Mitigating Data Imbalance in Time Series Classification Based on Counterfactual Minority Samples Augmentation

Lei Wang
Chongqing University
Chongqing, China
20222401006@stu.cqu.edu.cn

Shanshan Huang
Chongqing University
Chongqing, China
shanshanhuang@cqu.edu.cn

Chunyuan Zheng
Peking University
Beijing, China
cyzheng@stu.pku.edu.cn

Jun Liao
Chongqing University
Chongqing, China
liaojun@cqu.edu.cn

Xiaofei Zhu
Chongqing University of Technology
Chongqing, China
zxf@cqut.edu.cn

Haoxuan Li
Peking University
Beijing, China
hxli@stu.pku.edu.cn

Li Liu*
Chongqing University
Chongqing, China
dcsliuli@cqu.edu.cn

Abstract

Imbalanced Time-Series Classification is a critical, yet challenging task across a spectrum of real-world applications. Previous oversampling and generative approaches primarily target the minority class and often rely on static decision boundaries or similarity-based heuristics. However, these methods overlook the underlying causal factors that govern the distinction between majority and minority classes, particularly in scenarios with ambiguous class boundaries. As a result, the generated samples may fail to enhance class separability, thereby limiting improvements in classification performance. To this end, we propose a **C**ounter**F**actual **A**ugmentation **M**inority **G**eneration (**CFAMG**) method based on generative models that aims to discover the causal factors that determine different classes from a causality perspective. Specifically, our method first utilizes a disentangled classifier to distinguish between causal and non-causal factors. Next, we perform counterfactual intervention by replacing the causal factors of majority class samples with those from minority class samples, creating an intervened latent representation that reflects minority characteristics while preserving essential structures. Finally, the trained minority class decoder generates counterfactual minority samples that resemble real minority instances yet remain distinguishable from the original majority class. Extensive experiments demonstrate that our method outperforms state-of-the-art methods in both univariate and multivariate imbalanced time-series classification tasks. The code is published at https://github.com/WangLei-CQU/CFAMG.

CCS Concepts

• **Information systems → Data mining**.

Keywords

Imbalanced Time Series Classification; Counterfactual Augmentation; Disentangled Representation Learning

ACM Reference Format:
Lei Wang, Shanshan Huang, Chunyuan Zheng, Jun Liao, Xiaofei Zhu, Haoxuan Li, and Li Liu. 2025. Mitigating Data Imbalance in Time Series Classification Based on Counterfactual Minority Samples Augmentation. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2 (KDD '25), August 3–7, 2025, Toronto, ON, Canada.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3711896.3737049

1 Introduction

Imbalanced time-series classification (ITSC) is a significant challenge in machine learning and data mining, where the primary issue arises from the imbalanced distribution of class labels. In this paper, we focus on the binary classification setting of ITSC, aiming to distinguish two highly imbalanced classes. In various practical scenarios, including credit fraud detection [9], facial recognition [4], and industrial fault detection [7], the minority class—typically of heightened interest—remains significantly underrepresented relative to the majority class. Such an imbalance can bias models in favor of the majority class, leading to suboptimal predictive performance when it comes to the minority class, which in turn can result in poor generalization and heavy misclassification costs [16].

Methods for addressing imbalanced learning can be broadly categorized into data-based and algorithm-based approaches. Data-based methods, such as undersampling, oversampling, and hybrid techniques, adjust class distribution through resampling strategies, either by removing excess majority class samples or generating synthetic minority class samples [53]. In contrast, algorithm-based methods [38] modify learning algorithms to better handle imbalanced data using techniques like cost-sensitive learning and ensemble learning. This paper focuses on data-based methods. Beyond

[*] Li Liu is the corresponding author.

KDD '25, Toronto, ON, Canada

ACM ISBN 979-8-4007-1454-2/2025/08
https://doi.org/10.1145/3711896.3737049

traditional sampling-based approaches, deep generative models provide an alternative solution for addressing class imbalance [40]. These methods combine sampling strategies with generative models, allowing for the synthesis of minority class samples while embedding discriminative information from the classification task. By leveraging the power of deep generative models, these approaches can capture complex data distributions and generate highly realistic minority samples that preserve intricate structures [1].

Traditional oversampling methods, such as SMOTE and its variants [37], synthesize samples of the minority class by interpolating between existing samples and their nearest neighbors. These methods aim to improve class balance and reduce overfitting by creating more diverse synthetic samples, especially in regions such as decision boundaries or areas with fewer minority samples [16]. However, when methods focus solely on augmenting data from the minority class, they still inevitably lead to issues such as overfitting and lack of diversity, ultimately reducing the generalization performance of the classifier. One possible reason is that existing oversampling methods primarily focus on the correlation matrix such as distance and similarity between samples to generate new minority class instances [5, 6, 19, 29, 36, 61], without considering the causal factors that influence class labels. Consequently, correlation-based oversampling may yield samples that closely resemble the majority class, leading to sub-optimal performance, particularly when class boundaries are ambiguous.

To address these challenges, we propose a dual disentangled VAE framework to identify the latent causal factors that truly determine the class labels. Since causal and non-causal factors are entangled in the data, we first employ disentangled representation learning methods [51] to separate the causal factors. Additionally, we can leverage counterfactual thinking by intervening in the causal factors of the majority class to generate counterfactual minority class samples, thereby fully utilizing the information in the majority class and closely approximating the real data distribution. Following this line of thought, we propose a **C**ounter**F**actual **A**ugmentation **M**inority **G**eneration (**CFAMG**) for handling the ITSC task. Specifically, we first employ a temporal neural network structure to encode both minority and majority class samples into their respective latent representations. Each encoder is designed with a dual-output head, where one head is responsible for extracting the causal latent factors, while the other captures the non-causal factors corresponding to each class. Next, we train a disentangled classifier combined with mutual information regularization, which enables the model to learn causal factor representations. Once the disentanglement process is complete, we intervene on the causal factors of the majority class by replacing them with the causal factors from the minority class, and then generate counterfactual minority class samples through the trained minority class decoder, achieving more realistic and discriminative minority class instances generation. The contributions of our work are as follows:

- We reveal the limitations of existing oversampling methods and approach the ITSC problem from a causal perspective. Specifically, our work is the first to develop counterfactual generation for ITSC tasks.
- We utilize disentangled representation learning to uncover the causal factors of both the majority and minority classes in the latent space of the generative model, and intervene in the causal factors of the majority class to generate realistic and representative minority class samples.
- Extensive experiments conducted on several benchmark datasets, along with analytical results, validate the effectiveness and superiority of our approach.

2 Related Works

2.1 Oversampling Methods

Oversampling is a data-driven and effective method for addressing data imbalance. Prominent examples include SMOTE [6] and its various enhancements [2, 5, 32, 34, 61, 64], which typically generate synthetic samples based on K-nearest neighbors and class decision boundaries. However, since these methods do not involve a learning process, the generated samples can be highly similar when the data are extremely imbalanced, which results in overfitting. Additionally, while generating samples near the decision boundary aims to improve classification accuracy by focusing on critical samples, these methods may not always capture the underlying factors driving the classification, leading to degraded performance. Recent oversampling methods for ITSC, such as T-SMOTE [61], employ an LSTM-based [41] auxiliary classifier to assign scores to minority-class samples and select time subsequences with varying lead lengths for generation. While this design aims to enhance temporal representation, its effectiveness is highly dependent on the quality of the classifier. Furthermore, BFGAN (Boundary-Focused GAN) [29], as a generative model to address ITSC, also faces the challenge of classifier performance and parameter selection. Similarly, these methods overlook the underlying causal mechanisms that fundamentally distinguish the majority and minority classes when the decision boundary is ambiguous or difficult to model. To address the aforementioned issue, we focus on learning the causal factors that distinguish the minority and majority classes. Additionally, we employ a classifier whose performance does not heavily impact the process, encouraging the model to automatically learn the causal representations of different classes in the latent space. Leveraging these causal factors, we can generate realistic and discriminative samples to effectively tackle the ITSC task.

2.2 CounterFactual Generation & Explanations

Causality-based techniques [22, 30, 54, 55, 58, 59] provide a new perspective in data mining [31, 45–47, 60, 62, 63], among which counterfactual generation is a key approach. It creates hypothetical scenarios that differ from existing data to explore "what if" questions [35, 43, 48–50]. This approach enhances a model's causal reasoning and robustness to unforeseen situations. It has proven especially effective in improving interpretability in fields such as computer vision [23–26] and natural language processing [13, 39]. In time series analysis, counterfactuals involve modifying historical data to understand how changes in certain variables or events could alter the system's evolution. Therefore, counterfactual explanations have become one of the key research directions in this field [15, 42, 44]. TimeTuner [18] integrates counterfactual explanations to connect time series representations, features, and predictions, aiding in the understanding of model behavior. CounTS [57] generates counterfactual explanations via a variational Bayesian deep learning model, offering actionable insights while preserving prediction accuracy. This approach reveals causal relationships and enhances model interpretability by identifying key factors. While substantial progress has been made in the application of counterfactual explanations to this field, extending counterfactual techniques to other tasks is equally critical for further advancing research in this domain. Therefore, our work is the first to develop counterfactual generation to for ITSC tasks, exploring how generating counterfactual minority class samples from the majority class influence the performance of ITSC models.

3 Preliminary

3.1 Problem Setup

Consider an imbalanced time series dataset $X = \{X^P, X^N\}$, where $X^P = \{X_i^P\}_{i=1}^m$ and $X^N = \{X_j^N\}_{j=1}^n$ represent the minority and majority class samples with corresponding labels $y^P = 1$ and $y^N = 0$, respectively. Here, $m \ll n$ highlights the significant class imbalance, where the number of minority class samples is much smaller than the number of majority class samples. Specifically, a minority sample $X_i^P = \{x_i^1, \ldots, x_i^T\} \in \mathbb{R}^{T \times d}$ is a sequence of data with feature dimension d and sequence length T. The ITSC task focuses on developing a model to predict the category of time-series data, despite the inherent class imbalances.

3.2 Existing Oversampling Methods for ITSC

The existing oversampling techniques for ITSC task can be categorized into the following approaches:
Covariance-based methods. These methods regularize the covariance structure of the minority-class samples X^P [5] by computing their covariance matrix W_P:

$$W_P = \frac{1}{m}\sum_{i=1}^{m}(x_i^P - \bar{x}^P)(x_i^P - \bar{x}^P)^T,$$

where $\bar{x}^P$ is the mean of X^P. Then, these method performs eigen decomposition $W_P = VDV^T$ and adjusts the eigenvalues in D to obtain the regularized covariance $W_P^* = VD^*V^T$. Synthetic samples are generated from $x_{\text{new}}^P \sim \mathcal{N}(\bar{x}^P, W_P^*)$.
Clustering-based methods. These methods partition X^P into K clusters $C_1, C_2, \ldots, C_K$ based on Density-Ratio Shared Nearest Neighbor, each cluster C_i has mean μ_i^P and empirical covariance S_i [64]. Define shrinkage covariance matrix S_i^*:

$$S_i^* = \alpha F + (1-\alpha)S_i, \quad F_{ij} = \sqrt{(S_i)_{ii}(S_i)_{jj}},$$

where α is a regularization parameter. Synthetic samples are generated per cluster: $x_{\text{new}}^P \sim \mathcal{N}(\mu_i^P, S_i^*)$.
Leading-time-based methods. These methods model temporal dependencies in X^P via LSTM to predict sample scores s_i^l [61] and determines the maximum leading time L by a spy-based technique. For each time step l, generate synthetic samples via interpolation:

$$x_{\text{new}}^P = \alpha x_i^l + (1-\alpha)x_i^{l+1}, \quad \alpha \sim \mathcal{B}(s_i^l, s_i^{l+1}),$$

where $\mathcal{B}(\cdot)$ is a Beta distribution.

3.3 Generative Model-Based Methods for ITSC

Generative models generate minority class samples by learning the data distribution, addressing class imbalance while preserving discriminative features. Boundary information is extracted by calculating the neighborhood density of each minority sample X_i^P using k-nearest neighbors and local outlier factor, assigning importance labels $v_i = 1$ for cautious samples and $v_i = 0$ for certain samples [29]. The generator G takes inputs (z, c, v), where $z \sim \text{Uniform}(-1, 1)^d$ is noise, c is the class label, and v is the importance label, generating samples $G(z, c, v)$. The discriminator D classifies real X^P or generated samples and estimates their importance label. Finally, the generator generates minority class samples to balance the dataset.

4 Methodology

4.1 Motivation

Existing methods for ITSC primarily model correlations between minority and majority classes using techniques like similarity-based clustering or covariance modeling. However, these correlation-driven approaches risk capturing spurious associations, introducing bias in generated data and degrading classifier performance. Additionally, they focus solely on minority class characteristics while overlooking fundamental distinctions between majority and minority classes. To address this, we adopt a causal perspective for ITSC by disentangling causal factors that determine class labels from non-causal elements. This enables the model to generate minority samples with stronger discriminative power. Moreover, leveraging majority class information enables our method to produce samples with enhanced realism and quality.

4.2 Overall Model

Figure 1 (a) illustrates the objective of our proposed CFAMG, which aims to disentangle the latent causal and non-causal factors and intervene in the causal factors of the majority class to generate counterfactual minority class samples. Specifically, for X^P and X^N, we can decompose it into two independent representations, corresponding to the causal factor z_c^P, z_c^N that determines the class, and the non-causal factor z_{nc}^P, z_{nc}^N that is unrelated to the category. Then, we intervene in the causal factor z_c^N of samples from the majority class to generate counterfactual samples $\hat{X}_{cf}^P$ corresponding to the minority class label y^P through the generative model.

As shown in Figure 1 (b), we use a dual VAE architecture to represent latent factors and generate counterfactual samples. Unlike traditional VAE [27], our framework separately encodes minority and majority class samples into latent factors. It then performs disentanglement and counterfactual intervention in the latent space, followed by the generation of counterfactual minority class samples through a minority decoder. Further details are provided in the following sections.

4.3 Learning Disentangled Representation

In this section, we will introduce how to construct causal and non-causal factors through the framework of Figure 1 (b) and how to disentangle the two types of factors.

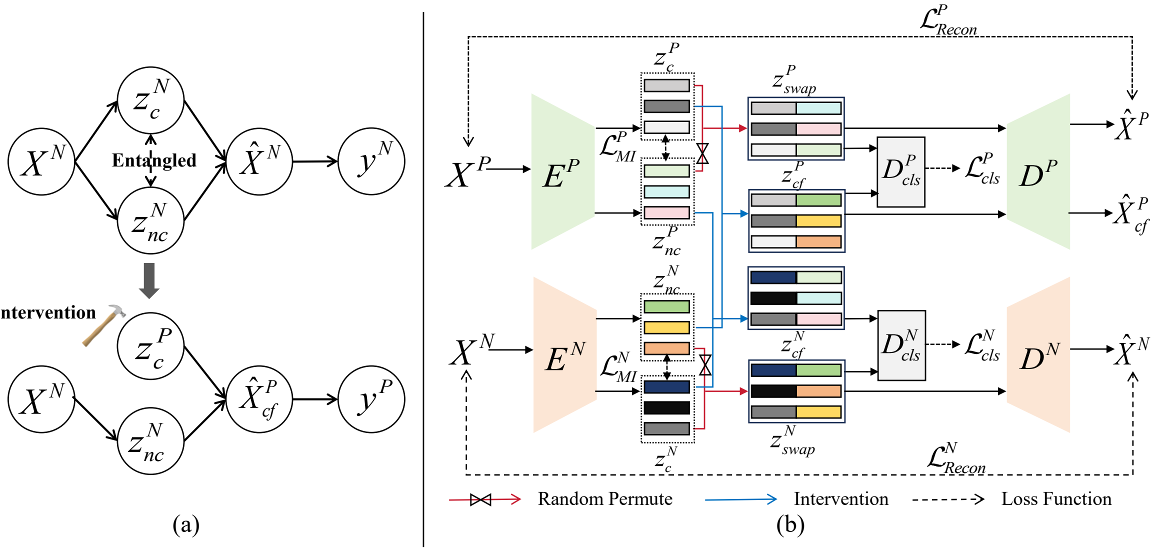


Figure 1: (a) The goal of CFAMG: By intervening with the latent causal factors (z_c^P) of the minority class (X^P) on the latent causal factors (z_c^N) of the majority class (X^N), counterfactual samples ($\hat{X}_{cf}^P$) are generated with reversed labels ($y^N \rightarrow y^P$). (b) The framework of CFAMG: X^N and X^P are encoded by encoders E^P and E^N to obtain z_c^P and z_c^N, respectively. These causal factors are further separated using disentangled classifiers D_{cls}^P and D_{cls}^N, along with mutual information regularization. Finally, the intervention on the latent factors z_{cf}^P is applied to generate $\hat{X}_{cf}^P$ using the trained decoder D^P. Details can be found in Section 4.

4.3.1 Causal and Non-causal Factor. To capture the latent causal and non-causal representations of both minority and majority classes, and to enable counterfactual intervention without flipping the ground truth label, we need to learn the latent representations of the minority and majority samples separately.

Given a batch, the minority and majority class samples are represented as $X^P \in \mathbb{R}^{B\times d\times T}$ and $X^N \in \mathbb{R}^{B\times d\times T}$, respectively, where $B = \min(m, B)$, with B being the batch size and m the number of minority class samples. Furthermore, we design dual output heads at the encoder's input to obtain the representations of causal factors and non-causal factors. This design serves two purposes: first, to distinguish different latent factors by obtaining diverse representations through different output heads, and second, to aid in disentangling the two types of factors. This can be expressed by

$$z_c^P, z_{nc}^P = E^P(X^P; \theta^P), \quad z_c^N, z_{nc}^N = E^N(X^N; \theta^N), \tag{1}$$

where θ^P and θ^N are the parameters of encoders E^P and E^N, respectively. The encoders output causal and non-causal factors for the minority and majority classes, i.e., $z_c^P, z_{nc}^P \in \mathbb{R}^{B\times k/2}$ and $z_c^N, z_{nc}^N \in \mathbb{R}^{B\times k/2}$, k is the latent space dimension. Next, we will construct an auxiliary classifier to assist in disentangling.

4.3.2 Disentangled Classifier. The disentangled classifier is designed to decouple causal and non-causal factors, ensuring they can be effectively utilized in counterfactual interventions and sample generation. To train the disentangled classifier, we combine causal and non-causal factors in different ways and input them into the classifier for training. Specifically, for the minority disentangled classifier D_{cls}^P, we construct two types of swaps: ① the internal swap $z_{swap}^P = \textbf{Concatenate}(z_c^P, z_{rp}^P) \in \mathbb{R}^{B\times k}$, where z_{rp}^P represents a randomly permuted z_{nc}^P; ② the counterfactual swap $z_{cf}^P = \textbf{Concatenate}(z_c^P, z_{nc}^N) \in \mathbb{R}^{B\times k}$. Internal random swapping is an enhancement strategy for disentangled learning, while external swapping is a key step in counterfactual intervention. Thus, for D_{cls}^P, our objective function is to minimize the following loss:

$$\mathcal{L}_{\text{cls}}^P = \sum_{i=1}^{\lceil m/B \rceil} [CE\left(D_{\text{cls}}^P\left(\{z_{cf}^P\}_i\right), y^P\right) + CE\left(D_{\text{cls}}^P\left(\{z_{\text{swap}}^P\}_i\right), y^P\right)], \tag{2}$$

where CE denotes the cross-entropy loss. Similarly, for the majority disentangled classifier D_{cls}^N, the input construction method is analogous to D_{cls}^P, so the objective function for D_{cls}^N is to minimize the following loss:

$$\mathcal{L}_{\text{cls}}^N = \sum_{i=1}^{\lceil n/B \rceil} \left[CE\left(D_{\text{cls}}^N\left(\{z_{cf}^N\}_i\right), y^N\right) + CE\left(D_{\text{cls}}^N\left(\{z_{\text{swap}}^N\}_i\right), y^N\right)\right]. \tag{3}$$

Note that our disentangled classifier differs significantly from those in the existing literature [29, 61] in both function and performance. Firstly, the goal of the disentangled classifier is to separate causal and non-causal factors. Consequently, it relies less on the inherent performance of the classifier to some extent. Secondly, the disentangled classifier is designed to be an overfitting classifier. For different classes of samples, the classifier's role is to ensure that the latent factor representations remain within their original classes, even after performing two types of swap operations. Therefore, in the disentanglement task, the high memory capacity and accuracy of a classifier that tends to overfit become advantageous. Finally, since the disentangled classifier is less dependent on the classifier's performance, its design is simpler compared to traditional classifier models. A single fully connected layer serves as the disentangled classifier in this work.

4.3.3 Mutual Information Regularization. To further ensure the disentanglement of causal and non-causal factors, we introduce a mutual information constraint on the latent representations of both factors. Specifically, we compute the loss over a batch of samples, ensuring that the disentanglement is enforced consistently across different instances within each batch. Accordingly, we minimize the following batch loss:

$$\mathcal{L}_{MI} = \mathcal{L}_{MI}^{P} + \mathcal{L}_{MI}^{N} = I(z_c^P, z_{nc}^P) + I(z_c^N, z_{nc}^N), \tag{4}$$

where $I(\cdot, \cdot)$ denotes the mutual information between two representations. The optimization strategy to minimize $I(z_c^P, z_{nc}^P)$ follows the approach outlined in [8]. First, we derive an upper bound for the mutual information $I(z_c^P, z_{nc}^P)$, which can be expressed as:

$$\begin{aligned} & I(z_c^P, z_{nc}^P) \\ & \leq \mathbb{E}_{P(z_c^P, z_{nc}^P)}[\log P(z_{nc}^P \mid z_c^P)] - \mathbb{E}_{P(z_c^P)P(z_{nc}^P)}[\log P(z_{nc}^P \mid z_c^P)] \\ & = I_{\text{CLUB}}^{P}(z_c^P, z_{nc}^P). \end{aligned} \tag{5}$$

Equality holds if and only if z_c^P and z_{nc}^P are independent variables. However, directly estimating the joint distribution $P(z_c^P, z_{nc}^P)$ and the conditional distribution $P(z_{nc}^P \mid z_c^P)$ is challenging. To address this, following the approach in [52], we introduce a parameterized variational distribution $Q(z_c^P, z_{nc}^P)$, which serves as an approximation to the true joint distribution. Hence, we define $Q(z_c^P, z_{nc}^P)$ as:

$$\begin{aligned} Q(z_c^P, z_{nc}^P) &= \frac{Q(z_c^P)Q(z_{nc}^P)}{Z_\beta} e^{\gamma_P(z_c^P, z_{nc}^P)}, \\ Z_\beta &= \mathbb{E}_{Q(z_c^P)Q(z_{nc}^P)}[e^{\gamma_P(z_c^P, z_{nc}^P)}], \end{aligned} \tag{6}$$

therefore, Eq (5) can be rewritten as:

$$\begin{aligned} & I_{\text{CLUB}}^{P}(z_c^P, z_{nc}^P) \\ & = \mathbb{E}_{Q(z_c^P, z_{nc}^P)}[\gamma_P(z_{nc}^P \mid z_c^P)] - \mathbb{E}_{Q(z_c^P)Q(z_{nc}^P)}[\gamma_P(z_{nc}^P \mid z_c^P)], \end{aligned} \tag{7}$$

where $\gamma_P(z_c^P, z_{nc}^P)$ is a learnable minority critic function, and Z_β is the normalization constant ensuring that $Q(z_c^P, z_{nc}^P)$ integrates to 1. Using this variational distribution, we can compute the expected values and obtain the upper bound for the mutual information. To ensure stability during optimization and prevent the upper bound from becoming negative, we add an L2 regularization to constrain the negative values of the upper bound. The final loss function $\mathcal{L}_{MI}^{P}$ can be expressed as follows:

$$\mathcal{L}_{MI}^{P} = I_{\text{CLUB}}^{P}(z_c^P, z_{nc}^P) + \|I_{\text{CLUB}}^{P}(z_c^P, z_{nc}^P)\|^2. \tag{8}$$

The optimization strategy for $I(z_c^N, z_{nc}^N)$ follows the same approach as described above and can be expressed as follows:

$$\begin{aligned} & I(z_c^N, z_{nc}^N) \leq I_{\text{CLUB}}^{N}(z_c^N, z_{nc}^N) \\ & = \mathbb{E}_{Q(z_c^N, z_{nc}^N)}[\gamma_N(z_{nc}^N \mid z_c^N)] - \mathbb{E}_{Q(z_c^N)Q(z_{nc}^N)}[\gamma_N(z_{nc}^N \mid z_c^N)], \end{aligned} \tag{9}$$

where $\gamma_N(z_c^N, z_{nc}^N)$ is a learnable majority critic function. The $\mathcal{L}_{MI}^{N}$ can be expressed as follows:

$$\mathcal{L}_{MI}^{N} = I_{\text{CLUB}}^{N}(z_c^N, z_{nc}^N) + \|I_{\text{CLUB}}^{N}(z_c^N, z_{nc}^N)\|^2. \tag{10}$$

4.3.4 Counterfactual Intervention and Generation. The purpose of counterfactual intervention is to observe the change in the outcome variable by altering the value of the causal variable. One of the goals of this study is to examine the impact of counterfactual samples on time series imbalance classification, particularly when the decisive features of majority samples are modified to convert their labels to minority. After the model is fully trained and the disentanglement of causal and non-causal factors for both minority and majority classes is completed, we perform counterfactual intervention on the latent causal factors z_c^N of majority samples to generate counterfactual samples. Specifically, the intervention is carried out by replacing z_c^N with the latent causal factors of minority samples (z_c^P), forming a joint representation Z_{cf}^P with z_{nc}^N. The counterfactual minority samples are then generated using the minority decoder D^P, expressed as:

$$\hat{X}_{cf}^{P} = D^P(z_{cf}^P). \tag{11}$$

In theory, CFAMG can generate $m \times n$ counterfactual samples. From these, we select $n - m$ samples to complement the original minority class samples, forming a balanced dataset.

4.4 Overall Objective Function

To ensure the learned latent factor representations faithfully capture the true representations of both minority and majority classes, we have the following objectives for a batch B of X^P and X^N:

$$\begin{aligned} \mathcal{L}_{\text{gen}}^{P} &= D_{\text{KL}}(q(z_c^P, z_{nc}^P | x^P) \parallel p(x^P | z_c^P, z_{nc}^P)), \\ \mathcal{L}_{\text{gen}}^{N} &= D_{\text{KL}}(q(z_c^N, z_{nc}^N | x^N) \parallel p(x^N | z_c^N, z_{nc}^N)), \end{aligned} \tag{12}$$

where D_{KL} denotes the KL divergence, q represents the learned representation distribution, and p represents the true representation distribution. Since the true distributions $p(x^P | z_c^P, z_{nc}^P)$ and $p(x^N | z_c^N, z_{nc}^N)$ cannot be directly calculated [11], we transform them into the following objectives:

$$\begin{aligned} \mathcal{L}_{\text{gen}}^{P} &= \mathcal{L}_{\text{Recon}}^{P} + \mathcal{L}_{\text{KL}}^{P} \\ &= -\mathbb{E}_{z_c^P, z_{nc}^P \sim q_{E^P}(z_c^P, z_{nc}^P | x^P)} \log p_{D^P}(x^P | z_c^P, z_{nc}^P) \\ &+ D_{\text{KL}}(q(z_c^P, z_{nc}^P | x^P) \parallel p(z \mid N(0, I))). \end{aligned} \tag{13}$$

Following the same approach, we can denote the $\mathcal{L}_{\text{gen}}^{N}$ as:

$$\begin{aligned} \mathcal{L}_{\text{gen}}^{N} &= \mathcal{L}_{\text{Recon}}^{N} + \mathcal{L}_{\text{KL}}^{N} \\ &= -\mathbb{E}_{z_c^N, z_{nc}^N \sim q_{E^N}(z_c^N, z_{nc}^N | x^N)} \log p_{D^N}(x^N | z_c^N, z_{nc}^N) \\ &+ D_{\text{KL}}(q(z_c^N, z_{nc}^N | x^N) \parallel p(z \mid N(0, I))), \end{aligned} \tag{14}$$

where E and D represent the encoder and decoder, respectively. The first term computes the reconstruction loss between the generated samples and the original samples using MSE, while the second term approximates the latent factor representation distribution using a standard normal distribution. In summary, our overall objectives are as follows:

$$\mathcal{L} = (\mathcal{L}_{\text{gen}}^{P} + \mathcal{L}_{\text{gen}}^{N}) + \alpha \cdot (\mathcal{L}_{\text{cls}}^{P} + \mathcal{L}_{\text{cls}}^{N}) + \beta \cdot \mathcal{L}_{MI}, \tag{15}$$

where α and β are a balancing factor. Unless otherwise specified, α and β are set to 1 in this paper.

Table 1: Experimental results on the average performance of MLP, LSTM, and ResNet classifiers across 53 UCR and UEA datasets. The best performances are highlighted in bold.

Classifier	**MLP**			**LSTM**			**ResNet**		
Methods	F1 ↑	G-Means ↑	AUC ↑	F1 ↑	G-Means ↑	AUC ↑	F1 ↑	G-Means ↑	AUC ↑
None	0.2851	0.3307	0.7594	0.1714	0.1416	0.5852	0.3689	0.3828	0.7695
SMOTE	0.5616	0.6578	0.8096	0.3540	0.3838	0.6362	0.5252	0.6007	0.7888
BL-SMOTE	0.5316	0.6292	0.7953	0.3356	0.3924	0.6511	0.5074	0.5841	0.7808
MWMOTE	0.5485	0.6484	0.8026	0.3773	0.4140	0.6103	0.5213	0.5990	0.7840
ADASYN	0.5507	0.6607	0.8207	0.3928	0.4104	0.6387	0.5547	0.6237	0.8007
MBS	0.5165	0.5924	0.8052	0.3435	0.3788	0.6412	0.3785	0.4280	0.7757
INOS	0.5524	0.6413	0.7994	0.3771	0.3998	0.5932	0.4758	0.5497	0.7829
OHIT	0.4773	0.5773	0.7577	0.3240	0.3672	0.5995	0.3638	0.4080	0.7520
T-SMOTE	0.4807	0.5469	0.7853	0.3273	0.3493	0.6280	0.4783	0.5414	0.7825
BFGAN	0.4969	0.5682	0.8134	0.2473	0.2506	0.5773	0.3309	0.3740	0.7619
CSMOTE	0.4391	0.5395	0.7625	0.2099	0.2855	0.5584	0.3319	0.3924	0.7500
TCGAN	0.4985	0.5457	0.7787	0.3047	0.3075	0.6038	0.3482	0.4058	0.7502
I-GAN	0.5458	0.6349	0.7885	0.3398	0.3732	0.6344	0.4030	0.4567	0.7716
H-SMOTE	0.5239	0.6337	0.7751	0.2956	0.3774	0.6191	0.4809	0.5677	0.7557
CFAMG	**0.6963**	**0.7839**	**0.8751**	**0.5743**	**0.6896**	**0.7787**	**0.6593**	**0.7379**	**0.8592**

5 Experiments

5.1 Experimental Setups

5.1.1 Datasets. In our experiments, we used 53 time series datasets sourced from the UCR and UEA time series repository [1] to evaluate the performance of the proposed method. To construct imbalanced datasets, the class(es) with the fewest samples were designated as the minority class, and the remaining classes were treated as the majority class for dataset construction. For datasets with a low imbalance ratio (IR) (less than 2), we chose to discard a portion of the minority class samples, but no fewer than 10 samples. After balancing the training set, we trained classification models using this balanced training set and evaluated the quality of the generated data with the original test set. Dataset statistics are provided in the **Appendix Table 6**.

5.1.2 Competitors. To evaluate the performance of CFAMG, we compared it against 13 classical oversampling methods, the latest ITSC method based on sampling and deep generation methods: *None*, *SMOTE* [6], *Borderline-SMOTE (BL-SMOTE)* [17], *MWMOTE* [2], *ADASYN* [19], *MBS* [33], *INOS* [5], *OHIT* [64], *T-SMOTE* [61], *BF-GAN* [29], CSMOTE [34], TCGAN [21], I-GAN [36] and H-SMOTE [32]. The *None* represents training and testing with the original data. Note that methods such as SMOTE, Borderline-SMOTE, MW-MOTE, ADASYN, MBS, OHIT, and H-SMOTE require the input to be explicitly reshaped into one-dimensional form.

5.1.3 Evaluation Metrics. In our experiments, we used three common metrics in the field of imbalanced learning: *F1 Score* [56], *G-mean* [3], and *AUC* [12], to evaluate the model's performance in handling imbalanced data, avoiding the misleading nature of accuracy. The F1 Score is the harmonic mean of precision and recall, reflecting the model's ability to distinguish both minority and majority classes. G-Mean measures the geometric mean of sensitivity and specificity, ensuring balanced performance across classes. It is computed as: $G\text{-}Mean = \sqrt{\frac{TP}{TP+FN} \times \frac{TN}{TN+FP}}$. AUC represents the Area Under the ROC Curve, assessing the model's overall performance by plotting the True Positive Rate against the False Positive Rate. Moreover, *MMD* [14] and *FID* [20] are employed to assess the quality of the generated data. We use the MMD metric with multi-bandwidth RBF kernels to measure the distributional difference between generated minority and real minority data. A lower MMD indicates a closer match to the real distribution. Since FID is commonly used for image quality assessment, we followed the FID setup from the literature [28], using a pretrained FCN model to extract features from real and generated time series, and then calculating the mean and covariance of these features. Finally, by computing the Frechet distance between the feature distributions, a lower FID indicates higher generation quality.

5.1.4 Classifier Specification. In our experiments, we utilized the proposed method and the comparative methods to achieve balanced datasets, which were then evaluated using a consistent classifier. Specifically, we employed the MLP, LSTM, and ResNet classifiers for this assessment. The MLP classifier consists of three fully connected layers with configurations [500, 500, 500], while the LSTM classifier uses a single layer with default parameters. The ResNet classifier is composed of three residual blocks with varying kernel sizes (7, 5, 3), followed by adaptive average pooling and a fully connected layer for classification.

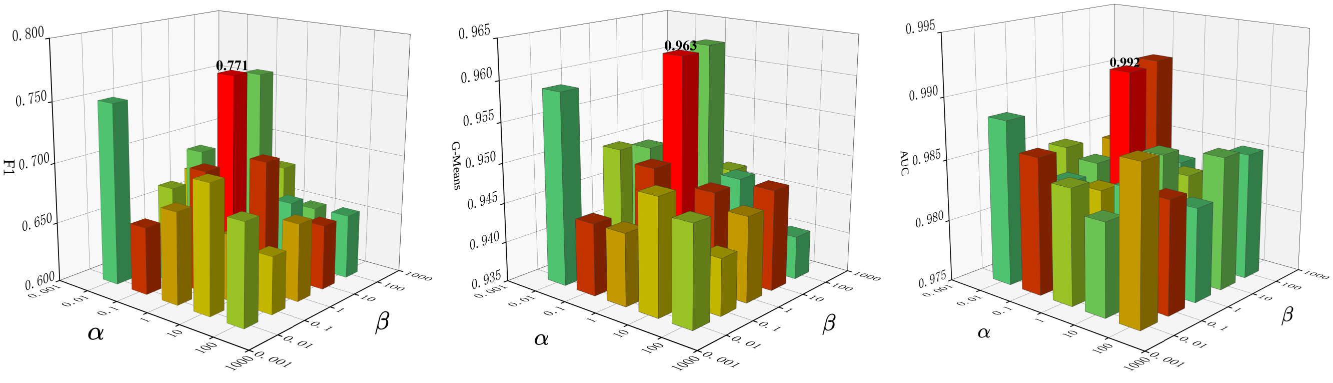
5.2 Experimental Results

5.2.1 Experimental Results on All Datasets. Table 1 presents a comparison of the average performance of CFAMG and baseline methods on the 53 constructed imbalanced datasets. The baseline methods are divided into two categories: non-temporal oversampling methods and temporal sampling methods.

[1] https://www.timeseriesclassification.com/dataset.php

Table 2: Performance of different imbalance rates on the MPOC dataset using MLP classifier.

IR (#Minority)	**5.54 (70, Original)**			**7.76 (50)**			**12.93 (30)**			**38.8 (10)**			**77.6 (5)**		
Methods	F1 ↑	G-Means ↑	AUC ↑	F1 ↑	G-Means ↑	AUC ↑	F1 ↑	G-Means ↑	AUC ↑	F1 ↑	G-Means ↑	AUC ↑	F1 ↑	G-Means ↑	AUC ↑
SMOTE	0.7339	0.7680	0.8644	0.6843	0.7276	0.8473	0.6069	0.6649	0.8152	0.4698	0.5568	0.7895	0.2802	0.4031	0.8088
BL-SMOTE	0.7104	0.7475	0.8721	0.6732	0.7174	0.8694	0.5415	0.6101	0.8687	0.5273	0.6007	0.8076	0.1918	0.3197	0.7954
ADASYN	0.7335	0.7681	0.8555	0.6903	0.7324	0.8475	0.5703	0.6359	0.8138	0.4594	0.5485	0.7875	0.2734	0.3975	0.8040
MWMOTE	0.7450	**0.7778**	0.8649	0.6981	0.7386	0.8498	0.6032	0.6616	0.8380	0.5070	0.5850	0.8238	0.2543	0.3812	0.8047
MBS	0.6743	0.7171	0.8930	0.6358	0.6854	0.8807	0.5658	0.6298	0.8498	0.4725	0.5577	0.8519	0.2240	0.3532	0.7760
INOS	0.7408	0.7725	0.8899	0.7080	0.7462	0.8751	0.6597	0.7059	0.8560	0.4757	0.5605	0.8010	0.3771	0.4817	**0.8382**
OHIT	0.7412	0.7739	0.8644	0.7104	0.7479	0.8522	0.6050	0.6624	0.8354	0.4112	0.5100	0.7783	0.2506	0.3782	0.7171
T-SMOTE	0.4205	0.5077	0.8429	0.1781	0.2828	0.8134	0.0646	0.1708	0.7959	0.0159	0.0894	0.6977	0.0143	0.0805	0.6380
BFGAN	0.3821	0.4812	0.8688	0.3616	0.4652	0.8740	0.1292	0.2613	0.7998	0.0675	0.1845	0.7345	0.0674	0.1826	0.7354
CSMOTE	0.1194	0.2522	0.8060	0.4699	0.5552	0.8692	0.3026	0.4238	0.8224	0.1061	0.2366	0.7785	0.0809	0.2022	0.8087
TCGAN	0.6237	0.6750	0.8838	0.4699	0.5552	0.8827	0.6735	0.7156	0.8638	0.4125	0.5107	**0.8676**	0.1113	0.2405	0.6947
I-GAN	0.6837	0.7232	0.8693	0.5349	0.6048	0.8972	0.4074	0.5076	**0.8944**	0.0000	0.0000	0.4005	0.0000	0.0000	0.0000
H-SMOTE	0.6977	0.7388	0.8631	0.7168	0.7549	0.8578	0.5550	0.6251	0.8087	0.0469	0.1549	0.6715	0.0739	0.1958	0.7867
CFAMG	**0.7464**	0.7755	**0.8954**	**0.7212**	**0.7557**	**0.8893**	**0.7065**	**0.7422**	0.8807	**0.5926**	**0.6530**	0.8222	**0.5027**	**0.5843**	0.7755



(a) F1 (b) G-Means (c) AUC

Figure 2: Parameter sensitivity analysis on the MP dataset using MLP classifier.

Result analysis. For non-temporal oversampling methods, CFAMG shows an average performance improvement of 24.0%, 18.6%, and 6.6% over the best non-temporal sampling method in the MLP classifier, 46.2%, 66.6% and 19.6% in the LSTM classifier, and 18.86%, 18.31%, and 7.31% in the ResNet classifier. For temporal oversampling methods, CFAMG outperforms the best method in this category by 26.0%, 22.2%, and 9.5% in the MLP classifier, 52.3%, 72.5%, and 24.0% in the LSTM classifier, and 37.84%, 34.24%, and 9.74% in the ResNet classifier. The experimental results show that CFAMG outperforms the latest baseline methods across all three classifiers, with particularly significant improvements in F1 score and G-Means. This is attributed to CFAMG's use of causal factors to generate distinguishable minority class samples. In addition, its consistently strong performance across multiple baseline classifiers suggests robustness to the choice of classifier.

5.2.2 Experimental Results on Different Imbalance Ratios. We conducted performance comparisons on the MiddlePhalanxOutlineCorrect (MPOC) dataset using different IRs, where the IRs were generated by selecting 70, 50, 30, 10, and 5 minority class samples, respectively. As shown in Table 2, the proposed method demonstrates a more evident advantage in terms of F1 and G-Means as the IR increases, although its AUC is slightly lower than that of the baseline methods at higher IRs. The improvements in F1 and G-Means are primarily due to CFAMG's ability to leverage majority class information to generate distinguishable minority class samples. While the proximity of these samples to the decision boundary—resulting from majority class influence—may lead to a slight decrease in AUC, their high separability is still evident, validating the overall effectiveness of the generated data.

5.2.3 Training Runtime, Convergence, and Parameter Sensitivity. **Training runtime.** Table 4 compares the training time of CFAMG and baselines on 9 high-IR datasets (same as in the ablation study). The "SMOTE & variants" reports the average runtime of SMOTE, BL-SMOTE, ADASYN, and MWMOTE. CFAMG shows markedly better efficiency than generation-based oversampling methods like BFGAN, I-GAN, and TCGAN.
Convergence. Figure 3 illustrates the convergence of the total loss and disentangled classifier loss for the majority and minority

Table 3: Ablation studies on different generating factors and the two disentangling components. The best results are bolded.

	Metrics	RS	NATOPS	FW	GMAD3	MI	MP	PAP	PPTW	Worms	Average
Random	F1 ↑	0.9254	0.6265	0.2326	0.3333	0.7027	0.6437	0.0800	0.0000	0.4444	0.4432
	G-Means ↑	0.9467	0.8430	0.3925	0.4472	0.8660	0.9433	0.2212	0.0000	0.6757	0.5928
	AUC ↑	0.9980	0.9142	0.6740	0.7675	0.9790	0.9638	0.8593	0.9458	0.7627	0.8738
Only z_c^P	F1 ↑	0.9552	0.6761	0.2727	0.1818	0.8065	0.6942	0.2941	0.0000	0.6667	0.5053
	G-Means ↑	0.9660	0.8422	0.4299	0.3162	0.8557	0.9494	0.4879	0.0000	0.7790	0.6251
	AUC ↑	0.9988	0.9298	0.7384	0.8358	0.9879	0.9832	0.8577	0.9606	**0.8714**	0.9071
$z_c^P + Random$	F1 ↑	0.9697	0.6552	0.2727	0.4444	0.7692	0.6307	0.2778	0.0000	**0.7143**	0.5260
	G-Means ↑	0.9701	0.8556	0.4299	0.6831	0.9133	0.9374	0.4852	0.0000	0.7848	0.6733
	AUC ↑	0.9985	0.9473	0.7306	0.8258	0.9859	0.9843	0.8529	0.9581	0.8496	0.9037
$z_c^P + z_{rp}^P$	F1 ↑	0.9851	0.6866	0.2632	0.2667	**0.8475**	0.6192	0.2286	0.0000	0.6667	0.5071
	G-Means ↑	0.9852	0.8337	0.3948	0.4416	0.9146	0.9377	0.4339	0.0000	0.7790	0.6356
	AUC ↑	0.9995	0.9411	0.7456	0.8200	0.9872	0.9826	0.8511	0.9631	0.8696	0.9066
w/MI	F1 ↑	0.9706	0.6667	0.2727	0.4444	0.8060	0.6792	0.4000	0.3333	0.5882	0.5735
	G-Means ↑	0.9810	0.8214	0.4299	0.7141	0.9172	0.9442	0.6119	0.7019	0.7673	0.7654
	AUC ↑	0.9998	0.9356	0.7697	0.8408	0.9887	0.9819	0.8598	0.9581	0.8370	0.9079
w/DC	F1 ↑	0.9565	**0.7105**	0.2917	0.5000	0.7937	0.6942	0.3636	0.3333	0.5000	0.5715
	G-Means ↑	0.9768	0.8866	0.4627	0.6922	0.9146	0.9494	0.6049	0.7019	0.7494	0.7709
	AUC ↑	0.9990	0.9382	0.7625	0.8125	0.9889	0.9837	0.8705	**0.9729**	0.8315	0.9066
CFAMG	F1 ↑	**0.9855**	**0.7105**	**0.2979**	**0.5556**	0.8406	**0.7706**	**0.3784**	**0.4000**	0.6667	**0.6229**
	G-Means ↑	**0.9958**	**0.8866**	**0.4633**	**0.7517**	**0.9197**	**0.9632**	**0.5757**	**0.7036**	**0.8076**	**0.7852**
	AUC ↑	**1.0000**	**0.9518**	**0.7848**	**0.8792**	**0.9910**	**0.9920**	**0.8734**	0.9680	0.8678	**0.9231**

Table 4: Average time consumed in the 9 high-IR univariate and multivariate datasets.

	SMOTE & Variants	MBS	INOS	OHIT	T-SMOTE	BFGAN	CSMOTE	TCGAN	I-GAN	H-SMOTE	CFAMG
Time (s)	0.12	94.95	311.11	1.28	1.63	78.43	11.86	30.22	1051.69	0.03	16.94

classes on the FiftyWords, demonstrating that CFAMG consistently achieves convergence during both training phases, which validates the stability and effectiveness of the proposed learning process. Furthermore, as training progresses, the MSE of the causal and non-causal factors for both majority and minority classes increases, indicating that the two types of factors are being disentangled.

Parameter sensitivity. Figure 2 illustrates the sensitivity of the parameters α and β evaluated over the set $\{0.01, 0.1, 1, 10, 100\}$ on the MelbournePedestrian (MP) dataset. Experimental results show CFAMG performs best when $\alpha = \beta = 1$, a trend consistent across various datasets; this setting is therefore selected as the default.

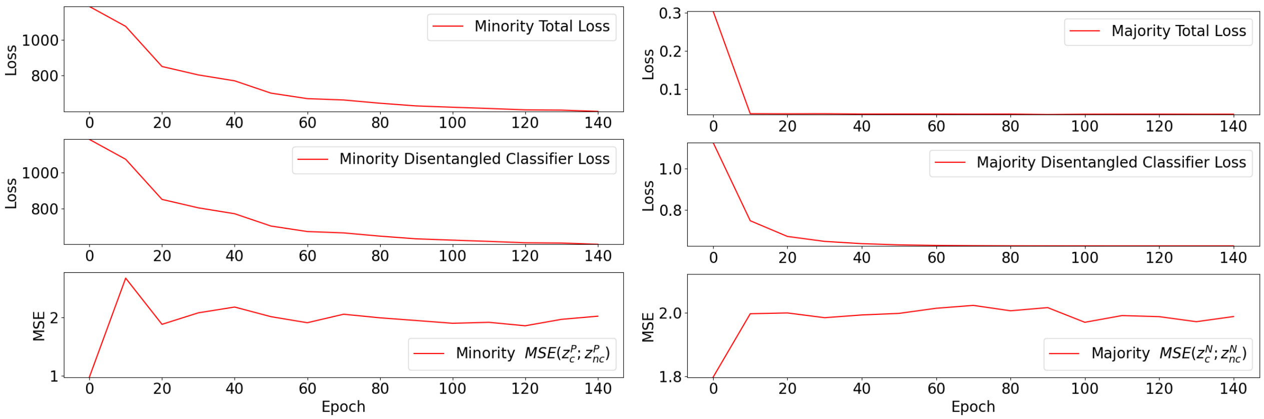
5.2.4 Results on Generation Quality. To validate the quality of the data generated by CFAMG, we compare our method with CSMOTE and other generative baselines (VAE-based and GAN-based). Table 5 presents the results on the *FiftyWords (FW)*, *GestureMidAirD3 (GMAD3)*, and *MelbournePedestrian (MP)* datasets. Our method achieves the second-best MMD score and the best FID score (while I-GAN attains the best MMD, it suffers from a substantially higher FID). These results demonstrate that CFAMG generates minority class samples that closely approximate the real data distribution, while maintaining high diversity among generated samples.

5.3 Ablation Studies

In the ablation study, we used 9 high-IR univariate and multivariate datasets, including *FiftyWords (FW)*, *GestureMidAirD3 (GMAD3)*, *MedicalImage (MI)*, *MelbournePedestrian (MP)*, *PigAirwayPressure (PAP)*, *ProximalPhalanxTW (PPTW)*, *RacketSports (RS)*, *NATOPS* and *Worms*, with the lowest IR of 3.44, the highest IR of 36.5, and an average IR of 15.43.

5.3.1 Ablation Study for Different Generating Factors. To investigate the contribution of minority class causal factors to classification performance, we design several generation configurations as follows: a) *Random* represents the use of random factors, which causes the model to degrade to a standard VAE model; b) only causal factors from the minority class (z_c^P) are used, with no additional auxiliary information; c) z_c^P is combined with random information, where the random information serves as non-causal data and noise; d) z_c^P is combined with the randomly permuted non-causal factors of the minority class (z_{rp}^P), using deterministic, underrepresented non-causal information and noise.

Result analysis. Based on the results in Table 3, the following observations can be made: 1) In case b), compared to case a), performance improved by 11.85%, 4.68%, and 3.37% across three metrics, indicating that using causal factors to generate samples is more



(a) Minority

(b) Majority

Figure 3: Convergence of total loss, disentangled classifier loss, and MSE of causal and non-causal factors (z_c, z_{nc}) in the latent space for majority and minority classes on the FiftyWords dataset.

Table 5: Experimental results on the assessment of generation quality with MMD and FID.

Datasets	**FW**		**GMAD3**		**MP**		**Average**	
Methods	MMD ↓	FID ↓	MMD ↓	FID ↓	MMD ↓	FID ↓	MMD ↓	FID ↓
CSMOTE	0.256	40.399	1.695	184.523	0.027	8.768	0.659	77.897
BFGAN	1.565	560.608	0.403	662.033	1.055	151.850	1.008	458.164
TCGAN	0.145	3.341	0.330	35.913	0.630	58.510	0.368	32.588
I-GAN	0.043	332.632	0.067	171.727	< 0.001	0.260	**0.055**	168.206
VAE	0.638	32.536	0.443	66.950	0.017	1.059	0.366	33.515
CFAMG	0.510	15.426	0.111	14.622	< 0.001	0.083	0.311	**10.044**

beneficial for classification; 2) When auxiliary information is used, the model performance improved on average by 2% in F1 score and 4% in G-Means, demonstrating that additional auxiliary information can enhance classification performance; 3) When using random auxiliary information, performance improved by 3.21% in F1 score and 5.14% in G-Means compared to using z_{rp}^P, suggesting that under imbalanced data, the random mechanism provides richer non-causal information than z_{rp}^P; 4) When majority class samples were used as counterfactual augmented minority class samples, performance improved by 15.97%, 14.51%, and 1.97% across the three metrics compared to the random mechanism. The above results indicate that causal factors can improve the classifier's performance, and that counterfactual minority samples generated using majority class samples are more effective for producing high-quality samples that enhance classification performance.

5.3.2 Ablaiton Study for Two Disentangling Components. To verify the impact of disentangled representations on classification performance, we removed the disentangling classifier (DC) and mutual information regularizer to observe the changes in classification performance. When no disentangling component is used, the model degrades to a standard VAE model. Therefore, when either disentangling component is used, classification performance improved by 24.67%, 25.4%, and 3.44% across the three metrics. When both disentangling components were used, classification performance improved by 7.715%, 1.965%, and 1.615% across the three metrics compared to using each disentangling component individually, validating the effectiveness of both disentangling components in separating causal factors.

6 Conclusion

In this paper, we propose a novel counterfactual augmented framework for imbalance time series classification, termed CFAMG, which focuses on leveraging the differences between majority and minority class samples from a causality perspective to balance the training dataset. Extensive experiments validate the effectiveness of utilizing causal factors and majority class information for ITSC tasks. The limitation of CFAMG remains an indirect causal modeling approach for time series data, which may limit its ability to capture fine-grained causal relationships. Future work will focus on identifying and modeling time-varying causal factors.

Acknowledgments

This work was supported by grants from the National Key R&D Program of China (2022YFB3303302), National Natural Science Foundation of China (62477004, 62377040, 62207007, 623B2002) and Natural Science Foundation of Chongqing (CSTB2022NSCQ-MSX1672).

References

[1] Mohanad Abukmeil, Stefano Ferrari, Angelo Genovese, Vincenzo Piuri, and Fabio Scotti. 2021. A survey of unsupervised generative models for exploratory data analysis and representation learning. *Comput. Surveys* 54 (2021), 1–40.

[2] Sukarna Barua, Md Monirul Islam, Xin Yao, and Kazuyuki Murase. 2012. MWMOTE–majority weighted minority oversampling technique for imbalanced data set learning. *IEEE Transactions on Knowledge and Data Engineering* 26 (2012), 405–425.

[3] Mohamed Bekkar, Hassiba Kheliouane Djemaa, and Taklit Akrouf Alitouche. 2013. Evaluation measures for models assessment over imbalanced data sets. *Journal of Information Engineering and Applications* 3 (2013), 1–13.

[4] Fadi Boutros, Vitomir Struc, Julian Fierrez, and Naser Damer. 2023. Synthetic data for face recognition: Current state and future prospects. *Image and Vision Computing* 135 (2023), 104688.

[5] Hong Cao, Xiao-Li Li, David Yew-Kwong Woon, and See-Kiong Ng. 2013. Integrated oversampling for imbalanced time series classification. *IEEE Transactions on Knowledge and Data Engineering* 25 (2013), 2809–2822.

[6] 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.

[7] Zhelun Chen, Zheng O'Neill, Jin Wen, Ojas Pradhan, Tao Yang, Xing Lu, Guanjing Lin, Shohei Miyata, Seungjae Lee, Chou Shen, et al. 2023. A review of data-driven fault detection and diagnostics for building HVAC systems. *Applied Energy* 339 (2023), 121030.

[8] Pengyu Cheng, Weituo Hao, Shuyang Dai, Jiachang Liu, Zhe Gan, and Lawrence Carin. 2020. Club: A contrastive log-ratio upper bound of mutual information. In *International Conference on Machine Learning*. PMLR.

[9] Asma Cherif, Arwa Badhib, Heyfa Ammar, Suhair Alshehri, Manal Kalkatawi, and Abdessamad Imine. 2023. Credit card fraud detection in the era of disruptive technologies: A systematic review. *Journal of King Saud University-Computer and Information Sciences* 35 (2023), 145–174.

[10] Matthew Dixon, Diego Klabjan, and Lan Wei. 2017. OSTSC: Over sampling for time series classification in R. *arXiv preprint arXiv:1711.09545* (2017).

[11] Carl Doersch. 2016. Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908* (2016).

[12] Tom Fawcett. 2004. ROC graphs: Notes and practical considerations for researchers. *Machine Learning* 31 (2004), 1–38.

[13] Zee Fryer, Vera Axelrod, Ben Packer, Alex Beutel, Jilin Chen, and Kellie Webster. 2022. Flexible text generation for counterfactual fairness probing. *arXiv preprint arXiv:2206.13757* (2022).

[14] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. 2012. A kernel two-sample test. *The Journal of Machine Learning Research* 13 (2012), 723–773.

[15] Riccardo Guidotti, Anna Monreale, Francesco Spinnato, Dino Pedreschi, and Fosca Giannotti. 2020. Explaining any time series classifier. In *International Conference on Cognitive Machine Intelligence*. IEEE.

[16] Guo Haixiang, Li Yijing, Jennifer Shang, Gu Mingyun, Huang Yuanyue, and Gong Bing. 2017. Learning from class-imbalanced data: Review of methods and applications. *Expert Systems With Applications* 73 (2017), 220–239.

[17] Hui Han, Wen-Yuan Wang, and Bing-Huan Mao. 2005. Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. In *International Conference on Intelligent Computing*. Springer.

[18] Jianing Hao, Qing Shi, Yilin Ye, and Wei Zeng. 2023. TimeTuner: Diagnosing Time Representations for Time-Series Forecasting with Counterfactual Explanations. *IEEE Transactions on Visualization and Computer Graphics* 30 (2023), 1183–1193.

[19] Haibo He, Yang Bai, Edwardo A Garcia, and Shutao Li. 2008. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*. IEEE.

[20] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*.

[21] Fanling Huang and Yangdong Deng. 2023. TCGAN: Convolutional Generative Adversarial Network for time series classification and clustering. *Neural Networks* 165 (2023), 868–883.

[22] Shanshan Huang, Haoxuan Li, Qingsong Li, Chunyuan Zheng, and Li Liu. 2023. Pareto invariant representation learning for multimedia recommendation. In *Proceedings of the 31st ACM International Conference on Multimedia*.

[23] Shanshan Huang, Haoxuan Li, Chunyuan Zheng, Mingyuan Ge, Wei Gao, Lei Wang, and Li Liu. 2025. Text-Driven Fashion Image Editing with Compositional Concept Learning and Counterfactual Abduction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

[24] Shanshan Huang, Haoxuan Li, Chunyuan Zheng, Lei Wang, Guorui Liao, Zhili Gong, Huayi Yang, and Li Liu. 2025. Visual Representation Learning through Causal Intervention for Controllable Image Editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

[25] Shanshan Huang, Lei Wang, Jun Liao, and Li Liu. 2024. Multi-attentional causal intervention networks for medical image diagnosis. *Knowledge-Based Systems* 299 (2024), 111993.

[26] Shanshan Huang, Yuanhao Wang, Zhili Gong, Jun Liao, Shu Wang, and Li Liu. 2024. Controllable image generation based on causal representation learning. *Frontiers of Information Technology & Electronic Engineering* 25 (2024), 135–148.

[27] Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).

[28] Daesoo Lee, Sara Malacarne, and Erlend Aune. 2023. Vector quantized time series generation with a bidirectional prior model. *arXiv preprint arXiv:2303.04743* (2023).

[29] Han Kyu Lee, Jiyoon Lee, and Seoung Bum Kim. 2022. Boundary-focused generative adversarial networks for imbalanced and multimodal time series. *IEEE Transactions on Knowledge and Data Engineering* 34, 9 (2022), 4102–4118.

[30] Haoxuan Li, Yanghao Xiao, Chunyuan Zheng, and Peng Wu. 2023. Balancing unobserved confounding with a few unbiased ratings in debiased recommendations. In *International World Wide Web Conference*.

[31] Meng Li and Haochen Sui. 2025. Causal Recommendation via Machine Unlearning with a Few Unbiased Data. In *AAAI Workshop on Artificial Intelligence with Causal Techniques*.

[32] Lawrence Chuin Ming Liaw, Shing Chiang Tan, Pey Yun Goh, and Chee Peng Lim. 2025. A histogram SMOTE-based sampling algorithm with incremental learning for imbalanced data classification. *Information Sciences* 686 (2025), 121193.

[33] Chien-Liang Liu and Po-Yen Hsieh. 2019. Model-based synthetic sampling for imbalanced data. *IEEE Transactions on Knowledge and Data Engineering* 32 (2019), 1543–1556.

[34] Pin Liu, Xiaohui Guo, Rui Wang, Pengpeng Chen, Tianyu Wo, and Xudong Liu. 2021. CSMOTE: Contrastive synthetic minority oversampling for imbalanced time series classification. In *Neural Information Processing: 28th International Conference, ICONIP*. Springer.

[35] James Mahoney and Rodrigo Barrenechea. 2019. The logic of counterfactual analysis in case-study explanation. *The British journal of sociology* 70 (2019), 306–338.

[36] Tingting Pan, Witold Pedrycz, Jie Yang, and Jian Wang. 2024. An improved generative adversarial network to oversample imbalanced datasets. *Engineering Applications of Artificial Intelligence* 132 (2024), 107934.

[37] Gede Angga Pradipta, Retantyo Wardoyo, Aina Musdholifah, I Nyoman Hariyasa Sanjaya, and Muhammad Ismail. 2021. SMOTE for handling imbalanced data problem: A review. In *International Conference on Informatics and Computing*. IEEE, 1–8.

[38] Salim Rezvani and Xizhao Wang. 2023. A broad review on class imbalance learning techniques. *Applied Soft Computing* 143 (2023), 110415.

[39] Marcel Robeer, Floris Bex, and Ad Feelders. 2021. Generating realistic natural language counterfactuals. In *Findings of the Association for Computational Linguistics: EMNLP 2021*. 3611–3625.

[40] Rick Sauber-Cole and Taghi M Khoshgoftaar. 2022. The use of generative adversarial networks to alleviate class imbalance in tabular data: a survey. *Journal of Big Data* 9 (2022), 98.

[41] Farhad Mortezapour Shiri, Thinagaran Perumal, Norwati Mustapha, and Raihani Mohamed. 2023. A comprehensive overview and comparative analysis on deep learning models: CNN, RNN, LSTM, GRU. *arXiv preprint arXiv:2305.17473* (2023).

[42] Xiangyu Sun, Raquel Aoki, and Kevin H Wilson. 2024. Counterfactual Explanations for Multivariate Time-Series without Training Datasets. *arXiv preprint arXiv:2405.18563* (2024).

[43] Fan Wang, Chaochao Chen, Weiming Liu, Tianhao Fan, Xinting Liao, Yanchao Tan, Lianyong Qi, and Xiaolin Zheng. 2024. CE-RCFR: Robust Counterfactual Regression for Consensus-Enabled Treatment Effect Estimation. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

[44] Fan Wang, Lianyong Qi, Weiming Liu, Bowen Yu, Jintao Chen, and Yanwei Xu. 2025. Inter- and Intra- Similarity Preserved Counterfactual Incentive Effect Estimation for Recommendation Systems. *ACM Transactions on Information Systems* (2025).

[45] Hao Wang. 2024. Improving Neural Network Generalization on Data-Limited Regression with Doubly-Robust Boosting. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

[46] Hao Wang, Zhichao Chen, Zhaoran Liu, Xu Chen, Haoxuan Li, and Zhouchen Lin. 2025. Proximity Matters: Local Proximity Enhanced Balancing for Treatment Effect Estimation. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

[47] Hao Wang, Jiajun Fan, Zhichao Chen, Haoxuan Li, Weiming Liu, Tianqiao Liu, Quanyu Dai, Yichao Wang, Zhenhua Dong, and Ruiming Tang. 2023. Optimal Transport for Treatment Effect Estimation. In *Advances in Neural Information Processing Systems*.

[48] Haotian Wang, Haoxuan Li, Hao Zou, Haoang Chi, Long Lan, Wanrong Huang, and Wenjing Yang. 2025. Effective and Efficient Time-Varying Counterfactual Prediction with State-Space Models. In *International Conference on Learning Representations*.

[49] Jun Wang, Haoxuan Li, Chi Zhang, Dongxu Liang, Enyun Yu, Wenwu Ou, and Wenjia Wang. 2023. CounterCLR: Counterfactual contrastive learning with non-random missing data in recommendation. In *IEEE International Conference on Data Mining.*

[50] Lei Wang, Shanshan Huang, Shu Wang, Jun Liao, Tingpeng Li, and Li Liu. 2024. A survey of causal discovery based on functional causal model. *Engineering Applications of Artificial Intelligence* 133 (2024), 108258.

[51] Xin Wang, Hong Chen, Si'ao Tang, Zihao Wu, and Wenwu Zhu. 2024. Disentangled representation learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2024).

[52] Zhiyuan Wang, Xovee Xu, Weifeng Zhang, Goce Trajcevski, Ting Zhong, and Fan Zhou. 2022. Learning latent seasonal-trend representations for time series forecasting. In *Advances in Neural Information Processing Systems.*

[53] Tarid Wongvorachan, Surina He, and Okan Bulut. 2023. A comparison of undersampling, oversampling, and SMOTE methods for dealing with imbalanced classification in educational data mining. *Information* 14 (2023), 54.

[54] Yuntian Wu, Yuntian Yang, Jiabao Sean Xiao, Chuan Zhou, Haochen Sui, and Haoxuan Li. 2024. Invariant Spatiotemporal Representation Learning for Cross-patient Seizure Classification. In *NeurIPS Workshop on NeuroAI.*

[55] Yanghao Xiao, Haoxuan Li, Yongqiang Tang, and Wensheng Zhang. 2024. Addressing Hidden Confounding with Heterogeneous Observational Datasets for Recommendation. In *NeurIPS.*

[56] Reda Yacouby and Dustin Axman. 2020. Probabilistic extension of precision, recall, and f1 score for more thorough evaluation of classification models. In *Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems.*

[57] Jingquan Yan and Hao Wang. 2023. Self-interpretable time series prediction with counterfactual explanations. In *International Conference on Machine Learning.* PMLR.

[58] Huayi Yang, Chunyuan Zheng, Guorui Liao, Shanshan Huang, Jun Liao, Zhili Gong, Haoxuan Li, and Li Liu. 2025. CAP: Causal Air Quality Index Prediction Under Interference with Unmeasured Confounding. In *International World Wide Web Conference.*

[59] Wenjing Yang, Haotian Wang, Haoxuan Li, Hao Zou, Ruochun Jin, Kun Kuang, and Peng Cui. 2024. Your Neighbor Matters: Towards Fair Decisions Under Networked Interference. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining.*

[60] Shuqiang Zhang, Yuchao Zhang, Jinkun Chen, and Haochen Sui. 2025. Addressing Correlated Latent Exogenous Variables in Debiased Recommender Systems. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining.*

[61] Pu Zhao, Chuan Luo, Bo Qiao, Lu Wang, Saravan Rajmohan, Qingwei Lin, and Dongmei Zhang. 2022. T-SMOTE: Temporal-oriented Synthetic Minority Oversampling Technique for Imbalanced Time Series Classification. In *IJCAI.* 2406–2412.

[62] Chunyuan Zheng, Hang Pan, Yang Zhang, and Haoxuan Li. 2025. Adaptive Structure Learning with Partial Parameter Sharing for Post-Click Conversion Rate Prediction. In *International ACM SIGIR Conference on Research and Development in Information Retrieval.*

[63] Chuan Zhou, Yaxuan Li, Chunyuan Zheng, Haiteng Zhang, Min Zhang, Haoxuan Li, and Mingming Gong. 2025. A Two-Stage Pretraining-Finetuning Framework for Treatment Effect Estimation with Unmeasured Confounding. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining.*

[64] Tuanfei Zhu, Cheng Luo, Zhihong Zhang, Jing Li, Siqi Ren, and Yifu Zeng. 2022. Minority oversampling for imbalanced time series classification. *Knowledge-Based Systems* 247 (2022), 108764.

Appendix

A Experiment Supplementary

For transparency and reproducibility, we provide the implementation links for all the baseline methods used in our experiments. The code repositories are publicly available and listed as follows:

- **SMOTE [6], BL-SMOTE [17], ADASYN [19], MWMOTE [2]:** https://github.com/analyticalmindsltd/smote_variants.
- **MBS [33]:** https://github.com/b10071007/Model-Based-Synthetic-Sampling.
- **INOS [5]:** The method can be implemented using the OSTSC package [10].
- **OHIT [64]:** https://github.com/zhutuanfei/OHIT.
- **T-SMOTE [61]:** The results presented are based on our reproduction of the original findings of T-SMOTE.
- **BFGAN [29]:** https://github.com/koogi303/Oversampling.
- **CSMOTE [34]:** https://github.com/liupin-source/csmote.
- **TCGAN [21]:** https://bitbucket.org/Lynn1/tcgan.
- **I-GAN [36]:** https://github.com/flowerbloom000/I-GAN.
- **H-SMOTE [32]:** https://github.com/Lawrence98/H-SMOTE-FAM.git.
- **CFAMG (Ours):** https://github.com/WangLei-CQU/CFAMG.

Algorithm 1 CFAMG Algorithm

```
Input: Minority data $\{X^P, y^P\}$, Majority data $\{X^N, y^N\}$, hyperparameters and initialized CFAMG parameters.
Output: Counterfactual minority samples $\hat{X}^P_{cf}$.
\* Training *\
while until convergence do
    for batch $\{X^P_i, y^P_i\}$, $\{X^N_i, y^N_i\}$ do
        \* Minority training *\
        Encoder $\{X^P_i, y^P_i\}$ to $z^P_c, z^P_{nc}$
        Reconstruct $X^P_i$, calculate loss according to Eq. (13) and (8), update the parameters of $E^P, D^P$.
        \* Majority training *\
        Encoder $\{X^N_i, y^N_i\}$ to $z^N_c, z^N_{nc}$
        Reconstruct $X^N_i$, calculate loss according to Eq. (14) and (10), update the parameters of $E^N, D^N$.
        \* Disentangled classifier training *\
        Shuffle $z^P_{nc}, z^N_{nc}$ to get $z^P_{swap}, z^P_{cf}$
        Calculate classifier loss according to Eq. (2) and (3), update the parameters of $D^P_{cls}, D^N_{cls}$.
    end for
end while
\* Generating *\
for batch $\{X^P_i, y^P_i\}$ in minority set do
    Encoder $\{X^P_i, y^P_i\}$ to $z^P_c, z^P_{nc}$
    for batch $\{X^N_i, y^N_i\}$ in majority set do
        Encoder $\{X^N_i, y^N_i\}$ to $z^N_c, z^N_{nc}$
        Generate $\hat{X}^P_{cf}$ from $Concatenate(z^P_c, z^N_{nc})$
    end for
end for
```

B Implementation Details of CFAMG

The proposed model is an extension of the basic VAE model. During training, a separate VAE model is used for the minority and negative class samples. The pseudo-codes of CFAMG covering training and generating phases are summarized in **Algorithm 1**. The latent representation dimension is set to 64, with 200 iterations, a learning rate of $1e-3$, and weight decay of $1e-4$. The proposed method has model-independent properties, meaning that the VAE encoder and decoder can be replaced with any network. In this work, the encoder employs a three-layer MLP with layer dimensions of [32, 64, 128] and corresponding dropout rates of [0.1, 0.1, 0.2]. The decoder is implemented as a single-layer MLP with a hidden dimension of 128 and a dropout rate of 0.1. All the experiments were implemented using Python software and executed on an i7-12700 CPU with 24GB of RAM and NVIDIA GeForce 4060.

Table 6: Univariate and multivariate datasets characteristics for training and test set. IR is the imbalance ratio.

Type	Dataset	Dimensions	Minority Class	#Length	Training			Test		
					#Maj	#Min	IR	#Maj	#Min	IR
Univariate	ACSF1	1	1	1460	90	10	9.00	90	10	9.00
Multivariate	ArticularyWordRecognition	9	25.0	144	264	11	24.00	288	12	24.00
Univariate	Car	1	3	577	49	11	4.45	41	19	2.16
Univariate	Computers	1	2	720	125	41	3.05	125	125	1.00
Univariate	DodgerLoopDay	1	5, 2	288	51	16	3.19	56	21	2.67
Univariate	Earthquakes	1	1	512	264	58	4.55	104	35	2.97
Univariate	ECG5000	1	1	140	208	97	2.14	1873	2627	0.71
Univariate	EthanolLevel	1	4	1751	378	126	3.00	376	124	3.03
Univariate	FaceAll	1	14	131	520	40	13.00	1658	32	51.81
Univariate	FacesUCR	1	11, 14	131	190	10	19.00	1940	110	17.64
Univariate	FiftyWords	1	41, 50, 49, 42, 25, 34	270	438	12	36.50	423	32	13.22
Univariate	Fish	1	4	463	154	21	7.33	146	29	5.03
Univariate	FreezerRegularTrain	1	2	301	75	25	3.00	1425	1425	1.00
Univariate	GestureMidAirD2	1	23, 26	360	192	16	12.00	120	10	12.00
Univariate	GestureMidAirD3	1	3, 13	360	192	16	12.00	120	10	12.00
Univariate	GesturePebbleZ1	1	2	455	112	20	5.60	147	25	5.88
Univariate	GesturePebbleZ2	1	2	455	123	23	5.35	136	22	6.18
Univariate	GunPoint	1	1	150	26	10	2.60	74	76	0.97
Univariate	GunPointAgeSpan	1	2	150	68	22	3.09	160	156	1.03
Univariate	Ham	1	1	431	57	17	3.35	54	51	1.06
Multivariate	HandMovementDirection	10	backward	400	120	40	3.00	59	15	3.93
Univariate	Haptics	1	1	1092	137	18	7.61	248	60	4.13
Univariate	Herring	1	2	512	39	10	3.90	38	26	1.46
Univariate	InlineSkate	1	1, 7	1882	80	20	4.00	446	104	4.29
Univariate	ItalyPowerDemand	1	2	24	34	11	3.09	513	516	0.99
Univariate	Lightning2	1	-1	637	40	20	2.00	33	28	1.18
Univariate	Mallat	1	5, 1, 3	1024	41	14	2.93	1459	886	1.65
Univariate	MedicalImages	1	8, 6	99	368	13	28.31	726	34	21.35
Univariate	MelbournePedestrian	1	9	24	1040	98	10.61	2129	190	11.21
Univariate	MiddlePhalanxOutlineCorrect	1	0	80	388	70	5.54	166	125	1.33
Univariate	MixedShapesRegularTrain	1	5	1024	400	100	4.00	2111	314	6.72
Univariate	MixedShapesSmallTrain	1	5	1024	80	20	4.00	2111	314	6.72
Multivariate	NATOPS	24	2.0	51	150	30	5.00	150	30	5.00
Univariate	PickupGestureWiimoteZ	1	10, 9	361	40	10	4.00	40	10	4.00
Univariate	PigAirwayPressure	1	52, 51, 50, 49, 48	2000	94	10	9.40	188	20	9.40
Univariate	PigArtPressure	1	52, 51, 50, 49, 48	2000	94	10	9.40	188	20	9.40
Univariate	ProximalPhalanxOutlineAgeGroup	1	1	80	328	72	4.56	188	17	11.06
Univariate	ProximalPhalanxTW	1	3	80	384	16	24.00	203	2	101.50
Multivariate	RacketSports	6	squash_backhandboast	30	117	34	3.44	118	34	3.47
Univariate	RefrigerationDevices	1	3	720	250	125	2.00	250	125	2.00
Univariate	ScreenType	1	3	720	250	125	2.00	250	125	2.00
Multivariate	SelfRegulationSCP2	7	positivity	1152	100	33	3.03	90	90	1.00
Univariate	SemgHandGenderCh2	1	2	1500	150	50	3.00	390	210	1.86
Univariate	SemgHandMovementCh2	1	6	1500	375	75	5.00	375	75	5.00
Univariate	ShakeGestureWiimoteZ	1	10, 9	385	40	10	4.00	40	10	4.00
Univariate	SmallKitchenAppliances	1	3	720	250	125	2.00	250	125	2.00
Univariate	SmoothSubspace	1	3	15	100	50	2.00	100	50	2.00
Univariate	Strawberry	1	1	235	394	73	5.40	238	132	1.80
Univariate	Trace	1	2	275	79	21	3.76	71	29	2.45
Univariate	TwoPatterns	1	2	128	763	237	3.22	2989	1011	2.96
Univariate	Wine	1	2	234	30	10	3.00	27	27	1.00
Univariate	Worms	1	5	900	164	17	9.65	69	8	8.62
Univariate	WormsTwoClass	1	1	900	105	25	4.20	44	33	1.33