
K-IPO: KENDALL-CONSTRAINED IMPORTANCE PRESERVING OVERSAMPLING FOR IMBALANCED TABULAR DATA

Marios Tyrovolas^{1,2,3,*} Argiris Sofotasios^{3,4} Dimitris Metaxakis^{3,4} Georgios Mermigkis^{2,3,4}
 George Georgoulas^{1,5} Panagiotis Hadjidoukas^{2,4} Chrysostomos Stylios^{1,2}

¹Department of Informatics and Telecommunications, University of Ioannina, Arta, Greece

²Industrial Systems Institute, Athena Research Center, Patras, Greece

³Archimedes Unit, Athena Research Center, Athens, Greece

⁴Department of Computer Engineering and Informatics, University of Patras, Patras, Greece

⁵Department of Mechanical Engineering and Aeronautics, University of Patras, Patras, Greece

*Corresponding author: tyrovolas@kic.uoi.gr

ABSTRACT

Oversampling is widely used to address class imbalance in tabular classification, but existing methods can distort the feature importance ranking underlying model explanations. Although recent studies have quantified this distortion by comparing real and synthetic data, none have actively sought to prevent it. In this paper, we introduce Kendall-constrained Importance-Preserving Oversampling (K-IPO), a generator-agnostic, "generate-then-select" framework that preserves the original data's feature importance ranking during augmentation. K-IPO iteratively generates minority-class candidates and accepts them only if their inclusion maintains a user-defined minimum Kendall's tau (τ) correlation with the reference ranking. Optionally, stricter constraints can be applied to the highest-ranked features. We evaluated K-IPO on 20 imbalanced binary classification datasets using three classifiers and multiple explanation methods. In most cases, K-IPO achieved the best or tied-best results in feature importance preservation, explanation consistency, and class separability. It also generally improved predictive performance while maintaining competitive computational overhead.

1 Introduction

Class imbalance, in which observations are unevenly distributed across class labels, remains a significant challenge in supervised classification [1]. Models trained on imbalanced data tend to favor majority-class patterns, resulting in skewed decision boundaries, reduced sensitivity to the minority class, and misleading performance estimates [2]. This issue is particularly critical when the minority class represents rare but high-impact events such as fraud, disease, safety incidents, or equipment failure [3], thereby limiting the real-world applicability of these models.

Common approaches for addressing class imbalance include data-level rebalancing, cost-sensitive or class-weighted learning, threshold adjustment, ensemble methods, and hybrid strategies [2, 4, 5, 6, 7]. However, the latter three typically require task- or classifier-specific cost calibration or introduce additional training and tuning complexity. Therefore, data-level rebalancing remains a widely adopted strategy, with oversampling being a practical option because it is model-agnostic, easy to integrate into standard training pipelines, and does not require modifications to the downstream classifier. Accordingly, numerous oversampling methods have been proposed, ranging from random replication [8] and local interpolation [9] to probabilistic [10] and deep-generative approaches [11, 12, 13]. Although these methods can improve class balance and predictive performance, they are designed primarily to optimize sample placement, distributional fidelity, or downstream utility rather than explicitly preserving the feature-target relationships that shape model reasoning [14].

This limitation becomes particularly important when evaluation considers not only predictive performance, but also whether a classifier relies on appropriate feature-target relationships rather than merely achieving accurate predictions [15]. Synthetic samples may improve minority-class metrics while altering which predictors appear to be the most influential to the trained model. Previous studies have shown that synthetic tabular data may fail to preserve deeper

structural properties, including logical or functional dependencies and feature importance rankings [16, 17, 18]. Such discrepancies have direct implications for eXplainable Artificial Intelligence (XAI), where feature importance serves as a primary tool for model interpretation, auditing, domain validation, and decision support [19]. If oversampling modifies the ranking of influential predictors, a model may remain accurate while producing explanations that no longer reflect the original prediction task. In high-stakes domains such as healthcare, manufacturing, and finance, this explanation drift can undermine trust, reproducibility, and practical reliability.

Recent frameworks have recognized the need to evaluate synthetic data beyond predictive utility by incorporating structural fidelity and explanation-aware criteria [20, 21]. However, existing approaches remain primarily diagnostic, measuring feature importance drift after synthetic data generation rather than preventing it during augmentation. This leaves a methodological gap: oversampling should not only generate plausible and useful minority-class samples but also preserve the feature importance structure derived from the original data.

To address this gap, we propose Kendall-constrained Importance-Preserving Oversampling (K-IPO), a "generate-then-select" framework for imbalanced tabular classification. K-IPO employs a chosen oversampling method or tabular generator to iteratively produce minority-class candidates, accepting only candidate sets whose inclusion preserves the feature importance ranking of the original data. Rank preservation is enforced through Kendall’s rank correlation coefficient, τ [22], with optional constraints applied to the top- k features. Thus, K-IPO integrates explanation stability directly into the oversampling process rather than assessing it only post hoc. Using 20 imbalanced binary tabular classification datasets, we evaluate K-IPO in terms of feature-importance ranking preservation and class separability of the generated data, as well as the predictive performance and explanation consistency of classifiers trained on the augmented data. Comparisons with representative baselines show that K-IPO achieves the strongest aggregate predictive-explanatory performance while incurring acceptable additional computational overhead.

The remainder of this paper is organized as follows. Section 2 reviews related work on imbalanced learning, tabular data generation, and explanation-aware evaluation. Section 3 presents K-IPO, while Section 4 describes the experimental setup, including datasets, preprocessing, baseline oversampling methods, classifiers, explainability methods, and evaluation metrics. Section 5 presents the empirical results, and Section 6 concludes the paper by discussing the limitations of the proposed framework and outlining future research directions.

2 Literature Review

This section reviews oversampling methods for imbalanced tabular data classification from the perspective of feature importance preservation. Existing methods can be broadly categorized into *replication-based oversampling*, *local interpolation*, *distribution-learning approaches*, and *deep generative models*. Although these categories differ in how they generate minority-class observations, they generally prioritize class balance, local geometry, distributional fidelity, or downstream predictive utility over preserving the feature-target relationships underlying model explanations.

2.1 Replication-based and interpolation-based oversampling

Random oversampling is the simplest data-level approach to address class imbalance, achieved by duplicating minority class observations until the desired class ratio is reached. Although this method increases minority-class representation, it introduces no new information about the minority distribution and may lead to overfitting when few distinct minority observations are available [8, 2]. This limitation motivated the development of synthetic approaches, such as the Synthetic Minority Oversampling Technique (SMOTE) [9]. SMOTE generates minority-class samples by interpolating between an observation and its nearest minority-class neighbors, making it classifier-independent and easy to integrate into standard learning pipelines.

Despite its success, SMOTE assumes that local linear interpolation between minority class observations produces plausible samples. This assumption may fail when minority regions are sparse, multimodal, noisy, or overlapping with the majority class. In such cases, synthetic samples may connect distinct minority subclusters, populate ambiguous regions, or smooth the minority distribution in ways that distort the original learning problem [23]. In response, several SMOTE variants have been proposed to refine the generation or selection process. For instance, Borderline-SMOTE focuses on minority observations near the decision boundary [24], ADASYN allocates more synthetic samples to minority observations that are harder to learn [25], and hybrid approaches, such as SMOTE-Tomek and SMOTE-ENN, combine oversampling with cleaning mechanisms that remove ambiguous, noisy, or borderline observations [8]. SMOTE with Boosting (SMOTEWB) similarly combines noise detection with boosting-guided generation [26]. Specifically, it uses boosting-derived weights to determine where samples should be generated and adaptively selects the number of neighbors for each observation. Clustering-based methods, such as KMeans-SMOTE, restrict generation to selected minority regions to reduce noise and address within-class imbalance [27]. Nevertheless, these approaches

remain primarily guided by geometric, neighborhood-based, or heuristic criteria and do not assess whether synthetic observations preserve the global feature importance ranking derived from the original training data.

2.2 Distribution-learning methods for tabular synthesis

Distribution learning methods explicitly model the minority class distribution. Instead of interpolating between neighboring observations, they estimate the statistical properties of the data and sample new records from the learned distribution. Copula-based methods are a prominent example because they separate marginal distributions from the dependence structure, allowing heterogeneous variables to be modeled more flexibly than methods based solely on Euclidean neighborhoods [10]. This flexibility is particularly valuable in tabular data, where variables may differ in scale, type, and dependency structure. Nevertheless, distribution learning remains challenging under class imbalance because the limited number of minority observations complicates the reliable estimation of marginal distributions and dependence structures. Common copula formulations may also struggle to capture nonlinear dependencies, higher-order interactions, and minority-specific structures [28, 29]. Consequently, these methods may reproduce some statistical regularities of the minority class without preserving the interactions that shape model behavior and feature attribution.

2.3 Deep generative oversampling and tabular diffusion models

Deep generative models have gained prominence in generating synthetic tabular data due to their ability to capture nonlinear dependencies, handle mixed continuous and categorical variables, and model complex feature interactions. Generative Adversarial Networks (GANs), Variational Autoencoders (VAEs), and diffusion models have all been employed for this purpose [30].

CTGAN and TVAE are widely used deep learning baselines for tabular synthesis. CTGAN adapts the GAN framework to address two common properties of real-world tabular data: non-Gaussian or multimodal continuous variables and highly imbalanced categorical variables. It combines mode-specific normalization for continuous columns with conditional generation and training-by-sampling for discrete columns, improving the representation of both frequent and rare categories [11]. Introduced in the same study, TVAE provides a VAE-based alternative for mixed-type tabular data. Instead of using adversarial training, it learns a latent probabilistic representation of the observed rows and generates new records by decoding samples from this space [11]. TabDDPM extends denoising diffusion probabilistic models to tabular data by formulating generation as a gradual denoising process adapted to heterogeneous feature spaces containing continuous and discrete variables. Therefore, it represents an important modern baseline connecting tabular synthesis with the broader success of diffusion-based generative modeling [12].

Although these models are powerful candidates for oversampling, their objectives do not guarantee feature importance preservation. GANs optimize adversarial indistinguishability, VAEs emphasize reconstruction and latent-likelihood objectives, and diffusion models optimize denoising. Thus, even realistic synthetic samples that improve downstream performance may weaken relevant dependencies, amplify spurious associations, or alter the ranking of influential predictors. This risk is especially pronounced under class imbalance, where minority-class structure must be inferred from limited, potentially sparse, noisy, or overlapping observations. Empirical studies further indicate that the benefits of deep generative oversampling may be modest relative to its complexity and strongly dependent on dataset characteristics [31, 32, 33]. Deep generative models are therefore valuable candidate generators, but preventing feature importance drift requires an additional acceptance criterion.

2.4 Synthetic-data evaluation and feature importance preservation

Synthetic tabular data are commonly evaluated in terms of fidelity, utility, and privacy [34, 35, 36]. Fidelity metrics assess whether synthetic data reproduce properties of the original data, including marginal distributions, correlations, and multivariate similarities. Utility metrics evaluate their suitability for downstream tasks, often through predictive performance comparisons or train-on-synthetic-test-on-real protocols. Privacy metrics estimate whether synthetic records disclose information about individuals in the original data using approaches such as nearest-neighbor and inference-risk analyses. These criteria are complementary rather than interchangeable: a synthetic dataset may support accurate predictions while failing to preserve structural relationships required for interpretation.

Recent research has therefore emphasized structure- and explanation-aware evaluation. Giles *et al.* [17] showed that feature importance rankings derived from synthetic data can differ from those obtained from real data, particularly in complex real-world scenarios. Umesh *et al.* [16] similarly found that synthetic tabular generators may fail to preserve the logical and functional dependencies among features. In imbalanced classification, Stando *et al.* [37] demonstrated that balancing methods can alter model behavior in ways not captured by conventional performance metrics. They compared models before and after rebalancing using variable importance, partial dependence, and accumulated local

Table 1: Conceptual positioning of oversampling families for imbalanced tabular classification.

Family	Representative methods	Generation principle	FI-rank preservation
Replication-based	Random oversampling	Duplicates existing minority observations.	○
Local interpolation	SMOTE, SMOTE-NC; Borderline-SMOTE, ADASYN; SMOTE-Tomek, SMOTE-ENN; KMeans-SMOTE	Interpolates between neighboring minority observations.	○
Distribution-learning	Gaussian Copula	Samples from an estimated minority-class distribution.	○
Deep generative	CTGAN, TVAE	Learns a neural generator for tabular records.	○
Diffusion-based	TabDDPM	Generates records through iterative denoising.	○
Proposed method	K-IPO	Generates candidate samples and accepts them only under a Kendall-based feature importance (FI) ranking constraint.	●

effects. Explanation-aware measures, such as SHapley Additive exPlanations (SHAP)-based Distance, have also shown that attribution patterns can reveal semantic discrepancies between real and synthetic data that distributional and predictive metrics may overlook [20].

3 K-IPO Framework

This section presents Kendall-constrained Importance-Preserving Oversampling (K-IPO), a framework designed for oversampling in imbalanced tabular classification that explicitly mitigates feature importance drift during data augmentation. Unlike the conventional methods discussed in Section 2, which primarily focus on balancing the training distribution, K-IPO accepts synthetic samples only when they preserve the feature importance ranking of the original training data within a user-defined tolerance (see Table 1). Fig. 1 illustrates the overall workflow, while Algorithm 1 summarizes the oversampling procedure of the proposed framework. The following subsections describe the oversampling objective, generator-agnostic candidate generation, acceptance mechanism, adaptive block-and-chunk search, and computational requirements.

3.1 Problem Formulation

Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ be an imbalanced binary tabular dataset, where $x_i \in \mathbb{R}^d$ is a d -dimensional feature vector, and $y_i \in \{0, 1\}$ is its class label. Without loss of generality, $y = 1$ denotes the minority class, and $y = 0$ denotes the majority class. Additionally, let n_{maj} and n_{min} denote the numbers of majority and minority class samples, respectively. For a target minority-to-majority ratio $r > 0$, the objective is to construct an augmented dataset containing approximately $r \cdot n_{maj}$ minority-class observations. Therefore, the required number of synthetic samples is

$$N^* = \max(0, \lfloor r \cdot n_{maj} \rfloor - n_{min}). \quad (1)$$

K-IPO aims to achieve this target while preserving the feature-importance structure of the original training data. Let $s^{ref} = \phi(\mathcal{D})$ denote the reference importance-score vector produced by an importance estimator $\phi(\cdot)$. The ranking induced by s^{ref} is used as the reference order throughout oversampling. The framework is agnostic to the choice of ϕ , which may be a filter-based estimator such as the ANOVA F -score [38], or an XAI method such as SHAP [39] or Permutation Feature Importance (PFI) [40]. In this study, ϕ was instantiated using the ANOVA F -score, which assigns larger values to features with greater between-class variation relative to within-class variation. This choice was made because it is model-independent, computationally efficient, and suitable for repeated evaluation during iterative oversampling [41].

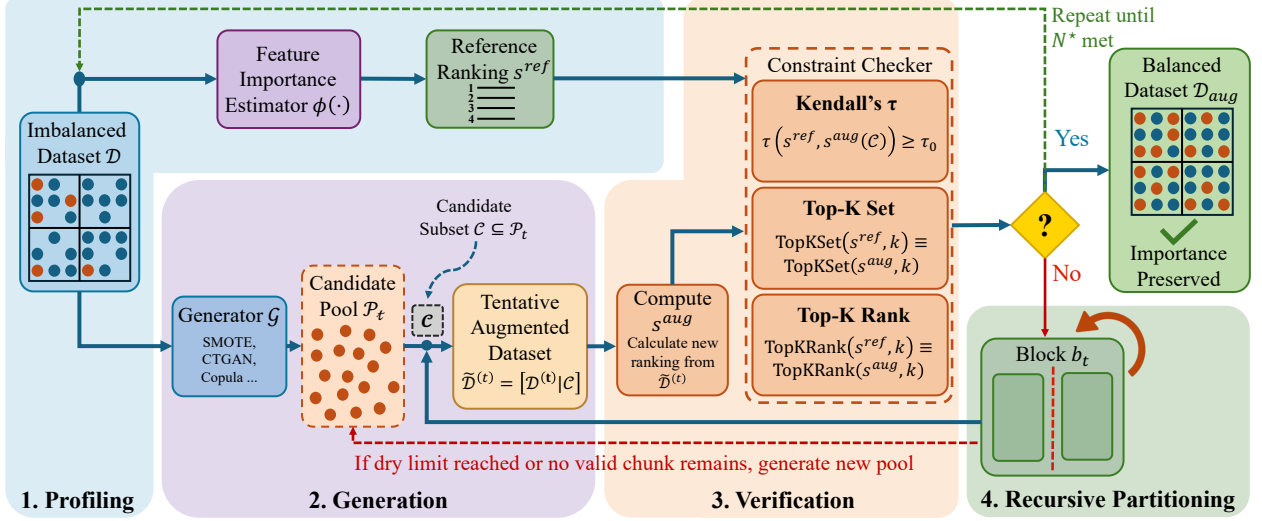


Figure 1: K-IPO workflow for importance-preserving oversampling. The method profiles the original data to compute a reference feature importance ranking, generates minority-class candidates, verifies their effect on the ranking using Kendall’s τ and optional top- k constraints, and recursively partitions rejected blocks until N^* valid synthetic samples are accepted.

3.2 Generator-Agnostic Candidate Pool Generation

At iteration t , let $\mathcal{D}^{(t)}$ denote the current augmented dataset and let N_{acc} be the number of synthetic samples accepted so far. K-IPO requests a candidate block of size

$$b_t = \min(B_{\max}, N^* - N_{acc}) \quad (2)$$

and generates

$$\mathcal{P}_t = \mathcal{G}(\mathcal{D}^{(t)}, b_t), \quad |\mathcal{P}_t| = b_t. \quad (3)$$

The generator $\mathcal{G}$ is a modular component that can be instantiated using conventional oversamplers, such as SMOTE or ADASYN, probabilistic tabular synthesizers, or deep generative models. K-IPO does not modify the generator itself; instead, it applies an importance-preserving selection layer to the generated candidate samples. As a result, the data generation method can be chosen based on the data, modelling constraints, and application requirements without changing the acceptance mechanism.

3.3 Importance-Preserving Acceptance Mechanism

For a candidate subset $\mathcal{C} \subseteq \mathcal{P}_t$, K-IPO forms the tentative dataset

$$\tilde{\mathcal{D}}^{(t)} = [\mathcal{D}^{(t)} | \mathcal{C}], \quad (4)$$

where $[\cdot | \cdot]$ denotes row-wise dataset concatenation, and recomputes the importance scores as

$$\mathbf{s}^{aug}(\mathcal{C}) = \phi(\tilde{\mathcal{D}}^{(t)}). \quad (5)$$

The subset is accepted only if the ranking induced by $\mathbf{s}^{aug}(\mathcal{C})$ is sufficiently aligned with the reference ranking. The primary acceptance criterion is Kendall’s rank correlation coefficient:

$$\tau(\mathbf{s}^{ref}, \mathbf{s}^{aug}(\mathcal{C})) \geq \tau_0, \quad (6)$$

where $\tau_0 \in (0, 1]$ is a user-defined minimum permitted Kendall rank correlation. Kendall’s τ measures ordinal agreement by comparing concordant and discordant feature pairs [22]. Values close to 1 indicate strong preservation of the reference ordering, whereas lower values indicate increasing rank drift. In addition to this global criterion, K-IPO can optionally enforce stricter local consistency through user-defined top- k constraints. Specifically, the top- k overlap

constraint requires that the set of the k most important features remains unchanged, while the top- k ordering constraint requires that the internal ordering of these features also remains unchanged:

$$\text{TopKSet}(s^{ref}, k) \equiv \text{TopKSet}(s^{aug}, k), \quad \text{TopKRank}(s^{ref}, k) \equiv \text{TopKRank}(s^{aug}, k). \quad (7)$$

A third, more restrictive option limits changes in the magnitudes of the relative importance scores of the top-ranked features. Because such a constraint is generally suitable only for small datasets with specific score characteristics, all three top- k criteria are optional and user-defined. Practitioners may enforce overlap alone, overlap and ordering, or an additional magnitude constraint. The K-IPO configuration evaluated in this study did not use magnitude preservation and was therefore omitted from Fig. 1.

3.4 Adaptive Block-and-Chunk Search Strategy

K-IPO first evaluates the complete candidate block $\mathcal{P}_t$. If the block satisfies Eq. (6) and any enabled top- k constraints, it is accepted in full. Otherwise, rejecting the entire block may be overly conservative when only a subset of samples violates the acceptance criteria. At the same time, evaluating samples individually can increase selection bias and require substantially more feature importance computations.

To balance these concerns, K-IPO uses an adaptive block-and-chunk search. It initializes the chunk size as $c = \lfloor |\mathcal{P}_t|/2 \rfloor$, partitions the remaining candidates into contiguous chunks of size c , and halves c after each pass until $c < B_{\min}$. Chunks are evaluated sequentially against the current augmented dataset. Each accepted chunk is immediately added to the dataset and removed from the candidate pool, ensuring that subsequent evaluations account for all samples accepted earlier in the iteration. In this way, every dataset update satisfies the preservation criteria.

The search is terminated and a new candidate pool is generated if no acceptable sub-block is found before c falls below $B_{\min}$, or if the number of consecutive chunk rejections reaches the maximum dry-attempt limit. This strategy improves candidate utilization and robustness to imperfect generators while balancing feature importance preservation with computational cost.

3.5 Computational Requirements

Two factors primarily determine the computational overhead of K-IPO: (i) the number of times the underlying generator produces candidate sample pools, and (ii) the number of times the importance-preserving acceptance mechanism is invoked.

Let G denote the number of calls to the underlying generator, and let M denote the number of calls to the acceptance function after the reference feature ranking has been computed. Furthermore, let $C_g(n, d, k)$ denote the cost of generating a candidate pool of k samples from a dataset with n samples and d features, and let $C_\phi(n, d)$ denote the cost of computing the feature importance ranking on n samples and d features. The additional computational cost introduced by K-IPO can therefore be approximated as $G \cdot C_g(n, d, k) + M \cdot C_\phi(n, d)$, plus lower-order costs associated with ranking comparison, Kendall’s τ computation, and optional top- k consistency checks.

When most generated candidate blocks are accepted without further partitioning, the number of pool generation steps is significantly reduced, and the number of acceptance evaluations is approximately given by

$$M \approx \left\lceil \frac{N^*}{B_{\max}} \right\rceil. \quad (8)$$

In this favorable case, the method benefits from block-level screening, requiring only a limited number of feature importance recomputations. However, when candidate blocks are repeatedly rejected and recursively split into smaller sub-blocks, the number of acceptance evaluations increases until either the minimum block size $B_{\min}$ is reached or the dry-attempt limit is exceeded. With the ANOVA F -score used in this study, $C_\phi(n, d)$ increases linearly with the number of samples in the dataset. Consequently, the overall runtime of K-IPO is mainly governed by the dataset scale, the block-size hyperparameters, the quality and computational complexity of the candidate generator, and the strictness of the Kendall’s τ and optional top- k preservation constraints. These factors induce a trade-off between computational efficiency and the degree of feature importance preservation.

4 Experimental Setup

This section describes the experimental protocol adopted to evaluate the proposed framework, including the datasets, preprocessing steps, baseline oversampling methods, predictive models, XAI techniques, and evaluation metrics. All

Algorithm 1 Kendall-constrained Importance-Preserving Oversampling (K-IPO)

Require: Dataset $\mathcal{D}$, target ratio r , generator $\mathcal{G}$, importance function ϕ , Kendall threshold τ_0 , optional top- k constraints, maximum block size $B_{\max}$, minimum block size $B_{\min}$, maximum dry attempts $A_{\max}$

Ensure: Balanced dataset $\mathcal{D}_{aug}$

```

1: Compute  $N^* = \max(0, \lfloor r \cdot n_{maj} \rfloor - n_{min})$ ,  $N_{acc} \leftarrow 0$ 
2:  $\mathcal{D}_{aug} \leftarrow \mathcal{D}$ ,  $\mathbf{s}^{ref} \leftarrow \phi(\mathcal{D})$ 
3: while  $N_{acc} < N^*$  do
4:    $b_t \leftarrow \min(B_{\max}, N^* - N_{acc})$ 
5:   Generate candidate pool  $\mathcal{P}_t \leftarrow \mathcal{G}(\mathcal{D}^{(t)}, b_t)$ 
6:   if IMPORTANCEPRESERVINGACCEPT( $\mathcal{P}_t, \mathcal{D}_{aug}, \mathbf{s}^{ref}$ ) then
7:      $\mathcal{D}_{aug} \leftarrow [\mathcal{D}_{aug} | \mathcal{P}_t]$ 
8:      $N_{acc} \leftarrow N_{acc} + |\mathcal{P}_t|$ 
9:   else
10:     $\mathcal{R} \leftarrow \mathcal{P}_t$ ,  $c \leftarrow \lfloor |\mathcal{R}|/2 \rfloor$ ,  $dry \leftarrow 0$ 
11:    while  $c \geq B_{\min}$  and  $dry < A_{\max}$  do
12:      Partition  $\mathcal{R}$  into contiguous chunks of size  $c$ 
13:      if  $\text{mod}(|\mathcal{R}|, c) \neq 0$  then
14:        Discard last chunk
15:      end if
16:      for each evaluated chunk  $\mathcal{C}$  do
17:        if IMPORTANCEPRESERVINGACCEPT( $\mathcal{C}, \mathcal{D}_{aug}, \mathbf{s}^{ref}$ ) then
18:           $\mathcal{D}_{aug} \leftarrow [\mathcal{D}_{aug} | \mathcal{C}]$ 
19:           $N_{acc} \leftarrow N_{acc} + |\mathcal{C}|$ 
20:           $\mathcal{R} \leftarrow \mathcal{R} \setminus \mathcal{C}$  ▷ Remove accepted chunk
21:           $dry \leftarrow 0$ 
22:        else
23:           $dry \leftarrow dry + 1$ 
24:          if  $dry \geq A_{\max}$  then
25:            break ▷ Abandon current block search
26:          end if
27:        end if
28:      end for
29:       $c \leftarrow \lfloor c/2 \rfloor$ 
30:    end while
31:  end if
32: end while
33: return  $\mathcal{D}_{aug} \leftarrow \mathcal{D}^{(t)}$ 
34: function IMPORTANCEPRESERVINGACCEPT( $\mathcal{C}, \mathcal{D}^{(t)}, \mathbf{s}^{ref}$ )
35:   Compute  $\mathbf{s}^{aug} = \phi([\mathcal{D}^{(t)} | \mathcal{C}])$ 
36:   Accept if:
37:      $\tau(\mathbf{s}^{ref}, \mathbf{s}^{aug}) \geq \tau_0 \wedge$  enabled top- $k$  constraints are satisfied
38: end function

```

experiments were conducted using Python 3.10.20 on a workstation equipped with an AMD Ryzen 9 7950x processor, 32 GB of RAM, an NVIDIA GeForce RTX 4070 Ti GPU with 12 GB VRAM and a Kingston KC3000 PCIe 4.0 NVMe SSD. The source code and all experimental results are publicly available on GitHub¹.

4.1 Datasets

We evaluated K-IPO on 20 tabular binary classification datasets from widely used repositories and collections, including UCI, OpenML, Kaggle, and imbalanced-learn, as well as domain-specific datasets from prior studies [42, 43]. The benchmark spans diverse application domains, sample sizes, feature dimensionalities and types, and class-imbalance ratios, including numerical, categorical, and mixed-type data. For seven datasets whose original class distributions were either balanced or only mildly imbalanced, namely airlines, bank-customer-churn-prediction, churn, rl, fried,

¹<https://github.com/CEID-HPCLAB/K-IPO>

bank32nh, and online shoppers purchasing intention, we artificially induced imbalance using imbalanced-learn’s `make_imbalance()` function, setting the minority-to-majority ratio to approximately 0.1.

For each dataset, we extracted the target variable and encoded the majority and minority classes as 0 and 1, respectively. Categorical features were label- or ordinal-encoded as appropriate, while numerical features were standardized using Z-score normalization. All transformations were fitted on the training data and applied to the corresponding test split. Table 2 summarizes the sample size, feature composition, imbalance ratio, and separability score, $1 - N3$, where $N3$ is the leave-one-out error rate of a 1-nearest-neighbor classifier [44]. Higher $1 - N3$ values indicate greater local class separability and, therefore, lower classification complexity.

Table 2: Detailed description of binary imbalanced datasets used in the experiments.

ID	Dataset	# N	# F_n	# F_c	Maj.	Min.	IR	$1 - N3$
D1	ai4i 2020	9981	5	1	9643	338	0.035	0.967
D2	abalone	4177	7	1	3786	391	0.103	0.866
D3	airlines	16427	3	4	14934	1493	0.100	0.842
D4	seismic-bumps	2584	11	4	2414	170	0.070	0.896
D5	bank customer churn prediction	8759	8	2	7963	796	0.100	0.891
D6	churn	4722	17	3	4293	429	0.100	0.903
D7	bank-marketing	4521	7	9	4000	521	0.130	0.872
D8	rl	2733	5	7	2485	248	0.100	0.867
D9	online shoppers purchasing intention	11464	15	2	10422	1042	0.100	0.897
D10	car_eval_4	1728	0	6	1663	65	0.039	0.965
D11	mammography	11183	6	0	10923	260	0.024	0.984
D12	ur3 cobotops	7355	20	0	6837	518	0.076	0.925
D13	nhanes	2105	7	0	1914	191	0.100	0.847
D14	magic gamma telescope	13565	10	0	12332	1233	0.100	0.925
D15	fried	22469	10	0	20427	2042	0.100	0.924
D16	bank32nh	6213	32	0	5649	564	0.100	0.877
D17	wilt	4839	5	0	4578	261	0.057	0.971
D18	lines-overload-50	4368	187	0	4046	322	0.080	0.978
D19	japanese vowels	9961	14	0	8347	1614	0.193	0.998
D20	pen_digits	10992	16	0	9937	1055	0.106	0.999

Note: # N denotes the number of samples; # F_n and # F_c denote the numbers of numerical and categorical features, respectively; Maj. and Min. denote the numbers of samples in the majority and minority classes, respectively; IR denotes the minority-to-majority class ratio, calculated as $\text{Min.}/\text{Maj.}$; and $1 - N3$ is a class-separability indicator based on nearest-neighbor misclassification.

4.2 Baseline oversampling and tabular generation methods

K-IPO was evaluated against representative baselines spanning the main oversampling and tabular generation paradigms reviewed in Section 2:

1. **SMOTE-family neighborhood-based oversampling.** The SMOTE variant was selected a priori based on the feature types of each dataset. Mixed numerical-categorical datasets used SMOTE-NC [9], which interpolates continuous features and assigns valid categorical values from the local minority neighborhood. The fully categorical car-eval-4 dataset used SMOTEN [9], which defines neighborhoods using categorical distance and generates nominal values without interpolating the numerical encodings. Numerical-only datasets used SMOTEB [26], a recent noise-aware variant whose original evaluation reported significant MCC improvements over the standard SMOTE and random oversampling. Its competitiveness is further supported by a recent study of 18 imbalanced datasets, in which it outperformed established variants, including Borderline-SMOTE, ADASYN, SVM-SMOTE, ProWSyn, and polyfit-Mesh, in terms of the average F1-score rank with an SVM classifier [45].
2. **Statistical distribution learning.** Gaussian Copula was included as a representative statistical generator. For a fixed training set, it fits the marginal distribution of each feature, models the dependencies among the transformed variables, and samples from the resulting joint distribution.
3. **GAN/VAE-based deep generative models.** CTGAN and TVAE were selected as representative neural tabular generators. Widely used as benchmarks for adversarial and variational generation, respectively, both are designed for mixed-type tabular data.

4. **Diffusion-based generation.** TabDDPM was included as a state-of-the-art denoising diffusion model for heterogeneous tabular data generation [12].

This selection enabled a comprehensive assessment of K-IPO against both traditional oversampling techniques and modern generative approaches [1]. For reproducibility and fair comparison across heterogeneous methods, all the baselines were evaluated using the default configurations recommended by their respective libraries.

4.3 Predictive Models

To mitigate the risk of biased results from using a single predictive model, all oversampling and tabular generation methods were evaluated using three classifiers of varying complexity and interpretability.

1. **Random Forest.** Random Forest serves as a robust ensemble baseline for tabular classification. By aggregating multiple decision trees trained on random subsets of samples and features, it reduces variance, enhances robustness against overfitting, and captures nonlinear feature interactions, making it a standard benchmark for tabular prediction tasks.
2. **Multi-Layer Perceptron (MLP).** An MLP was included as a black-box neural classifier. It consists of stacked hidden layers with nonlinear activation functions and can learn complex hierarchical mappings between the input features and class labels.
3. **GAMI-Net.** GAMI-Net was used as an explainable neural model based on disentangled additive subnetworks, where each subnetwork models either a single-feature effect or a pairwise interaction [46]. By first learning the main effects and then modeling interactions through residuals, GAMI-Net enables the inherent attribution of feature contributions.

For these models, standard configuration parameters were used: 100 trees for Random Forest, default scikit-learn settings for the MLP, and default PiML configuration with available CPU parallelism for GAMI-Net [47]. This ensured that the performance differences could be attributed primarily to the oversampling methods rather than classifier-specific optimization. Each classifier was trained independently on each augmented training set, and the final dataset-level results were obtained by averaging the performances across the three classifiers.

4.4 XAI Methods and Feature Importance Extraction

The objective of this study was not only to improve predictive performance but also to preserve the feature importance ranking of the original data. Therefore, each classifier was paired with an explanation method suitable for extracting feature importance.

For the Random Forest classifier, we employed **SHAP**, a game-theoretic post-hoc explanation framework that quantifies each feature’s contribution using Shapley values, representing its average marginal effect across all possible feature subsets [39]. Global feature importance rankings were obtained using TreeSHAP, with importance scores computed as the mean absolute SHAP values across test instances.

For the MLP, feature importance was estimated using **Sparseness-Optimized Feature Importance (SOFI)**, a post-hoc method that measures the impact of controlled feature marginalization on predictive performance [48]. Prior work has shown that SOFI produces more faithful and accurate feature rankings than alternative post-hoc methods, as indicated by the stronger correspondence between feature removal and performance degradation.

For **GAMI-Net**, feature importance was extracted directly from the fitted model through its intrinsic interpretability, eliminating the need for a separate explanation procedure.

To prevent data leakage, feature importance was extracted after model training within each experimental repetition, using only the model fitted to the corresponding augmented training set. When an explanation method required access to evaluation data, it was applied only after model training, and all preprocessing steps were completed. Consequently, any differences in the feature importance rankings can be attributed to the effects of oversampling rather than contamination by information outside the intended data split.

For each model–XAI pair, the pipeline for training, evaluation, and feature importance ranking extraction is independent, enabling parallel execution to reduce the overall runtime. As the computational costs of each predictive model and XAI technique varied significantly, the resulting workload was inherently irregular. To address this, we adopted a task-based parallel programming paradigm, which is well-suited for irregular workloads, in contrast to traditional parallel programming models that assume regular computation patterns. Our implementation is based on torcpy [49], an MPI-based Python runtime for task-based parallelism on both shared and distributed memory platforms.

4.5 Evaluation Metrics

The evaluation considered four complementary dimensions: predictive performance, feature importance preservation, explainability consistency, and class separability. Predictive performance was assessed using Balanced Accuracy and the Matthews Correlation Coefficient (MCC) as the primary threshold-dependent metrics, as both provide robust performance estimates under class imbalance. The F1-score was also used to summarize positive-class performance. To evaluate threshold-independent discrimination, the Precision–Recall Area Under the Curve (PR-AUC) was computed from the predicted probabilities, as it is particularly informative when the minority class is underrepresented. Because the conventional decision threshold of 0.5 is often suboptimal for imbalanced datasets, binary predictions were generated using a threshold that maximized the geometric mean (G-mean) of sensitivity and specificity. This criterion balances minority-class recall and majority-class specificity, providing a more equitable assessment of classification performance.

Feature importance preservation was quantified using Kendall’s τ , which measures the ordinal agreement between the feature importance rankings obtained before and after synthetic oversampling. Finally, explainability consistency was evaluated to determine whether the model-derived feature rankings remained aligned with an external reference ranking. Following the functionality-grounded evaluation framework of Elkhawaga *et al.* [50], reference rankings were constructed from a diverse set of filter-based feature selection methods and used as proxies for ground-truth feature relevance. Model-based and reference rankings were then compared to quantify the extent to which the explanations reflected feature importance supported by the data. The selected filter methods span statistical dependence, information-theoretic, neighborhood-based, impurity-based, and manifold-structure criteria, thereby capturing complementary notions of feature relevance.

1. **Distance Correlation (dCor):** A nonparametric measure of statistical dependence based on the pairwise distances among observations [51]. Unlike conventional correlation coefficients, which primarily characterize linear relationships, distance correlation can capture both linear and nonlinear associations. For each feature X_j , its relevance score was computed as $\text{dCor}(X_j, y)$, where y denotes the binary target variable, and features were ranked in descending order of their distance-correlation values.
2. **Mutual Information (MI):** An information-theoretic filter that quantifies the statistical dependency between each feature and the target variable, capturing both linear and non-linear relationships. Features with higher mutual information are considered more informative [52].
3. **Laplacian Score:** An unsupervised method that evaluates each feature’s ability to preserve local neighborhood structures in the data [53]. It favors features that reflect the intrinsic geometric layout and assigns lower scores to less informative features.
4. **ReliefF:** A neighbor-based filter that assesses how well features differentiate between nearby instances from different classes [54]. For each sample, the algorithm compares the feature values with its nearest neighbors from the same and different classes. It increases a feature’s weight when this feature helps distinguish classes and decreases it when it varies within the same class. ReliefF is robust against noise and sensitive to feature interactions.
5. **Gini Index:** A measure of impurity used to evaluate the quality of a feature based on how well it separates different classes. It calculates the probability that a random data instance is incorrectly classified when labelled according to the class distribution in a subset. A Gini Index of 0 indicates perfect purity (all data instances belong to one class), while higher values indicate increasing impurity [55].
6. **Joint Mutual Information (JMI):** An information-theoretic criterion that selects features by maximizing the information shared between a candidate feature, the already selected feature set, and the target class. It is designed to account for feature interactions, aiming to provide a balance between the relevance of a feature to the class and its redundancy with features already in the subset [56].
7. **Maximum Relevance and Max-Independence (MRI):** A greedy multivariate filter method selects features by considering both their relevance to the target and the amount of new, independent classification information they add concerning the features already selected. MRI emphasizes whether a candidate feature contributes additional class-discriminative information beyond the currently selected subset [57].

For each dataset, explainability consistency was measured as the average overlap between the top-ranked features identified by the model and the reference methods. The subset size was determined as the smallest number of top features accounting for at least 90% of the cumulative ANOVA F -score, and this subset was used for the overlap calculation. Fig. 2 provides a toy example in which, for a dataset with 10 features, the top 4 features account for more than 90% of the cumulative ANOVA F -score and are therefore selected. Table 3 reports the selected subset sizes and their proportions relative to the total number of features for each dataset. Finally, because imbalance severity should not be characterized solely by class ratios [58], we also assessed class separability as an additional source of classification difficulty using the $1 - N3$ indicator, as described in Section 4.1.

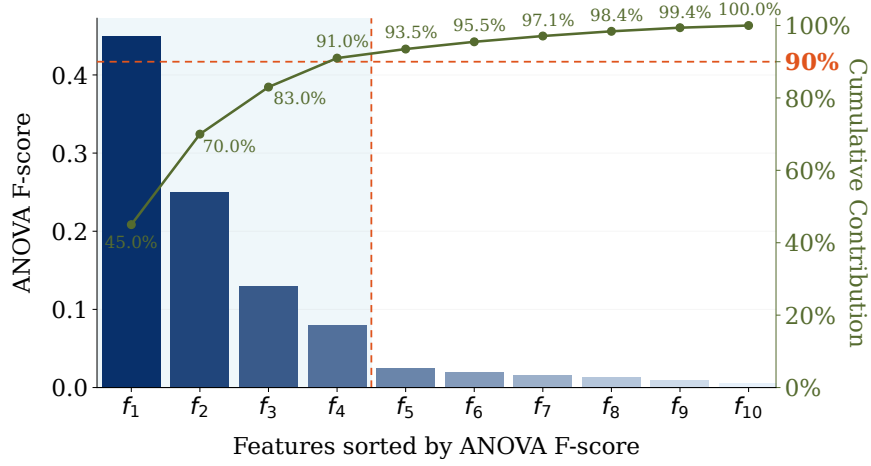


Figure 2: Toy example of feature selection using cumulative ANOVA F-score: the smallest subset reaching 90% cumulative contribution is selected (here, the first four features).

Table 3: Number and proportion of selected features per dataset, defined as the smallest subset accounting for at least 90% of the cumulative ANOVA F-score.

Dataset	# Selected	Total Features	Proportion
ai4i 2020	3	6	0.50
abalone	6	8	0.75
airlines	5	7	0.71
seismic-bumps	10	15	0.67
bank customer churn prediction	4	10	0.40
churn	7	20	0.35
bank-marketing	5	16	0.31
rl	5	12	0.42
online shoppers purchasing intention	6	17	0.35
car_eval_4	4	6	0.67
mammography	3	6	0.50
ur3 cobotops	12	20	0.60
nhanes	4	7	0.57
magic gamma telescope	4	10	0.40
fried	3	10	0.30
bank32nh	3	32	0.09
wilt	3	5	0.60
lines-overload-50	110	187	0.59
japanese vowels	6	14	0.43
pen_digits	9	16	0.56

5 Experimental Results

This section evaluates the effectiveness of K-IPO in enhancing minority-class representation while maintaining the original feature importance ranking. The evaluation was conducted in two stages. First, a sensitivity analysis examined how key K-IPO design choices, namely the candidate generator and the strength of the importance preservation constraints, affect the trade-off between predictive performance and interpretability. Second, K-IPO was compared with the baseline oversampling methods using the predictive and interpretability metrics defined above.

5.1 Sensitivity Analysis of K-IPO Design Components

We first examined the sensitivity of K-IPO to its two main design components: the candidate generator and importance preservation constraints. At the beginning of each repetition, the original data were stratified into training and test sets using an 80/20 ratio. All preprocessing transformations were fitted exclusively to the training set and subsequently applied to the test set. Each of the three classifiers was then fitted independently to the resulting augmented training set. The classification threshold that maximized the G-mean was estimated from an internal validation split of the training data and fixed before evaluation. Finally, the predictive performance and model-derived feature importance were evaluated using the untouched original test set.

Regarding the most suitable candidate data generator within K-IPO, we compared the methods discussed in Section 4.2. Performance was assessed using Balanced Accuracy, F1-score, PR-AUC, and MCC. Table 4 reports the corresponding results, with the best and second-best values for each dataset–metric pair shown in bold and underlined, respectively. In this experiment, Kendall’s agreement threshold was fixed at $\tau_0 = 0.7$, enforcing a strong but not overly restrictive agreement between the original and augmented feature importance rankings. This configuration allows K-IPO to reject synthetic samples that distort the feature importance structure while still permitting sufficient flexibility for minority-class augmentation. The reported values were averaged across repetitions and the three classifiers. A failure was recorded whenever K-IPO could not obtain the target number N^* of accepted synthetic samples within a reasonable runtime under the available computational resources. These cases are indicated by a dash in Table 4.

The results indicate that K-IPO performs more consistently when paired with SMOTE-based generators. Specifically, SMOTE achieved the highest number of wins for Balanced Accuracy, F1-score, and MCC, with 13 wins in each category, while also securing the second-highest number of PR-AUC wins (8). Notably, SMOTE was the only generator that completed the requested augmentation for all 20 datasets, indicating reliable candidate generation under the experimental conditions. This can be attributed to the generation of synthetic samples in the local neighborhood of existing minority class observations, thereby preserving the minority-class geometry and producing candidates that are more likely to satisfy the imposed constraints. This behavior is particularly advantageous in small- and medium-sized tabular datasets, where more complex deep generative models may not have sufficient minority class evidence to learn stable data distributions [59, 60].

Among the deep generative models, CTGAN was the strongest performer. It achieved the second-highest number of wins in terms of Balanced Accuracy, F1-score, and MCC (5 wins each) and the highest number of wins in terms of PR-AUC (9 wins). This indicates that CTGAN is particularly effective when performance depends on the quality of probability estimates and instance ranking, likely because its conditional generation mechanism can capture complex nonlinear dependencies in tabular data. However, CTGAN also recorded 3 failures and exhibited lower overall robustness, suggesting that its greater expressive power comes at the cost of increased training instability [61]. Gaussian Copula was less competitive, achieving only 2 wins in Balanced Accuracy, F1-score, and MCC, and 3 wins in PR-AUC, while failing on 6 datasets. These results highlight the limitations of its parametric dependence assumptions, which may be insufficient for heterogeneous datasets with complex nonlinear relationships. TVAE showed limited effectiveness, with no wins in Balanced Accuracy, F1-score, or MCC, only 2 wins in PR-AUC, and 9 failures. Although VAEs are generally considered more stable and easier to train than GANs, latent space regularization often produces overly smooth synthetic samples, reducing their discriminative value [62]. This drawback becomes particularly problematic in highly imbalanced settings, where VAEs frequently underperform both GAN-based approaches and simpler statistical generators [63]. TabDDPM was the weakest method overall, achieving no wins on any metric and failing on 11 datasets, indicating its poor compatibility with the proposed framework. Overall, the results suggest that within K-IPO, simpler interpolation-based generators provide the best trade-off between predictive performance, stability, and compatibility with importance-preserving acceptance. More expressive deep generative models can deliver occasional performance gains, but their benefits are offset by reduced robustness and greater sensitivity across diverse and imbalanced tabular datasets.

Subsequently, we examined the importance of preservation constraints, focusing on the Kendall’s threshold τ_0 and the top- k ordering constraint. We emphasized top- k ordering because it provides the most informative middle ground between the two alternative top- k constraints. On the one hand, top- k overlap is comparatively weak because it

Table 4: Generator-sensitivity analysis of K-IPO at a fixed Kendall’s agreement threshold of $\tau_0 = 0.7$. Reported values are averaged across 10 independent repetitions and three classifiers.

Dataset	Balanced Accuracy				F1-score				PR AUC				MCC			
	SMOTE	CTGAN	TVAE	GC	TabDDPM	SMOTE	CTGAN	TVAE	GC	TabDDPM	SMOTE	CTGAN	TVAE	GC	TabDDPM	SMOTE
Mixed numerical–categorical datasets																
D1	0.9774	0.9659	0.9626	0.9655	0.8996	0.9730	0.9616	0.9558	0.9618	0.8865	0.9916	0.9923	0.9897	0.9913	0.9540	0.9526
D2	0.8954	0.9311	0.9056	0.8436	—	0.8802	0.9241	0.8922	0.8238	—	0.8967	0.9787	0.9640	0.8617	—	0.7847
D3	0.8263	0.7847	<u>0.8210</u>	0.7769	—	0.8033	0.7557	<u>0.7954</u>	0.7406	—	0.8566	0.8380	0.8762	0.8168	—	0.6534
D4	0.9016	0.9518	—	—	—	0.8874	0.9482	—	—	—	0.9317	0.9769	—	—	—	0.7997
D5	0.9036	0.9100	—	—	—	0.8893	0.8987	—	—	—	0.9507	0.9581	—	—	—	0.8033
D6	0.9520	—	—	0.8874	—	0.9446	—	—	0.8695	—	0.9850	—	—	0.9390	—	0.9030
D7	0.9037	0.9120	—	0.9073	—	0.8898	0.9007	—	0.8957	—	0.9492	0.9675	—	0.9643	—	0.8052
D8	0.9032	0.8978	—	0.8558	—	0.8901	0.8842	—	0.8354	—	0.9434	0.9452	—	0.9183	—	0.8094
D9	0.9362	0.9447	—	—	—	0.9256	0.9383	—	—	—	0.9690	0.9847	—	—	—	0.8682
Categorical-only datasets																
D10	1.0000	0.9850	<u>0.9995</u>	0.9850	0.4742	1.0000	0.9804	<u>0.9994</u>	0.9804	0.4310	1.0000	0.9849	<u>0.9999</u>	0.9856	0.4456	1.0000
Numerical-only datasets																
D11	0.9871	0.9786	<u>0.9852</u>	—	0.9361	0.9839	0.9753	<u>0.9821</u>	—	0.9281	0.9949	0.9966	<u>0.9954</u>	—	0.9692	0.9718
D12	0.9436	—	—	0.9566	—	0.9345	—	—	0.9521	—	0.9736	—	—	—	—	0.8838
D13	0.8774	0.8944	—	0.9304	0.7237	0.8612	0.8813	—	0.9232	0.6861	0.9530	0.9369	—	0.9877	0.7618	0.7578
D14	0.9772	0.9656	0.9387	0.9507	0.8978	0.9741	0.9619	0.9299	0.9442	0.8836	0.9961	0.9900	0.9783	0.9850	0.9561	0.9547
D15	0.9696	0.9430	0.9595	0.8676	0.8849	0.9651	0.9366	0.9547	0.8484	0.8689	0.9946	0.9803	<u>0.9895</u>	0.9333	0.9483	0.9389
D16	0.9758	—	—	—	—	0.9720	—	—	—	—	0.9828	—	—	—	—	0.9512
D17	0.9910	0.9436	0.9387	0.9648	0.9768	0.9890	0.9357	0.9305	0.9592	0.9722	0.9980	0.9819	0.9797	0.9907	0.9943	0.9807
D18	0.9979	0.9955	0.9969	0.9964	—	0.9971	0.9944	0.9962	0.9954	—	0.9990	0.9998	0.9998	0.9998	—	0.9950
D19	0.9977	0.9887	0.9891	0.9858	0.9876	0.9973	0.9871	0.9871	0.9838	0.9858	0.9999	0.9990	0.9990	0.9985	0.9988	0.9953
D20	0.99998	0.9975	<u>0.9981</u>	—	0.9909	0.99998	0.9972	<u>0.9978</u>	—	0.9898	1.0000	0.9997	<u>0.9999</u>	—	0.9989	0.9996
Wins: SMOTE = (13, 13, 8, 13), CTGAN = (5, 5, 9, 5), TVAe = (0, 0, 2, 0), GC = (2, 2, 3, 2), TabDDPM = (0, 0, 0, 0) Failures: SMOTE = 0, CTGAN = 3, TVAe = 9, GC = 6, TabDDPM = 11																

Note. Bold and underlined values indicate the best- and second-best-performing generators, respectively, for each dataset and metric. A dash (—) indicates failure to generate the requested N^* accepted synthetic samples within a practical execution time. Failures count such non-completions across datasets. Wins are ordered as Balanced Accuracy, F1-score, PR AUC, and MCC. The SMOTE-family variant was selected according to feature composition: SMOTENC for mixed numerical–categorical datasets, SMOTEN for categorical-only datasets, and SMOTENB for numerical-only datasets.

requires only that the same features appear in the top set without preserving their internal order. Moreover, part of this information is already reflected by Kendall’s τ , as a high global rank correlation generally implies substantial agreement among the most influential features. Additionally, top- k overlap does not scale monotonically in an interpretable way with k : preserving the top- k set for a larger value of k does not necessarily guarantee the preservation of smaller, more critical subsets, limiting its interpretability as a progressive constraint. In contrast, top- k magnitude preservation is considerably stricter because it constrains relative importance differences in addition to the rank order. This increases the computational burden, especially for larger and high-dimensional datasets that require repeated candidate evaluations, while offering limited additional benefit to the objective of this study, where preserving the order of influential features is the primary concern.

To jointly assess predictive performance and interpretability, we computed the harmonic mean of a predictive component P and an interpretability component I for each configuration of τ_0 and top- k ordering across all datasets. The predictive component was defined as the average of Accuracy and PR-AUC, while the interpretability component measured the consistency between model-derived and reference feature rankings, as described in Section 4.5. We combined these components as $\frac{2PI}{P+I}$, favoring configurations that achieved strong performance in both dimensions rather than improving one at the expense of the other.

The results, summarized in Fig. 3, reveal a consistent overall trend, along with dataset-specific variations. Generally, the harmonic mean improves as Kendall’s threshold becomes stricter, indicating that stronger preservation of the original importance ranking enhances the aggregate predictive and explanatory power of the trained models. In contrast, the effect of the top- k ordering constraint is weaker and more dependent on the dataset. In several cases, including airlines, pen_digits, and seismic-bumps, performance improves as the top- k ratio increases but eventually plateaus, even exhibiting diminishing returns. Across all datasets, increasing the top- k ratio generally leads to performance improvements or stabilization; however, the rate of improvement and saturation point vary, reflecting differences in dataset sensitivity to this constraint. Overall, these findings indicate that the most substantial gains arise from enforcing a sufficiently strict, global rank-preservation constraint. While preserving the ordering of a larger subset of important features is beneficial, the advantages of expanding the top- k constraint tend to saturate, with limited additional gains beyond a dataset-dependent threshold.

5.2 Comparison with baseline oversampling techniques

After analyzing the sensitivity of K-IPO to its design parameters, we compared it with the baseline methods described in Section 4.2. All methods were applied before model training. For K-IPO, the importance-preserving parameter values were selected separately for each dataset based on the aforementioned sensitivity analysis, ensuring operation within the most suitable regime. In contrast, all baseline methods were evaluated using their default library-recommended configuration parameters to maintain a fair and method-agnostic comparison. As in Section 5.1, the augmentation process for each dataset-generator pair was repeated 10 times, and the results are reported as the average over all runs.

Tables 5 and 6 summarize the results across all datasets, reporting the Balanced Accuracy, F1-score, MCC, composite predictive score (P), explainability consistency score (I), Kendall’s τ after augmentation, and separability indicator ($1 - N3$). All values represent the averages of the three predictive models coupled with their corresponding XAI methods for 10 independent experimental repetitions.

The results indicate that K-IPO offers the most favorable trade-off between predictive performance and interpretability. Despite the structural constraints imposed during augmentation, K-IPO remains highly competitive, achieving the highest number of wins in Balanced Accuracy and leading in F1-score, MCC, and P (9 wins each). These findings suggest that the sample-selection mechanism preserves the downstream classifier utility by filtering out synthetic instances that could distort the decision boundary. However, this advantage was not uniform across all datasets, as expected, given the restrictive acceptance criterion. In some cases, unconstrained generators, such as CTGAN or TabDDPM, achieved higher predictive scores; however, these gains often came at the cost of weaker feature importance preservation. For instance, in D12, CTGAN obtained the best predictive performance, but its Kendall’s τ was only 0.0211, indicating a severe distortion of the original feature importance ranking. Gaussian Copula showed a stable but weaker predictive performance, achieving only one win in Balanced Accuracy, F1-score, MCC, and P . This suggests that its lightweight statistical fit-and-sample mechanism can be effective in isolated cases, but its distributional assumptions may limit its ability to capture dataset-specific minority class structures. Moreover, the zero standard deviations reported for the Gaussian Copula arise from the design of the Synthetic Data Vault (SDV) library [64], which was used to implement this generator. Specifically, the SDV API does not expose the random state used during sample generation and instead relies on its internal default random state. Because the same train/test split was used across all experimental runs, the generator produced identical augmented datasets in every repetition. Consequently, the evaluation metrics remained unchanged throughout the runs, leading to a zero standard deviation.

Table 5: Comparison of synthetic data generation methods on datasets D1–D10 across seven evaluation metrics. Entries represent mean $\pm$ standard deviation over multiple runs. **Bold** values indicate the best performance per metric per dataset.

Method	Datasets D1–D5							Datasets D6–D10						
	Bal. Acc.	F1	MCC	P	Kendall's τ	I	1–N3	Bal. Acc.	F1	MCC	P	I	Kendall's τ	1–N3
D1														
CTGAN	0.9567 $\pm$ 0.0097	0.9512 $\pm$ 0.0113	0.9152 $\pm$ 0.0197	0.9758 $\pm$ 0.0068	0.6318 $\pm$ 0.0891	0.2267 $\pm$ 0.2335	0.9309 $\pm$ 0.0138	0.9673 $\pm$ 0.0018	0.9634 $\pm$ 0.0020	0.9366 $\pm$ 0.0035	0.9790 $\pm$ 0.0009	0.4082 $\pm$ 0.0314	0.1148 $\pm$ 0.1172	0.8539 $\pm$ 0.0209
Gaussian Copula	0.9567 $\pm$ 0.0097	0.9512 $\pm$ 0.0113	0.9152 $\pm$ 0.0197	0.9758 $\pm$ 0.0068	0.6318 $\pm$ 0.0891	0.2267 $\pm$ 0.2335	0.9309 $\pm$ 0.0138	0.9673 $\pm$ 0.0018	0.9634 $\pm$ 0.0020	0.9366 $\pm$ 0.0035	0.9790 $\pm$ 0.0009	0.4082 $\pm$ 0.0314	0.1148 $\pm$ 0.1172	0.8539 $\pm$ 0.0209
K-IPO	0.9812 $\pm$ 0.0016	0.9777 $\pm$ 0.0019	0.9698 $\pm$ 0.0034	0.9870 $\pm$ 0.0012	0.8666 $\pm$ 0.0010	1.0000 $\pm$ 0.0000	0.9828 $\pm$ 0.0007	0.9673 $\pm$ 0.0018	0.9634 $\pm$ 0.0020	0.9366 $\pm$ 0.0035	0.9790 $\pm$ 0.0009	0.4082 $\pm$ 0.0314	0.1148 $\pm$ 0.1172	0.8539 $\pm$ 0.0209
SMOTE	0.9734 $\pm$ 0.0027	0.9688 $\pm$ 0.0030	0.9453 $\pm$ 0.0053	0.9820 $\pm$ 0.0015	0.7801 $\pm$ 0.0751	0.6000 $\pm$ 0.0000	0.9370 $\pm$ 0.0008	0.9528 $\pm$ 0.0017	0.9456 $\pm$ 0.0021	0.9050 $\pm$ 0.0036	0.9690 $\pm$ 0.0012	0.4449 $\pm$ 0.0121	0.7810 $\pm$ 0.0317	0.9393 $\pm$ 0.0023
TaDDPDM	0.8730 $\pm$ 0.0420	0.8558 $\pm$ 0.0482	0.7492 $\pm$ 0.0888	0.9060 $\pm$ 0.0390	0.7810 $\pm$ 0.0751	0.4667 $\pm$ 0.2266	0.8330 $\pm$ 0.0549	0.9639 $\pm$ 0.0055	0.9509 $\pm$ 0.0061	0.9310 $\pm$ 0.0103	0.9769 $\pm$ 0.0039	0.4061 $\pm$ 0.0291	0.9726 $\pm$ 0.1378	0.9200 $\pm$ 0.0115
TVAE	0.9458 $\pm$ 0.0044	0.9365 $\pm$ 0.0050	0.8877 $\pm$ 0.0090	0.9628 $\pm$ 0.0032	0.6730 $\pm$ 0.0778	0.4400 $\pm$ 0.0562	0.9125 $\pm$ 0.0042	0.9645 $\pm$ 0.0016	0.9601 $\pm$ 0.0021	0.9307 $\pm$ 0.0040	0.9769 $\pm$ 0.0010	0.4082 $\pm$ 0.0187	0.4684 $\pm$ 0.0181	0.9008 $\pm$ 0.0043
D2														
CTGAN	0.9320 $\pm$ 0.0027	0.9251 $\pm$ 0.0037	0.8759 $\pm$ 0.0084	0.9587 $\pm$ 0.0022	0.8175 $\pm$ 0.0259	0.2572 $\pm$ 0.2054	0.8069 $\pm$ 0.0051	0.9095 $\pm$ 0.0064	0.8975 $\pm$ 0.0079	0.8248 $\pm$ 0.0153	0.9404 $\pm$ 0.0058	0.4809 $\pm$ 0.0322	0.3083 $\pm$ 0.1028	0.8083 $\pm$ 0.0370
Gaussian Copula	0.8436 $\pm$ 0.0000	0.8238 $\pm$ 0.0000	0.6803 $\pm$ 0.0000	0.8497 $\pm$ 0.0000	0.8095 $\pm$ 0.0000	0.8571 $\pm$ 0.0000	0.7301 $\pm$ 0.0000	0.9073 $\pm$ 0.0000	0.8957 $\pm$ 0.0000	0.8219 $\pm$ 0.0000	0.9382 $\pm$ 0.0000	0.4667 $\pm$ 0.0000	0.7607 $\pm$ 0.0000	0.7796 $\pm$ 0.0000
K-IPO	0.9008 $\pm$ 0.0058	0.8863 $\pm$ 0.0062	0.7961 $\pm$ 0.0115	0.9002 $\pm$ 0.0042	0.8810 $\pm$ 0.0242	1.0000 $\pm$ 0.0000	0.9192 $\pm$ 0.0018	0.9166 $\pm$ 0.0048	0.9051 $\pm$ 0.0056	0.8549 $\pm$ 0.0105	0.9390 $\pm$ 0.0039	0.4753 $\pm$ 0.0239	0.5400 $\pm$ 0.0238	0.9069 $\pm$ 0.0022
SMOTE	0.8940 $\pm$ 0.0048	0.8788 $\pm$ 0.0051	0.7824 $\pm$ 0.0092	0.8940 $\pm$ 0.0035	0.8444 $\pm$ 0.0146	0.8571 $\pm$ 0.0000	0.9045 $\pm$ 0.0024	0.9041 $\pm$ 0.0026	0.8896 $\pm$ 0.0029	0.8040 $\pm$ 0.0057	0.9237 $\pm$ 0.0031	0.4457 $\pm$ 0.0219	0.6933 $\pm$ 0.0353	0.8937 $\pm$ 0.0018
TaDDPDM	0.8669 $\pm$ 0.0351	0.8482 $\pm$ 0.0423	0.7487 $\pm$ 0.0776	0.9051 $\pm$ 0.0309	0.8270 $\pm$ 0.0272	−0.0214 $\pm$ 0.1013	0.8464 $\pm$ 0.0328	0.8176 $\pm$ 0.0016	0.9073 $\pm$ 0.0022	0.8428 $\pm$ 0.0063	0.9473 $\pm$ 0.0015	0.4905 $\pm$ 0.0299	0.2717 $\pm$ 0.0933	0.9036 $\pm$ 0.0026
TVAE	0.9186 $\pm$ 0.0028	0.9074 $\pm$ 0.0030	0.8394 $\pm$ 0.0051	0.9470 $\pm$ 0.0019	0.7595 $\pm$ 0.0195	0.4714 $\pm$ 0.0694	0.8647 $\pm$ 0.0047	0.9145 $\pm$ 0.0016	0.9027 $\pm$ 0.0020	0.8311 $\pm$ 0.0045	0.9399 $\pm$ 0.0020	0.3867 $\pm$ 0.0247	0.2700 $\pm$ 0.0422	0.8746 $\pm$ 0.0042
D3														
CTGAN	0.7904 $\pm$ 0.0132	0.7616 $\pm$ 0.0136	0.5799 $\pm$ 0.0280	0.8189 $\pm$ 0.0173	0.8114 $\pm$ 0.0481	0.4857 $\pm$ 0.1118	0.6932 $\pm$ 0.0208	0.9273 $\pm$ 0.0080	0.9193 $\pm$ 0.0098	0.8650 $\pm$ 0.0178	0.9487 $\pm$ 0.0058	0.7043 $\pm$ 0.0639	0.4515 $\pm$ 0.1542	0.8006 $\pm$ 0.0208
Gaussian Copula	0.7688 $\pm$ 0.0000	0.7427 $\pm$ 0.0000	0.5352 $\pm$ 0.0000	0.7923 $\pm$ 0.0000	0.7571 $\pm$ 0.0000	0.7143 $\pm$ 0.0000	0.6627 $\pm$ 0.0000	0.9065 $\pm$ 0.0000	0.8941 $\pm$ 0.0000	0.8162 $\pm$ 0.0000	0.9335 $\pm$ 0.0000	0.7714 $\pm$ 0.0000	0.5758 $\pm$ 0.0000	0.7427 $\pm$ 0.0000
K-IPO	0.8113 $\pm$ 0.0093	0.7877 $\pm$ 0.0113	0.6219 $\pm$ 0.0188	0.8271 $\pm$ 0.0111	0.7705 $\pm$ 0.0240	0.9524 $\pm$ 0.0502	0.8253 $\pm$ 0.0138	0.9467 $\pm$ 0.0041	0.9400 $\pm$ 0.0047	0.8965 $\pm$ 0.0083	0.9648 $\pm$ 0.0028	0.0714 $\pm$ 0.0270	0.8364 $\pm$ 0.0256	0.9082 $\pm$ 0.0028
SMOTE	0.7964 $\pm$ 0.0039	0.7706 $\pm$ 0.0032	0.5900 $\pm$ 0.0090	0.8106 $\pm$ 0.0040	0.7467 $\pm$ 0.0281	0.7143 $\pm$ 0.0449	0.8025 $\pm$ 0.0031	0.9346 $\pm$ 0.0061	0.9256 $\pm$ 0.0075	0.8708 $\pm$ 0.0141	0.9548 $\pm$ 0.0047	0.6285 $\pm$ 0.0530	0.3303 $\pm$ 0.0390	0.8840 $\pm$ 0.0026
TaDDPDM	0.9315 $\pm$ 0.0002	0.9264 $\pm$ 0.0002	0.8845 $\pm$ 0.0004	0.9488 $\pm$ 0.0004	0.7352 $\pm$ 0.0378	−0.2857 $\pm$ 0.1634	0.9022 $\pm$ 0.0008	0.9126 $\pm$ 0.0267	0.9026 $\pm$ 0.0320	0.8425 $\pm$ 0.0514	0.9374 $\pm$ 0.0203	0.7743 $\pm$ 0.0598	0.3606 $\pm$ 0.1233	0.8613 $\pm$ 0.0452
TVAE	0.8381 $\pm$ 0.0042	0.8158 $\pm$ 0.0052	0.6783 $\pm$ 0.0086	0.8670 $\pm$ 0.0038	0.6705 $\pm$ 0.0295	0.4381 $\pm$ 0.1821	0.7034 $\pm$ 0.0041	0.9076 $\pm$ 0.0104	0.8954 $\pm$ 0.0123	0.8223 $\pm$ 0.0225	0.9333 $\pm$ 0.0071	0.4357 $\pm$ 0.0422	−0.0697 $\pm$ 0.0249	0.8170 $\pm$ 0.0126
D4														
CTGAN	0.9521 $\pm$ 0.0016	0.9485 $\pm$ 0.0019	0.9141 $\pm$ 0.0034	0.9672 $\pm$ 0.0012	0.7295 $\pm$ 0.0287	0.3752 $\pm$ 0.1253	0.8677 $\pm$ 0.0111	0.9432 $\pm$ 0.0031	0.9363 $\pm$ 0.0037	0.8002 $\pm$ 0.0069	0.9649 $\pm$ 0.0025	0.3548 $\pm$ 0.0545	0.4177 $\pm$ 0.2129	0.8550 $\pm$ 0.0197
Gaussian Copula	0.9383 $\pm$ 0.0000	0.9320 $\pm$ 0.0000	0.8859 $\pm$ 0.0000	0.9568 $\pm$ 0.0000	0.7190 $\pm$ 0.0000	0.6190 $\pm$ 0.0000	0.8405 $\pm$ 0.0000	0.9332 $\pm$ 0.0000	0.9250 $\pm$ 0.0000	0.8709 $\pm$ 0.0000	0.9571 $\pm$ 0.0000	0.4683 $\pm$ 0.0000	0.4853 $\pm$ 0.0000	0.8151 $\pm$ 0.0000
K-IPO	0.9046 $\pm$ 0.0061	0.8909 $\pm$ 0.0073	0.8073 $\pm$ 0.0135	0.9227 $\pm$ 0.0066	0.7667 $\pm$ 0.0087	0.8400 $\pm$ 0.0359	0.9069 $\pm$ 0.0057	0.9411 $\pm$ 0.0022	0.9315 $\pm$ 0.0023	0.8791 $\pm$ 0.0038	0.9558 $\pm$ 0.0018	0.5682 $\pm$ 0.0412	0.8493 $\pm$ 0.0388	0.9319 $\pm$ 0.0037
SMOTE	0.8904 $\pm$ 0.0040	0.8745 $\pm$ 0.0048	0.7759 $\pm$ 0.0088	0.9035 $\pm$ 0.0037	0.7510 $\pm$ 0.0194	0.7257 $\pm$ 0.0326	0.8922 $\pm$ 0.0049	0.9360 $\pm$ 0.0028	0.9254 $\pm$ 0.0030	0.8681 $\pm$ 0.0055	0.9516 $\pm$ 0.0020	0.5349 $\pm$ 0.0443	0.6822 $\pm$ 0.0408	0.9245 $\pm$ 0.0025
TaDDPDM	0.9540 $\pm$ 0.0025	0.9517 $\pm$ 0.0028	0.9215 $\pm$ 0.0041	0.9688 $\pm$ 0.0017	0.7181 $\pm$ 0.0359	0.0248 $\pm$ 0.0739	0.9294 $\pm$ 0.0016	0.9495 $\pm$ 0.0007	0.9436 $\pm$ 0.0010	0.9030 $\pm$ 0.0025	0.9689 $\pm$ 0.0006	0.5016 $\pm$ 0.1058	0.2956 $\pm$ 0.0910	0.9158 $\pm$ 0.0017
TVAE	0.9394 $\pm$ 0.0040	0.9334 $\pm$ 0.0044	0.8880 $\pm$ 0.0068	0.9592 $\pm$ 0.0026	0.6843 $\pm$ 0.0260	0.2152 $\pm$ 0.0465	0.8708 $\pm$ 0.0039	0.9395 $\pm$ 0.0006	0.9319 $\pm$ 0.0008	0.8824 $\pm$ 0.0018	0.9603 $\pm$ 0.0008	0.4143 $\pm$ 0.0282	−0.0073 $\pm$ 0.0915	0.9088 $\pm$ 0.0038
D5														
CTGAN	0.9330 $\pm$ 0.0104	0.9022 $\pm$ 0.0124	0.8359 $\pm$ 0.0213	0.9391 $\pm$ 0.0091	0.4417 $\pm$ 0.0440	0.1200 $\pm$ 0.3479	0.7877 $\pm$ 0.0278	0.9851 $\pm$ 0.0005	0.9805 $\pm$ 0.0007	0.9659 $\pm$ 0.0012	0.9839 $\pm$ 0.0020	0.5548 $\pm$ 0.0404	0.3600 $\pm$ 0.0644	0.9678 $\pm$ 0.0026
Gaussian Copula	0.9286 $\pm$ 0.0100	0.9234 $\pm$ 0.0101	0.8319 $\pm$ 0.0193	0.9352 $\pm$ 0.0090	0.4378 $\pm$ 0.0406	0.0677 $\pm$ 0.0890	0.7394 $\pm$ 0.0000	0.9860 $\pm$ 0.0000	0.9804 $\pm$ 0.0000	0.9637 $\pm$ 0.0000	0.9842 $\pm$ 0.0000	0.8214 $\pm$ 0.0000	1.0000 $\pm$ 0.0000	0.9878 $\pm$ 0.0000
K-IPO	0.9091 $\pm$ 0.0072	0.8843 $\pm$ 0.0097	0.7914 $\pm$ 0.0076	0.9233 $\pm$ 0.0026	0.5557 $\pm$ 0.0106	0.9545 $\pm$ 0.0529	0.9393 $\pm$ 0.0027	1.0000 $\pm$ 0.0000	1.0000 $\pm$ 0.0000	1.0000 $\pm$ 0.0000	1.0000 $\pm$ 0.0000	0.8214 $\pm$ 0.0000	0.9831 $\pm$ 0.0000	0.9878 $\pm$ 0.0000
SMOTE	0.8901 $\pm$ 0.0072	0.8843 $\pm$ 0.0097	0.7914 $\pm$ 0.0076	0.9233 $\pm$ 0.0026	0.5557 $\pm$ 0.0106	0.9545 $\pm$ 0.0529	0.9393 $\pm$ 0.0027	1.0000 $\pm$ 0.0000	1.0000 $\pm$ 0.0000	1.0000 $\pm$ 0.0000	1.0000 $\pm$ 0.0000	0.8214 $\pm$ 0.0000	0.9831 $\pm$ 0.0000	0.9878 $\pm$ 0.0000
TaDDPDM	0.9493 $\pm$ 0.0015	0.9453 $\pm$ 0.0017	0.9088 $\pm$ 0.0027	0.9669 $\pm$ 0.0008	0.5333 $\pm$ 0.0411	−0.1467 $\pm$ 0.2341	0.9216 $\pm$ 0.0017	0.4770 $\pm$ 0.0164	0.4328 $\pm$ 0.0179	−0.0457 $\pm$ 0.0327	0.4579 $\pm$ 0.0158	0.6000 $\pm$ 0.0373	0.4533 $\pm$ 0.2626	0.5983 $\pm$ 0.0122
TVAE	0.9257 $\pm$ 0.0030	0.9163 $\pm$ 0.0033	0.8548 $\pm$ 0.0059	0.9420 $\pm$ 0.0027	0.4798 $\pm$ 0.0416	0.4533 $\pm$ 0.1222	0.8404 $\pm$ 0.0058	0.9996 $\pm$ 0.0006	0.9995 $\pm$ 0.0006	0.9991 $\pm$ 0.0011	0.9998 $\pm$ 0.0003	0.8464 $\pm$ 0.0379	0.8890 $\pm$ 0.0422	0.9846 $\pm$ 0.0007

Table 6: Comparison of synthetic data generation methods on datasets D11–D20 across seven evaluation metrics. Entries represent mean $\pm$ standard deviation over multiple runs. **Bold** values indicate the best performance per metric per dataset.

Method	Datasets D11–D15							Datasets D16–D20						
	Bal. Acc.	F1	MCC	P	I	Kendall's τ	1 – N3	Bal. Acc.	F1	MCC	P	I	Kendall's τ	1 – N3
D11														
CTGAN	0.9819 $\pm$ 0.0049	0.9791 $\pm$ 0.0054	0.9634 $\pm$ 0.0094	0.9896 $\pm$ 0.0029	0.5651 $\pm$ 0.0609	0.5867 $\pm$ 0.2127	0.9696 $\pm$ 0.0096	0.8840 $\pm$ 0.0093	0.8022 $\pm$ 0.0146	0.9270 $\pm$ 0.0073	0.9270 $\pm$ 0.0073	0.9270 $\pm$ 0.0073	0.1827 $\pm$ 0.0925	0.7472 $\pm$ 0.0335
Gaussian Copula	0.9836 $\pm$ 0.0049	0.9782 $\pm$ 0.0091	0.9619 $\pm$ 0.0090	0.9891 $\pm$ 0.0029	0.4444 $\pm$ 0.0099	0.5600 $\pm$ 0.0070	0.9647 $\pm$ 0.0090	0.8771 $\pm$ 0.0090	0.7941 $\pm$ 0.0040	0.9194 $\pm$ 0.0040	0.9194 $\pm$ 0.0040	0.9194 $\pm$ 0.0040	0.1827 $\pm$ 0.0925	0.7480 $\pm$ 0.0335
K-IPO	0.9873 $\pm$ 0.0009	0.9842 $\pm$ 0.0010	0.9728 $\pm$ 0.0018	0.9905 $\pm$ 0.0005	0.6206 $\pm$ 0.0117	1.0000 $\pm$ 0.0000	0.9947 $\pm$ 0.0001	0.9854 $\pm$ 0.0019	0.9743 $\pm$ 0.0083	0.9843 $\pm$ 0.0013	0.9843 $\pm$ 0.0013	0.9843 $\pm$ 0.0013	0.7746 $\pm$ 0.0082	0.9543 $\pm$ 0.0005
SMOTE	0.9868 $\pm$ 0.0037	0.9833 $\pm$ 0.0048	0.9710 $\pm$ 0.0014	0.9902 $\pm$ 0.0004	0.5905 $\pm$ 0.0222	0.8667 $\pm$ 0.0000	0.9246 $\pm$ 0.0000	0.9609 $\pm$ 0.0099	0.9209 $\pm$ 0.0196	0.9719 $\pm$ 0.0047	0.9719 $\pm$ 0.0047	0.9719 $\pm$ 0.0047	0.5544 $\pm$ 0.0382	0.8409 $\pm$ 0.0382
TaDDPDM	0.9481 $\pm$ 0.0117	0.9416 $\pm$ 0.0128	0.8993 $\pm$ 0.0094	0.9639 $\pm$ 0.0091	0.6127 $\pm$ 0.0250	0.4933 $\pm$ 0.1968	0.9206 $\pm$ 0.0520	0.8347 $\pm$ 0.0196	0.8110 $\pm$ 0.0222	0.8708 $\pm$ 0.0418	0.8708 $\pm$ 0.0418	0.8708 $\pm$ 0.0418	0.2105 $\pm$ 0.1528	0.6878 $\pm$ 0.0311
TVAE	0.9857 $\pm$ 0.0007	0.9827 $\pm$ 0.0009	0.9697 $\pm$ 0.0016	0.9904 $\pm$ 0.0005	0.5397 $\pm$ 0.0225	0.7466 $\pm$ 0.0422	0.9728 $\pm$ 0.0007	0.9253 $\pm$ 0.0034	0.9157 $\pm$ 0.0037	0.8553 $\pm$ 0.0062	0.9511 $\pm$ 0.0021	0.9127 $\pm$ 0.0383	0.1582 $\pm$ 0.0562	0.8670 $\pm$ 0.0084
D12														
CTGAN	0.9572 $\pm$ 0.0013	0.9527 $\pm$ 0.0017	0.9188 $\pm$ 0.0033	0.9742 $\pm$ 0.0009	0.5817 $\pm$ 0.0174	0.0211 $\pm$ 0.0802	0.9374 $\pm$ 0.0032	0.9887 $\pm$ 0.0000	0.9802 $\pm$ 0.0000	0.9938 $\pm$ 0.0002	0.9938 $\pm$ 0.0002	0.6508 $\pm$ 0.0000	1.0000 $\pm$ 0.0000	0.9953 $\pm$ 0.0000
Gaussian Copula	0.9568 $\pm$ 0.0000	0.9519 $\pm$ 0.0000	0.9168 $\pm$ 0.0000	0.9736 $\pm$ 0.0000	0.6270 $\pm$ 0.0000	0.5474 $\pm$ 0.0000	0.9031 $\pm$ 0.0000	0.9648 $\pm$ 0.0000	0.9287 $\pm$ 0.0000	0.9778 $\pm$ 0.0000	0.9778 $\pm$ 0.0000	0.6349 $\pm$ 0.0000	1.0000 $\pm$ 0.0000	0.9187 $\pm$ 0.0000
K-IPO	0.9488 $\pm$ 0.0067	0.9412 $\pm$ 0.0072	0.8962 $\pm$ 0.0125	0.9620 $\pm$ 0.0027	0.6190 $\pm$ 0.0109	0.8105 $\pm$ 0.0489	0.9547 $\pm$ 0.0111	0.9918 $\pm$ 0.0011	0.9898 $\pm$ 0.0012	0.9945 $\pm$ 0.0007	0.9945 $\pm$ 0.0007	0.6508 $\pm$ 0.0000	1.0000 $\pm$ 0.0000	0.9950 $\pm$ 0.0003
SMOTE	0.9376 $\pm$ 0.0024	0.9284 $\pm$ 0.0028	0.8737 $\pm$ 0.0049	0.9552 $\pm$ 0.0013	0.5972 $\pm$ 0.0073	0.7274 $\pm$ 0.0376	0.9402 $\pm$ 0.0021	0.9905 $\pm$ 0.0003	0.9884 $\pm$ 0.0004	0.9796 $\pm$ 0.0006	0.9796 $\pm$ 0.0006	0.6508 $\pm$ 0.0000	1.0000 $\pm$ 0.0000	0.9953 $\pm$ 0.0000
TaDDPDM	0.9181 $\pm$ 0.0041	0.9067 $\pm$ 0.0047	0.8387 $\pm$ 0.0075	0.9443 $\pm$ 0.0031	0.6377 $\pm$ 0.0123	–0.0710 $\pm$ 0.1492	0.8686 $\pm$ 0.0103	0.9767 $\pm$ 0.0019	0.9720 $\pm$ 0.0021	0.9369 $\pm$ 0.0037	0.9369 $\pm$ 0.0037	0.6317 $\pm$ 0.0123	0.5000 $\pm$ 0.2539	0.8827 $\pm$ 0.0325
TVAE	0.9455 $\pm$ 0.0014	0.9383 $\pm$ 0.0019	0.8930 $\pm$ 0.0037	0.9656 $\pm$ 0.0010	0.6329 $\pm$ 0.0111	0.3968 $\pm$ 0.0884	0.9106 $\pm$ 0.0026	0.9376 $\pm$ 0.0085	0.9292 $\pm$ 0.0100	0.8769 $\pm$ 0.0178	0.8769 $\pm$ 0.0178	0.6190 $\pm$ 0.0146	0.9800 $\pm$ 0.0532	0.8810 $\pm$ 0.0061
D13														
CTGAN	0.8830 $\pm$ 0.0109	0.8687 $\pm$ 0.0120	0.7779 $\pm$ 0.0239	0.9074 $\pm$ 0.0111	0.5547 $\pm$ 0.0413	0.2857 $\pm$ 0.1918	0.7915 $\pm$ 0.0268	0.9553 $\pm$ 0.0003	0.9542 $\pm$ 0.0004	0.9898 $\pm$ 0.0006	0.9974 $\pm$ 0.0002	0.6415 $\pm$ 0.0078	0.7233 $\pm$ 0.0081	0.9314 $\pm$ 0.0137
Gaussian Copula	0.9304 $\pm$ 0.0000	0.9232 $\pm$ 0.0000	0.8726 $\pm$ 0.0000	0.9515 $\pm$ 0.0000	0.5357 $\pm$ 0.0000	0.9048 $\pm$ 0.0000	0.8647 $\pm$ 0.0000	0.9951 $\pm$ 0.0000	0.9940 $\pm$ 0.0000	0.9893 $\pm$ 0.0000	0.9971 $\pm$ 0.0000	0.6355 $\pm$ 0.0000	0.7785 $\pm$ 0.0000	0.9756 $\pm$ 0.0000
K-IPO	0.9030 $\pm$ 0.0060	0.8909 $\pm$ 0.0065	0.8052 $\pm$ 0.0142	0.9123 $\pm$ 0.0077	0.7500 $\pm$ 0.0202	0.9714 $\pm$ 0.0460	0.9279 $\pm$ 0.0042	0.9981 $\pm$ 0.0003	0.9974 $\pm$ 0.0004	0.9955 $\pm$ 0.0007	0.9985 $\pm$ 0.0002	0.6434 $\pm$ 0.0036	0.8509 $\pm$ 0.0059	0.9885 $\pm$ 0.0004
SMOTE	0.8776 $\pm$ 0.0095	0.8615 $\pm$ 0.0108	0.7582 $\pm$ 0.0181	0.8918 $\pm$ 0.0079	0.7226 $\pm$ 0.0313	0.7238 $\pm$ 0.0301	0.9279 $\pm$ 0.0045	0.9979 $\pm$ 0.0003	0.9972 $\pm$ 0.0004	0.9951 $\pm$ 0.0008	0.9984 $\pm$ 0.0002	0.6369 $\pm$ 0.0051	0.8309 $\pm$ 0.0027	0.9890 $\pm$ 0.0001
TaDDPDM	0.7553 $\pm$ 0.0167	0.6997 $\pm$ 0.0228	0.4703 $\pm$ 0.0314	0.7612 $\pm$ 0.0177	0.6976 $\pm$ 0.0347	0.5428 $\pm$ 0.1405	0.6413 $\pm$ 0.0124	0.9951 $\pm$ 0.0009	0.9941 $\pm$ 0.0010	0.9897 $\pm$ 0.0017	0.9973 $\pm$ 0.0005	0.6310 $\pm$ 0.0090	0.1289 $\pm$ 0.0575	0.7126 $\pm$ 0.0569
TVAE	0.8554 $\pm$ 0.0077	0.8393 $\pm$ 0.0085	0.7188 $\pm$ 0.0148	0.8667 $\pm$ 0.0086	0.4774 $\pm$ 0.0467	0.6190 $\pm$ 0.0000	0.6988 $\pm$ 0.0124	0.9968 $\pm$ 0.0006	0.9961 $\pm$ 0.0008	0.9932 $\pm$ 0.0014	0.9982 $\pm$ 0.0004	0.6371 $\pm$ 0.0061	0.7728 $\pm$ 0.0080	0.9869 $\pm$ 0.0001
D14														
CTGAN	0.9652 $\pm$ 0.0011	0.9618 $\pm$ 0.0011	0.9343 $\pm$ 0.0019	0.9789 $\pm$ 0.0006	0.4976 $\pm$ 0.0689	0.2534 $\pm$ 0.1929	0.9393 $\pm$ 0.0034	0.9895 $\pm$ 0.0030	0.9878 $\pm$ 0.0037	0.9786 $\pm$ 0.0065	0.9943 $\pm$ 0.0018	0.6016 $\pm$ 0.0513	0.4484 $\pm$ 0.1170	0.9865 $\pm$ 0.0056
Gaussian Copula	0.9507 $\pm$ 0.0000	0.9412 $\pm$ 0.0000	0.9029 $\pm$ 0.0000	0.9687 $\pm$ 0.0000	0.5714 $\pm$ 0.0000	1.0000 $\pm$ 0.0000	0.9159 $\pm$ 0.0000	0.9858 $\pm$ 0.0000	0.9838 $\pm$ 0.0000	0.9716 $\pm$ 0.0000	0.9928 $\pm$ 0.0000	0.5556 $\pm$ 0.0000	0.9121 $\pm$ 0.0000	0.9910 $\pm$ 0.0000
K-IPO	0.9378 $\pm$ 0.0074	0.9291 $\pm$ 0.0087	0.8765 $\pm$ 0.0155	0.9591 $\pm$ 0.0052	0.5119 $\pm$ 0.0194	1.0000 $\pm$ 0.0000	0.9509 $\pm$ 0.0007	0.9963 $\pm$ 0.0005	0.9958 $\pm$ 0.0006	0.9926 $\pm$ 0.0010	0.9981 $\pm$ 0.0003	0.6071 $\pm$ 0.0180	0.9626 $\pm$ 0.0208	0.9980 $\pm$ 0.0002
SMOTE	0.9363 $\pm$ 0.0088	0.9274 $\pm$ 0.0102	0.8735 $\pm$ 0.0179	0.9582 $\pm$ 0.0063	0.4940 $\pm$ 0.0369	0.9689 $\pm$ 0.0396	0.9506 $\pm$ 0.0068	0.9975 $\pm$ 0.0006	0.9970 $\pm$ 0.0006	0.9947 $\pm$ 0.0011	0.9986 $\pm$ 0.0003	0.5968 $\pm$ 0.0346	0.9341 $\pm$ 0.0000	0.9984 $\pm$ 0.0001
TaDDPDM	0.8169 $\pm$ 0.1286	0.7905 $\pm$ 0.1483	0.6374 $\pm$ 0.2593	0.8451 $\pm$ 0.1333	0.5536 $\pm$ 0.0470	0.7511 $\pm$ 0.0843	0.7643 $\pm$ 0.1213	0.9879 $\pm$ 0.0018	0.9860 $\pm$ 0.0019	0.9756 $\pm$ 0.0033	0.9935 $\pm$ 0.0009	0.6008 $\pm$ 0.0495	0.6989 $\pm$ 0.1294	0.9891 $\pm$ 0.0013
TVAE	0.9383 $\pm$ 0.0017	0.9299 $\pm$ 0.0019	0.8779 $\pm$ 0.0033	0.9592 $\pm$ 0.0013	0.5547 $\pm$ 0.0556	0.9156 $\pm$ 0.0328	0.8603 $\pm$ 0.0041	0.9885 $\pm$ 0.0014	0.9865 $\pm$ 0.0018	0.9763 $\pm$ 0.0032	0.9937 $\pm$ 0.0009	0.5913 $\pm$ 0.0273	0.8549 $\pm$ 0.0362	0.9898 $\pm$ 0.0007
D15														
CTGAN	0.9287 $\pm$ 0.0134	0.9201 $\pm$ 0.0153	0.8643 $\pm$ 0.0248	0.9532 $\pm$ 0.0102	0.5143 $\pm$ 0.1010	0.3378 $\pm$ 0.1820	0.8203 $\pm$ 0.0363	0.9977 $\pm$ 0.0004	0.9975 $\pm$ 0.0004	0.9956 $\pm$ 0.0007	0.9988 $\pm$ 0.0002	0.6323 $\pm$ 0.0311	0.4083 $\pm$ 0.1550	0.9928 $\pm$ 0.0010
Gaussian Copula	0.9481 $\pm$ 0.0030	0.8488 $\pm$ 0.0094	0.7535 $\pm$ 0.0090	0.9608 $\pm$ 0.0000	0.6508 $\pm$ 0.0000	0.7778 $\pm$ 0.0000	0.7734 $\pm$ 0.0000	0.9981 $\pm$ 0.0000	0.9977 $\pm$ 0.0000	0.9960 $\pm$ 0.0000	0.9980 $\pm$ 0.0000	0.6349 $\pm$ 0.0000	0.6500 $\pm$ 0.0000	0.9979 $\pm$ 0.0000
K-IPO	0.931 $\pm$ 0.0038	0.9453 $\pm$ 0.0047	0.9009 $\pm$ 0.0024	0.9893 $\pm$ 0.0004	0.6746 $\pm$ 0.0135	0.9697 $\pm$ 0.0363	0.9694 $\pm$ 0.0013	1.0000 $\pm$ 0.0000	1.0000 $\pm$ 0.0000	1.0000 $\pm$ 0.0000	1.0000 $\pm$ 0.0000	0.6344 $\pm$ 0.0008	0.9600 $\pm$ 0.0005	0.9979 $\pm$ 0.0000
SMOTE	0.9750 $\pm$ 0.0015	0.9659 $\pm$ 0.0023	0.9109 $\pm$ 0.0023	0.9859 $\pm$ 0.0004	0.6693 $\pm$ 0.0117	0.8133 $\pm$ 0.1061	0.9289 $\pm$ 0.0000	0.9999 $\pm$ 0.0001	0.9999 $\pm$ 0.0001	0.9999 $\pm$ 0.0001	0.9999 $\pm$ 0.0001	0.6349 $\pm$ 0.0000	0.9600 $\pm$ 0.0005	0.9993 $\pm$ 0.0001
TaDDPDM	0.8830 $\pm$ 0.0015	0.8676 $\pm$ 0.0052	0.7739 $\pm$ 0.0091	0.9154 $\pm$ 0.0034	0.6693 $\pm$ 0.0117	0.8133 $\pm$ 0.1061	0.7925 $\pm$ 0.0033	0.9909 $\pm$ 0.0001	0.9899 $\pm$ 0.0001	0.9899 $\pm$ 0.0001	0.9929 $\pm$ 0.0006	0.6503 $\pm$ 0.0326	0.6817 $\pm$ 0.1026	0.9806 $\pm$ 0.0021
TVAE	0.9723 $\pm$ 0.0014	0.9696 $\pm$ 0.0016	0.9476 $\pm$ 0.0028	0.9844 $\pm$ 0.0009	0.6699 $\pm$ 0.0164	0.5556 $\pm$ 0.0983	0.9441 $\pm$ 0.0016	0.9979 $\pm$ 0.0006	0.9976 $\pm$ 0.0006	0.9959 $\pm$ 0.0011	0.9988 $\pm$ 0.0003	0.6455 $\pm$ 0.0112	0.9883 $\pm$ 0.0226	0.9963 $\pm$ 0.0005

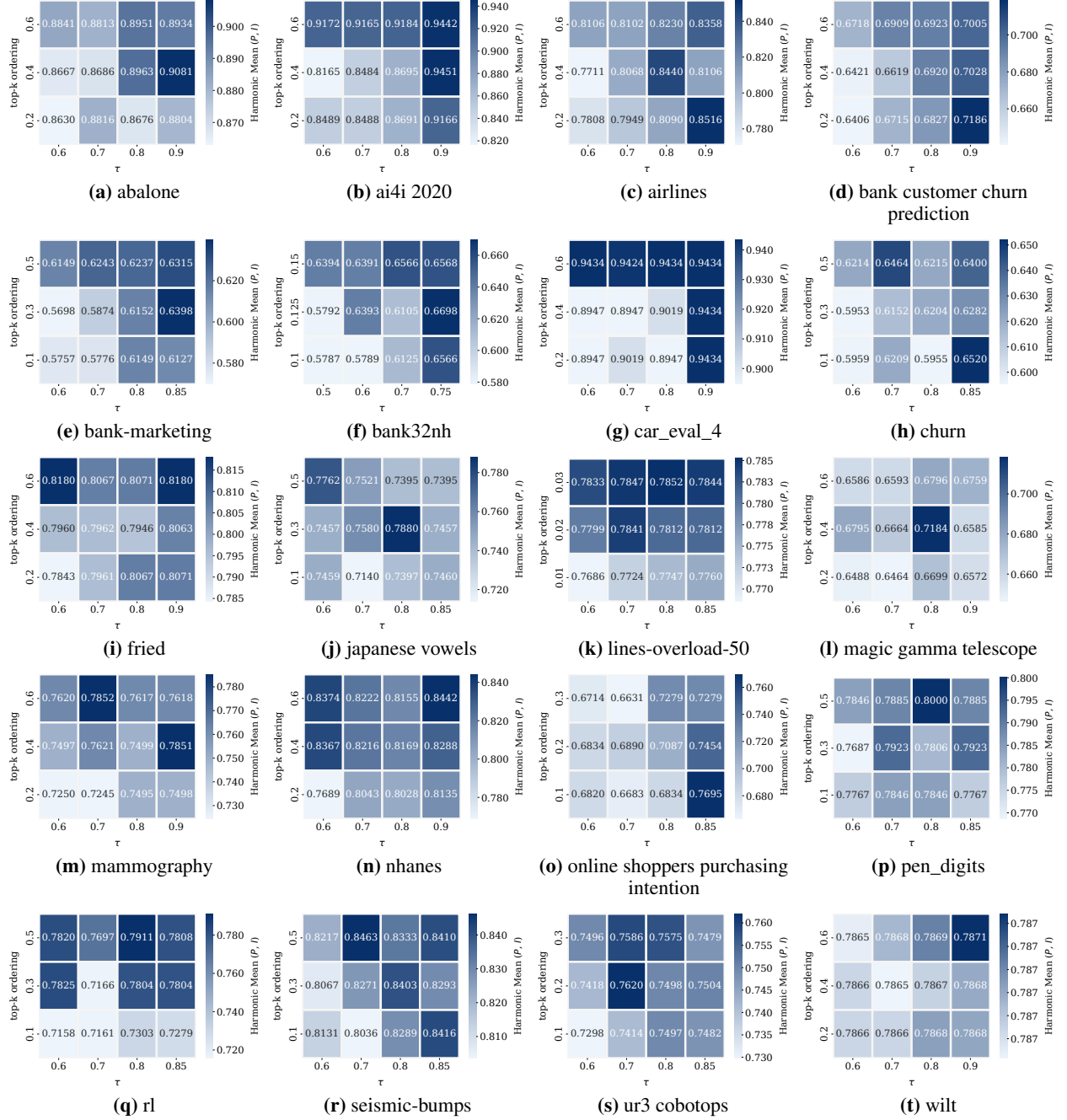


Figure 3: Harmonic mean $H(P, I)$ heatmaps over $(\tau, \text{top-}k \text{ ordering})$ for the benchmark datasets. The decimal values in the top- k ordering axis denote the proportion of top-ranked features whose ordering is constrained, e.g., 0.2 corresponds to the top 20% of features.

The interpretability results provide stronger evidence for the main contribution of K-IPO. K-IPO achieves the best or tied-best Kendall's τ on all 20 datasets, substantially outperforming other methods in terms of feature importance rank preservation. It also achieves the highest explainability consistency score I in 15 out of 20 datasets, whereas the closest competitor, TabDDPM, achieves only 3 wins. These results support the main premise of this study: synthetic samples that appear useful from a predictive or distributional perspective may still alter the feature importance structure underlying model decisions when no explicit importance preservation constraint is imposed. In contrast, K-IPO directly controls the source of the explanation drift through its Kendall-based acceptance mechanism.

Table 7: Number of wins achieved by each method across all datasets and selected evaluation metrics.

Method	Bal. Acc.	F1	MCC	P	I	Kendall's τ	$1 - N3$
CTGAN	4	4	4	4	2	1	1
Gaussian Copula	1	1	1	1	1	3	0
K-IPO	9	9	9	9	15	20	13
SMOTE	2	2	3	3	1	1	5
TabDDPM	5	5	5	5	3	0	3
TVAE	0	0	0	0	0	0	0

The separability results further support this conclusion. K-IPO achieved the best or tied-best $1 - N3$ value in 13 out of 20 datasets, surpassing SMOTE, which ranked second with 5 wins, and TabDDPM with 3 wins. This suggests that the proposed acceptance mechanism tends to reject synthetic samples that would blur class boundaries or increase the overlap between minority and majority regions. This behavior explains why K-IPO can preserve interpretability without sacrificing predictive performance. By selecting only candidate samples that maintain the original importance structure, this method tends to preserve a more favorable class geometry.

Table 7 summarizes the number of wins achieved by each method across all datasets and evaluation metrics. K-IPO clearly dominates the interpretability-related metrics, recording 15 wins for I and 20 for Kendall's τ , far exceeding all competing methods. It also achieved the strongest class separability, with 13 wins for $1 - N3$. K-IPO also leads across the predictive metrics, with 9 wins each in Balanced Accuracy, F1-score, MCC, and P . TabDDPM was the closest predictive competitor, achieving 5 wins in each metric, whereas TVAE recorded no wins in any category. Overall, K-IPO is the only method that consistently combines strong predictive performance with explanation consistency, rank preservation, and class separability.

5.3 Computational Overhead

Regarding the computational cost, Fig. 4 shows the mean elapsed time for each dataset and method, while Table 8 summarizes the average and median wall-clock times across all datasets. The results revealed clear differences in computational efficiency among method families. Gaussian Copula achieved the lowest overall mean runtime at 0.561 s, followed by SMOTE at 1.082 s. This is expected, as Gaussian Copula relies on a lightweight statistical fit-and-sample procedure, whereas SMOTE generates synthetic minority samples through nearest-neighbor interpolation without training a neural generative model. In contrast, CTGAN, TVAE, and TabDDPM require an explicit model-fitting stage before sampling: CTGAN trains an adversarial network with conditional generation and training-by-sampling, TVAE trains a variational autoencoder over multiple epochs, and TabDDPM trains a diffusion-based generator. Therefore, their runtimes reflect both model learning and sampling costs, which explains their substantially higher computational burden.

K-IPO had the highest mean runtime (9.605 s), primarily due to the repeated feature importance evaluations required by the Kendall-constrained acceptance mechanism. For each candidate block, the resulting feature importance ranking is compared with the reference ranking, and rejected blocks are recursively split into smaller chunks before being accepted or discarded. Thus, K-IPO incurs a computational cost absent from the unconstrained baseline methods. However, the median runtime provides a more nuanced picture. The median runtime of K-IPO was 3.184 s, which was close to those of TVAE (2.768 s) and CTGAN (2.808 s), and lower than that of TabDDPM (4.316 s). Therefore, K-IPO is not consistently slower than neural or diffusion-based generators. Instead, its higher mean runtime is mainly driven by a small number of complex datasets, where the acceptance mechanism becomes highly selective, leading to repeated block rejection and recursive splitting. This pattern is further illustrated in Fig. 4, where K-IPO exhibited moderate runtimes on several datasets but showed sharp increases in specific cases.

Importantly, K-IPO can be faster than deep generative baselines on datasets where valid candidate blocks are accepted early. In such cases, the expensive training phase required by CTGAN, TVAE, and TabDDPM is avoided, with costs mainly limited to local candidate generation and feature importance verification. In contrast, deep generative methods must first fit a model, even when only a small number of synthetic samples are needed to rebalance the minority class. Thus, the relative runtime of K-IPO depends on the strictness of the importance preservation constraint, the quality of the candidate pool, the number of required synthetic samples, and dataset dimensionality. Overall, although K-IPO introduces additional computational costs, this overhead remains acceptable, given its gains in predictive performance, explainability consistency, feature importance rank preservation, and class separability.

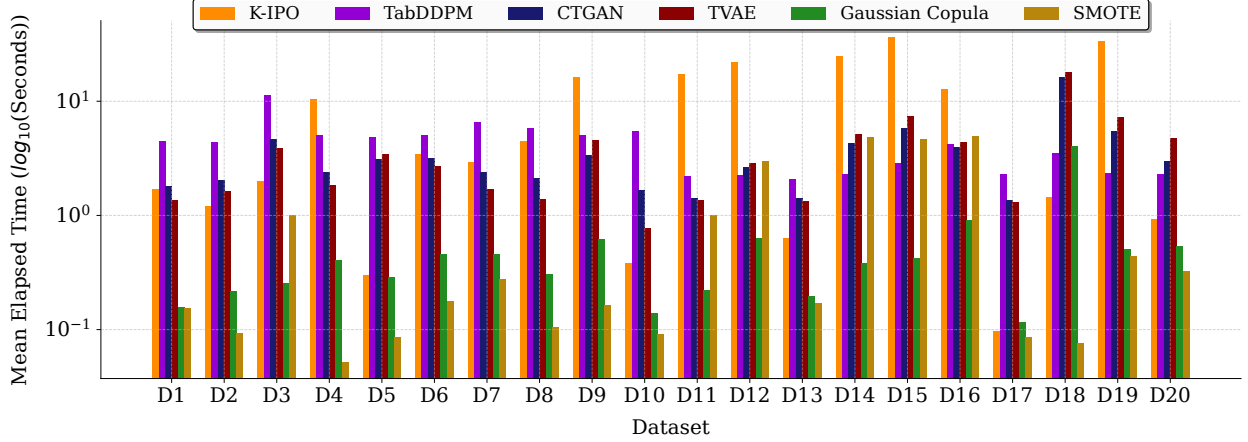


Figure 4: Runtime comparison by dataset and oversampling method on a logarithmic scale.

5.4 Statistical Analysis

To complement the descriptive evaluation, we performed a non-parametric statistical analysis across the examined datasets and competing methods. Following the standard protocol for comparing multiple algorithms over multiple datasets [65], the methods were ranked within each dataset for each evaluation metric, with lower ranks indicating better performance. We then applied the Friedman test [66] to assess whether the observed rank differences were statistically significant. When the null hypothesis was rejected, we conducted the Nemenyi post-hoc test [67] to identify the methods whose average ranks did not differ significantly.

Table 9 presents the Friedman test statistics and the corresponding p -values. At $\alpha = 0.05$, the null hypothesis of equal performance across the baseline methods was rejected for all examined metrics, indicating statistically significant differences among the methods. This pattern holds for predictive metrics, including Balanced Accuracy, F1-score, MCC, and the predictive score P , as well as for interpretability-oriented metrics. The strongest effects were observed for explainability consistency I , Kendall’s τ , and separability $1 - N3$. Overall, the results show that the choice of synthetic data generation method significantly influences both predictive performance and interpretability.

The Nemenyi critical difference diagrams in Fig. 5 offer a more detailed view of the pairwise post-hoc comparisons. For the predictive metrics, K-IPO achieved the best average rank for Balanced Accuracy, F1-score, MCC, and P . However, these differences were not always significant enough to statistically distinguish K-IPO from all competing methods. Specifically, CTGAN, TVAE, SMOTE, and TabDDPM remained statistically comparable to K-IPO for some predictive metrics, indicating that unconstrained oversampling and generative methods can still deliver competitive classification performance on specific datasets.

A clearer pattern emerged for the interpretability-related metrics. For explainability consistency I , K-IPO achieved the best average rank by a wide margin (1.70) and formed a distinct top group, demonstrating statistically significant superiority over all competing baselines. The substantial gap from the remaining methods, whose average ranks exceeded 3.4, indicates that the proposed Kendall-constrained acceptance mechanism markedly improved the alignment between feature-attribution explanations and the feature importance ranking of the original data. K-IPO also obtained the best average rank for Kendall’s τ (1.12), reflecting the strongest preservation of the original feature importance ordering. The post-hoc analysis ranked K-IPO at the top, clearly separating it from lower-performing methods. Although

Table 8: Summary of computational cost across oversampling methods.

Method	Mean time (s)	Median time (s)
SMOTE	1.082	0.173
Gaussian Copula	0.561	0.393
TVAE	3.816	2.768
CTGAN	3.606	2.808
K-IPO	9.605	3.184
TabDDPM	4.199	4.316

Table 9: Friedman test results for the examined datasets and oversampling methods.

Metric	Friedman statistic	p -value
Balanced Accuracy	20.5651	1×10^{-3}
F1-score	19.9213	1.3×10^{-3}
MCC	17.3605	3.9×10^{-3}
P	14.1702	1.46×10^{-3}
I	27.8674	3.86×10^{-5}
Kendall’s τ	72.4090	4.21×10^{-14}
Separability ($1 - N3$)	56.7024	5.82×10^{-11}

the Gaussian Copula and SMOTE constitute the closest competing group, their average rank (2.70) remains substantially worse, highlighting K-IPO’s superior ability to maintain feature importance consistency. This finding aligns directly with the design objective of K-IPO, where synthetic samples are accepted only if they satisfy an explicit rank-preservation criterion, whereas competing methods do not directly constrain feature importance drift.

The separability results further highlight the benefits of our approach. K-IPO achieved the best average rank for $1 - N3$ (1.40) and was statistically comparable only to the SMOTE-based baseline (2.08), while clearly outperforming all other methods, whose average ranks ranged from 3.95 to 4.80. This indicates that the proposed acceptance mechanism effectively filters out synthetic samples that would increase class overlap or distort the local neighborhood structure of the data. As a result, K-IPO preserves class separability more effectively while maintaining the interpretability characteristics of the original dataset.

Finally, the critical difference diagram for the harmonic mean of predictive performance and interpretability provides the strongest overall evidence in favor of K-IPO. K-IPO achieved the best average rank (1.60) and was not connected to any competing method, indicating its statistically significant superiority over all baseline methods. This result shows that K-IPO does not sacrifice interpretability for predictive utility; instead, it offers the most effective overall trade-off between these two objectives. Thus, although some baselines remain competitive in individual predictive metrics, K-IPO is the only method that consistently combines strong predictive performance, explanation consistency, feature importance rank preservation, and class separability.

6 Conclusions

This paper introduced Kendall-constrained Importance-Preserving Oversampling (K-IPO), a generator-agnostic framework for imbalanced tabular classification that explicitly controls feature importance drift during augmentation. Unlike conventional oversampling methods, which focus primarily on class balance, sample plausibility, or predictive performance, K-IPO follows a generate-then-select strategy, accepting candidate minority-class samples only when their inclusion sufficiently preserves the feature importance ranking of the original data. By doing so, K-IPO incorporates explanation stability directly into the augmentation process, rather than treating it solely as a post-hoc evaluation criterion.

Experiments across 20 binary imbalanced datasets, three predictive models, and multiple explanation methods showed that K-IPO achieved the best or tied-best Kendall rank preservation on all examined datasets, the highest explainability-consistency score on 15 datasets, and the strongest class-separability result on 13 datasets. It also remained competitive in predictive performance, recording the most wins in Balanced Accuracy and leading in F1-score, MCC, and the composite predictive score. These findings indicate that explicitly preserving feature importance ranking can improve explanation consistency without systematically compromising classification performance.

Future research should address the computational cost of K-IPO, particularly for large datasets, where repeated feature importance evaluations and candidate subset assessments may limit scalability. Promising directions include parallel implementations for both shared and distributed memory computing environments, faster candidate screening mechanisms, and data-driven methods for the automatic selection of appropriate preservation constraints. The framework could also be extended through uncertainty-aware rank constraints, acceptance criteria that preserve local explanations, and support for multiclass and multilabel classification. Finally, broader evaluations should examine whether the generated samples protect sensitive information, avoid reinforcing unfair biases, preserve causal relationships, and satisfy domain-specific validity requirements.

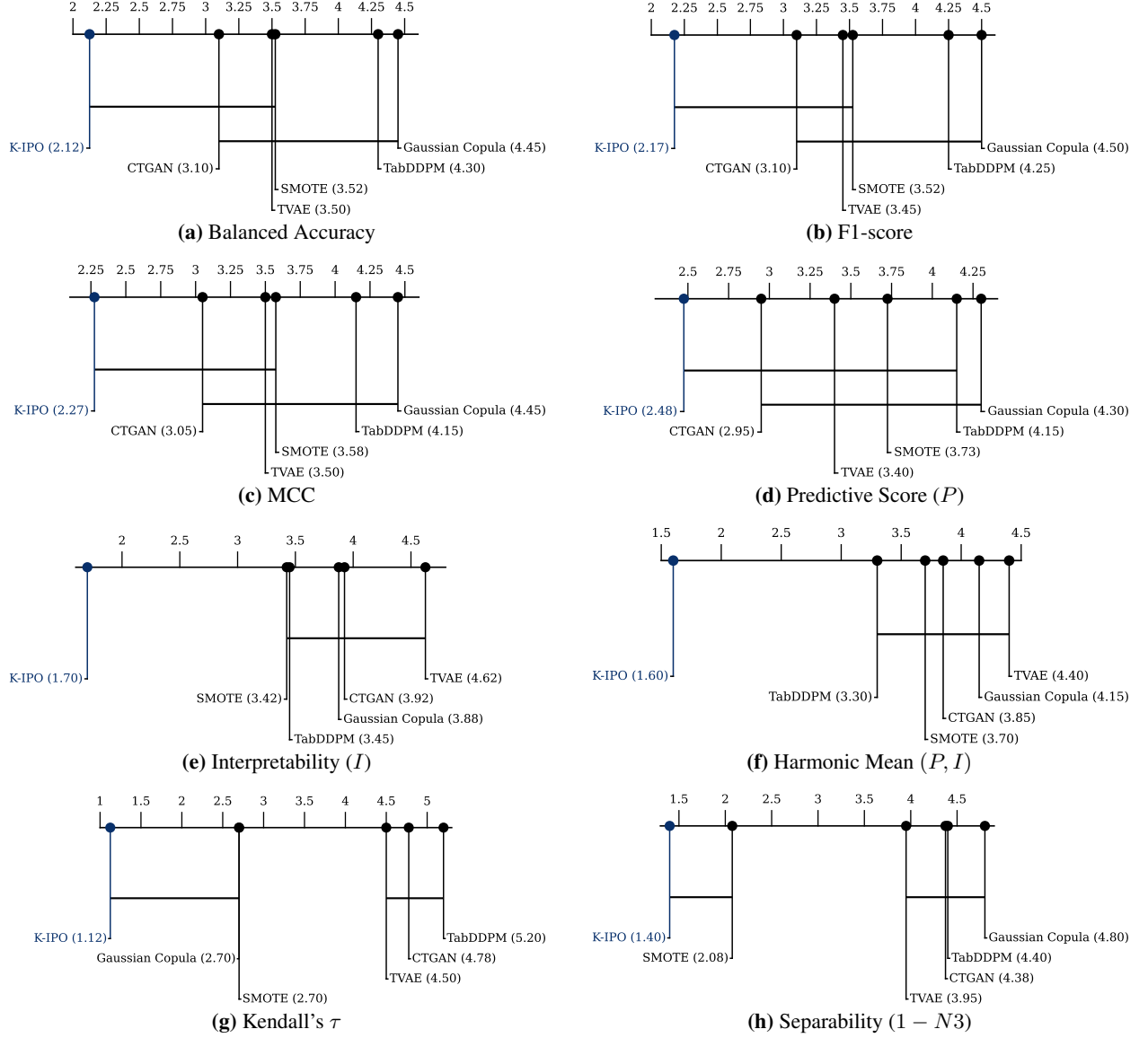


Figure 5: Critical difference diagrams from Nemenyi post-hoc tests on predictive, interpretability, and structural metrics. Lower average ranks indicate better performance; methods connected by a horizontal bar are not significantly different at $\alpha = 0.05$.

References

- [1] Wuxing Chen, Kaixiang Yang, Zhiwen Yu, Yifan Shi, and C. L. Philip Chen. A survey on imbalanced learning: latest research, applications and future directions. *Artif. Intell. Rev.*, 57(6):137, 2024.
- [2] Haibo He and Edwardo A. Garcia. Learning from imbalanced data. *IEEE Trans. Knowl. Data Eng.*, 21(9):1263–1284, 2009.
- [3] Jaber Qezelbash-Chamak, Karen Hicklin, and Minhee Kim. KANBalance: Kolmogorov–Arnold network mitigates class imbalance. *Pattern Recognit.*, 171:112325, 2026.
- [4] Charles Elkan. The foundations of cost-sensitive learning. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI-01)*, volume 2, pages 973–978. Morgan Kaufmann, 2001.
- [5] Zhi-Hua Zhou and Xu-Ying Liu. Training cost-sensitive neural networks with methods addressing the class imbalance problem. *IEEE Trans. Knowl. Data Eng.*, 18(1):63–77, 2006.

- [6] Xu-Ying Liu, Jianxin Wu, and Zhi-Hua Zhou. Exploratory undersampling for class-imbalance learning. *IEEE Trans. Syst. Man Cybern. B Cybern.*, 39(2):539–550, 2009.
- [7] Lingyun Zhao, Fei Han, Qinghua Ling, Yubin Ge, Yuze Zhang, Qing Liu, and Henry Han. Contribution-based imbalanced hybrid resampling ensemble. *Pattern Recognit.*, 164:111553, 2025.
- [8] Gustavo E. A. P. A. Batista, Ronaldo C. Prati, and Maria Carolina Monard. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Explor. Newsl.*, 6(1):20–29, 2004.
- [9] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. SMOTE: synthetic minority over-sampling technique. *J. Artif. Intell. Res.*, 16:321–357, 2002.
- [10] Agnideep Aich, Md Monzur Murshed, Bruce Wade, and Sameera Hewage. CopulaSMOTE: A copula-based oversampling approach for imbalanced classification in diabetes prediction. *arXiv*, 2025.
- [11] Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Modeling tabular data using conditional GAN. In *Advances in Neural Information Processing Systems*, volume 32, pages 7333–7343, 2019.
- [12] Akim Kotelnikov, Dmitry Baranchuk, Ivan Rubachev, and Artem Babenko. TabDDPM: modelling tabular data with diffusion models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 17564–17579. PMLR, 2023.
- [13] Kristian Schultz, Saptarshi Bej, Waldemar Hahn, Markus Wolfien, Prashant Srivastava, and Olaf Wolkenhauer. ConvGeN: a convex space learning approach for deep-generative oversampling and imbalanced classification of small tabular datasets. *Pattern Recognit.*, 147:110138, 2024.
- [14] Lasse Hansen, Nabeel Seedat, Mihaela van der Schaar, and Andrija Petrovic. Reimagining synthetic tabular data generation through data-centric AI: a comprehensive benchmark. In *Advances in Neural Information Processing Systems*, volume 36, pages 33781–33823, 2023.
- [15] Andrew Slavin Ross, Michael C. Hughes, and Finale Doshi-Velez. Right for the right reasons: Training differentiable models by constraining their explanations. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI-17)*, pages 2662–2670, 2017.
- [16] Chaithra Umesh, Kristian Schultz, Manjunath Mahendra, Saptarshi Bej, and Olaf Wolkenhauer. Preserving logical and functional dependencies in synthetic tabular data. *Pattern Recognit.*, 163:111459, 2025.
- [17] Oscar Giles, Kasra Hosseini, Grigorios Mingas, Oliver Strickson, Louise Bowler, Camila Rangel Smith, Harrison Wilde, Jen Ning Lim, Bilal Mateen, Kasun Amarasinghe, Rayid Ghani, Alison Heppenstall, Nik Lomax, Nick Malleson, Martin O’Reilly, and Sebastian Vollmert. Faking feature importance: A cautionary tale on the use of differentially-private synthetic data. *arXiv*, 2022.
- [18] Henny Leidiyana. Analisis dampak SMOTE terhadap feature importance pada klasifikasi data migraine menggunakan Random Forest dan Extra Trees. *J. Komtika (Komputasi Inform.)*, 10(1):72–82, 2026.
- [19] Jeremy Goldwasser and Giles Hooker. Statistical significance of feature importance rankings. In *Proceedings of the Forty-first Conference on Uncertainty in Artificial Intelligence*, volume 286 of *Proceedings of Machine Learning Research*, pages 1476–1496. PMLR, 2025.
- [20] Ke Yu, Shigeru Ishikura, Yukari Usukura, Yuki Shigoku, and Teruaki Hayashi. SHAP distance: an explainability-aware metric for evaluating the semantic fidelity of synthetic tabular data. In *2025 IEEE International Conference on Big Data (BigData)*, pages 3132–3141. IEEE, 2025.
- [21] Jan Kapar, Niklas Koenen, and Martin Jullum. What’s wrong with your synthetic tabular data? using explainable AI to evaluate generative models. In *Explainable Artificial Intelligence*, volume 2578 of *Communications in Computer and Information Science*, pages 19–43, Cham, 2026. Springer.
- [22] M. G. Kendall. A new measure of rank correlation. *Biometrika*, 30(1–2):81–93, 1938.
- [23] Dina Elreedy, Amir F. Atiya, and Firuz Kamalov. A theoretical distribution analysis of synthetic minority oversampling technique (SMOTE) for imbalanced learning. *Mach. Learn.*, 113(7):4903–4923, 2024.
- [24] Hui Han, Wen-Yuan Wang, and Bing-Huan Mao. Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. In *Advances in Intelligent Computing*, volume 3644 of *Lecture Notes in Computer Science*, pages 878–887, Berlin, Heidelberg, 2005. Springer.
- [25] Haibo He, Yang Bai, Edwardo A. Garcia, and Shutao Li. ADASYN: adaptive synthetic sampling approach for imbalanced learning. In *2008 IEEE International Joint Conference on Neural Networks (IJCNN)*, pages 1322–1328. IEEE, 2008.
- [26] Fatih Sağlam and Mehmet Ali Cengiz. A novel SMOTE-based resampling technique through noise detection and the boosting procedure. *Expert Syst. Appl.*, 200:117023, 2022.

- [27] Georgios Douzas, Fernando Bacao, and Felix Last. Improving imbalanced learning through a heuristic oversampling method based on k-means and SMOTE. *Inf. Sci.*, 465:1–20, 2018.
- [28] Joseph Feldman and Daniel R. Kowal. Nonparametric copula models for multivariate, mixed, and missing data. *J. Mach. Learn. Res.*, 25(164):1–50, 2024.
- [29] David Huk, Mark Steel, and Ritabrata Dutta. Your copula is a classifier in disguise: classification-based copula density estimation. In *Proceedings of the 28th International Conference on Artificial Intelligence and Statistics*, volume 258 of *Proceedings of Machine Learning Research*, pages 3790–3798. PMLR, 2025.
- [30] Ruxue Shi, Yili Wang, Mengnan Du, Xu Shen, Yi Chang, and Xin Wang. A comprehensive survey of synthetic tabular data generation. *arXiv*, 2025.
- [31] Ramiro D. Camino, Radu State, and Christian A. Hammerschmidt. Oversampling tabular data with deep generative models: is it worth the effort? In *Proceedings on “I Can’t Believe It’s Not Better!” at NeurIPS Workshops*, volume 137 of *Proceedings of Machine Learning Research*, pages 148–157. PMLR, 2020.
- [32] Justin Engelmann and Stefan Lessmann. Conditional Wasserstein GAN-based oversampling of tabular data for imbalanced learning. *Expert Syst. Appl.*, 174:114582, 2021.
- [33] Annie D’souza, Swetha M, and Sunita Sarawagi. Synthetic tabular data generation for imbalanced classification: the surprising effectiveness of an overlap class. *Proc. AAAI Conf. Artif. Intell.*, 39(15):16127–16134, 2025.
- [34] Michael Platzter and Thomas Reutterer. Holdout-based empirical assessment of mixed-type synthetic data. *Front. Big Data*, 4:679939, 2021.
- [35] Anton D. Lautrup, Tobias Hyrup, Arthur Zimek, and Peter Schneider-Kamp. Syntheval: a framework for detailed utility and privacy evaluation of tabular synthetic data. *Data Min. Knowl. Discov.*, 39(1):6, 2025.
- [36] Mikel Hernandez, Pablo A. Osorio-Marulanda, Mikel Catalina, Lorea Loinaz, Gorka Epelde, and Naiara Aginako. Comprehensive evaluation framework for synthetic tabular data in health: fidelity, utility and privacy analysis of generative models with and without privacy guarantees. *Front. Digit. Health*, 7:1576290, 2025.
- [37] Adrian Stando, Mustafa Cavus, and Przemyslaw Biecek. The effect of balancing methods on model behavior in imbalanced classification problems. In *Proceedings of the Fifth International Workshop on Learning with Imbalanced Domains: Theory and Applications*, volume 241 of *Proceedings of Machine Learning Research*, pages 16–30. PMLR, 2024.
- [38] R. A. Fisher. Statistical methods for research workers. In *Breakthroughs in Statistics: Methodology and Distribution*, Springer Series in Statistics, pages 66–70. Springer, New York, NY, 1992.
- [39] Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, volume 30, pages 4765–4774, 2017.
- [40] Leo Breiman. Random forests. *Mach. Learn.*, 45(1):5–32, 2001.
- [41] Ghada Elkhawaga, Omar M. Elzeki, Mervat Abu-Elkheir, and Manfred Reichert. Why should I trust your explanation? an evaluation approach for XAI methods applied to predictive process monitoring results. *IEEE Trans. Artif. Intell.*, 5(4):1458–1472, 2024.
- [42] Marios Tyrovolas, Chrysostomos Stylios, Khurshid Aliev, and Dario Antonelli. Leveraging information flow-based fuzzy cognitive maps for interpretable fault diagnosis in industrial robotics. In *Technological Innovation for Human-Centric Systems*, volume 716 of *IFIP Advances in Information and Communication Technology*, pages 98–110, Cham, 2024. Springer.
- [43] Konstantinos Parginos, Marios Tyrovolas, Ricardo J. Bessa, X. San Liang, Chrysostomos Stylios, Simon Camal, and George Kariniotakis. Interpretable power grid overload detection with information flow-based fuzzy cognitive maps. *CSEE J. Power Energy Syst.*, pages 1–12, 2025.
- [44] Tin Kam Ho and Mitra Basu. Complexity measures of supervised classification problems. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(3):289–300, 2002.
- [45] Mahesh Kumbhar, Sunith Bandaru, and Alexander Karlsson. Imbalanced data oversampling through subspace optimization with bayesian reinforcement. *Artif. Intell. Rev.*, 59(1):1, 2026.
- [46] Zebin Yang, Aijun Zhang, and Agus Sudjianto. GAMI-Net: an explainable neural network based on generalized additive models with structured interactions. *Pattern Recognit.*, 120:108192, 2021.
- [47] Agus Sudjianto, Aijun Zhang, Zebin Yang, Yu Su, and Ningzhou Zeng. PiML toolbox for interpretable machine learning model development and diagnostics. *arXiv*, 2023.
- [48] Isel Grau and Gonzalo Nápoles. Sparseness-optimized feature importance. In *Explainable Artificial Intelligence*, volume 2154 of *Communications in Computer and Information Science*, pages 393–415, Cham, 2024. Springer.

- [49] P.E. Hadjidoukas, A. Bartezzaghi, F. Scheidegger, R. Istrate, C. Bekas, and A.C.I. Malossi. torcpy: Supporting task parallelism in python. *SoftwareX*, 12:100517, 2020.
- [50] Ghada Elkhawaga, Omar Elzeki, Mervat Abuelkheir, and Manfred Reichert. Evaluating explainable artificial intelligence methods based on feature elimination: a functionality-grounded approach. *Electronics*, 12(7):1670, 2023.
- [51] Ranit Das, Gregor Kasieczka, and David Shih. Feature selection with distance correlation. *Phys. Rev. D*, 109(5), 2024.
- [52] R. Battiti. Using mutual information for selecting features in supervised neural net learning. *IEEE Trans. Neural Netw.*, 5(4):537–550, 1994.
- [53] Xiaofei He, Deng Cai, and Partha Niyogi. Laplacian score for feature selection. In *Advances in Neural Information Processing Systems*, volume 18, pages 507–514, 2005.
- [54] Igor Kononenko. Estimating attributes: analysis and extensions of RELIEF. In *Machine Learning: ECML-94*, volume 784 of *Lecture Notes in Computer Science*, pages 171–182, Berlin, Heidelberg, 1994. Springer.
- [55] Leo Breiman, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone. *Classification and Regression Trees*. Wadsworth, Belmont, CA, 1984.
- [56] Howard Hua Yang and John Moody. Data visualization and feature selection: new algorithms for nongaussian data. In *Advances in Neural Information Processing Systems*, volume 12, pages 687–693, 1999.
- [57] Jun Wang, Jin-Mao Wei, Zhenglu Yang, and Shu-Qin Wang. Feature selection by maximizing independent classification information. *IEEE Trans. Knowl. Data Eng.*, 29(4):828–841, 2017.
- [58] Mingming Han, Husheng Guo, and Wenjian Wang. A new data complexity measure for multi-class imbalanced classification tasks. *Pattern Recognit.*, 157:110881, 2025.
- [59] Alex X. Wang, Stefanka S. Chukova, Colin R. Simpson, and Binh P. Nguyen. Challenges and opportunities of generative models on tabular data. *Appl. Soft Comput.*, 166:112223, 2024.
- [60] Dong-Hyun Won, Kwang-Seong Shin, and Sungkwan Youm. Synthetic data augmentation for imbalanced tabular data: a comparative study of generation methods. *Electronics*, 15(4):883, 2026.
- [61] Mozafar Hayaean Shirvan, Mohammad Hossein Moattar, and Mehdi Hosseinzadeh. Deep generative approaches for oversampling in imbalanced data classification problems: a comprehensive review and comparative analysis. *Appl. Soft Comput.*, 170:112677, 2025.
- [62] Yuhta Takida, Wei-Hsiang Liao, Chieh-Hsin Lai, Toshimitsu Uesaka, Shusuke Takahashi, and Yuki Mitsufuji. Preventing oversmoothing in VAE via generalized variance parameterization. *Neurocomputing*, 509:137–156, 2022.
- [63] A. Kiran and S. Saravana Kumar. A comparative analysis of GAN and VAE based synthetic data generators for high dimensional, imbalanced tabular data. In *2023 2nd International Conference for Innovation in Technology (INOCON)*, pages 1–6. IEEE, 2023.
- [64] Neha Patki, Roy Wedge, and Kalyan Veeramachaneni. The synthetic data vault. In *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 399–410, 2016.
- [65] Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.*, 7(1):1–30, 2006.
- [66] Milton Friedman. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *J. Am. Stat. Assoc.*, 32(200):675–701, 1937.
- [67] Peter Bjorn Nemenyi. *Distribution-free multiple comparisons*. PhD thesis, Princeton University, Princeton, NJ, USA, 1963.