supervised learning framework designed to generate meaningful representations without relying on labelled data. It achieves this objective by maximising agreement between

augmented views of the same data sample using a contrastive loss in the latent space (Chen et al., 2020). The new contrastive loss measure **Normalised Temperature-Scaled Cross Entropy Loss** (NT-Xent) largely contributed to the overall success of SimCLR.

Given a sampled batch of N data points, SimCLR applied a multiple data augmentation operations to each data point resulting in two augmented views of the same data point, $\tilde{x}_i$ and $\tilde{x}_j$, which are treated as a positive pair. These positive pair samples are considered as negative pairs with the remaining $2(N-1)$ augmented samples. For a positive pair i and j, the NT-Xent loss is defined as:

$$\ell_{i,j} = -\log \frac{\exp\left(\frac{\text{sim}(z_i, z_j)}{\tau}\right)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp\left(\frac{\text{sim}(z_i, z_k)}{\tau}\right)} \tag{9}$$

where:

- z_i and z_j are latent vector representations of augmented samples $\tilde{x}_i$ and $\tilde{x}_j$
- $\text{sim}(z_i, z_j)$ is cosine similarity between the two latent vectors
- τ is the temperature. A hyperparameter that determines the impact of the penalty
- $1_{[k \neq i]}$ is a self similarity mask, that excludes the sample latent vector itself from the total

Minimising the NT-Xent loss draws latent representations of positive pairs closer together, while simultaneously pushing the representations of negative pairs away. This contrastive objective encourages the model to learn robust and discriminative features. When incorporated into ContraCTGAN, this strategy enhances the discriminator's ability to detect subtle statistical differences between real and synthetic data distributions, leading to more robust adversarial training.

2 Methodology

2.1 Motivation

The ContraCTGAN approach proposed in this paper aims to address the CTGAN frameworks difficulty in capturing subtle inter-feature dependencies in financial fraud detection data. It extends CTGAN by integrating contrastive learning via a normalized temperature-scaled cross entropy loss (NT-Xent) as a regularization term during training. This objective enhances the discriminator's sensitivity to subtle differences in the data, enabling it to learn more robust latent representations of real data. The underlying hypothesis is that a more refined discriminator will provide more nuanced feedback to the generator, improving the quality of the generated synthetic data and ensuring that it better preserves the inter-feature dependencies present in the real data distribution.

2.2 Integration of Contrastive Learning

Contrastive learning trains models to distinguish between positive and negative pairs, resulting in more discriminative representations. The proposed ContraCTGAN framework integrates the Normalized Temperature-Scaled Cross Entropy (NT-Xent) loss from SimCLR into the training process for tabular data. For this the discriminator needs to learn positive pairs of augmented data points (z_i and z_j). The underlying rationale is that by encouraging the discriminator to learn from augmented representations of the original data sample, the model can more effectively capture the subtle inter-feature dependencies.

Unlike the image data which was used in SimCLR, tabular data does not lend itself well to complex transformation techniques such as rotation, cutout and jitter. So i chose to apply a minimal Gaussian noise augmentation. This approach generates realistic but slightly different representations of the real data. For a given real data point x, two augmented variants are produced as

$$x_{aug1} = x + \epsilon_1 \ , \ x_{aug2} = x + \epsilon_2$$

2.3 Network Architectures

The ContraCTGAN framework is comprised of three neural networks; the conditional generator, the discriminator and an embedder.

2.3.1 Conditional Generator

The Conditional Generator $G(z, cond)$ takes an latent vector z concatenated with a conditional vector $cond$ as input, feeds it forward through two full connected layers with ReLU activation functions and batch normalisation. Its then outputted through three different output heads; $\hat{\alpha}_i$ with the tanh activation for continuous outputs, $\hat{\beta}_i$ for mode indicators and $\hat{d}_i$ for discrete outputs which both use the Gumbel Softmax activation function.

$$
\begin{aligned}
h_0 &= z \oplus cond \\
h_1 &= \text{ReLU}\big(\text{BatchNorm}\big(\text{FC}_{|cond|+|z|\to 256}(h_0)\big)\big) \\
h_2 &= \text{ReLU}\big(\text{BatchNorm}\big(\text{FC}_{256\to 256}(h_1)\big)\big) \\
\hat{\alpha}_i &= \tanh\big(\text{FC}_{256\to|\alpha_i|}(h_2)\big), \quad 1 \le i \le N_c \\
\hat{\beta}_i &= \text{Gumbel}_{0.2}\big(\text{FC}_{256\to|\beta_i|}(h_2)\big), \quad 1 \le i \le N_c \\
\hat{d}_i &= \text{Gumbel}_{0.2}\big(\text{FC}_{256\to|D_i|}(h_2)\big), \quad 1 \le i \le N_d
\end{aligned}
$$

where:

- z is a latent space representation vector
- $cond$ is a conditional one-hot vector indicate the desired categories
- N_c and N_d represent the total number of continuous and discrete columns respectively
- FC denotes a fully connected layer
- $BatchNorm$ represents the batch normalisation layers
- $ReLU$ and $Gumbel_{0.2}$ represent the RelU and Gumbel Softmax activation functions respectively. The subscripted 0.2 indicates the temperature used in the Gumbel Softmax.

2.3.2 Discriminator with PacGAN

The discriminator takes a group of 10 inputs in a pack using the PacGAN mechanism to mitigate mode collapse. The $pac\ size$ was left as 10 which is the value used in the original code. The discriminator can be represented as:

$$
\begin{aligned}
h_0 &= r_1 \oplus \cdots \oplus r_{10} \oplus cond_1 \oplus \cdots \oplus cond_{10} \\
h_1 &= \text{Dropout}\big(\text{LeakyReLU}_{0.2}\big(\text{FC}_{10|r|+10|cond|\to 256}(h_0)\big)\big) \\
h_2 &= \text{Dropout}\big(\text{LeakyReLU}_{0.2}\big(\text{FC}_{256\to 256}(h_1)\big)\big) \\
C(\cdot) &= \text{FC}_{256\to 1}(h_2)
\end{aligned}
$$

where:

- r_i and $cond_i$ represent the ith data sample (either real or synthetic) and its corresponding conditional vector respectively.
- $Dropout$ represents the dropout regularisation to promote generalisability
- $LeakyReLU$ represents the leaky ReLU activation function

2.3.3 Embedder for Contrastive Learning

To embed the augmented positive pairs in latent space for contrastive learning, a dedicated embedder is introduced. The embedder network is comprised of two fully connected layers, one which uses a ReLU activation function. It can be represented as:

$$
\begin{aligned}
e_1 &= \text{ReLU}\big(\text{FC}_{input\to 128}(x)\big) \\
Embedder(x) &= \text{FC}_{128\to embed\_dim}(e_1)
\end{aligned}
$$

where:

- x is the augmented data point as input

- e_1 is the embedded latent space representation of the input
- 128 was chosen as the embedding dimension size for computational efficiency
- $embed\_dim$ represents the input dimension for the discriminator. 256 was used in this project.

2.4 Mathematical Formulation

2.4.1 Wasserstein Distance with Gradient Penalty

Wasserstein Distance is used as the primary loss function, where the objective is to minimise the distance between real and synthetic distributions. A gradient penalty is used to stabilise the gradients by enforcing Lipschitz continuity. The WGAN-GP loss was defined previously in Equation (8).

2.4.2 Normalised Temperature-Scaled Cross Entropy Loss

The NT-Xent contrastive loss is used as a regularisation on the WGAN-GP loss. Minimising the NT-Xent loss guides the discriminator to learn better representations of the real distribution, mitigating the impact of noise. When applied, the positive augmented pairs will pull close together, while the negative pairs (different samples) will remain further away. The NT-Xent loss was defined previously in Equation (9),

2.4.3 Combined WGAN-GP + NT-Xent Loss

ContraCTGAN trains the discriminator with a combined WGAN-GP and NT-Xent loss, with a contrastive hyperparameter λ to balance the impact of the NT-Xent regularisation. This combined loss ensures that the discriminator learns to distinguish between real and synthetic data well. This objective results in better feedback for the generator, which helps it produce higher quality data with greater fidelity which can improve the classification performance of a model trained with it.

$$L_D^{\text{total}} = L_D^{WGAN-GP} + \lambda_{\text{contrastive}} \cdot L_{\text{NT-Xent}} \tag{10}$$

2.5 Experimentation

2.5.1 The Dataset

The experiments for this project were conducted on the publicly available Credit Card Fraud Detection dataset from Kaggle (https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud), a well-established dataset for benchmarking models in terms of downstream classification performance. This dataset is comprised of 284,807 transactions made by European credit cardholders in September 2013 of which only 492 were fraud transactions. The extreme class imbalance in this financial dataset made it an excellent candidate to evaluate the effectiveness of synthetic data generation frameworks with a focus on improving downstream fraud classification models. Each transaction sample has 28 principal component features ($V1 - V28$), a continuous $Time$ feature representing the elapsed time between the each transaction and the first transaction in the dataset, a continuous $Amount$ feature indicating the transaction amount in euros, and a discrete $Class$ label which indicates whether the transaction was legitimate ($Class = 0$) or fraud ($Class = 1$).

2.5.2 Data Pre-processing and Train-Test Split

The CTGAN framework upon which ContraCTGAN was built cannot handle null values in continuous columns, so data validation is performed via functions in the script to check for missing data to ensure compatibility with the framework.

The dataset was split 80/20 into training and test sets using stratified sampling using the scikit-learn library. A stratified sampling approach was used to maintain the proportional representation of legitimate and fraud transactions. The splitting was performed outside of the script to mitigate the risk of test set leakage due to operational oversight.

Algorithm 1 ContraCTGAN Training Procedure

for each epoch **do**
  for each training batch **do**
    for each discriminator step **do**
      Sample mini-batch of real data: $\{x^{(1)}, \ldots, x^{(m)}\}$
      Generate two augmented views for each $x^{(i)}$:
      $x^{(i)}_{\text{aug1}} = x^{(i)} + \epsilon_1, \quad x^{(i)}_{\text{aug2}} = x^{(i)} + \epsilon_2$
      Sample latent vectors $\{z^{(1)}, \ldots, z^{(m)}\}$ and conditional vectors $\{cond^{(1)}, \ldots, cond^{(m)}\}$
      Generate synthetic samples: $x^{(i)}_{\text{fake}} = G(z^{(i)}, cond^{(i)})$
      Compute discriminator outputs on real and fake PacGAN packs
      Compute WGAN-GP loss using Eq. (8):

$$L_D^{\text{WGAN-GP}} = -\left[\mathbb{E}(C(x_{\text{real}})) - \mathbb{E}(C(x_{\text{fake}}))\right] + \lambda_{gp} \cdot GP$$

      Project augmented views via the embedder network:

$$z^{(i)}_{\text{real}} = \text{Embedder}(x^{(i)}_{\text{aug1}}), \quad z'^{(i)}_{\text{real}} = \text{Embedder}(x^{(i)}_{\text{aug2}})$$

      Compute NT-Xent loss across all pairs:

$$L_{\text{NT-Xent}} = \frac{1}{2m} \sum_{i=1}^{2m} \ell_{i,j(i)} \quad (\ell_{i,j} \text{ is from Eq. (9)})$$

      Update discriminator and embedder with the combined loss:

$$L_D^{\text{total}} = L_D^{\text{WGAN-GP}} + \lambda_{\text{contrastive}} \cdot L_{\text{NT-Xent}}$$

    end for
    Sample new latent vectors and conditional vectors
    Generate synthetic samples: $x_{\text{fake}} = G(z, cond)$
    Compute generator loss:

$$L_G = -\mathbb{E}[C(x_{\text{fake}})]$$

    Update generator using Adam optimiser
  end for
end for

2.5.3 Hyperparameters and Optimisers

The embedding dimensions for the noise vector passed to the generator, the dimensions for both the generator and the discriminator, the PacGAN pac size, learning rates and weight decay for both the generator and discriminator, the Adam optimiser with betas 0.5 and 0.9 and batch size were kept the same as the original CTGAN code. Additionally the discriminator steps were kept at $K = 1$ since its a common hyperparameter value in GAN literature including Goodfellow et al. (2014) which claimed it was the least computationally expensive and a viable option.

For the proposed ContraCTGAN framework additional parameters; contrastive lambda, contrastive temperature, standard deviation for data augmentation noise and a dimension value for the embedder used to embed the augmented pairs was introduced. Mixed precision training was also introduced into the training loop via PyTorch's Automated Mixed Precision (AMP) library to reduce the memory usage and reduce training time on the GPU used for this project, with negligible impact to the model performance. Below are the additional hyper parameter values

- $contrastive\_lambda = 0.1$
- $contrastive\_temperature = 0.5$
- $noise\_std = 0.01$
- $embed\_dim = 128$ for the embedder
- $use\_amp = True$

2.5.4 Technical Specifications

All parts of this project were conducted on a local workstation equipped with an AMD Ryzen 7 7800X3D CPU and an NVIDIA RTX 3090 GPU (24GB VRAM), supporting CUDA for hardware-accelerated training. The system used Ubuntu 24.04.2, Python 3.12.9 and Pytorch 2.6.0+cu124.

2.5.5 Evaluation Metrics

Wasserstein Distance, Jensen-Shannon Divergence and L2 Distance between Pearson Correlation Matrices were used to evaluate the fidelity of the synthetic data generated by the ContraCTGAN model. Accuracy, Area Under the ROC Curve (AUC) and F1 score were used to evaluate the utility of the synthetic data. This decision was motivated by previous academic work in this domain, primarily (Zhao et al., 2021) that also extended the CTGAN model to produce their more robust CTAB-GAN model.

Wasserstein Distance (WD) measures the effort needed to transform a probability distribution (the synthetic distribution) into another (the real distribution). Applied to numerical data, it measures the discrepancies between the two distributions. Its implemented through a custom $compute\_wasserstein\_distance$ function which iterates through each feature with numerical data and computes the WD, then averages the distances to return the mean WD over all features in the dataset. A lower WD value indicates that the two distributions are closer together, suggested greater fidelity.

Jensen-Shannon Divergence (JSD) measures the divergence between two frequency distributions. While commonly used for categorical data, I produced a custom variant suited for numerical data as my dataset only contained one categorical column, the class label. The function iterates through a feature and bins the continuous values into a set of 50 bins (the set default), which approximates a distribution of the data as a histogram. The histogram is then normalised to convert it to a probability distribution. A divergence measure is then computed on the real and synthetic distributions to quantify the magnitude of difference between the two. The JSD for all features are averaged to produce a mean JSD. A lower JSD value indicates that the distributions have greater similarity.

An **L2 Distance between Pearson Correlation Matrices** was used to evaluate whether inter-feature dependencies were maintained, measuring how well these dependencies in the synthetic data match their real data counterpart. This was implemented through a custom $compute\_l2dist\_pearson$ function that produces correlation matrices for each feature in both real and synthetic datasets which are flattened, then computes the euclidean distance between the flattened matrices. A lower L2 distance indicates that the synthetic data has better captured the inter-feature dependencies.

Downstream classification performance was evaluated to measure whether the synthetic data can be used to train classifiers that generalise well to an unseen holdout dataset. This aligns closely with the primary objective of synthetic data stated in the introduction - to act as a replacement for real data for modelling. An XGBoost Classifier, an efficient and frequently utilised model in this domain was trained on the synthetic training data and evaluated on a holdout real test dataset.

Accuracy measures the proportion of samples that were correctly classified. Arguably the most commonly used metric for evaluating classification performance. A higher accuracy indicates that the model is able to correctly classify more samples. Unfortunately this metric alone can be misleading when working with imbalanced datasets since the model could just be classifying the samples as the majority class (e.g. legitimate). It can be expressed in terms of true and false positives/negatives as

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}. \tag{11}$$

AUC represents the Area Under the ROC Curve (AUC), which plots the True Positive Rate against the False Positive Rate for different thresholds. It evaluates the classifier's ability to distinguish between the legitimate and fraud classes considering the rates. An AUC value closer to 1 indicates that the model is able to better distinguish between the classes.

F1 Score measures the harmonic mean between precision and recall. For context precision measures the accuracy of positive predictions while recall measures the models ability to identify all positive samples. It is a valuable metric in fraud detection where its important to correctly identify fraud instances but equally as important to avoid misclassifying legitimate instances as fraud. This is a

popular metric to use when working with imbalanced datasets. A higher F1 score is preferred as it indicates a better balance between precision and recall.

$$\text{Precision} = \frac{TP}{TP + FP} \quad , \quad \text{Recall} = \frac{TP}{TP + FN}. \tag{12}$$

$$\text{F1 Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{13}$$

2.6 Competitor Frameworks

To benchmark the performance of the proposed ContraCTGAN model, the CTGAN and TVAE, two established frameworks from Xu et al.(2019) for tabular synthetic data generation were considered. CTGAN uses a GAN architecture while TVAE uses a variational auto-encoder (VAE), however both address the challenge continuous and categorical variables in tabular data.

2.6.1 CTGAN Model

As explained in the literature review CTGAN aims to capture the multimodal characteristics of tabular datasets (e.g. financial fraud detection) by performing mode-specific normalisation utilising a variational Gaussian mixture model. Additionally, it addresses the issue of imbalanced classes through a conditional generator and a training-by-sampling approach. The framework also integrates the mechanisms from WGAN-GP to stabilise the training process and PacGAN to mitigate mode collapse. The framework is able to improve the quality and diversity of synthesised data considerably compared to a vanilla GAN and the other frameworks it benchmarked against.

2.6.2 TVAE Model

TVAE is a tabular variational auto-encoder framework that learns a latent space representation of data using an encoder and decoder to model $p_\theta(r_j|z_j)$ and $q_\phi(z_j|r_j)$ and train them using an evidence lower-bound (ELBO) loss (Xu et al., 2019). r_j denotes a sample from the dataset which is encoded to a latent vector z_j by the encoder, and decoded back to sample by the decoder. The objective is to maximise the ELBO loss which balances the reconstruction loss with a Kullback-Leibler (KL) divergence regularisation, to improve the latent space structure.

$$\begin{aligned}
\text{ELBO}(q) &= \mathbb{E}_q[\log p(x,z) - \log q(z)] \\
&= \mathbb{E}_q[\log p(x,z)] - \mathbb{E}_q[\log q(z)] \\
&= \mathbb{E}_q[\log p(x|z) + \log p(z)] - \mathbb{E}_q[\log q(z)] \\
&= \mathbb{E}_q[\log p(x|z)] + \mathbb{E}_q[\log p(z)] - \mathbb{E}_q[\log q(z)] \\
&= \mathbb{E}_q[\log p(x|z)] \; + \; \int q(z) \; \log p(z) \, \mathrm{d}z \; - \; \int q(z) \; \log q(z) \, \mathrm{d}z \\
&= \mathbb{E}_q[\log p(x|z)] \; + \; \int q(z) \; \log\Big(\frac{p(z)}{q(z)}\Big) \mathrm{d}z \\
&= \mathbb{E}_q[\log p(x|z)] \; - \; \mathrm{KL}\big(q(z) \,\|\, p(z)\big).
\end{aligned} \tag{14}$$

Note: This derivation of ELBO loss is adapted from (Patacchiola, 2021).

3 Results

3.1 Data Fidelity Results VS Baselines

The objective data fidelity metrics are to ensure that the synthetic data accurately captures the statistical properties and inter-feature dependencies of the real data distribution. In this project, fidelity is evaluated using Wasserstein Distance (WSD), Jenson-Shannon Divergence (JSD), and the L2 distance between Pearson correlation matrices. Fidelity is an important to fraud data as downstream classifiers depend on high quality data that captures inter-feature dependencies to perform well.

Model	WSD	JSD	L2 Pearson	Accuracy	AUC	F1
Real Data	–	–	–	0.9996	0.9777	0.8990
Baseline CTGAN	167.66	0.19825	8.19410	0.9792	0.9762	0.1392
Baseline TVAE	240.79	0.16331	NaN	0.9982	0.5	0.0
ContraCTGAN Variants						
ContraCTGAN (λ=0.5, τ=0.5)	109.73	0.19842	7.73692	0.9919	0.9673	0.2883
ContraCTGAN (λ=0.5, τ=0.1)	122.52	0.20566	8.18397	0.9894	0.9774	0.2418
ContraCTGAN (λ=0.2, τ=1.0)	187.77	0.20333	7.92103	0.9931	0.9808	0.3160
ContraCTGAN (λ=0.8, τ=0.8)	81.34	0.19956	8.56982	0.9884	0.9690	0.2259
ContraCTGAN (λ=1.0, τ=0.07)	96.35	0.19801	7.97287	0.9870	0.9773	0.2019

Table 1: Comparison of Baseline CTGAN, Baseline TVAE, and ContraCTGAN Variants

The results table implies that the baseline TVAE fails to capture these properties and maintain inter-feature dependencies resulting in a WSD of 240.79 and an null L2 Pearson distance. Despite it exceptional performance in downstream classification, it is not an ideal choice when aiming to generate synthetic data that mimics the real distribution.

In contrast the baseline CTGAN model achieves a lower WSD of 167.66 an L2 distance of 8.19410, with the trade-off being a slightly higher JSD of 0.19825. This suggests that the model is able to generate decent synthetic data, however it diverges considerably from the real distribution. Not being able to full capture the inter-feature dependencies, may pose challenges when dealing with financial data where these dependencies play a critical role in distinguishing between legitimate and fraud cases.

The proposed ContraCTGAN variant with a $\lambda = 0.5$ and $\tau = 0.5$ shows notable improvements in data fidelity. It averages a lower WSD of 109.73 and L2 Pearson measure of 7.73692, at the expensive of a marginally higher JSD. From these results a claim can be made that the model demonstrates a closer match with the real data distribution, suggesting that the integration of contrastive learning allowed the discriminator to learn a more refined representation of the real data.

3.2 Data Utility Results VS Baselines

The objective of data utility metrics are to measure the downstream classification performance of a model trained on the generated synthetic data. XGBoost classifiers were trained on the real data and synthetic data from CTGAN, TVAE and 36 variants of the ContraCTGAN of which five are presented in the results table. The models were evaluated using classification accuracy, AUC and F1 score metrics.

As a upper-bound baseline, the XGBoost classifier was trained on the real data and achieved close to perfect performance with an accuracy of 0.9996, an AUC of 0.9777 and an F1 score of 0.8990.

In contrast the baseline CTGAN achieves a reasonable accuracy of 0.9792, AUC of 0.9762 but a considerably lower F1 score of 0.1392 compared to the real data baseline. The low F1 score suggests that the model struggles to correctly identify the fraud minority class and find a balance between precision and recall.

The baseline TVAE presents a misleading accuracy of 0.9982, while achieving a low AUC of 0.5 and F1 score of 0.0. This reinforces the idea that classification accuracy alone is not a viable measure of discriminative performance when dealing with imbalanced datasets.

The ContraCTGAN variant with a $\lambda = 0.5$ and $\tau = 0.5$ improves on the CTGAN's downstream classification performance with a accuracy of 0.9919 (+1.30%) and an F1 score of 0.2883 (+1.07%). The increase in F1 implies a better balance between correctly identifying fraud cases (recall) and not misclassifying legitimate cases as fraud (precision), which if executed incorrectly would have serious repercussion in the financial domain. It is important to note that the classifier trained on this synthetic data attained a slightly lower AUC of 0.9673 (-9.12%). However, this marginal decrease can be justified given the significant gains in accuracy and F1.

3.3 Overall Performance vs Baselines

While, none of the generative models came close to the high F1 performance achieved by the real data, the proposed ContraCTGAN($\lambda = 0.5$, $\tau = 0.5$) conveyed the strongest balance of classification accuracy, AUC and F1 among its competition, hence making it a strong candidate in terms of the measured metrics. It's impressive downstream classification performance combined with its reasonably high data fidelity emphasises the importance of better capturing the underlying characteristics and inter-dependencies in the real data distribution.

4 Discussion

4.1 Interpretation of results

The performance of ContraCTGAN variant with $\lambda = 0.5$ and $\tau = 0.5$ compared to its competition in terms of the measured metrics showed that integrating contrastive learning into the CTGAN framework improved both data fidelity and utility. The improved Wasserstein Distance and L2 Pearson values support the hypothesise that the contrastive (NT-Xent) regularisation enhanced the discriminators ability to capture subtle inter-feature dependencies, which resulted in the generation of synthetic data that aligns closer with the true distribution. The improvements in downstream classification metrics, most notably F1 score demonstrates that the synthetic data produced by ContraCTGAN facilitated training of more refined fraud detection models.

It is critically important for a fraud detection system to correctly identify uncommon fraud transactions within an imbalanced dataset. Like the other generative models, ContraCTGAN was was not able to reach the high benchmark set by training the classifier on the real data, but it was able to narrow the gap by outperforming its competition in this project.

4.2 Limitations of the project

The results were promising, but there exist several limitations of this project. The primary limitation is that while the ContraCTGAN model improves upon the CTGAN model, the synthetic data produced by it fails to meet the same quality as the real data resulting it weaker downstream classification performance. Secondly, the experiment was conducted on a single credit card fraud dataset, this decision was made due to the scope and time constraints of this project. A broader evaluation across multiple datasets would be needed to determine the frameworks generalisability. Finally, the computational constraints influenced the chosen hyperparameter values, training epochs and tuning strategies which had an impact on quality of the generated synthetic data.

4.3 Future Work

Future developments on this project should focus on enhancing the quality of the generated synthetic data for better fraud detection performance. Proposed developments include:

- **Broader Evaluation**: Evaluating the performance of ContraCTGAN on multiple tabular datasets to determine the generalisability of the framework.
- **Advanced Augmentation**: Investigating more advanced augmentation methods on data points that produce positive pairs in latent space, to better simulate the variability in real world data.
- **Alternative regularisations and Hybrid models**: Explore and evaluate the performance of the model combined with other regularisation methods besides gradient penalty and Wasserstein distance based loss such as reconstruction loss, or VAE-GAN hybrid models.
- **Privacy-Preserving Mechanisms**: Privacy preservation is a key focus in synthetic data generation. Future work could explore the integration of mechanisms like differential privacy into the ContraCTGAN model to produce synthetic data that preserves inter-feature dependencies and statistical properties while protecting sensitive information. Such a model could be benchmarked against others like DP-CTGAN and CTAB-GAN+ (Zhao et al., 2024) which successfully implemented these mechanisms into their frameworks. This would require sufficient computation resources and careful tuning of the privacy budget to achieve a balance between privacy and utility.

5 Bibliography

Chen, T., Kornblith, S., Norouzi, M. and Hinton, G., 2020, November. A simple framework for contrastive learning of visual representations. In International conference on machine learning (pp. 1597-1607). PmLR.

Fang, M.L., Dhami, D.S. and Kersting, K., 2022. DP-CTGAN: Differentially Private Medical Data Generation Using CTGANs. In: M. Michalowski, S.S.R. Abidi and S. Abidi, eds. Artificial Intelligence in Medicine. AIME 2022. Lecture Notes in Computer Science, vol. 13263. Cham: Springer, pp. 178-188. doi:10.1007/978-3-031-09342-5_17.

Financial Conduct Authority (2024). Report: Using Synthetic Data in Financial Services. [online] Available at: https://www.fca.org.uk/publications/corporate-documents/report-using-synthetic-data-financial-services [Accessed 1 April 2025].

Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y., 2014. Generative adversarial nets. Advances in neural information processing systems, 27.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V. and Courville, A.C., 2017. Improved training of wasserstein gans. Advances in neural information processing systems, 30.

Lin, Z., Khetan, A., Fanti, G. and Oh, S., 2018. Pacgan: The power of two samples in generative adversarial networks. Advances in neural information processing systems, 31.

Patacchiola, R. (2021). An Introduction to Variational Inference. [online] Available at: https://mpatacchiola.github.io/blog/2021/01/25/intro-variational-inference.html [Accessed 1 April 2025].

Xu, L., Skoularidou, M., Cuesta-Infante, A. and Veeramachaneni, K., 2019. Modeling tabular data using conditional gan. Advances in neural information processing systems, 32.

Zhao, Z., Kunar, A., Birke, R. and Chen, L.Y., 2021, November. Ctab-gan: Effective table data synthesizing. In Asian Conference on Machine Learning (pp. 97-112). PMLR.

Zhao, Z., Kunar, A., Birke, R., Van der Scheer, H. and Chen, L.Y., 2024. Ctab-gan+: Enhancing tabular data synthesis. Frontiers in big Data, 6, p.1296508.