Imb-FinDiff: Conditional Diffusion Models for Class Imbalance Synthesis of Financial Tabular Data

Marco Schreyer
International Computer Science Institute
Berkeley, USA
marco@icsi.berkeley.edu

Timur Sattarov
Deutsche Bundesbank
Frankfurt am Main, Germany
timur.sattarov@bundesbank.de

Alexander Sim
Lawrence Berkeley National Laboratory
Berkeley, USA
asim@lbl.gov

Kesheng Wu
Lawrence Berkeley National Laboratory
Berkeley, USA
kwu@lbl.gov

ABSTRACT

Handling imbalanced datasets remains a critical challenge in financial machine-learning applications such as loan approval, credit scoring, and fraud detection. We present *Imbalanced Financial Diffusion* (*Imb-FinDiff*), a novel denoising diffusion framework designed to address class imbalance in financial tabular data. Our framework leverages embedding encodings for categorical and numerical attributes, effectively managing the complexities of mixed-type financial datasets. By incorporating a dual learning objective, (i) diffusion timestep noise and (ii) class label prediction, we synthesize minority class samples. Extensive experiments on diverse and real-world financial datasets demonstrate that *Imb-FinDiff* maintains the statistical properties of the original data while reducing bias caused by class imbalance. The minority class samples generated by *Imb-FinDiff* enhance the utility and fidelity of downstream machine learning classifiers.

CCS CONCEPTS

• **Computing methodologies** → **Supervised learning**; **Neural networks**; **Learning latent representations**.

KEYWORDS

neural networks, denoising diffusion probabilistic models, imbalanced learning, synthetic data generation, mixed-type tabular data

ACM Reference Format:
Marco Schreyer, Timur Sattarov, Alexander Sim, and Kesheng Wu. 2024. *Imb-FinDiff*: Conditional Diffusion Models for Class Imbalance Synthesis of Financial Tabular Data. In *5th ACM International Conference on AI in Finance (ICAIF '24), November 14–17, 2024, Brooklyn, NY, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3677052.3698659

1 INTRODUCTION

Learning from imbalanced datasets is a critical challenge in many domains of financial machine learning, such as loan approval, credit scoring, or fraud detection [1, 18]. Imbalance manifests in two common forms: minority interests and rare instances [5, 6, 17]. Minority interests arise in domains where rare but crucial events, such as fraudulent transactions, must be identified accurately. Rare instances concern situations where data representing a particular event is limited compared to other distributions. In financial auditing, detecting fraudulent activities in a vast pool of transactions exemplifies the minority interest challenge [8, 47].

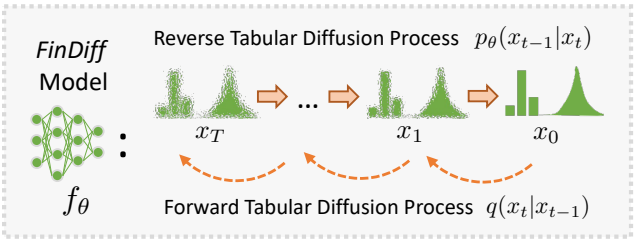


Figure 1: Overview of the financial diffusion model (*FinDiff*) approach for synthesizing mixed-type tabular data [46].

Tabular financial datasets are often imbalanced, with certain classes significantly underrepresented compared to general records [5, 6]. This imbalance can severely affect the true-positive performance of the minority class in supervised machine learning settings [17, 18]. Additionally, underrepresented or minority classes can lead to biased models that fail to identify risk factors or misclassify behaviors [41]. Ensuring a balanced representation of classes can help create fairer financial models [1].

A promising solution lies in generating high-quality synthetic data [1, 41]. Synthetic data mimics the statistical properties of real data, mitigating complications arising from imbalanced classes [32, 63]. This approach offers to model rare but impactful events like fraud and biased predictions in credit scoring [8, 41]. Generating high-fidelity synthetic tabular data offers to improve the performance of machine learning models on imbalanced datasets, fostering collaboration among financial entities and modelling rare but impactful events like fraud. However, real-world tabular data is often characterized by inherent complexities [26, 45, 64]:

- **Mixed Attribute Types:** Tabular data comprises diverse attribute types. Modelling these complexities requires synthesizing different data types into a generative model.

ICAIF '24, November 14–17, 2024, Brooklyn, NY, USA

ACM ISBN 979-8-4007-1081-0/24/11.
https://doi.org/10.1145/3677052.3698659

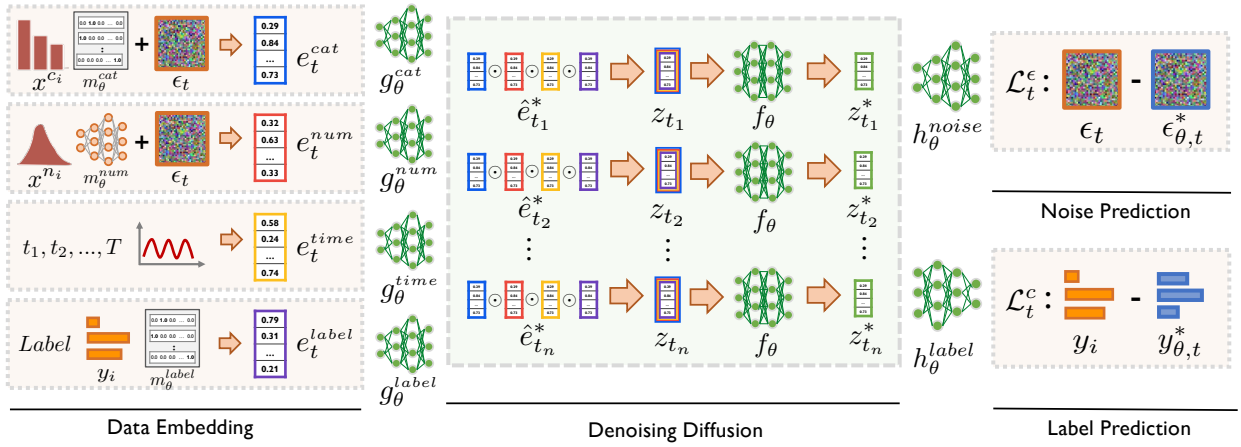


Figure 2: Overview of the *Imb-FinDiff* model architecture. The model extends denoising diffusion probabilistic models, as introduced in [20, 51], to oversample minority classes through a structured diffusion process, thereby improving class balance and enhancing machine learning model performance on imbalanced datasets.

- **Implicit Relationships:** Tabular data includes implicit relationships between records and attributes, necessitating models that can capture these (inter-) dependencies.
- **Distribution Imbalance:** Tabular data often has skewed distributions and imbalances, demanding advanced modelling techniques to represent the nuanced patterns accurately.

Recent advancements in deep generative models have demonstrated impressive capabilities in creating diverse and realistic content across various domains. Notably, *denoising diffusion probabilistic models* [20, 43, 51] have shown the ability to generate high-quality synthetic images, videos, and more. Such models have recently been proposed in synthetic financial tabular data synthesis [26, 45, 46]. Figure 1 illustrates the *FinDiff* [46] approach for synthesizing mixed-type tabular data from financial datasets.

In this work, inspired by these advances, we explore the potential of diffusion models to mitigate class imbalance. We introduce *Imbalanced Financial Diffusion* (*Imb-FinDiff*), a denoising diffusion framework tailored to synthesize minority class samples in financial tabular data. Our evaluation assesses three research questions: (RQ 1) Can *Imb-FinDiff*'s synthesized data improve the performance of financial machine learning tasks? (RQ 2) Can *Imb-FinDiff* maintain high fidelity in synthetic data, preserving the statistical properties of real-world data? (RQ 3) How effectively does *Imb-FinDiff* capture the diversity of minority class samples? Our study makes the following contributions to address these questions:

(1) **Learning:** We propose a learning framework synthesizing minority-class financial tabular data (RQs 1,2).
(2) **Sampling:** Our approach mitigates class imbalance in synthetic datasets through effective oversampling (RQs 1,3).
(3) **Evaluation:** We demonstrate the framework's utility and fidelity through evaluations on diverse datasets (RQs 2,3).

The remainder of this paper is structured as follows: section 2 provides an overview of related work. section 3 describes the basics of diffusion models and outlines the proposed methodology. section 4 and section 5 detail the experimental setup and results, respectively. We conclude with a summary and future research directions in section 6.

2 RELATED WORK

In recent years, synthesizing financial data has gained significant interest. This survey addresses (1) techniques for mitigating class imbalance in machine learning, (2) models for financial data synthesis, and (3) diffusion models for data synthesis.

2.1 Class Imbalance Learning Techniques

Sampling strategies involve oversampling or undersampling to balance data distributions. For instance, cost curves [11] study the interaction of sampling with decision trees, while JOUS-Boost [49] combines adaptive boosting with jittering sampling. *Synthetic data generation methods*, like SMOTE [5], interpolate between existing minority classes to create synthetic minority class examples. Extensions such as SMOTEBoost [6] and DataBoost-IM [16] combine synthetic generation with boosting. Additionally, Adasyn [17] generates synthetic examples based on feature space density distributions. *Cost-sensitive learning* uses cost matrices to address imbalance without altering data distributions. Techniques include cost-sensitive trees [33], Metacost [10], and threshold-moving for neural networks [69]. *Active learning* selects informative instances to reduce computational costs. Methods based on SVMs [13, 54] and word sense disambiguation strategies [42] have shown effectiveness in imbalanced scenarios. *Kernel-based methods* optimize model generalization for imbalanced data. Techniques include the ROWLS estimator [61] and kernel-boundary alignment [59], which adjust the kernel matrix based on data distribution.

2.2 Data Synthesis Using Generative Models

Generative models [15] have improved the creation of realistic content, including images [3, 43], videos [4, 50], audio [28, 60], code [7, 31], and natural language [37, 56]. *For financial data synthesis*, Wiese et al. [63] introduced Quant GANs to capture long-range dependencies. Ni et al. [34, 35] and Liao et al. [32] used signature

Wasserstein GANs for high-fidelity time-series generation. Using generative models, Dogariu et al. [9] synthesized realistic financial time-series. *For tabular financial data*, variational autoencoder (VAE) [25] and GAN-based models [12, 65] have been proposed. Xu et al. [65] introduced CTGAN, a conditional generator that handles mixed data types. Engelmann and Lessmann [12] addressed class imbalances using conditional Wasserstein GANs. Jordon et al. [22] developed PATE-GAN for enhanced data synthesis privacy with differential privacy guarantees. Torfi et al. [55] presented a differentially private framework for synthetic healthcare data. Zhao et al. [68] developed CTAB-GAN to address data imbalance, while Kim et al. [24] used neural ODEs to improve synthetic tabular data utility. Wen et al. [62] introduced Causal-TGAN, leveraging causal relationships to enhance data quality. Zhang et al. [67] proposed GANBLR for understanding feature importance, and Nock and Guillame-Bert [36] suggested a tree-based approach as an alternative.

2.3 Data Synthesis using Diffusion Models

Lately, diffusion models for data synthesis have gained momentum [19, 51]. *For image synthesis*, Rombach et al. [44] demonstrated that latent diffusion models can generate high-quality images with reduced computation time. These advancements have expanded the applicability of diffusion models beyond image synthesis, showcasing their robustness and versatility [66]. *For text generation*, Strudel et al. [52] introduced self-conditioned embedding diffusion that rivals autoregressive models. Gao et al. [14] applied embeddings for discrete text data generation, addressing challenges like the collapse of the denoising objective and imbalanced embedding norms. Li et al. [30] introduced Diffusion-LM to control complex, fine-grained text outputs. *For tabular data synthesis*, Kotelnikov et al. [26], Ouyang et al. [38], and Sattarov et al. [46] proposed multinomial and conditional diffusion models [21]. Recently, this was extended by integrating federated learning, enhancing privacy [45].

To the best of our knowledge, this work is the first attempt to develop a diffusion model for synthesizing financial tabular that explicitly addresses the challenge of class imbalance mitigation.

3 METHODOLOGY

In this section, we introduce *Imbalanced Financial Diffusion* (*Imb-FinDiff*), an extension of *Financial Diffusion* (*FinDiff*) by Sattarov et al. [46], to synthesize imbalanced tabular data. We first introduce the foundational concepts of diffusion models, followed by a detailed description of the proposed *Imb-FinDiff* framework.

3.1 Gaussian Diffusion Models

Denoising diffusion probabilistic models [19, 51] are latent variable models that gradually add Gaussian noise to data in a forward process and learn to reverse this process to generate samples. The forward process transforms data $\mathbf{x}_0 \in \mathbb{R}^d$ into latent variables $\mathbf{x}_1, ..., \mathbf{x}_T$ via a Markov chain, ultimately reaching Gaussian noise $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Each Markov transition has the form:

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}) \tag{1}$$

where β_t is the noise level added at timestep t. The process allows for direct sampling of $\mathbf{x}_t$ given $\mathbf{x}_0$:

$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\hat{\beta}_t}\mathbf{x}_0, \hat{\beta}_t\mathbf{I}) \tag{2}$$

where $\hat{\beta}_t = 1 - \prod_{i=0}^{t}(1-\beta_i)$. In the reverse process, the model incrementally denoises the latent variables x_t to recover the data x_0. To approximate this process, we train a neural network f_θ with parameters θ, and each denoising step is parameterized as:

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \Sigma_\theta(\mathbf{x}_t, t)) \tag{3}$$

where μ_θ and Σ_θ are the mean and covariance of $q(x_t|x_{t-1})$. Since Σ_θ is diagonal, following Ho et al. [19], $\boldsymbol{\mu}_\theta$ can be parametrized as:

$$\boldsymbol{\mu}_\theta(\mathbf{x}_t, t) = \frac{1}{\sqrt{\alpha_t}}(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\hat{\alpha_t}}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)) \tag{4}$$

where $\alpha_t := 1 - \beta_t$, $\hat{\alpha_t} := \prod_{i=0}^{t}\alpha_i$, and $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)$ predicts the noise component. The mean-squared error between the ground truth ϵ and the estimated $\epsilon(x_t, t)$ has been empirically shown to produce better results compared to the variational lower bound $\log p_\theta(x_0)$:

$$\mathcal{L}_t = \mathbb{E}_{x_0,\epsilon,t}\left[|\epsilon - \epsilon_\theta(x_t, t)|_2^2\right] \tag{5}$$

While this framework is effective for continuous data, particularly images, and videos, it cannot be directly applied to discrete data, such as categorical attributes in tabular datasets.

3.2 Financial Tabular Diffusion Framework

The *Imb-FinDiff* framework extends the foundational concepts of *FinDiff* [46] by introducing a dual learning objective, integrating both (i) a *timestep noise loss* and (ii) a novel *class label loss*. This learning objective improves the framework's capability to accurately generate data for a desired class. Formally, let X be a population of $i = 1, 2, \ldots, K$ tabular records, where each record is defined as:

$$x_i = (x_i^{cat_1}, \ldots, x_i^{cat_N}; x_i^{num_1}, \ldots, x_i^{num_M}; y_i) \tag{6}$$

consisting of N categorical attributes x^{cat}, M numerical attributes x^{num}, and the class label y_i. The proposed learning framework comprises three modules, as illustrated in Fig. 2.

3.2.1 Embedding Module. The first module embeds the (i) tabular record attributes and labels as well as the (ii) diffusion timestep in continuous embedding spaces $E \in \mathbb{R}^{D_1}$ respectively:

- *Categorical Attribute Embedding:* The individual categorical attributes x^{cat} are transformed into continuous embeddings, denoted as $e^{cat} \in \mathbb{R}^{D_1}$, via designated embedding layers m_θ^{cat}.
- *Numerical Attribute Embedding:* The individual numerical attributes x^{num} are transformed into continuous embeddings, denoted as $e^{num} \in \mathbb{R}^{D_1}$, via a sequence of linear layers m_θ^{num}.

Next, Gaussian timestep noise is added to the categorical and numerical embeddings. The timesteps t are randomly sampled from a uniform distribution. Following Eq. 1, this is achieved by:

$$e_t^{cat} = \sqrt{1-\beta_t} \cdot e^{cat} + \sqrt{\beta_t} \cdot \epsilon, \quad \epsilon \sim \mathcal{N}(0, I), \tag{7}$$

$$e_t^{num} = \sqrt{1-\beta_t} \cdot e^{num} + \sqrt{\beta_t} \cdot \epsilon, \quad \epsilon \sim \mathcal{N}(0, I), \tag{8}$$

where β_t denotes the noise at timestep t, and ϵ is Gaussian noise.

- *Diffusion Timestep Embedding:* The diffusion noise timesteps t are transformed into continuous embeddings, denoted as $e_t^{time} \in \mathbb{R}^{D_1}$, using positional encodings, proposed by Vaswani *et al.* [58].
- *Data Class Embedding:* The categorical class labels y, are transformed into continous embeddings, denoted as $e^{label} \in \mathbb{R}^{D_1}$, via embedding layers m_θ^{label}, proposed by Kotelnikov *et al.* [26].

The derived embeddings e^{cat}, e^{num}, e^{time}, and $e^{label} \in \mathbb{R}^{D_1}$ are then processed by the denoising diffusion module.

3.2.2 Denoising Diffusion Module. The second module processes the synthesized embeddings to remove noise and accurately predict the class labels. It operates in three stages, as described below:

- *Embedding Projection:* Each embedding e^* is projected into a joint embedding space $Z \in \mathbb{R}^{D_2}$, where $D_2 > D_1$, using a set of embedding head functions $\hat{e}^* = g_\theta^*(e^*)$ with parameters θ.
- *Embedding Synthesis:* Following, a combined embedding vector $z_t = e_t^{cat} \odot e_t^{num} \odot e_t^{time} \odot e^{label}$ is constructed, where $z_t \in \mathbb{R}^{D_2}$ and $\odot$ denotes the Hadamard product. Given the combined embeddings z_t, a synthesizer network learns a joint representation that enables accurate noise and class label prediction.
- *Prediction De-Embedding:* The synthesized embeddings are projected back into the original embedding space $E \in \mathbb{R}^{D_1}$, using two projection head functions, as given by:

$$\boldsymbol{\epsilon}_{\theta,t}^* = h_\theta^{noise}(z_t^*), \quad y_{\theta,t}^* = h_\theta^{label}(z_t^*), \tag{9}$$

where $\boldsymbol{\epsilon}_{\theta,t} \in \mathbb{R}^{D_2}$ denotes the predicted noise and $y_{\theta,t} \in \mathbb{R}^{D_2}$ the predicted class label of the synthesized embedding.

The derived de-embedded predictions $\hat{\boldsymbol{\epsilon}}_{\theta,t}$ and $\hat{y}_{\theta,t}$ in $\mathbb{R}^{D_1}$ are then processed by the prediction module.

3.2.3 Prediction Module. The third module computes the framework's dual learning objective: accurate (i) timestep noise and (ii) class label predictions. By incorporating the *Class Label Loss*, the *Imb-FinDiff* model is guided towards robust class-specific learning, enhancing generative capability for minority class synthesis.

- *Timestep Noise Prediction:* The timestep noise loss $\mathcal{L}_t^\epsilon$, ensuring accurate denoising, is defined as:

$$\mathcal{L}_t^\epsilon = \mathbb{E}_{\epsilon,t}\left[|\epsilon - \epsilon_{\theta,t}^*|_2^2\right], \tag{10}$$

and computes the mean-squared-error loss between the true noise ϵ and the predicted noise $\epsilon_{\theta,t}^*$.

- *Class Label Prediction:* The class label loss $\mathcal{L}_t^y$, ensuring class-specific accuracy, is defined as:

$$\mathcal{L}_t^y = \mathbb{E}_{y,t}\left[|y - y_{\theta,t}^*|_2^2\right], \tag{11}$$

and measures the mean-squared-error loss between the true class label y and the predicted class label $y_{\theta,t}^*$.

The combination of both losses $\mathcal{L}_t = \mathcal{L}_t^\epsilon + \mathcal{L}_t^y$ constitutes a generative model learning framework, effectively addressing class imbalance complexities in mixed-type tabular data.

Given an optimized *Imb-FinDiff* model, a reverse diffusion process $p_\theta(\mathbf{z}_{t-1}|\mathbf{z}_t)$ can be initiated, as defined in Eq. 4, to sample and progressively reconstruct minority class tabular data records.

4 EXPERIMENTAL SETUP

In this section, we describe the experiments conducted. We outline the datasets, data preprocessing steps, diffusion model training setup, baselines, and evaluation metrics.

4.1 Datasets and Data Preparation

We benchmarked the developed technique using four real-world mixed-type tabular datasets selected to provide diversity in the proportion of categorical and numeric attributes. The details of the datasets are as follows:

- **Adult Income ($\mathcal{D}_1$)**: This dataset,[1] extracted from the 1994 Census database, contains records of individuals' income along with attributes such as age, education, and native country. It is commonly used as a benchmark for imbalanced learning tasks [5].
- **Accounting Entries ($\mathcal{D}_2$)**: This dataset,[2] is an excerpt of the synthetic dataset presented in [48]. It resembles accounting data and comprises attributes such as posting date, account, posting type, posting amount, and currency.
- **City Payments ($\mathcal{D}_3$)**: This dataset,[3] contains nearly a quarter-million lines of payments data from city offices, departments, boards, and commissions. It covers the City's payments during the 2017 fiscal year, totalling nearly $4.2 billion.
- **Card Transactions ($\mathcal{D}_4$)**: This dataset,[4] includes an extensive collection of credit card transactions. The transaction records contain various features, including transaction details and device information, recorded over two months in 2018.

Table 1 presents the descriptive statistics of the datasets. Notably, dataset $\mathcal{D}_1$ exhibits a more balanced class distribution compared to datasets $\mathcal{D}_2$, $\mathcal{D}_3$, and $\mathcal{D}_4$, which exhibit fewer minority samples.

Table 1: Statistics of the mixed-type tabular datasets, including the count and percentage of minority samples.

		Attributes		Records	Minority	
Data	Name	Cat.	Num.	Abs.	Abs.	Rel.
$\mathcal{D}_1$	Adult Income	8	6	48,842	11,687	23.93%
$\mathcal{D}_2$	Accounting Entries	6	2	533,010	30	0.01%
$\mathcal{D}_3$	City Payments	15	1	238,895	200	0.08%
$\mathcal{D}_4$	Card Transactions	20	372	590,542	20,663	3.50%

4.2 Diffusion Model Training

The diffusion models are trained on samples from all classes within each dataset, which is divided into 80% for training and 20% for testing. During training, the models learn the attribute distributions of each class, enabling the generative model to produce synthetic instances that augment the minority class and address the class imbalance.

4.2.1 Architecture Setup. The *Imb-FinDiff* model architecture includes several key components optimized for mixed-type tabular datasets as detailed in Tab. 2. The architecture encompasses three

[1] https://archive.ics.uci.edu/ml/datasets/adult
[2] https://www.kaggle.com/ntnu-testimon/paysim1
[3] https://opendataphilly.org/datasets/city-payments
[4] https://www.kaggle.com/c/ieee-fraud-detection/data

networks: (i) the embedding net (g_θ), (ii) the backbone net (f_θ), and (iii) the projection net (h_θ). The networks consist of fully connected layers. In g_θ and h_θ, we apply SELU non-linear activations while Leaky ReLU activations with $\alpha = 0.4$ are used in f_θ.[5]

4.2.2 Hyperparameters. The following general hyperparameters were chosen: 100 diffusion steps with a linear schedule and diffusion betas ranging from 0.0001 to 0.1. During training, each model was optimized for a maximum of 5,001 iterations with a mini-batch size of 64, using the Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$. A learning rate of 0.0005 was used, coupled with a cosine learning rate scheduler. The weights were initialized using Xavier uniform distributions, and a weight decay of 1e-6 was applied.

Table 2: Architectural configurations of the *Imb-FinDiff* models optimized for mixed-type tabular datasets ($\mathcal{D}_1$ to $\mathcal{D}_4$).

	Architectural Setup		
Data	Emb.-Net g_θ	Backbone-Net f_θ	Proj.-Net h_θ
$\mathcal{D}_1$	{4, 256, 512, 1,024}	{1,024, 1,024, 1,024, 1,024}	{512, 256, 4}
$\mathcal{D}_2$	{4, 256, 512, 1,024}	{512, 512, 512, 512}	{512, 256, 4}
$\mathcal{D}_3$	{4, 256, 512, 1,024}	{1,024, 1,024, 1,024, 1,024}	{512, 256, 4}
$\mathcal{D}_4$	{4, 512, 1,024, 2,048}	{2,048, 2,048, 2,048, 2,048}	{1,024, 512, 4}

4.2.3 Baselines. To evaluate the effectiveness of the proposed *Imb-FinDiff* model, we compared it against three well-established baseline methods for handling class imbalance:

- **Oversampling** [18]: An effective technique that increases the number of instances in the minority class by randomly selecting samples with replacements from the minority instances, potentially leading to better model generalization.
- **SMOTE** [5]: A widely used technique that generates synthetic samples for the minority class by interpolating between existing minority instances. This approach balances the class distribution by creating new instances in the feature space.
- **ADASYN** [17]: An adaptive synthetic sampling approach that generates synthetic data based on the density distribution of the minority class. By focusing on harder-to-learn examples, ADASYN adaptively changes the decision boundary.

Additionally, we also compared against a setting in which no minority class sampling is applied (denoted as 'none').[6]

4.3 Diffusion Model Evaluation

The trained diffusion models are used to generate 100,000 minority class samples, which are then merged with 100,000 random samples of the original dataset. This results in a balanced training set that effectively addresses class imbalance.

4.3.1 Machine Learning Classifiers. To evaluate the effectiveness of the synthetic data generated by the *Imb-FinDiff* model, we trained several machine learning classifiers on the balanced training set.[7] The models' hyperparameters are grid-searched:

- **Gaussian Naive Bayes (GNB)**: The grid search for GNB optimized the `var_smoothing` parameter over the range [1e-9, 1e-8, 1e-7, 1e-6, 1e-5, 1e-4, 1e-3, 1e-2, 1e-1, 1].
- **Logistic Regression (LogReg)**: The grid search for logistic regression included the parameters `C` [0.001, 0.01, 0.1, 1, 10, 100, 1000], `penalty` ['l1', 'l2'], and `solver` ['lbfgs'].
- **XGBoost (XGB)**: The grid search for XGBoost included the parameters `n_estimators` [100, 200, 500], `learning_rate` [0.01, 0.05, 0.1], and `gamma` [0.1, 0.5].
- **Decision Tree (DTree)**: The grid search for decision trees included the parameters `criterion` ['gini', 'entropy'], `max_depth` [3, 5, 7, 9], and `max_features` ['sqrt', 'log2'].
- **k-Nearest Neighbors (kNN)**: The grid search for kNN included the parameters `n_neighbors` [3, 5, 7, 9, 11], `weights` ['uniform', 'distance'], and `metric` ['minkowski'].

Each hyperparameter combination was evaluated using five random seeds to ensure result robustness.

4.3.2 Evaluation Measures. To assess the quality of the synthesized data, we evaluate the trained models on (i) performance in downstream machine learning tasks and (ii) synthetic data fidelity.

Utility. Utility measures the functional equivalence of synthetic data to real-world data. We assessed utility by training classifiers on the generated synthetic data (S_{train}) and evaluating them against the actual test set (X_{test}). The overall utility of S_{train} is represented by the average F_1-Score across all classifiers, formalized as:

$$F_1^* = \frac{1}{N}\sum_{i=1}^{N} F_1^i(S_{train}, X_{test}), \tag{12}$$

where F_1^* represents the utility score of the synthetic data, and F_1^i denotes the F_1-Score of the i-th classifier. The F_1-Score determines the harmonic mean of classifier model precision and recall. The evaluation determines whether the synthetic data effectively enables effective classification model learning.

Fidelity. Fidelity measures how closely synthetic data emulates real-world data at column and row levels.[8] Numeric attributes are evaluated using the *Wasserstein Similarity* (WS) [23], and categorical attributes using the *Jensen-Shannon Divergence* (JS) [27]. The column fidelity score ϕ_{col}^d is defined as:

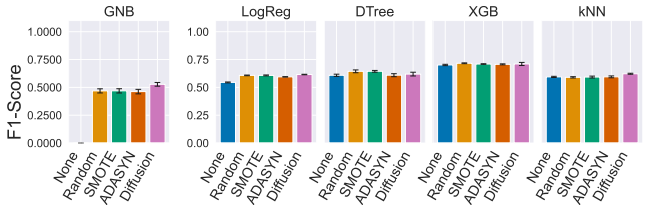
$$\phi_{col}^d = \begin{cases} 1 - WS(X^d, S^d) & \text{if } d \text{ is numeric} \\ 1 - JS(X^d, S^d) & \text{if } d \text{ is categorical} \end{cases} \tag{13}$$

where X denotes the original data, S the synthetic data, and d denotes the column index. The synthetic dataset's column fidelity is the mean of ϕ_{col}^d across all attributes. Row fidelity evaluates correlations between column pairs. For numeric attributes, it uses *Pearson Correlation* (PC) [2], and for categorical attributes, it uses the *Theil U* (TU) coefficient [53]. The row fidelity score $\phi_{row}^{a,b}$ is defined as:

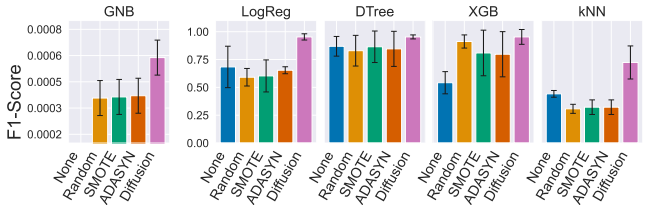
$$\phi_{row}^{a,b} = \begin{cases} 1 - PC(X^{a,b}, S^{a,b}) & \text{if } a,b \text{ are numeric} \\ 1 - TU(X^{a,b}, S^{a,b}) & \text{if } a,b \text{ are categorical} \end{cases} \tag{14}$$

where X denotes the original data, S the synthetic data, and a, b denote the column indices. The synthetic dataset's row fidelity

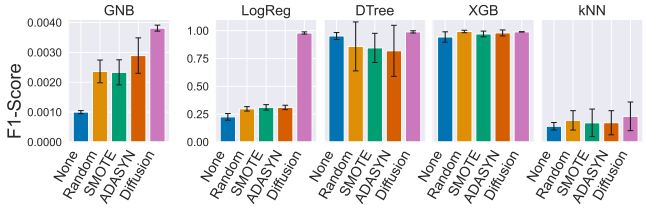
[5] We implemented the architecture using the *PyTorch* library [39].
[6] We used the baselines as implemented in the *Imbalanced-Learn* library [29].
[7] We used the classifiers as implemented in the *Scikit-Learn* library [40].
[8] We used the measures as implemented in the *Dython* library [70].



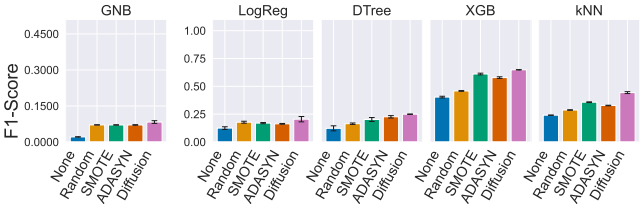
(a) Utility Scores: Adult Income Dataset ($\mathcal{D}_1$).



(b) Utility Scores: Accounting Entries Dataset ($\mathcal{D}_2$).



(c) Utility Scores: City Payments Dataset ($\mathcal{D}_3$).



(d) Utility Scores: Card Transactions Dataset ($\mathcal{D}_4$).

Figure 3: Utility scores (F_1^*) for five classifiers (GNB, Logistic Regression, Decision Tree, XGB, and kNN) under different minority sampling techniques (None, Random, SMOTE, ADASYN, and Diffusion). Each visualization (a)-(d) highlights the classifier performance across the sampling techniques for the evaluated mixed-type tabular datasets ($\mathcal{D}_1$ to $\mathcal{D}_4$).

score is the mean of $\phi_{row}^{a,b}$ across all attribute pairs. A dataset's aggregate fidelity score, denoted as ϕ^*, is the mean of column and row fidelities. The evaluation determines whether the synthetic data replicates the statistical properties of the original data.

5 EXPERIMENTAL RESULTS

In this section, we present and assess the results of three research questions (RQs) when evaluating the *Imb-FinDiff* framework for generating financial tabular data to address class imbalance.

RQ 1: *How does the utility of Imb-FinDiff compare to other sampling techniques in handling class imbalance?*

We compare *Imb-FinDiff* to the baselines. The average utility performance (F_1^*) across classifiers per method and dataset is presented in Tab. 3, while detailed results are illustrated in Fig. 3.

Table 3: Utility metric comparison of imbalance sampling methods on mixed-type tabular datasets ($\mathcal{D}_1$ to $\mathcal{D}_4$). The most effective technique for each dataset is highlighted in bold.

	Imbalance Sampling Technique				
Data	None ↑	Random ↑	SMOTE ↑	ADASYN ↑	Diffusion ↑
$\mathcal{D}_1$	0.49 ± 0.28	0.61 ± 0.09	0.60 ± 0.09	0.59 ± 0.09	0.61 ± **0.08**
$\mathcal{D}_2$	0.51 ± 0.33	0.53 ± 0.38	0.52 ± 0.36	0.52 ± 0.36	0.72 ± **0.41**
$\mathcal{D}_3$	0.45 ± 0.46	0.47 ± 0.43	0.46 ± 0.43	0.46 ± 0.42	0.64 ± **0.48**
$\mathcal{D}_4$	0.18 ± 0.15	0.23 ± 0.15	0.28 ± 0.21	0.27 ± 0.19	0.33 ± **0.22**
Avg.	0.41 ± 0.30	0.46 ± 0.26	0.47 ± 0.27	0.46 ± 0.27	0.57 ± **0.30**

*Variances originate from evaluating using five classifiers and seeds.

Results. The results indicate distinct performance differences among the three groups of sampling techniques.

- **None and Random Sampling**: Both methods show similar utility scores. None sampling has an average utility score of 0.41, while Random sampling improves to 0.46.
- **SMOTE and ADASYN**: Both methods only slightly outperform Random sampling, with SMOTE achieving an average utility score of 0.47 and ADASYN at 0.46.
- **Diffusion**: Diffusion consistently outperforms other methods, achieving the highest utility score of 0.57.

The obtained results suggest that the proposed diffusion models effectively handle class imbalance in mixed-type tabular data. Moreover, the stability across different datasets highlights robustness and generalizability.

Table 4: Fidelity metric comparison of imbalance sampling methods on mixed-type tabular datasets ($\mathcal{D}_1$ to $\mathcal{D}_4$). The most effective technique for each dataset is highlighted in bold.

	Imbalance Sampling Technique				
Data	None ↑	Random ↑	SMOTE ↑	ADASYN ↑	Diffusion ↑
$\mathcal{D}_1$	0.26 ± 0.24	0.26 ± 0.24	0.27 ± 0.24	0.28 ± 0.23	0.34 ± **0.18**
$\mathcal{D}_2$	0.35 ± 0.11	0.34 ± 0.11	0.34 ± 0.13	0.34 ± 0.13	0.42 ± **0.11**
$\mathcal{D}_3$	0.36 ± 0.19	0.38 ± 0.23	0.47 ± 0.17	0.47 ± 0.17	0.51 ± **0.13**
$\mathcal{D}_4$	0.26 ± 0.23	0.26 ± 0.24	0.29 ± 0.22	0.31 ± 0.21	0.42 ± **0.56**
Avg.	0.31 ± 0.19	0.31 ± 0.21	0.34 ± 0.19	0.35 ± 0.19	0.42 ± **0.25**

*Variances originate from evaluating using five classifiers and seeds.

RQ 2: *How well does the Imb-FinDiff model maintain fidelity in synthetic data compared to real-world data?*

We compare *Imb-FinDiff* to the baselines. The fidelity performance for each method and dataset, represented as the aggregated fidelity score (ϕ^*), is presented in Tab. 4.

Results. The results indicate distinct performance differences among the three groups of sampling techniques.

- **None and Random Sampling**: Both methods show similar fidelity scores. None sampling has an average fidelity score of 0.31, while Random sampling scores 0.31.

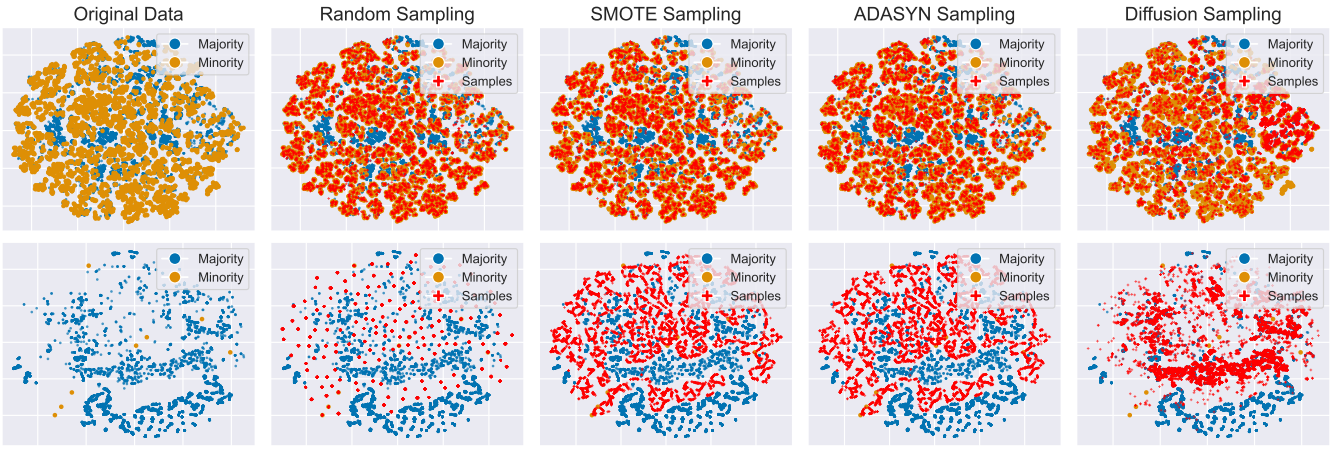


Figure 4: Example t-SNE [57] visualizations of majority class samples (blue), minority class samples (orange), and oversampled minority class samples (red) using different sampling techniques. The top-row results correspond to the (more balanced) Adult Income dataset ($\mathcal{D}_1$), and the bottom-row results to the (less balanced) City Payments dataset ($\mathcal{D}_3$). From left to right: (a) original data, (b) random sampling, (c) SMOTE, (d) ADASYN, and (e) *Imb-FinDiff* diffusion sampling.

- **SMOTE and ADASYN**: These methods slightly outperform Random sampling, with SMOTE achieving an average fidelity score of 0.34 and ADASYN at 0.35.
- **Diffusion**: Diffusion achieves the highest fidelity score of 0.42, consistently outperforming other methods.

The results suggest that the proposed diffusion models effectively replicate the real-world data's statistical properties. Additionally, the fidelity scores highlight the potential for accurate data synthesis in different financial datasets.

RQ 3: *Does Imb-FinDiff effectively capture the diversity of minority class samples compared to other sampling techniques?*

To evaluate the efficacy of oversampling techniques, we examine t-SNE[57] visualizations of the majority, minority, and oversampled minority class of the datasets $\mathcal{D}_1$ to $\mathcal{D}_3$ as shown in Fig. 4.

Results. The result indicates that the diffusion model samples more effectively augment the datasets' local structures.

- **Random Sampling**: Introduces new samples that mitigate global imbalance but do not improve local structures, showing some overlap and scattered minority points.
- **SMOTE and ADASYN**: Generate synthetic samples by interpolating between existing minority points, enhancing local density for minority points but maintaining distinct clusters.
- **Diffusion Sampling**: Introduces synthetic samples in a diffused manner, providing improved local and resulting in a balanced spread of minority samples across local structures.

The results suggest that diffusion sampling provides a more balanced and denser representation in the dataset's local structure regions. In these regions, classifiers often fail to discriminate between majority and minority class samples. Consequently, diffusion sampling enhances downstream model performance and generalization.

6 CONCLUSION AND FUTURE WORK

In this study, we introduced *Imb-FinDiff*, a diffusion-based generative technique for synthesizing high-fidelity financial tabular data. The *Imb-FinDiff* architecture effectively handles mixed-type data by embedding categorical and numerical attributes, addressing the complexities of financial datasets. Our evaluation showed that the learned model replicates feature distributions and maintains feature correlations, supporting the utility of the synthetic data for downstream tasks. Additionally, the model mitigates class imbalance, enhancing the performance of machine learning models trained on the synthesized data. In future work, we envision advanced conditioning mechanisms to (i) improve the granularity of minority class sample generation and (ii) extend the technique to handle temporal dependencies observable in financial datasets.

ACKNOWLEDGMENTS

This work was supported in parts by the Office of Advanced Scientific Computing Research, Office of Science, U.S. Department of Energy, under Contract No. DE-AC02-05CH11231, and utilized resources from the National Energy Research Scientific Computing Center (NERSC). Marco's research is funded by an IFI fellowship (No. 57515245) from the German Academic Exchange Service (DAAD). The views expressed in this work are those of the authors and do not necessarily represent the view of the Deutsche Bundesbank.

REFERENCES

[1] Samuel T Assefa, Sayantan Banerjee, Sahika Celik, Jinsong Du, and Carsten Eickhoff. 2020. Generating Synthetic Data to Mitigate Diversity Constraints in Imbalanced Learning. *arXiv preprint arXiv:2010.01537* (2020).
[2] Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. 2009. Pearson Correlation Coefficient. *Noise Reduction in Speech Processing* (2009), 1–4.
[3] Andrew Brock, Jeff Donahue, and Karen Simonyan. 2019. Large Scale GAN Training for High Fidelity Natural Image Synthesis. In *International Conference on Learning Representations*.
[4] Caroline Chan, Shiry Ginosar, Tinghui Zhou, and Alexei A Efros. 2019. Everybody Dance Now. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 5933–5942.

[5] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. 2002. SMOTE: Synthetic Minority Over-Sampling Technique. *Journal of Artificial Intelligence Research* 16 (2002), 321–357.

[6] Nitesh V Chawla, Aleksandar Lazarevic, Lawrence O Hall, and Kevin W Bowyer. 2003. SMOTEBoost: Improving Prediction of the Minority Class in Boosting. In *European Conference on Principles of Data Mining and Knowledge Discovery*. Springer, 107–119.

[7] Zhipeng Chen, Michael R Lyu, and Irwin King. 2022. CodeT: Code Generation with Pre-trained Transformer Language Models. *arXiv preprint arXiv:2203.08501* (2022).

[8] Yinan Cheng, Chi-Hua Wang, Vamsi K Potluru, Tucker Balch, and Guang Cheng. 2024. Downstream Task-Oriented Generative Model Selections on Synthetic Data Training for Fraud detection Models. *arXiv preprint arXiv:2401.00974* (2024).

[9] Andrei Dogariu, Parth Patil, and Wolfgang K Härdle. 2021. Synthetic Data Generation for Financial Applications Using Generative Models. *Journal of Financial Econometrics* (2021).

[10] Pedro Domingos. 1999. MetaCost: A General Method for Making Classifiers Cost-Sensitive. *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining* (1999), 155–164.

[11] Chris Drummond and Robert C Holte. 2003. C4.5, Class Imbalance, and Cost Sensitivity: Why Under-Sampling Beats Over-Sampling. *Workshop on Learning from Imbalanced Datasets II* 11 (2003), 1–8.

[12] Johann Engelmann and Stefan Lessmann. 2021. Conditional Wasserstein GAN-based Oversampling of Tabular Data for Imbalanced Learning. *Expert Systems with Applications* 174 (2021), 114582.

[13] Seyda Ertekin, Jian Huang, Léon Bottou, and C Lee Giles. 2007. Active Learning for Class Imbalance Problem. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 823–824.

[14] Sheng Gao, Wei Han, Ming Liu, Huaixiu Yang, and Yu Duan. 2023. Differmer: Discrete Diffusion Model for Text Generation. *arXiv preprint arXiv:2301.07817* (2023).

[15] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. *Generative Adversarial Networks*. arXiv preprint arXiv:1406.2661.

[16] Haibo Guo and Herna L Viktor. 2004. Learning from Imbalanced Data Sets with Boosting and Data Generation: The DataBoost-IM Approach. In *International Conference on Discovery Science*. Springer, 99–111.

[17] Haibo He and Yang Bai. 2008. ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning. *2008 IEEE International Joint Conference on Neural Networks* (2008), 1322–1328.

[18] Haibo He and Yunqian Ma. 2013. *Imbalanced Learning: Foundations, Algorithms, and Applications*. Wiley-IEEE Press.

[19] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising Diffusion Probabilistic Models. *arXiv preprint arXiv:2006.11239*.

[20] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2021. Diffusion Models Beat GANs on Image Synthesis. *arXiv preprint arXiv:2105.05233* (2021).

[21] Emiel Hoogeboom, Rianne van den Berg, and Max Welling. 2021. Argmax Flows and Multinomial Diffusion: Learning Categorical Distributions. *arXiv preprint arXiv:2102.05379* (2021).

[22] James Jordon, Jinsung Yoon, and Mihaela van der Schaar. 2018. PATE-GAN: Generating Synthetic Data with Differential Privacy Guarantees. In *International Conference on Learning Representations*.

[23] Leonid V Kantorovich. 1958. On the Translocation of Masses. *Management Science* 5, 1 (1958), 1–4.

[24] Sungho Kim, Taesup Kwon, and Jaekyun Yoo. 2021. Tabular Data Generation with Ordinary Differential Equations. *arXiv preprint arXiv:2110.13142* (2021).

[25] Diederik P Kingma and Max Welling. 2014. Auto-Encoding Variational Bayes. In *International Conference on Learning Representations*.

[26] Ilya Kotelnikov, Igor Melnyk, Vaibhav Garg, Neil Lawrence, Kari Torkkola, and Yin-Cheung Wong. 2022. TabDDPM: Modeling Tabular Data with Diffusion Models. *arXiv preprint arXiv:2203.09467* (2022).

[27] Gaetano Lamberti and Ana P Majtey. 2007. Jensen's Inequality and Quantum Information Geometry. *Physics Letters A* 365, 5-6 (2007), 458–463.

[28] Quoc V Le, Mark Chen, Ryan Prenger, Javier Valle, Wei Han, Jong Wook Kim, and Tom Sercu. 2023. Voicebox: A Generative Model for Speech Synthesis and Enhancement. *arXiv preprint arXiv:2305.00592* (2023).

[29] Guillaume Lemaître, Fernando Nogueira, and Christos K. Aridas. 2017. Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning. *Journal of Machine Learning Research* 18, 17 (2017), 1–5. http://jmlr.org/papers/v18/16-365

[30] Jiatao Li, Haoran Li, Pengcheng Xie, Tongshuang Zhao, Yizhe Li, Thamme Gowda, and Lawrence Carin. 2022. Diffusion-LM: Controlled Text Generation with Diffusion Models. *arXiv preprint arXiv:2208.05199* (2022).

[31] Yujia Li, Sreejan Kadavath, Paul Smolensky, Samira Ebrahimi Kahou, Saad Mohammad, Pierre Castonguay, Oriol Vinyals, Dragomir Radev, Ankur Bapna, Matthew Hoffman, et al. 2022. Competition-Level Code Generation with AlphaCode. *Science* 377, 6603 (2022), 705–708.

[32] Shujian Liao, Hao Ni, Lukasz Szpruch, Magnus Wiese, Marc Sabate-Vidales, and Baoren Xiao. 2020. Conditional Sig-Wasserstein GANs for Time Series Generation. *arXiv preprint arXiv:2006.05421* (2020).

[33] Charles X Ling and Victor S Sheng. 2004. Decision trees with minimal costs. In *Proceedings of the 21st international conference on Machine learning*. ACM, 69.

[34] Xiao Ni, Zhiguang Chen, Tianwen Yang, Linlin Zheng, Jingyi Yu, and Tianshou Zhao. 2020. Conditional Sig-Wasserstein GANs for Time Series Generation. *arXiv preprint arXiv:2002.12184* (2020).

[35] Xiao Ni, Zhiguang Chen, Tianwen Yang, Linlin Zheng, Jingyi Yu, and Tianshou Zhao. 2021. SigWGAN: Generative Modeling of Financial Time-Series. *Quantitative Finance* 21, 4 (2021), 527–543.

[36] Richard Nock and Marianne Guillame-Bert. 2022. Generative Trees for Interpretable Synthetic Data Generation. *Machine Learning* 111 (2022), 203–228.

[37] OpenAI. 2023. GPT-4 Technical Report. *arXiv preprint arXiv:2303.08774* (2023).

[38] Xiao Ouyang, Xiao Ni, and Tianshou Zhao. 2023. MissDiff: Missing Data Imputation with Diffusion Models. *arXiv preprint arXiv:2301.12345* (2023).

[39] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. *Advances in Neural Information Processing Systems* 32 (2019), 8026–8037.

[40] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011).

[41] Vamsi K Potluru, Daniel Borrajo, Andrea Coletta, Niccolò Dalmasso, Yousef El-Laham, Elizabeth Fons, Mohsen Ghassemi, Sriram Gopalakrishnan, Vikesh Gosai, Eleonora Kreačić, et al. 2023. Synthetic Data Applications in Finance. *arXiv preprint arXiv:2401.00081* (2023).

[42] Sameer S Pradhan, Edward Loper, Dmitriy Dligach, and Martha Palmer. 2005. Active Learning for Word Sense Disambiguation with Methods for Addressing the Class Imbalance Problem. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 397–404.

[43] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-Resolution Image Synthesis with Latent Diffusion Models. *arXiv preprint arXiv:2112.10752* (2022).

[44] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-Resolution Image Synthesis with Latent Diffusion Models. *arXiv preprint arXiv:2112.10752* (2022).

[45] Timur Sattarov, Marco Schreyer, and Damian Borth. 2023. FedTabDiff: Federated Learning of Diffusion Probabilistic Models for Synthetic Mixed-Type Tabular Data Generation. *arXiv preprint arXiv:2401.06263* (2023).

[46] Timur Sattarov, Marco Schreyer, and Damian Borth. 2023. Findiff: Diffusion Models for Financial Tabular Data Generation. In *Proceedings of the Fourth ACM International Conference on AI in Finance*. 64–72.

[47] Marco Schreyer, Timur Sattarov, Damian Borth, Andreas Dengel, and Bernd Reimer. 2018. Detection of Anomalies in Large Scale Accounting Data Using Deep Autoencoder Networks. *Nvidia's GPU Technology Conference (GTC)* (2018).

[48] Marco Schreyer, Timur Sattarov, Damian Borth, Benedikt Reimer, Michael Wursthorn, and Miklos A. Vasarhelyi. 2019. Detection of Accounting Anomalies in the Latent Space Using Adversarial Autoencoder Neural Networks. *arXiv preprint arXiv:1912.02757* (2019).

[49] Chris Seiffert, Taghi M Khoshgoftaar, Jason Van Hulse, and Amri Napolitano. 2009. RUSBoost: Improving Classification Performance when Training Data is Skewed. *2009 IEEE International Conference on Data Mining* (2009), 1–9.

[50] Yonatan Singer, Adam Polyak, Tali Dekel Hayes, Harsh N Padnos, Guy Sella, Galia Omer, Avraham Navon, Devi Parikh, Maor Ofri-Amar, Ameer Mahajerin, et al. 2022. Make-a-Video: Text-to-Video Generation without Text-Video Data. *arXiv preprint arXiv:2209.14792* (2022).

[51] Jascha Sohl-Dickstein, Eric A Weiss, Niru Maheswaranathan, and Surya Ganguli. 2015. Deep Unsupervised Learning using Nonequilibrium Thermodynamics. *arXiv preprint arXiv:1503.03585* (2015).

[52] Robin Strudel, Mathilde Caron, Hugo Touvron, Matthieu Cord, and Ivan Laptev. 2022. Self-conditioned Embedding Diffusion for Text Generation. *arXiv preprint arXiv:2205.14217* (2022).

[53] Henri Theil. 1972. *Statistical Decomposition Analysis: With Applications in the Social and Administrative Sciences*. North-Holland Publishing Company.

[54] Simon Tong and Edward Chang. 2001. Support Vector Machine Active Learning for Image Retrieval. In *Proceedings of the Ninth ACM International Conference on Multimedia*. ACM, 107–118.

[55] Atabak Torfi, Emily B Fox, and Charles Elkan. 2022. Differentially Private Synthetic Data Generation for Unstructured Data Using Generative Models: Application in Healthcare. In *Proceedings of the 36th Conference on Neural Information Processing Systems*.

[56] Hugo Touvron, Théo Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Ferhan

Azhar, et al. 2023. LLaMA: Open and Efficient Foundation Language Models. *arXiv preprint arXiv:2302.13971* (2023).

[57] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing Data Using t-SNE. *Journal of Machine Learning Research* 9, Nov (2008), 2579–2605.

[58] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. In *Advances in Neural Information Processing Systems*. 5998–6008.

[59] Shouhong Wang and Xin Yao. 2007. Improving Classification Performance on Imbalanced Data using Kernel-based Methods. In *Proceedings of the 19th IEEE International Conference on Tools with Artificial Intelligence*. IEEE, 647–654.

[60] Zi Wang, Long Kang, and Shiyin Huang. 2023. Neural Vocoder: Neural Network-based Vocoding and Speech Synthesis. *arXiv preprint arXiv:2304.07997* (2023).

[61] Xu Wei and Stephen A Billings. 2003. Regularized Orthogonal Least Squares. *International Journal of Modeling and Simulation* 23, 2 (2003), 88–99.

[62] Xiao Wen, Hang Zhao, and Kun Yi. 2022. Causal-TGAN: A Model for Generating High-Quality Synthetic Tabular Data Using Causal Relationships. *arXiv preprint arXiv:2205.11343* (2022).

[63] Moritz Wiese, René Knobloch, Ralf Korn, and Paul Kretschmer. 2019. Quant GANs: Deep Generation of Financial Time Series. *Quantitative Finance* 20, 9 (2019), 1419–1431.

[64] Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. 2019. Modeling Tabular Data Using Conditional GAN. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. 7335–7345.

[65] Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. 2019. Modeling Tabular Data using Conditional GAN. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. 7335–7345.

[66] Ling Yang, Zhilong Zhang, Yang Song, Wentao Hong, ..., Bin Cui, and Ming-Hsuan Yang. 2023. Diffusion Models: A Comprehensive Survey of Methods and Applications. *Computing Surveys* 56, 4 (2023).

[67] Yu Zhang, Wei Zheng, Qing Liu, and Xiaohui Li. 2021. GANBLR: Generative Adversarial Networks for Balancing and Interpreting Class Imbalanced Data. *IEEE Transactions on Neural Networks and Learning Systems* 33, 6 (2021), 2366–2378.

[68] Hang Zhao, Kun Yi, and Anshul Tiwari. 2021. CTAB-GAN: Effective Table Data Synthesis via Auxiliary Classifier GANs. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence*. 4483–4490.

[69] Zhi-Hua Zhou and Xu-Ying Liu. 2006. Training Cost-Sensitive Neural Networks with Methods Addressing the Class Imbalance Problem. *IEEE Transactions on Knowledge and Data Engineering* 18, 1 (2006), 63–77.

[70] Shaked Zychlinski and contributors. 2024. Dython: A collection of Data Science Tools in Python. https://github.com/shakedzy/dython