# A Green AI Perspective on Data-Centric Approaches for Imbalanced Classification


Bárbara Nóbrega Galiza
Master in Data Science
Department of Computer Science
Faculty of Sciences of the University of Porto
2026

# A Green AI Perspective on Data-Centric Approaches for Imbalanced Classification

Bárbara Nóbrega Galiza

Dissertation carried out as part of the Master in Data Science

Department of Computer Science

2026

**Supervisor**
Rita P. Ribeiro, Assistant Professor, Faculty of Sciences of the University of Porto

**Co-supervisor**
Miriam Seoane Santos, Assistant Professor, Faculty of Sciences of the University of Porto

*“ It’s just a spark, but it’s enough*

*To keep me going ”*

Paramore

*Em memória das vítimas dos terremotos na Venezuela de 2026.*

# Acknowledgements

Em primeiro lugar, agradeço à Professora Rita Ribeiro pela confiança depositada em mim para elaborar e desenvolver esse tema, pelo qual tenho grande apreço. Desde a escolha do mestrado, tinha como objetivo fazer uma dissertação que contribuísse para a causa da sustentabilidade e para a sociedade como um todo, e fico muito feliz em ver que consegui fazê-lo. Agradeço também à Professora Miriam pelo interesse em embarcar nessa jornada e pela dedicação em participar de todos os momentos. Agradeço às professoras por me ajudarem a manter o rumo certo para a realização de um bom trabalho e pela disponibilidade, paciência e apoio incessantes durante todo o desenvolvimento dessa dissertação.

Agradeço aos meus pais, Gentil e Christine, e ao meu irmão Tiago por todo o amor e carinho dados não só ao longo desse percurso, mas durante toda a minha vida. Aos meus pais, agradeço pela oportunidade de alcançar essa conquista acadêmcia em Portugal e por seguirem me apoiando mesmo quando a saudade apertou. Mesmo de longe, o amor de vocês foi sentido fortemente ao longo desses cinco anos. Ao meu irmão, agradeço pelo suporte dado de perto durante os últimos dois anos e pelo companheirismo no dia a dia.

Agradeço aos meus amigos de Fortaleza, que estiveram sempre presentes, independentemente da distância, e aos meus amigos e colegas de curso, que me fizeram sentir menos sozinha durante meu percurso acadêmico, tanto em Aveiro como no Porto. Obrigada por fazerem meus dias mais felizes.

Agradeço à Verónica pela companhia constante e por todo o suporte durante essa jornada. Obrigada por me ajudar a levar a vida com mais leveza, por sempre acreditar em mim e por sempre me motivar a seguir em frente.

Por fim, agradeço a Deus por todas as bênçãos na minha vida e por ter colocado essas e tantas outras pessoas especiais no meu caminho.

# Resumo

Os avanços recentes na área de inteligência artificial têm levado ao aumento das exigências computacionais, despertando a preocupação dos investigadores acerca do consequente impacto ambiental. Em resposta a esse problema, a área denominada como *green AI* (IA sustentável) surgiu com o objetivo de tornar os sistemas de inteligência artificial mais sustentáveis. Uma linha de investigação promissora é a *data-centric green AI* (IA sustentável centrada em dados) cujo foco é a manipulação dos dados a fim de tornar o processo de aprendizagem automática mais eficiente. No entanto, a literatura relativa a abordagens centradas em dados aplicadas à classificação desbalanceada permanece limitada, e não existem estudos prévios que comparem múltiplos métodos de reamostragem em termos de consumo energético. Nesta dissertação, investigamos o compromisso entre energia e desempenho de métodos de manipulação dos dados para classificação desbalanceada e analisamos como a complexidade dos dados influencia essa relação utilizando múltiplos conjuntos de dados de referência. Utilizamos vários métodos tradicionais já estabelecidos na literatura, além de um método baseado em aprendizagem profunda. Os resultados mostram que, embora exista frequentemente uma associação negativa entre energia e desempenho, há métodos que consomem menos energia que outros sem prejudicar o desempenho, constituindo assim uma alternativa mais sustentável. Em particular, o método baseado em aprendizagem profunda apresentou um baixo desempenho tanto em termos de consumo energético como de capacidade preditiva. Ademais, verificou-se que as características dos dados influenciam os compromissos encontrados entre os métodos, sendo as abordagens de sobre-amostragem mais adequadas para conjuntos de dados mais complexos do que abordagens de sub-amostragem ou sem reamostragem.

Palavras-chave: IA Sustentável, IA Centrada em Dados, Classificação Desbalanceada, Reamostragem, Métodos de Pré-processamento, Complexidade dos Dados

# Abstract

The recent advancements in artificial intelligence have led to an increase in computational demands, raising concerns regarding environmental impact. To address this issue, the field of green AI has emerged with the goal of improving the sustainability of artificial intelligence systems. A promising direction is data-centric green AI, which focuses on modifying data to improve overall pipeline efficiency. However, the literature on data-centric approaches for imbalanced classification remains limited, and no previous work has addressed comparisons between multiple resampling methods in terms of energy consumption. In this dissertation, we investigate the energy-performance trade-off of data-level methods for imbalanced classification and analyze how dataset complexity affects this relationship across multiple benchmark datasets. We consider several well-established traditional resampling methods, alongside one deep learning-based method. Our results show that although energy and performance are often negatively associated, there are methods that consume less energy than others without compromising performance, making them a more sustainable alternative. In particular, the deep learning-based method performed poorly in terms of both energy consumption and performance. In addition, we found that dataset characteristics impact the trade-offs between methods, with oversampling approaches being better suited than undersampling and no-resampling approaches for datasets with higher complexity.

Keywords: Green AI, Data-Centric AI, Imbalanced Classification, Resampling, Data-level Approaches, Dataset Complexity

# Table of Contents

# List of Tables

# List of Figures

# List of Abbreviations

**ADASYN** Adaptive Synthetic Sampling Approach.

**AI** Artificial Intelligence.

**CD** Critical Difference.

**CTGAN** Conditional Tabular GAN.

**DCAI** Data-Centric Artificial Intelligence.

**ENN** Edited Nearest Neighbors.

**GANs** Generative Adversarial Networks.

**IR** Imbalance Ratio.

**ML** Machine Learning.

**NVML** NVIDIA Management Library.

**RAPL** Running Average Power Limit.

**ROS** Random Oversampling.

**RUS** Random Undersampling.

**SMOTE** Synthetic Minority Oversampling Technique.

# 1. Introduction

The field of artificial intelligence (AI) has experienced rapid growth in recent years, leading to increased computational demands and larger training datasets. As these are directly related to energy consumption and carbon emissions, the concern about the environmental impact of this technology grew among researchers, leading to the creation of the field of green AI [1, 2]. The goal of green AI is to improve the sustainability of AI systems by developing techniques that reduce their carbon footprint and by promoting the inclusion of efficiency as an evaluation criterion alongside performance.

Given the relationship between data size and computational cost, a natural solution has been to use data-centric AI, which aims to improve machine learning (ML) systems by enhancing data quality rather than focusing solely on model tuning [3]. Although data-centric AI originally emerged with a focus on improving performance, studies exploring its potential to enable more sustainable AI systems show promising results. These studies demonstrate how data reduction can be used to lower energy consumption while maintaining performance [4], and how existing data-centric methods, such as active learning, knowledge transfer, and curriculum learning, can also improve computational efficiency [5].

Despite this demonstrated potential, an important set of data-centric techniques remains underexplored: data-level methods for imbalanced classification. Imbalanced classification refers to learning from data with an imbalanced class distribution, where the underrepresented class is also the class of interest [6]. This scenario is common in various applications, such as medicine, fault detection, and security, and is therefore an important research problem in ML [7]. In this scope, data-level methods are used to rebalance the dataset by adding or removing samples.

The current literature on the efficiency of data-centric approaches for imbalanced classification is scarce and shows significant gaps: in most studies, efficiency is either driven by application-specific resource constraints or treated as secondary to performance when comparing models; authors only report time or do not report any efficiency measurement; and there is a lack of benchmarking studies involving multiple resampling methods. To overcome these gaps, this dissertation provides an energy-performance trade-off comparison across well-established resampling methods and examines how dataset complexity

influences the results. With this, we aim to identify which methods offer the best balance between sustainability and performance, and to provide insights into their suitability based on dataset characteristics.

## 1.1. Research Objectives and Contributions

This dissertation aims to achieve the following goals:

1. Characterize the literature on the intersections between green AI, data-centric AI and imbalanced classification;
2. Conduct a comprehensive comparison of well-established resampling methods in terms of energy consumption and classification performance, identifying methods that offer an advantageous combination of sustainability and performance;
3. Provide insights into the sustainable use of resampling methods across different dataset complexity groups, supporting researchers and practitioners in selecting the most suitable method for their use case.

In this work, we conduct, to the best of our knowledge, the first benchmark study of resampling methods considering both classification performance and energy consumption. We provide an analysis of the energy-performance trade-off across 11 well-established resampling methods evaluated on 100 standard imbalanced classification datasets, including a stratified analysis over three dataset complexity groups. All experiments are implemented in a reproducible pipeline, and both the source code and the corresponding experimental results are publicly available in a dedicated repository.[1] Furthermore, a related paper has been submitted to the ECML-PKDD workshop on Data-Centric Artificial Intelligence (DEARING 2026).

## 1.2. Structure of the Document

This document is structured as follows. Chapter 2 presents the background of the three research areas considered in this study and reviews the related work on their pairwise and three-way intersections. Chapter 3 describes the experimental setup, highlighting the data collection process, dataset complexity measures, resampling techniques, classification algorithms, tools, hardware, and evaluation metrics. Chapter 4 presents and discusses the results of our experimental evaluation, and Chapter 5 presents the conclusions and potential directions for future work.

[1] https://github.com/Barb02/Green-AI-Data-Centric-Approaches-for-Imbalanced-Classification

# 2. State of the Art

In this chapter, we introduce each of the three areas considered in this study and discuss the related work at their pairwise and three-way intersections. Additionally, we provide a brief overview of the tools available for measuring efficiency metrics.

## 2.1. Background

This section provides the theoretical foundations necessary to understand the research motivation and the methods considered in this study.

### 2.1.1 Green AI

Green AI is an emergent field that aims to improve the sustainability of AI systems by relying on more energy-efficient practices. Its creation was motivated by the recent AI breakthroughs in deep learning and generative AI, which lead to a substantial rise in the carbon footprint [1, 2, 8]. The generative AI race is the greatest example of this, with models being built to surpass others in terms of accuracy, while disregarding efficiency and environmental concerns. To improve accuracy, new models are often developed with a larger number of parameters than the last, which then increases energy and water usage [8].

Following that scenario, green AI advocates for the importance of taking efficiency measures into account when making decisions about the ML pipeline (being it the data, model, hardware, software, datacenter, or algorithm) [9]. By doing so, practitioners would make more informed decisions by considering not only performance metrics, but also environmental impact and the associated trade-offs. On the research side, the focus is to create new methods or identify existing techniques that lower the computational cost of machine learning pipelines.

There are several ways one could measure model efficiency. Schwartz et al. [1] highlighted the following: carbon emission, electricity usage, elapsed time, number of parameters, and floating point operations (FPO). The most natural when it comes to the environmental perspective is to consider carbon emission, which can be understood more easily by non-technical audiences. However, its value depends on the place and time of the measurement, which makes it difficult for comparisons. Some good alternatives are electricity usage and elapsed time, which overcome time and location dependency, correlate

to carbon emission and are still straightforward for the general public. Their disadvantage lies on still being hardware dependent, which is what the number of parameters and FPO metrics have as their benefit. Although the authors point FPO as the best measure, it has important limitations: it is hard to interpret, difficult to understand for audiences without a technical background and hard to compute with existing tools. We argue that, despite its limitations, energy consumption remains the most suitable measure, given its ease of computation and intuitive interpretation. These properties are essential for the development of the field and for ensuring its impact on researchers, practitioners, policymakers, and society. Indeed, most studies in the field use either carbon emissions, energy usage, or elapsed time [8–12].

### 2.1.2 Data-Centric AI

Data-centric AI (DCAI) is a fairly new AI paradigm that focuses on improving ML systems by enhancing the quality of the data. It can be defined as an alternative to the traditional model-centric AI paradigm, where the model is the main subject of study and refinement [5]. In model-centric AI, improvements are driven almost exclusively by model design and tuning, with data quality generally remaining outside the optimization process. Data-centric AI posits that using high-quality data leads to better results, which would otherwise be hard to obtain by solely relying on model improvements.

Although the term “data-centric AI” has only recently gained traction, some of the methods it includes were already commonly adopted by ML practitioners, such as data cleaning, data labeling, and feature engineering, all included in what is referred as the “preprocessing phase” [3]. One of the purposes of DCAI is to expand the importance of these practices, making the process systematic and iterative. On a model-centric approach, data is processed once and left on a locked state, while the model is improved through several iterations of tuning. In contrast, DCAI incorporates iterative refinement of the dataset, where each cycle may include error analysis, data modification, and quality assessment. This enables continuous improvement of the data, which can lead to greater results.

There has been some effort in categorizing DCAI principles and challenges. Zha et al. [13] draw the big picture of DCAI by grouping common tasks under three large groups, which they call “missions”: training data development, inference data development, and data maintenance. Normally, more attention is given to the first, since its tasks are included in the usual preprocessing phase. The authors argue that inference data and

data maintenance should be equally valued: the former to ensure reliable validation and model understanding, and the latter to ensure data remains consistent as time passes. Alternatively, Jarrahi et al. [3] propose six principles to guide DCAI research. The first two correspond to the systematic improvement of data fit and data consistency, which includes checking for data feature coverage, class balance, and annotation convergence. The third highlights how both data and model must be iteratively monitored and improved, which relates closely to the concept of machine learning operations (MLOps). And finally, the last three focus on how the process of collecting, annotating, developing, and monitoring must be human-centered, in order to ensure more fair, explainable, and precise results.

### 2.1.3 Imbalanced Classification

A very common issue in classification tasks is the unequal distribution of class labels within the dataset, that is, a significant difference in the number of records belonging to each class. Adding to this, it is often the case that the most underrepresented class(es) are also the most relevant. When both these conditions hold true, the problem is classified as an imbalanced learning problem [6]. Examples of this scenario are usually found in fields such as anomaly detection, flagging spam, fault prediction, and medical diagnosis [7].

The main issues that arise from learning from an imbalanced dataset are model bias to the majority class and misleading performance estimates produced by standard evaluation metrics such as accuracy [7]. The first happens because classifiers are built to minimize classification error, thus making them prioritize the correct classification of the dominating classes, while treating minority class instances as noise. The second is a consequence of how metrics are defined: in the case of accuracy, it computes the ratio between the number of correctly classified instances and the total number of instances. In a dataset with 99 negative instances and only one positive instance, a model that predicts every instance as negative would achieve 99% accuracy, despite misclassifying all positive instances (which are the most relevant for the problem). Therefore, accuracy would be a misleading metric in this setting.

The evaluation metric problem is comparatively easier to address than the learning difficulties caused by class imbalance, as alternative metrics such as recall, precision, ROC-AUC, F1-score, and G-mean can be used instead of accuracy. Hence, most of the research is concentrated on improving model performance under class imbalance [7]. There

are several taxonomies in the literature, but recent reviews usually divide methods into three groups: data-level, algorithm-level, and hybrid approaches [14, 15]. Data-level approaches work by rebalancing the class distribution in the dataset before it is passed to the model; algorithm-level approaches focus on fixing the bias problem directly at the model level by developing new algorithms or modifying existing ones; and hybrid approaches encompass techniques from both data- and algorithm level approaches, combining their advantages. In particular, the data-level alternative can be seen as a data-centric approach used to deal with the imbalanced learning problem, which makes it the alternative of interest for this study.

Data-level approaches, also known as resampling methods, can rebalance the dataset by either oversampling the minority class, undersampling the majority class, or combining both strategies. Oversampling refers to the addition of minority class instances, and undersampling to the removal of majority class instances. There are several ways to perform these tasks. The simplest alternative is to randomly replicate/remove instances, until the number of records of each class correspond to the desired ratio. Another very popular method is Synthetic Minority Oversampling Technique (SMOTE) [16], which works by creating synthetic samples resulting from interpolation. SMOTE is usually preferred over random oversampling (ROS) and random undersampling (RUS) , because it attenuates the main issues with these two methods: overfitting and information loss, respectively. Since its introduction in 2002, SMOTE has been widely extended through several methods aimed at addressing its limitations. Two well-established examples are Borderline-SMOTE [17], which focuses on generating samples near the class boundary, where classification is harder, and Adaptive Synthetic Sampling Approach (ADASYN) [18], which also focuses on generating samples where the classification problem is harder, but based on minority class density (verifying how many neighbors of the minority class samples are from the majority class).

## 2.2. Related Work

In this section, we present related work that combines the areas presented in 2.1. We start by reviewing works at the pairwise intersections, followed by those addressing all three areas. Fig. 2.1 illustrates this organization. This approach is motivated by the fact that data-centric green AI for imbalanced classification is still a relatively new area, requiring a better understanding of its sub-areas.

Initially, we conducted a simple literature review by searching for papers in the databases Scopus, Web of Science, IEEE Xplore, and ACM Digital Library, as well as using the Google Scholar platform. However, papers corresponding to Regions B and D (Fig. 2.1) were difficult to find, which led us to adopt a more structured strategy. For these two areas, we constructed queries for the same four databases and selected papers based on predefined selection criteria. The detailed process and results for each region are described in their respective sections (2.2.2 and 2.2.4).

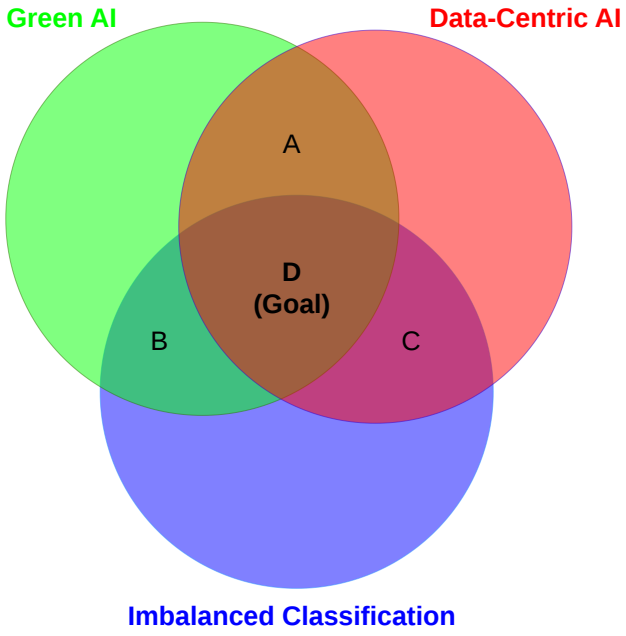



Figure 2.1: Venn diagram of the three areas included in this study. Region A corresponds to data-centric green AI (Section 2.2.1), Region B corresponds to imbalanced classification and green AI (Section 2.2.2), Region C corresponds to data-centric approaches for imbalanced classification (Section 2.2.3), and Region D corresponds to data-centric green AI for imbalanced classification (Section 2.2.4).

### 2.2.1 Data-Centric Green AI

The intersection of data-centric AI and green AI corresponds to the analysis of the impact that data handling has on the computational cost of AI systems, as well as the use of data manipulation techniques to obtain a more energy-efficient ML pipeline. The potential of the latter arises naturally, as big data and the increase in the amount of data needed for training are often credited with the rising environmental impact of AI. The idea is, if data is one of the main issues, data tuning can serve as a logical solution.

Thus, despite being an emergent field, relevant studies already exist in the literature. In this section, we highlight those which belong to Region A of the Venn diagram (Fig. 2.1).

Related work [5, 19, 20] identifies existing data-centric techniques that can be applied to green AI, namely active learning (AL), coresets, curriculum learning, dataset distillation (DD), data augmentation, knowledge transfer/sharing and gradient pruning (represented by Selective-Backdrop). They reduce computing costs by either performing a type of data reduction (AL, coresets, DD), allowing a faster convergence (curriculum learning, data augmentation), avoiding training from scratch (knowledge transfer/sharing), or omitting samples from gradient updates (gradient pruning via Selective-Backdrop). It is noteworthy that data augmentation, despite increasing the amount of data, can still reduce overall computational costs. This suggests that other factors, such as avoiding additional data collection and reducing the number of training epochs, can have a stronger impact on energy consumption, even when the dataset size increases. In addition, it is worth noting that related work often addresses deep learning specifically [19, 20], which reflects the growing interest in applying green AI to this specific area. This also explains why some of the methods mentioned are specific to deep learning algorithms.

One example of an empirical study in this domain is provided by Verdecchia et al. [4], who investigate whether dataset modification can reduce energy consumption. The results show that, for the selected dataset, the removal of rows and columns reduced energy consumption while maintaining a good F1-score, although the magnitude of this effect varied across models. The techniques used were feature selection and random sampling (regardless of class), with the second achieving a greater reduction in energy consumption than the first. Although the dataset used was imbalanced, this study was included in Region A instead of Region D of Fig. 2.1 as the selected methods did not aim to rebalance the dataset.

Other relevant works addressing data reduction include Alswaitti et al. [21], who propose an evolutionary-based sampling framework to identify a small subset of the training data that maintains the predictive performance of the full dataset (referred to as “elite samples”) and thereby reduce the amount of data used for training; Dhabe et al. [22], who introduce a data reduction technique termed “bonsai datasets”, which removes redundant samples from the training set; and Anselmo and Vitali [23], who present a general methodology for deep learning problems based on a human-in-the-loop approach, allowing researchers to specify the desired reduction-precision trade-off.

### 2.2.2 Imbalanced Classification and Green AI

The intersection between imbalanced classification and green AI includes the study of the sustainability impact of current imbalanced classification solutions and the creation of new techniques that are both efficient and successful at addressing the imbalance.

As previously mentioned, we conducted a query-based review for Region B (Fig. 2.1), which excludes data-centric approaches, due to the scarcity of studies found on the initial review. We combined keywords related to imbalanced classification and green AI by an “AND” operator, and excluded data-centric solutions by using an “AND NOT” operator. Search fields were adapted to each database’s functionality, and the selection criteria were: relates green AI (or efficiency concerns) and imbalanced classification through algorithm-level approaches.

A total of 61 papers were obtained after combining results and removing duplicates, but none remained after screening, as they did not meet the selection criteria. Although a reasonable portion of papers addressed both efficiency and imbalanced data, they did so independently, not relating both concepts, which explains the numerical discrepancy between matches and selected papers.

This shows that there is a lack of studies addressing how algorithm-level approaches for imbalanced classification can be more efficient. This condition can attributed to the fact that green AI is still emerging, and has not yet reached all fields within AI and machine learning. Unlike data-centric methods, imbalanced classification is not itself a means of achieving greener systems, but rather a research field in which green AI can be applied to develop more efficient methods. This makes the connection between green AI and imbalance less trivial. Yet, we call for more studies addressing this area, as they could lead to the development of more sustainable algorithm-level solutions.

### 2.2.3 Data-Centric Approaches for Imbalanced Classification

As mentioned in Section 2.1.3, when the concept of data-centric AI is applied to the imbalanced classification context, we obtain the intersection corresponding to the data-level approaches, which modify the data in order to improve performance. This is a much more explored area than the previous two, since data-level approaches existed before the term “data-centric” began to be widely used (e.g., SMOTE original paper is from 2002, and data-centric as a field emerged around 2021-2022) [3, 16]. In this section, we present some of the most recent advancements (2019 onward) within Region C in Fig. 2.1.

SMOTE continues to inspire the development of improved variants to this day [24]. Zhang et al. [25] propose Instance Weighted SMOTE, which focuses on data distribution by estimating location information for each instance. Instances are classified as noise, borderline or safe based on the error rate of an UnderBagging-like undersampling ensemble, and then are either removed, prioritized or underemphasized, respectively. Maldonado et al. [26] introduce Feature Weighted-SMOTE (FW-SMOTE), which uses the weighted Minkowski distance to define the neighborhood of minority class instances, and consequently prioritizes the most significant features. Both methods have been shown to outperform the original SMOTE and some other popular methods, such as ADASYN.

Another important breakthrough in this area was the introduction of Generative Adversarial Networks (GANs) [27], in 2014. GANs are generative models formed by two neural networks: a generator and a discriminator. The generator creates synthetic data based on the original data distribution, and the discriminator distinguishes if a sample comes from the original data or if it was produced by the generator, which leads to the minimization of the divergence between the generated and the real data distribution. In the context of imbalanced classification, they can be used to oversample the minority class [3]. More recently, some variants were introduced, such as Conditional Tabular GAN (CTGAN) [28] and Single Attribute guided Conditional GAN (SA-CGAN) [29], which are specifically tailored for tabular data. CTGAN addresses challenges that arise when applying GANs to tabular data, such as mixed data types and multi-modal non-Gaussian continuous columns, while SA-CGAN improves on previous GANs and traditional methods (SMOTE, ADASYN, random sampling) by incorporating single-attribute patterns in multi-class imbalanced scenarios.

### 2.2.4 Data-Centric Green AI for Imbalanced Classification

By uniting the three concepts, we define the focus of this study: a sustainability analysis of data-centric approaches used to deal with imbalanced classification problems. This sustainability analysis can be described as a energy-performance trade-off analysis, in which data-level methods are evaluated not only based on the performance metrics, but also in terms of their energy usage and consequential environmental impact.

As mentioned, a query-based search was conducted to overcome the difficulty of finding related papers in Region D of Fig 2.1. It was constructed by using “AND” operators between a set of keywords representing imbalanced classification, green AI, and data-centric

approaches. Similar to the procedure used in Region B, search fields were adapted to each database's functionality, and the selection criterion was to relate all three areas directly. After combining results and removing duplicates, 67 papers were obtained, which were then reduced to six after screening. Three other papers [30–32] had been already obtained by the initial research mentioned in Section 2.2, which were added to total. The nine papers are described in detail in Table 2.1, which includes whether the paper explicitly addressed green AI (or if it aimed at efficiency for other reasons), the efficiency metric considered, the employed data-centric technique, a summary of the main contributions, and the observed limitations. The studies are grouped by the type of data-centric approach employed, with the first six rows corresponding to undersampling techniques and the remaining three to oversampling SMOTE-based approaches.

Table 2.1: Overview of the papers addressing data-centric green AI for imbalanced classification.

| Paper Ref. | Green AI Motivation | Efficiency Metric | Data-Centric Technique | Summary | Limitations |
|---|---|---|---|---|---|
| Tae et al. (2015) [33] | No | - | Undersampling | Applies resource-efficient algorithms to predict high-cost patients. Undersampling is primarily used to deal with imbalance, but also to achieve efficiency. | Undersampling technique not specified. No explicit measurements, efficiency improvement influenced by the classification algorithm, not limited to undersampling. |
| Meriem Amina et al. (2018) [12] | No | Training and testing time | Undersampling, flow size variation, feature selection | Combines undersampling with different packet sizes to reduce inference time for real-time imbalanced traffic classification. Bagged Random Forest performed best in the optimal scenario: 500 samples/class with 8 packets/flow, achieving a good balance between time and F1-score. | Undersampling technique not specified. Time reduction also influenced by flow size, not limited to undersampling. |

*Continued on next page*

| Paper Ref. | Green AI Motivation | Efficiency Metric | Data-Centric Technique | Summary | Limitations |
|---|---|---|---|---|---|
| Yan et al. (2022) [32] | No | Resampling time and total training time | Spatial Distribution-based UnderSampling (SDUS) | Presents a method that uses sphere neighborhoods to maintain majority-class patterns across sampling subsets. Two variants were created based on different sample selection strategies. They show better/comparable performance to other methods, but have a higher runtime cost. | Does not include total training time for all of the methods, only for those that have specific embedded classifiers. |
| Kamalov et al. (2024) [10] | Yes | Training time | Undersampled Random Forest (URF) | Introduction of URF, which undersamples the majority class dynamically within each decision tree. Although URF reduces data size, it maintains high recall rates for minority class instances. | Reduction in overall accuracy; Improvements influenced by parallelism, not limited to undersampling; Might decrease time but maintain energy because of the parallelism, since more cores are in use. |
| Reddy et al. (2024) [30] | No | - | Undersampling (six undersampling-based ensembles) | Assesses the effectiveness of undersampling techniques in handling the imbalance present in Cyber-Physical systems. Those systems require a low computational cost, which is also achieved by undersampling. | No explicit measurements. |
| Sanderson and Kalganova (2025) [11] | Yes | Training time | Pure reduction (sample removal regardless of class) and undersampling using random and centroid-distance-based sample removal | Evaluation of several data-centric techniques on a time-series dataset. Identifies that class density may be used as a metric to qualify a dataset for reduction. Large and highly imbalanced datasets - reducing improves performance and time. Small and balanced datasets - worsens. Either highly imbalanced or large datasets - reducing maintains performance. | Dataset with good performance despite imbalance (maybe due to well-separated classes). Informs general CO2 per hour and based on power approximation, but does not use it to compare the methods. |

*Continued on next page*

| Paper Ref. | Green AI Motivation | Efficiency Metric | Data-Centric Technique | Summary | Limitations |
|---|---|---|---|---|---|
| Guo et al. (2020) [34] | No | Batch training time | SMOTE-Top-K. | Proposes STAN, a graph neural network, which uses SMOTE-Top-K. STAN achieves higher accuracy than other models with comparable training time. The study also shows how SMOTE-Top-K achieves higher accuracy than random sampling, as well as it helps on lowering training time per batch and memory usage. | Training time also influenced by other mechanisms; Comparison with other models made only in terms of batch training time; Focused more on batch training time and memory to accommodate bigger graphs, instead of the resampling time of SMOTE-Top-K. |
| Sanap and Aher (2024) [35] | No | - | Seagull-optimized SMOTE | Introduction of a deep learning approach for detecting Distributed Denial of Service (DDoS) attacks in cloud computing. The approach consists of a classifier combined with an optimized SMOTE, and it aims to solve imbalance and lower time complexity. The optimized SMOTE also reduces overfitting, and the proposed approach outperforms existing solutions. | No explicit measurements, time complexity improvement also influenced by other mechanisms. |
| Liaw et al. (2025) [31] | No | Resampling time and total training time | Histogram SMOTE | Presents a classifier combined with an adapted SMOTE that uses histograms in order to mitigate intra-class imbalance and small-disjuncts. The adapted SMOTE utperforms other SMOTE techniques in terms of G-mean. Usage of SMOTE increases runtime when compared to the standalone novel classifier. | Execution time comparison limited to baseline classifier. |

### 2.2.5 Discussion

From the related work presented, we can draw important conclusions. Naturally, the intersection area of data-centric imbalanced classification (Region C of Fig. 2.1) is the most established. In contrast, the area of data-centric green AI (Region A of Fig. 2.1) is in an emerging stage, while the imbalanced classification green AI domain (Region B of Fig. 2.1) remains largely unexplored. Interestingly, the connection between green AI and imbalanced classification was only found via data-level approaches (e.g. Region D, with nine papers, against Region B, with zero). This suggests that data-centric approaches may be more effective than algorithm-level approaches for jointly addressing efficiency constraints and class imbalance. Still, the overlap between imbalanced classification and green AI, both including and excluding DCAI (Region D and B, respectively), is where there is the most lack of studies.

When it comes to the papers that fully relate to the research focus of this work (Table 2.1), only two explicitly mention green AI as the motivation for achieving more efficient results and one third of the studies do not report any measurements. This shows two important gaps, which might be connected. Measuring is one of the core principles emphasized in the definition of green AI [1], but since a lot of papers do not target green AI specifically, a portion of them also fail to include measurements. This is concerning, as the lack of reporting hinders reproducibility and limits the comparability of different methods across studies.

Another reason for the lack of measurements stems from the natural assumption that less data always leads to less computation and greater efficiency. Researchers rely on this assumption to claim that their undersampling methods are more efficient and consequently avoid conducting experiments [30, 33]. However, the actual impact may be negligible and can only be properly assessed through measurements.

An additional gap is the fact that only two papers [31, 32] explicitly measure the computational cost of the resampling step itself, allowing its contribution to the total computational cost to be assessed. While it could be negligible in simple techniques such as random undersampling, more elaborate methods based on SMOTE or GANs might have a more considerable impact. Therefore, it is important to measure this consumption in order to either confirm it is negligible or add to the total energy consumed.

It is also noticeable that five [10, 12, 33–35] studies employ extra mechanisms that also

influence the decrease in execution time. Consequently, their efficiency improvements that cannot be attributed solely to the resampling technique. This is an issue because the results are not isolated, which may lead to an overestimation of the impact of resampling.

An additional threat to the validity of results is the choice of the datasets. Some studies, such as [11], have high accuracy results that might indicate that the dataset used is not be so negatively affected by the imbalance, which is usually due to classes being well separated.

From the table we can see that more papers focus on undersampling (six), than oversampling (three), indicating a stronger focus on undersampling methods, which may also be due to the idea of "less data leads to less energy" mentioned previously. This highlights the need for a more comprehensive analysis of undersampling, oversampling, and hybrid methods.

Finally, it is worth mentioning that there might exist more papers similar to to those of Liaw et al. [31] and Yan et al. [32] presented in Table 2.1 that were not identified during our search process. Notably, those two papers were found on the general search, and were not caught by the query due to not matching with the green AI keywords. They both conduct a trade-off analysis based on runtime, but do not highlight the practice on the main fields (title, abstract and keywords) as its not their main focus. Therefore, there might be more papers that similarly perform this analysis but do not consider it important enough to promote it as a contribution.

## 2.3. Software Tools for Efficiency Measurement

While numerous solutions for measuring efficiency metrics are available, the current literature [36, 37] indicates that their results may vary when assessing the same task. Therefore, selecting the most appropriate tool is a critical step. The tools considered in this study were sourced from the literature [4, 36, 37] and a related community repository [38]. They include GreenAlgorithms [39], MLCO2 Impact [40], CodeCarbon [41], Carbon Tracker [42], Zeus [43], and pyJoules [44].

These solutions can be divided into two categories: static and code-based approaches [38, 45]. Static approaches estimate energy consumption from elapsed time and manually provided hardware specification, while code-based tools measure energy in real time during program execution. Since static approaches estimate the energy consumed only after running the code, they are less consistent than code-based tools [45]. In addition,

they are less suitable for repeated usage across multiple experiments, as they require manual input. From the tools analyzed, GreenAlgorithms and MLCO2 Impact belong to this category.

In order to measure energy in real time, code-based tools use Running Average Power Limit (RAPL), a technology provided by Intel that enables the measurement of the energy consumption of processor components. This mechanism has been tested and validated in prior studies [46, 47]. For GPU measurements, they use the NVIDIA Management Library (NVML), which exposes energy consumption counters enabling direct GPU energy measurement. All remaining tools belong to the code-based category.

Although CodeCarbon, Carbon Tracker and Zeus can be used for measuring the energy of any code, their focus lies primarily on deep learning problems. So while their overhead is negligible for those scenarios, it becomes a problem for traditional machine learning models and resampling methods, as they have fast execution times. This issue is illustrated by a small preliminary comparison between CodeCarbon and pyJoules: for training a Logistic Regression model that originally took 0.1 seconds, CodeCarbon increased the runtime to approximately 7 seconds, whereas pyJoules preserved the original execution time.

For this study, the chosen tool must satisfy some predefined criteria. It should:

- Measure energy
- Measure CPU
- Measure GPU (for the deep learning-based method)
- Measure the resampling technique and the training separately
- Be available on Python (if code-based)
- Be compatible with Arch Linux

The tools described above were evaluated against these criteria and their respective limitations, resulting in pyJoules being selected. The selection process is summarized in Table 2.2.

Table 2.2: Summary of tool selection process.

| Tools | Type | Reason for elimination |
|---|---|---|
| GreenAlgorithms, MLCO2 Impact | Static | Poor consistency, hard to use |
| CodeCarbon, Carbon Tracker, Zeus | Code-based | Large overhead |
| pyJoules | Code-based | - |

# 3. Experimental Setup

Based on the gaps identified in the Discussion section of Related Work (2.2.5), we designed an experimental study to compare multiple data-level methods in terms of performance and energy consumption. Well-established methods were selected from the literature, combining simple, classical approaches with more advanced techniques, and a pipeline was created to collect the energy consumed during the resampling and training phases for each method. To address the research objectives outlined in Section 1.1, we formulated the following research questions:

**RQ1** How do different resampling techniques compare in terms of energy consumption and classification performance? Can a simple model with a complex resampling method achieve superior energy-performance results compared to a complex model with a simple resampling method or without resampling?

**RQ2** Is there a significant gain in performance that justifies the energy costs of using deep learning-based approaches?

**RQ3** How does dataset complexity influence the established trade-offs?

In the following sections, we describe all aspects related to the experimental setup, namely the datasets, dataset complexity measures, resampling techniques, algorithms, tools, hardware, and evaluation metrics.

## 3.1. Data Collection

We selected the KEEL dataset repository [48], one of the most widely used repositories for imbalanced datasets, as the source for this study. All binary-class imbalanced datasets were included, totaling 100 datasets. These datasets exhibit a wide range of imbalance ratios (1.82 to 129.44), numbers of samples (92 to 5472), and numbers of features (3 to 41), enabling a broad empirical analysis. Tables A.1, A.2, A.3 (available in Appendix A) summarize the main characteristics of the datasets, grouped by complexity level. Details on how these groups were defined and what the “score” metric represents are presented in the following section.

All datasets used correspond to the `.dat` files available in the KEEL repository. The only modification we made concerned *ecoli-0_vs_1*, whose class labels were initially inverted

(with “positive” representing the majority class and “negative” the minority class); these labels were flipped to ensure consistency across all datasets.

## 3.2. Dataset Complexity Measures

It is common for imbalanced learning studies to use imbalance ratio (IR), defined as the number of negative instances divided by the number of positive instances, as a defining factor for classification difficulty. However, previous work has shown it to be a limited measure [49]. For example, a dataset with high IR and well-separated classes might be easier to classify than one with medium IR and highly overlapping classes. Therefore, by combining multiple complexity measures, we can better measure the difficulty of each dataset.

Although multiple ways of measuring complexity exist, we follow the approach of Lorena et al. [50], one of the seminal papers on the subject. The authors group 22 complexity measures into six categories: feature-based, linearity, neighborhood, network, dimensionality, and class imbalance. A brief description of each group is presented next (Table 3.1), with a detailed description of each measure provided in Table B.1, in Appendix B.

Table 3.1: Brief description of complexity measure categories.

| Category | Usage |
|---|---|
| Feature-based | Determine how informative the dataset features are for class separability |
| Linearity | Quantify the linear separability of classes |
| Neighborhood | Characterize class density and presence inside local neighborhoods |
| Network | Obtain structural relationship upon analyzing samples as a graph |
| Dimensionality | Evaluate data sparsity by comparing number of samples and dimensionality |
| Class imbalance | Characterize the balance between the number of samples of each class |

In order to obtain the previously described measures for our datasets, we used the library problexity [51], which implements them in Python based on the definitions provided by Lorena et al. [50]. Although the package provides all 22 measures, the *hubs* measure was excluded from our analysis due to a runtime warning indicating that the returned values might not be meaningful. *Hubs* is a network measure that quantifies node connectivity weighted by the connectivity of neighboring nodes, by using a $\varepsilon$-NN graph representation

of the dataset. This warning was consistently raised across all datasets. Moreover, upon testing with the original version of the Iris dataset [52], which is balanced, we confirmed the problem persisted even outside of the scope of the imbalanced datasets used in this study.

The issue stems from the fixed $\varepsilon = 0.15$ used to construct the $\varepsilon$-NN graph. It imposes a global threshold over normalized distances, which, together with the removal of inter-class edges and the effects of moderate-to-high dimensionality, can lead to a highly sparse connectivity structure where many nodes exhibit no or very limited connectivity. As a result, more than 30% of the hub scores are zero, leading to a potential non-meaningful result, as indicated in the warning message. This observation is consistent with the remark by Lorena et al. [50], who note that a fixed value of 0.15 may not be suitable for all datasets. As the problexity library does not provide a mechanism to tune this parameter, and considering that other network and neighborhood measures capture similar structural properties of the data, we excluded this measure from the analysis.

The library also provides a single combined metric, referred to as the *score*, computed as the average of all measures. Since our goal was to obtain a general estimate of dataset complexity, we used *score* to group datasets into low, medium, and high complexity. It is important to note that, given the previously mentioned limitation, the *hubs* measure was excluded from the *score* computation. In the absence of a standard threshold, we employed equal-frequency partitioning to obtain the groups. Table 3.2 summarizes the number of datasets and the quantile boundaries for each group.

Table 3.2: Distribution of datasets across complexity groups defined by score terciles.

| Complexity Group | Number of Datasets | Score Interval |
|---|---|---|
| Low | 33 | [0.123, 0.282) |
| Medium | 34 | [0.282, 0.337) |
| High | 33 | [0.337, 0.548] |

## 3.3. Resampling Techniques

As mentioned, our goal is to provide a trade-off comparison between the most commonly used and well-established resampling techniques in the imbalanced classification field.

We aimed to use both undersampling and oversampling techniques while also considering the complexity of their internal mechanisms. Thus, we considered two random sampling methods, two informed undersampling methods, four SMOTE-based methods, and two hybrid methods.[1] All traditional methods considered in the study were taken from the imblearn library [53] and the smote-variants library [54]. All hyperparameters were set to their respective default values. However, in specific cases, the *k_neighbors* parameter had to be dynamically lowered (from 5 to 3, or lower for Cluster-SMOTE) to enable the resampler to function, given the small number of minority samples in some datasets. We also employed a deep learning-based approach, namely CTGAN, using the implementation provided by the Synthetic Data Vault (SDV) library [55]. For our analysis, we utilized *CTGANSynthesizer* with default parameters (Table 3.3). This ensured that the measured energy corresponded to a single fit (no hyperparameter tuning loop) and enabled a more equitable comparison with traditional methods. Next, we provide a brief description of the chosen methods for this study, which will allow for a better understanding of the experimental results.

Table 3.3: Default parameters used for *CTGANSynthesizer*.

| Parameter | Default |
|---|---|
| enforce_min_max_values | True |
| enforce_rounding | True |
| locales | ['en_US'] |
| embedding_dim | 128 |
| generator_dim | (256, 256) |
| discriminator_dim | (256, 256) |
| generator_lr | 2e-4 |
| generator_decay | 1e-6 |
| discriminator_lr | 2e-4 |
| discriminator_decay | 1e-6 |
| batch_size | 500 |
| discriminator_steps | 1 |
| log_frequency | True |
| verbose | False |
| epochs | 300 |
| pac | 10 |
| enable_gpu | True |

[1] Some widely used methods, namely ADASYN and K-Means SMOTE, were excluded because they consistently raised runtime exceptions when applied to datasets with very few minority samples.

### 3.3.1 Random Undersampling and Random Oversampling

Random Undersampling (RUS) is a trivial resampling method that works by randomly removing majority samples until achieving the desired balance ratio. It is agnostic to dataset structure, meaning it does not take into account whether or not the sample to be removed is informative for classification. This can lead to loss of information, and hurt prediction performance [7]. However, its simplicity could translate into a lower energy consumption, and it might perform well for less complex, well-separated datasets.

Random Oversampling (ROS) is very similar to RUS, but instead of removing majority samples, it randomly replicates minority samples until the goal ratio is achieved. Again, the randomness can be problematic, and in this case, it can lead to overfitting, as less representative samples or noisy instances may be replicated, causing the decision boundary to become skewed toward them [7]. Like RUS, it may still be useful for datasets with low complexity and be a reasonable low-energy solution.

### 3.3.2 Tomek Links and Edited Nearest Neighbors

Tomek Links [56] and Edited Nearest Neighbors (ENN) [57] are undersampling methods that work by cleaning samples that are close to samples of the other class. Depending on the configuration, they can either remove both majority and minority samples or only majority samples, the latter being the implementation we used, as it was the default on the chosen library. In particular, Tomek Links removes majority samples whose closest neighbor belongs to the minority class, while ENN removes majority samples whose nearest neighbors all (by the library default) belong to the minority class. Compared to RUS, both Tomek Links and ENN are more data-informed ways of undersampling the majority class, which helps reduce information loss.

Nevertheless, both methods are still simple and not very strong at handling imbalance, as they only remove few specific majority samples, typically near the decision boundary. Their advantage against more complex models lies in their cost-efficiency, as they are simpler and their data reduction might be beneficial for lowering training costs.

### 3.3.3 Synthetic Minority Oversampling Technique (SMOTE)

SMOTE [16] is one of the most popular methods for dealing with class imbalance. Proposed to reduce the overfitting that can result from ROS, it creates new synthetic samples by interpolating between minority samples and their nearest neighbors, which leads to the

introduction of synthetic examples along the “line segments” connecting minority class samples. Because it creates new samples instead of duplicating existing ones, as done in ROS, it leads to the expansion of the minority decision region and to lower overfitting.

However, SMOTE also has relevant drawbacks: interpolation near class boundaries can create points that fall into majority-class regions, introducing noisy samples (referred to as the problem of “overgeneralization”); the generated synthetic points concentrate where minority examples already cluster, reinforcing local arrangement of minority points rather than the full minority class support; and the linear interpolation may introduce artificial linear patterns not present in the original distribution. To attempt overcoming some of these limitations, several SMOTE adaptations have been proposed on the literature. In our study, we include Borderline-SMOTE, Safe-Level SMOTE, and Cluster-SMOTE.

### 3.3.4 Borderline-SMOTE

Borderline-SMOTE [17] generates synthetic samples based on local neighborhood structure, rather than uniformly across all minority points as in SMOTE. It focuses on creating new minority samples near the decision boundary, given their importance and higher chance of being misclassified, while also avoiding the introduction of samples in “noisy” and in “safe” regions. To identify these regions, the method examines the class composition of the nearest neighbors of each minority sample. If all nearest neighbors belong to the majority class, the sample is considered noise; if fewer than half belong to the majority class, it is considered safe; otherwise, it is classified as being in danger, meaning that at least half, but not all, of its nearest neighbors belong to the majority class. Only samples in the danger region are used to generate synthetic examples.

There are two variants of Borderline-SMOTE. In this study, we use type 1, as it is the default in the library used and more similar to the original SMOTE and other SMOTE adaptations. The difference between both variants is which samples are used for nearest neighbor computation: type 2 is more aggressive by using samples of both classes, while type 1 only uses minority class samples.

The main advantages of Borderline-SMOTE are a reduced overgeneralization (one of SMOTE’s drawbacks) and a better-defined decision boundary, which may improve classification performance. Nevertheless, the other two problems related to SMOTE, namely artificial linear patterns and local arrangement reinforcement, remain unsolved. An additional issue is the fact that nearby instances might be attributed to different regions (noise,

danger, or safe), which makes the method sensitive to small local variations and potentially leads to the misclassification of informative samples.

### 3.3.5 Safe-Level SMOTE

Differently than Borderline-SMOTE, Safe-Level SMOTE [58] focuses on creating new samples in “safe” regions, where there is a higher density of minority samples. The main goal is to avoid generating noisy samples. This is achieved by using the safe-level (SL) and safe level ratio values to define if or where the new samples should be generated. The SL value corresponds to the number of minority instances in the nearest neighbors, and the safe level ratio corresponds to the SL of the selected minority instance p divided by the SL of its nearest neighbor n. If the SL(p) = SL(n) = 0, no sample is generated. If SL(p) is not 0, then the safe level ratio is used to define the limits of the interpolation, in order to place the new sample closest to the “safest” sample.

In comparison with Borderline-SMOTE type 1 and SMOTE, Safe-Level SMOTE provides a more effective mitigation of noisy sample generation. In addition, it directs the placement of new samples toward safer regions, which is not done in Borderline-SMOTE type 1 or SMOTE. In particular, this mechanism directly addresses the sensitivity of Borderline-SMOTE’s discrete region assignments, aiming for a more consistent generation of samples. However, Safe-Level SMOTE still inherits the artificial linear patterns and local arrangement reinforcement issues from SMOTE, and, by favoring safer regions, may generate fewer samples near the decision boundary, potentially weakening its representation.

### 3.3.6 Cluster-SMOTE

Cluster-SMOTE [59] takes a more distinct approach than the previous SMOTE adaptations. First, the minority samples are grouped into clusters using the K-Means algorithm, and then SMOTE is applied within each cluster. The goal is to perform oversampling locally within homogeneous regions of the minority class, which can help preserve subcluster structure and improve learning in cases where the minority class is unevenly distributed.

By restricting synthetic generation to intra-cluster neighbors, Cluster-SMOTE reduces the generation of noisy samples, in particular those arising between minority class subregions, which are not explicitly addressed by Borderline-SMOTE or Safe-Level SMOTE.

In addition, it better preserves the density of clusters and reduces the impact of linear interpolation artifacts by restricting interpolation to local regions. However, its effectiveness depends on the quality of the clustering step, which itself also increases computational cost and reinforces the local arrangement issue found in SMOTE.

### 3.3.7 SMOTE-ENN and SMOTE-Tomek

SMOTE-ENN and SMOTE-Tomek [60] are hybrid methods, which means they combine oversampling with undersampling. In both, SMOTE is performed first, and the cleaning method is performed after. Unlike the previously presented standalone methods, here the ENN and Tomek Links algorithms clean both the minority and majority class samples (as per the library default). The goal is to clean the noisy samples that might have been created by SMOTE on the interpolation process, usually between clusters of same-class data.

Given that there is no modification on the SMOTE method, the same issues of SMOTE persist, with the exception of overgeneralization, which the cleaning process directly aims to solve. In contrast to Borderline-SMOTE, Safe-Level SMOTE, and Cluster-SMOTE, which modify where and how synthetic samples are generated, SMOTE-Tomek and SMOTE-ENN operate after generation and focus on cleaning the resulting dataset rather than controlling the generation process. This separation between generation and cleaning may lower the risk of over-constraining the minority distribution and the sensitivity to data structure assumptions, but may be insufficient to generate higher quality samples (e.g. near the boundary samples, as in Borderline-SMOTE).

### 3.3.8 Conditional Tabular GAN

As mentioned, the Conditional Tabular GAN (CTGAN) [28] is a type of GAN used for generating tabular data. It addresses challenges that arise when applying GANs to tabular data, namely non-gaussian distributions, multimodal distributions, sparse one-hot-encoded vectors and highly imbalanced categorical columns. These are handled by the conditional generation and mode-specific normalization mechanisms. The conditional generation handles the imbalance present on discrete columns, ensuring that samples from rare categories can be generated effectively and avoiding learning from sparse one-hot-encoded vectors; the mode-specific normalization handles the non-gaussian and multimodal distributions by modeling continuous features as a mixture of Gaussian distributions and transforming them into a normalized latent space.

While CTGAN was not primarily created for dealing with target class imbalance, its conditional generation mechanism can also be used for generating samples of the underrepresented class, therefore acting as an oversampling method. Prior work has explored this possibility [61–64], with CTGAN showing superior performance over SMOTE in some works, and inferior performance in others. However, the important issue with CTGAN utilization is the GPU usage, which increases substantially its energy consumption.

## 3.4. Classification Algorithms

To compare how different resampling techniques interact with models of different complexity, we selected Logistic Regression and Random Forest. Logistic Regression was chosen as a simple model, while Random Forest is a widely used, more complex model. Both were used with their default hyperparameters, as no tuning was performed. Our goal is to achieve a fairer comparison and to isolate the energy consumption to the resampling and training phases, excluding the contribution of hyperparameter optimization.

## 3.5. Tools, Hardware, and Evaluation Metrics

The tool used to measure energy consumption was pyJoules, as previously discussed in Section 2.3. As for the underlying hardware and software, all experiments were conducted on a machine with the following specifications:

- CPU: Intel Core i5-13500H (12 cores, 16 threads, up to 4.7 GHz)
- RAM: 16GB, DDR5, 5200 MT/s
- GPU: NVIDIA GeForce RTX 4060 Max-Q / Mobile (8 GB VRAM, 60W TGP, Driver Version: 595.58.03, CUDA version: 13.2)
- OS: EndeavourOS Linux x86_64, Kernel 6.19.11-arch1-1

The main evaluation metric used was the F1-score (Equation 3.1), which favors the true positives (TP), i.e., the correct identification of the minority (positive) class while balancing precision and recall, making it suitable for imbalanced classification. We also used recall (Equation 3.2) to further investigate the behavior of the CTGAN method.

$$F1\text{-}score = \frac{2 \times precision \times recall}{precision + recall} = \frac{TP}{TP + 1/2 \times (FP + FN)} \tag{3.1}$$

$$Recall = \frac{TP}{TP + FN} \tag{3.2}$$

## 3.6. Workflow

We divided our study into two steps. First, we compared traditional resampling techniques, and then we extended our analysis to include CTGAN. In the first group of methods, we conducted a direct comparison between performance and energy; however, with CTGAN, we focused solely on comparing performance. This is because differences in underlying hardware and algorithms make a direct comparison unreasonable. Moreover, since CTGAN differs from traditional methods in terms of preprocessing, execution time, and implementation, we made some adjustments to the initial pipeline, resulting in the use of two distinct pipelines. To ensure that our energy measurements were valid and as realistic as possible, we took the following aspects into consideration:

1. RAPL energy counters have limited resolution, which leads to noisy readings for fast-execution code snippets such as resampling and model fitting. To enhance the signal, we run the resampling and fitting in a loop and average the results. After testing with 10, 100, and 1000 iterations, we chose 100, as it proved stable.

2. RAPL monitors energy usage across all processes on the machine. Although all applications were closed, this does not guarantee the elimination of fluctuations due to potential interference from background processes. To mitigate this effect, experiments (dataset, algorithm, resampling technique) were repeated 10 times in a random order. The repetitions (“runs”) also served as way to capture the variability from random processes in resampling methods, with each run using a different random seed specified in the method’s *random_state* parameter.

3. As noted by Verdecchia et al. [4], the hardware temperature may also be a source of bias. As such, prior to the execution of the experiments, a CPU-intensive warm-up is performed by computing a Fibonacci sequence, followed by a 1-second sleep to let the CPU settle. This avoids a “cold boot” state. In addition, a 0.05-second sleep is introduced before each 100x fitting loop to help stabilize the RAPL counter.

4. Some models and methods support parallel execution, reducing computation time by leveraging multiple processors. Since this capability is not available for all methods, enabling it would compromise the fairness of the comparison. Moreover, parallel execution may introduce additional variability in energy measurements, making

them less stable and harder to interpret. Therefore, all methods and models were configured to use a single processor.

The main pipeline used in our experimental study is illustrated in Figure 3.1. After generating and shuffling the experiments and completing the warm-up phase, we begin the loop of experiments. For each experiment, we apply a 5-fold stratified cross-validation split. Each split is preprocessed and sequentially passed to a function that performs resampling and training, while also measuring energy consumption and time.

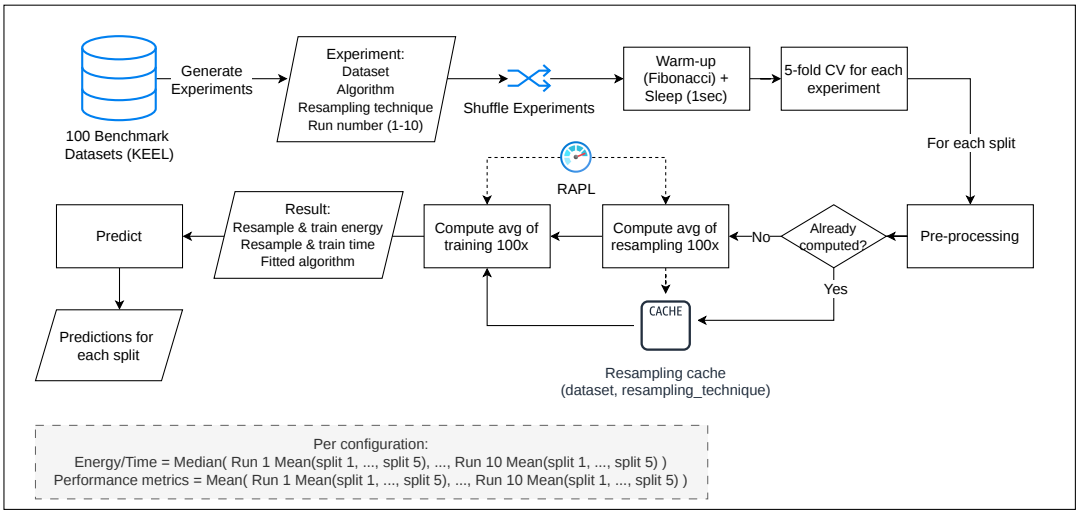



Figure 3.1: Experimental pipeline used to compare traditional resampling techniques.

Within this function, we first check a cache to determine whether the resampling energy and duration have already been computed for the corresponding dataset-resampling combination. This was done to reduce the pipeline's total execution time, under the assumption that energy and duration do not vary substantially between runs and folds, which is generally reasonable for most traditional methods. To validate this assumption, we analyzed the variability of Borderline-SMOTE, as its computation is more sensible to samples in each split. The results showed that training energy exhibited a higher coefficient of variation across folds (40%) compared to resampling (13%), and that the coefficient of variation across runs for resampling energy was 5%. Added to the fact that the training energy also includes the effect of the resampling (by adding/removing samples), and accounts for 88.0% of the total energy (median), we decided to keep the caching strategy. Note that training energy is never cached.

After obtaining both the measurements and the fitted model, predictions are performed on the corresponding test fold. Performance is evaluated using the F1-score. The final

F1 per configuration (dataset, resampling technique, algorithm) is computed as the mean across runs, with each run's value corresponding to the mean across its folds, while the final energy and time are computed as the median across runs, with each run again corresponding to the mean across its folds. For energy/time, the median was used to exclude noisy readings, whereas for F1, the mean was appropriate, as variations across runs represent legitimate outcomes of the stochastic learning process.

The second pipeline, tailored to fit the specifics of CTGAN, is similar to the previous one (Fig. 3.2). The modifications lie in when preprocessing is performed, in the usage of NVML for GPU measurement and in the strategies for energy validity. On the traditional methods' pipeline, the preprocessing is done before the resampling, since their implementation requires data to be only numerical. In this pipeline however, given the CTGAN handles nominal data (one of its main strengths), the preprocessing is applied after the resampling, before being sent to the models. Next, to make use of NVML, we added the GPU device to the list of the devices being measured in pyJoules. Finally, the 100x loop was removed as resampling execution time is no longer below the RAPL and NVML measurement resolution, and the cache was removed since resampling energy now accounts for a larger portion of total energy and exhibits greater variability.

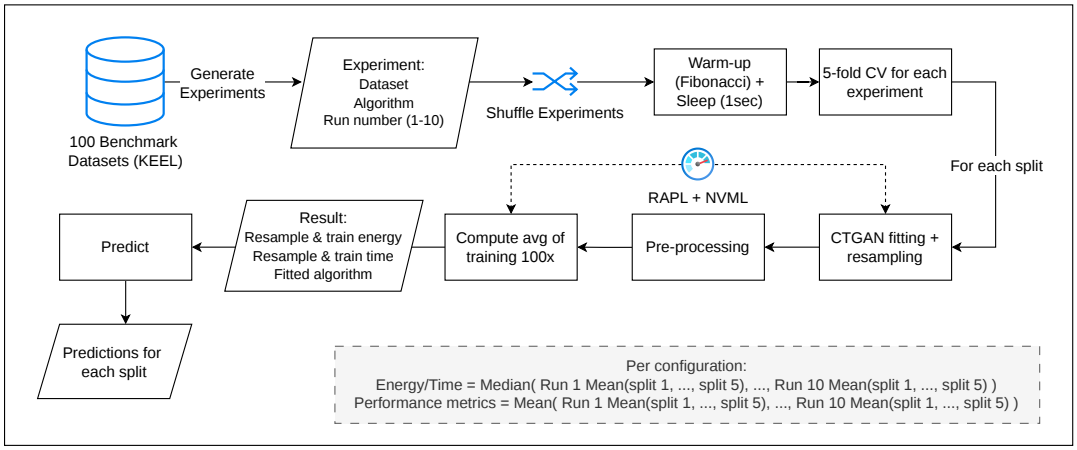



Figure 3.2: Experimental pipeline with modifications to accommodate CTGAN.

# 4. Results and Discussion

In this section, we present the results for the traditional resampling methods and the CTGAN approach, addressing **RQ1** and **RQ2**, respectively. For each case, we analyze how data complexity impacts the obtained results, answering **RQ3**. For the comparisons, we focused exclusively on energy consumption, as it provides a more interpretable measure of environmental impact. Unlike execution time, which only reflects how long a computation takes, energy captures the actual resource usage of the system by accounting for both runtime and power consumption. This makes it a more meaningful metric for comparing methods with different hardware utilization patterns and for assessing their potential environmental cost. Furthermore, execution time was found to be almost perfectly correlated with energy consumption ($R = 0.99841$ for the traditional methods and $R = 0.99785$ for CTGAN), providing no additional information beyond the energy measurements.

An additional insight from the strong correlation obtained for the traditional methods is that average power consumption showed minimal variation across them, causing differences in energy consumption to be driven mainly by execution time. This indicates that the traditional resampling methods do not vary substantially in terms of hardware utilization patterns.

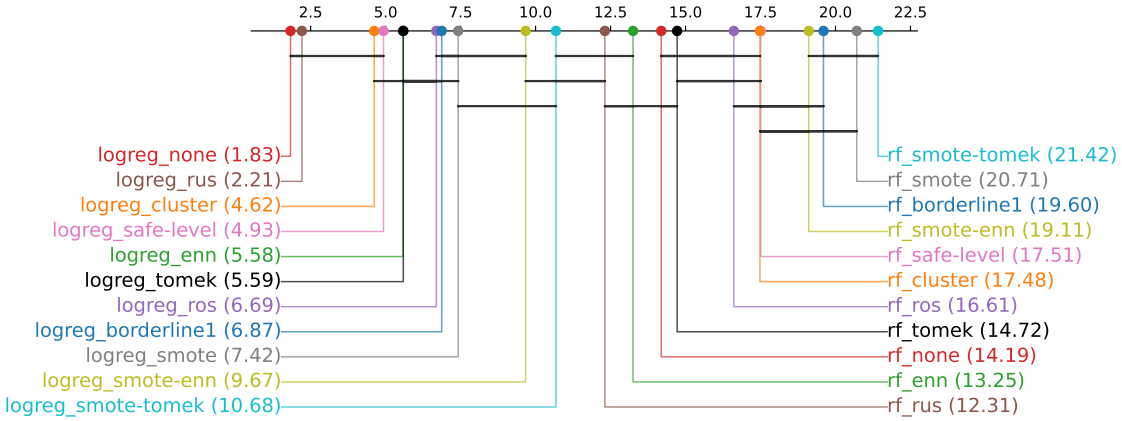
To compare the results of each resampling technique and model combination, we used the Friedman test [65], a non-parametric statistical test for detecting differences across multiple conditions applied to the same set of subjects (datasets). Each condition in this case was a combination of a resampling method and a modeling technique (Random Forest or Logistic Regression), and, to serve as a baseline, we also included the results of both these models combined with the original imbalanced dataset, denoted as *rf_none* and *logreg_none*. Upon verifying statistical significance, we applied the Nemenyi post-hoc test, which identifies which pairs of methods differ significantly.

## 4.1. General Results

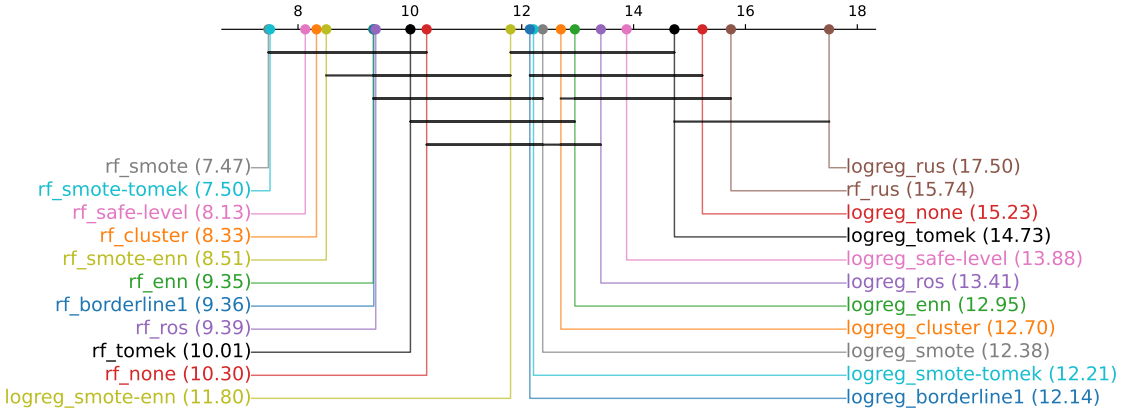
Figure 4.1 presents total energy (resampling + training energy) and F1 critical difference (CD) diagrams based on the Nemenyi post-hoc test for the comparison across all datasets, not distinguishing between complexity levels. In the energy plot, a lower ranking corresponds to a lower energy consumption, while in the F1 plot, a lower ranking corresponds to a higher F1 score. Each color represents a resampling method, with every

color appearing once for Random Forest and once for Logistic Regression. The numbers in the parenthesis following the methods represent the rankings rounded to 2 decimals. For connections that are harder to visualize, refer to Table C.2 in Appendix C, which show the pairwise rank differences between all methods.

A remark regarding the results is that the resampling methods Tomek Links, Edited Nearest Neighbors, and Borderline-SMOTE, due to their algorithmic behavior, returned the unchanged original dataset for some folds and datasets (usually not in all folds of the same dataset). For Tomek Links, this occurred in 43 datasets; for ENN, in 16 datasets; and for Borderline-SMOTE, in 7 datasets. For the latter two methods, the affected datasets include one with medium complexity and the remaining ones with low complexity. For Tomek Links, no clear relationship with complexity is observed, with 28 low-, 10 medium-, and 5 high-complexity datasets affected.




(a) Total energy CD diagram.




(b) F1-score CD diagram.

Figure 4.1: CD diagrams for energy and F1-score across all datasets (100 datasets). CD: 3.2995.

It is noticeable that methods that achieve better F1 scores tend to rank worse on energy

consumption, and vice versa. Generally, that would indicate that better-performing methods consume more energy, while worse-performing methods consume less. However, the significance of the rank differences needs to be considered, requiring a method-by-method analysis to identify clear advantage scenarios.

Another very noticeable characteristic of the plots is the almost complete separation between Random Forest and Logistic Regression. This shows that the model used not only impacts F1, but also energy. Random Forest, being a more complex model, results in higher energy consumption, whereas Logistic Regression consumes less. However, there is one important exception. For Random Forest with Random Undersampling (*rf_rus*), the F1 results place it in the worst-performing group, with significantly lower performance than four Logistic Regression configurations (*logreg_smote*, *logreg_smote-enn*, *logreg_smote-tomek*, and *logreg_borderline1*), and no significant difference compared to *logreg_rus*. Regarding energy consumption, *rf_rus* also performs poorly, showing significantly higher energy than *logreg_rus* and *logreg_borderline1*, and no significant difference from the other Logistic Regression configurations mentioned. Overall, these results indicate that *rf_rus* is a poor solution, as multiple methods achieve better performance at comparable energy or similar performance at lower energy.

The poor performance of the combination between Random Forest and Random Undersampling can be explained by the interaction between Random Forest's bootstrap mechanism and RUS's structure-agnostic reduction. Random Forest builds each tree using bootstrap samples from the original data distribution. In turn, RUS removes a large portion of majority-class instances without preserving their original distribution, which may lead to the generation of biased base learners that provide a poor approximation of the true decision boundary. Consequently, the benefit of the majority voting mechanism is reduced, as predictions are derived from biased trees rather than complementary views of the data distribution.

Given that the F1 diagram shows more overlap than the energy diagram, most observed dominance cases arise when methods have similar F1 performance but significantly different energy consumption. For example, *rf_none* is not significantly different from *rf_smote* in terms of F1 (Fig. 4.1b), but consumes significantly less energy (Fig. 4.1a). In this case, Random Forest without resampling should be preferred due to its lower energy consumption. Similar patterns are observed for *logreg_smote* compared to *rf_enn* and *rf_tomek*. In contrast, the opposite situation (differences in F1 without corresponding differences

in energy) is only observed in the *rf_rus* case. Multiple combinations show that simple models with more complex resampling can yield better solutions than complex models with simple resampling or no resampling, addressing the question posed in **RQ1**. Many pairs serve as examples, such as *logreg_smote* and *rf_none*, *logreg_smote* and *rf_rus*, *logreg_smote-tomek* and *rf_rus*, and *logreg_smote-enn* and *rf_enn*.

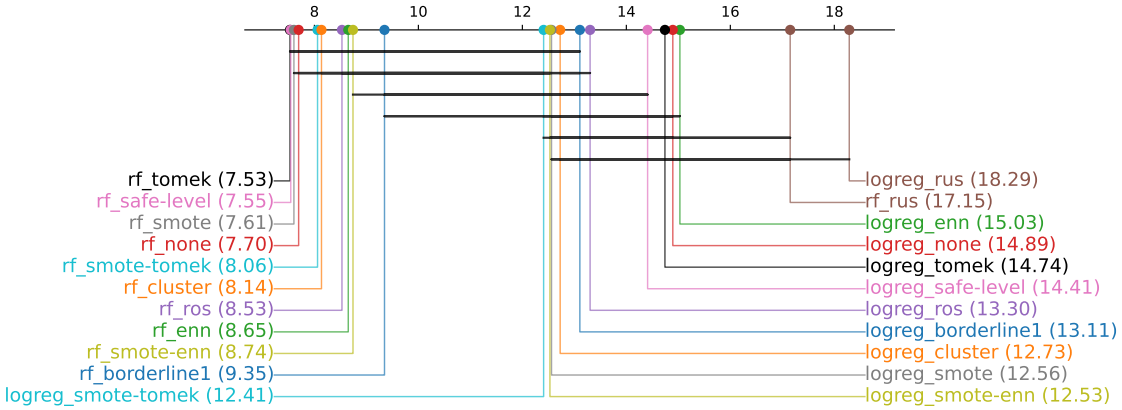
An extra insight can be drawn by comparing baseline and undersampling settings. Unlike "pure reduction" techniques (sample removal regardless of class, without a rebalancing effect), which have been shown to lower computational cost in other data-centric green AI studies [4], none of the undersampling methods were able to do the same. That is, no undersampling method consumed less energy than the baseline (no resampling) for the same model. This may be explained by the varying degrees of reduction achieved by undersampling techniques, which depend on the imbalance level of each dataset rather than on a fixed reduction ratio as used in class-agnostic reduction techniques. The impact of reduction is further diminished by the generally small dataset sizes (most have fewer than 2000 samples).

## 4.2. Complexity Analysis

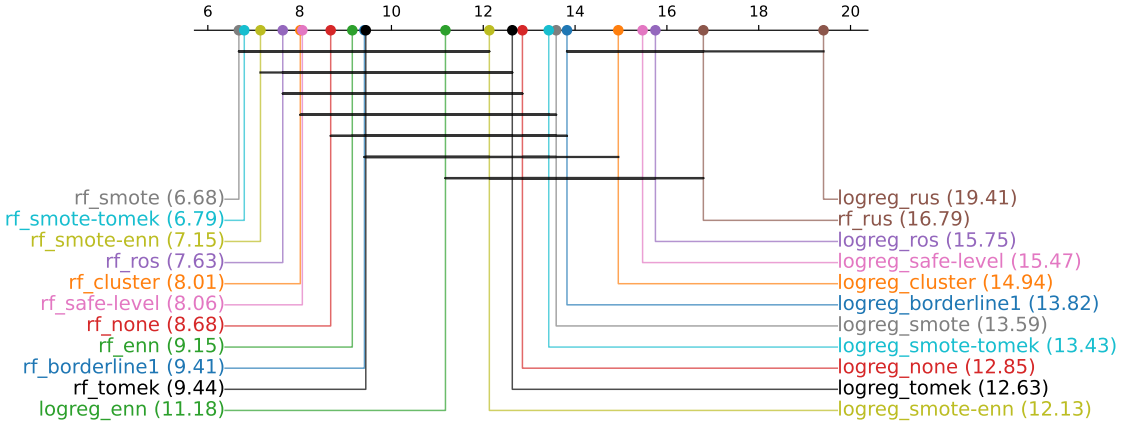
Figure 4.2 shows the F1 critical difference diagrams per dataset complexity. Note that the critical difference values are higher than the critical difference of the tests using all datasets (Fig. 4.1), as there are fewer datasets for each Friedman test. As a result, there are fewer significant differences.

Regarding energy, all three energy diagrams were very similar to Fig. 4.1a, with only minor ranking changes and fewer significant differences. They can be consulted in the Appendix C (Figure C.1 and Tables C.3-C.5). This leads to an important conclusion: complexity does not meaningfully affect the comparison of energy consumption across resampling methods.

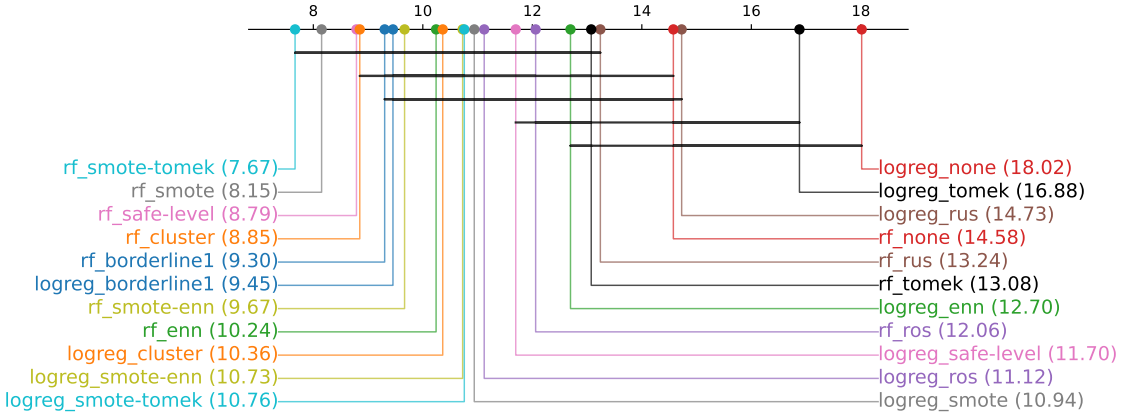
Low-complexity F1 results (Fig. 4.2a) are not very different from the general results (Fig. 4.1b). Although there were a few ranking changes, those mostly did not translate to new significant different pairs. A secondary insight is that, for some methods, if we compare the Logistic Regression and Random Forest versions, a significant difference is observed in the general results, but not in the low-complexity setting (e.g. SMOTE, SMOTE-Tomek and Cluster SMOTE). This suggests that the performance gap between simpler and more complex classification models decreases for low-complexity datasets.




(a) F1-score CD diagram for low-complexity datasets (33 datasets). CD: 5.7437.




(b) F1-score CD diagram for medium-complexity datasets (34 datasets). CD: 5.6586.




(c) F1-score CD diagram for high-complexity datasets (33 datasets). CD: 5.7437.

Figure 4.2: CD diagrams for F1-score by dataset complexity groups. For connections that are harder to visualize, refer to Tables C.3-C.5 in Appendix C.

In the medium-complexity CD diagram (Fig 4.2b), the highlight is *logreg_enn*. It ranks higher than in the general and low-complexity CD diagrams and is equivalent to top methods such as *rf_smote* and *rf_smote-tomek*. In addition, it consumes less energy than all Random Forest combinations, making it more advantageous than all Random Forest settings with equivalent performance. When compared with the low-complexity result, this *logreg_enn* advantage gain could be due to ENN having more noisy samples to clean, making it more effective.

Finally, in the high-complexity CD diagram, many methods are connected, as in the low-complexity diagram. This contrasts with the expectation that high-complexity datasets would lead to greater separation between methods and that more complex approaches would perform better. However, there is a clear divide between oversampling and undersampling/no-resampling methods, with oversampling methods generally achieving higher ranks. In addition, *rf_smote*, *rf_smote-tomek* and *rf_safe-level* are significantly better than *rf_none*, and all oversampling methods are significantly better than *logreg_none*. This indicates that, unlike in medium- and low-complexity datasets, there is a significant difference between resampling (oversampling) the data and not resampling it in high-complexity datasets. Given this setting, a trade-off emerges: the no-resampling combinations are still better in terms of energy, but are now worse in terms of F1. Here, advantageous scenarios arise from using Logistic Regression instead of Random Forest with the same resampling method (this is evident from the color pairs, which are closer to each other than in the other diagrams). Examples of this are the pairs, *rf_borderline1* vs. *logreg_borderline1* and *rf_cluster* vs. *logreg_cluster*. This suggests that, for highly complex datasets, opting for a simpler model combined with an oversampling method is more beneficial than using the same resampling method with a more complex model, as the former consumes less energy and does not lower performance.

## 4.3. CTGAN

In this section, we address **RQ2** by analyzing how a deep learning approach (CTGAN) compares to traditional methods in terms of energy and performance. Looking at the energy, we compare the total energy median across all traditional methods with that of CTGAN across all datasets: **0.66J** for the traditional methods versus **140.89J** for CTGAN. The CTGAN consumes a little over 200$\times$ higher energy than the traditional methods. Although both values look small when compared to daily activities such as phone charging,

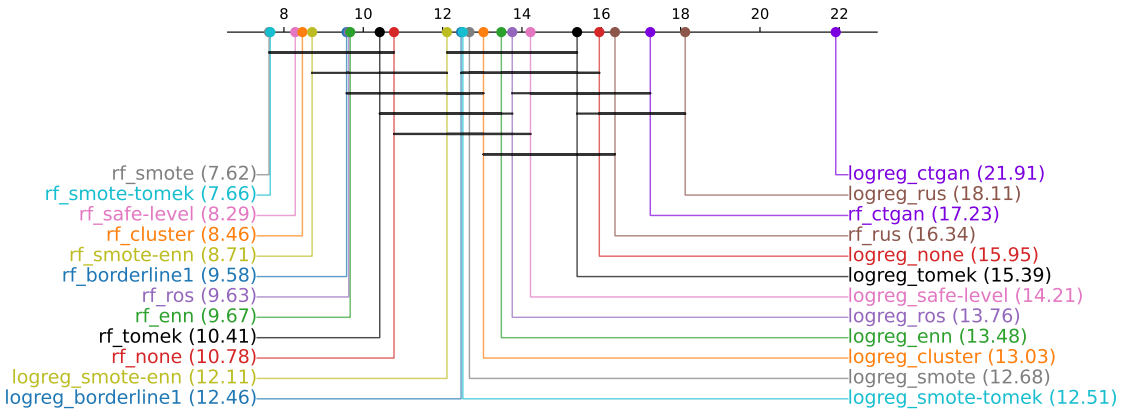



Figure 4.3: CD diagram for F1 across all datasets (100 datasets), with CTGAN results included. CD: 3.6373.

which would consume 68400J approximately[1], in practice, tuning, ML experimentation and retraining could multiply these costs substantially.

Still regarding energy, a major difference between the traditional methods and the CTGAN is what phase contributes the most to total energy. For the traditional methods, the median percentage of contribution is 12% for resampling and 88% for training, while for the CTGAN, it is 99.8% for resampling and 0.02% for training. This confirms expectations, as deep learning methods generally consume more resources than traditional ones.

In terms of F1-score, the CTGAN did not show superior results. Figure 4.3 illustrates this scenario: *logreg_ctgan* performed significantly worse than all other combinations, while *rf_ctgan* achieved performance equivalent to the worst traditional methods. In the case of Logistic Regression, convergence failed in some runs and folds for the datasets *kddcup-buffer_overflow_vs_back* and *kddcup-rootkit-imap_vs_back*. The difference between the two models is expected: CTGAN tends to perform worse with Logistic Regression because synthetic samples can distort global linear relationships, while Random Forests are more robust to local noise and distributional shifts introduced by generative models, due to their ensemble mechanism and non-linear decision boundaries.

To better understand what may have caused these results, we investigated the quality of synthetic samples from a subset of datasets in each complexity group using the sdmetrics library [66]. The results showed that although CTGAN preserved individual feature distributions reasonably well, it had greater difficulty maintaining inter-variable dependencies. This likely resulted in synthetic minority samples that did not accurately reflect the

[1]Value calculated from `https://agirpourlatransition.ademe.fr/particuliers/economiser/energie/consommation-appareils-menagers`

true minority decision regions, leading to lower F1 scores. This issue is aggravated by the fact that CTGAN generates minority samples by approximating the overall minority distribution, without explicitly targeting borderline, overlapping, or noisy regions, unlike advanced traditional methods. Other factors might also have contributed: no tuning was performed, which may have limited the achievement of optimal results; most datasets were small, which makes it harder for the neural network to learn the underlying distribution; and our results were averaged across different seeds to ensure a fair and more realistic comparison, unlike studies that report results based on the best-performing seed.

The results for different levels of complexity did not substantially differ from those shown in Figure 4.3, with the only major distinction being that *logreg_ctgan* was equivalent to more methods (as seen in Figure C.2, in Appendix C). This pattern appeared across all complexity levels, which is probably due to the increase in the critical difference value.

To further investigate CTGAN's results, we conducted extra Friedman tests over recall (Fig. 4.4). Both *logreg_ctgan* and *rf_ctgan* performed better in this metric. The latter performed significantly better than several methods and was not different than *rf_smote*, for example. These results were consistent across all complexity levels, as can be seen in Figure C.3 of Appendix C. This suggests that the drop in F1-score is mainly due to lower precision, indicating an increase in false positives. Therefore, although the generated samples helped identify more true positives, they may also have introduced noisy or borderline minority instances, leading to over-prediction of the positive class.

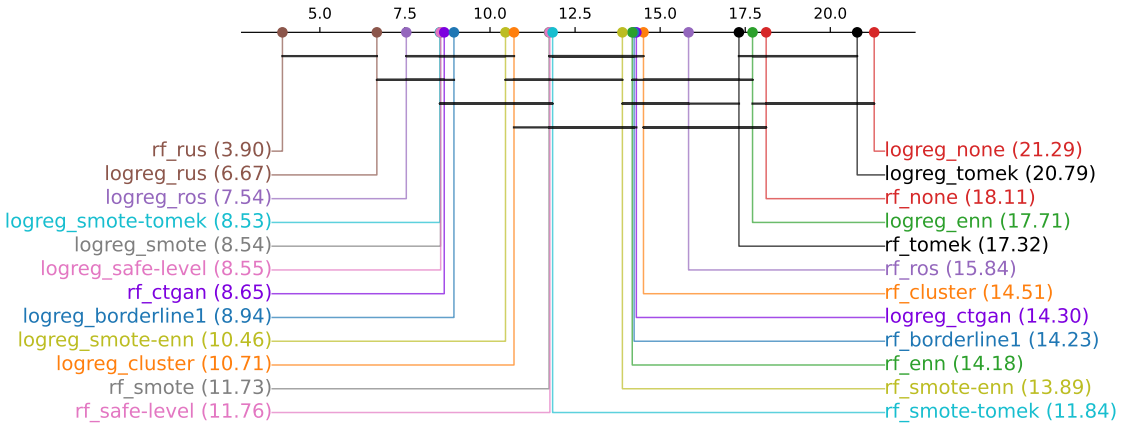



Figure 4.4: CD diagram for recall across all datasets (100 datasets), with CTGAN results included. CD: 3.6373.

Based on these findings, traditional methods seem more advantageous than CTGAN for both energy and performance when resampling small to medium-sized datasets with the established defaults. This conclusion remains true for all dataset complexities. More

work is needed to assess whether this conclusion remains valid when performing tuning, although this would further increase CTGAN's energy disadvantage. In addition, a study with larger datasets would be needed to obtain more general conclusions.

# 5. Conclusion

In this dissertation, we address the intersection between data-level methods for imbalanced learning and green AI. Despite data-level methods being a established field of research, there is a lack of studies regarding the efficiency of these methods, specially when referring to energy consumption and environmental impact. This dissertation fills that gap by exploring how a representative set of resampling methods compare in terms of energy consumption and the potential trade-offs with performance. In order to strengthen our conclusions and provide more useful suggestions, we also analyzed how dataset complexity influences these comparisons.

## 5.1. Results and Contributions

The main contribution of our work is the first benchmark study of resampling methods considering both classification performance and energy consumption. In addition, we obtained the following conclusions, related to the objectives established in the introduction:

- The thorough characterization of the literature highlighted the lack of studies on the relationship of green AI and data-centric approaches for imbalanced classification, as well as important gaps, including the limited explicit framing of efficiency within a green AI perspective, inconsistent reporting of efficiency metrics, and the tendency for efficiency improvements to be motivated by domain-specific constraints rather than sustainability concerns. The analysis of pairwise intersections also revealed that the connection between imbalanced classification and green AI is dependent on the inclusion of the data-centric aspect, as no work was found on the intersection of green AI and algorithmic-level imbalanced classification;
- The comprehensive energy-performance analysis of well-established resampling methods showed a general trade-off between performance and energy consumption, with more complex methods typically achieving better performance at a higher energy cost. However, several methods exhibited comparable performance while consuming significantly different amounts of energy, making lower-energy alternatives preferable. The analysis of CTGAN as a deep learning-based oversampling approach showed that, without hyperparameter tuning, traditional resampling methods seem to outperform CTGAN both in terms of performance and energy consumption on small to medium tabular datasets;

- The study on the influence of data complexity demonstrated that oversampling approaches tend to outperform undersampling and no-resampling approaches for high-complexity datasets, a trend not observed for low- and medium-complexity datasets. In addition, for those datasets, opting for a simpler model combined with an oversampling method is often more beneficial than using the same resampling method with a more complex model. With that, we are able to provide insights into the sustainable use of resampling methods based on dataset complexity.

## 5.2. Limitations and Future Work

It is important to acknowledge some limitations of this study. First, the usage of primarily small datasets may affect the difference in performance between traditional methods and CTGAN. Secondly, the inclusion of only one deep learning-based approach limits the range of the obtained conclusions. Finally, the lack of hyperparameter tuning may affect the results in terms of performance and prevents us from analyzing the impact of the ML pipeline as a whole.

Future work could focus on exploring additional GAN-based methods, such as SA-CGAN, Tabular Variational Autoencoder (TVAE) [28], GAN-SMOTE [67] and more privacy focused methods such as MedGAN [68]. This would allow for assessing if the poor performance of CTGAN is a reflection of GAN-based oversamplers as a whole or an individual characteristic, and for selecting the best GAN-based method in terms of energy-performance trade-off among different methods. In addition, an extension study including hyperparameter tuning and a larger, more diverse set of datasets could be conducted to strengthen the obtained conclusions, uncover more significant differences, and provide a more representative assessment of the environmental impact. Ultimately, the results of this work could support the development of energy-sensitive systems, in which resampling techniques may be selected based on dataset characteristics and predefined energy-performance trade-off objectives.

# References

[1] R. Schwartz, J. Dodge, N. A. Smith, and O. Etzioni, “Green ai,” *Commun. ACM*, vol. 63, no. 12, p. 54–63, 2020. [Online]. Available: https://doi.org/10.1145/3381831 [Cited on pages 1, 3, and 14.]

[2] E. Strubell, A. Ganesh, and A. McCallum, “Energy and policy considerations for deep learning in NLP,” in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2019, pp. 3645–3650. [Online]. Available: https://doi.org/10.18653/v1/P19-1355 [Cited on pages 1 and 3.]

[3] M. H. Jarrahi, A. Memariani, and S. Guha, “The principles of data-centric ai,” *Communications of the ACM*, vol. 66, no. 8, pp. 84–92, 2023. [Online]. Available: https://doi.org/10.1145/3571724 [Cited on pages 1, 4, 5, 9, and 10.]

[4] R. Verdecchia, L. Cruz, J. Sallou, M. Lin, J. Wickenden, and E. Hotellier, “Data-centric green ai an exploratory empirical study,” in *2022 International Conference on ICT for Sustainability (ICT4S)*. IEEE, 2022, pp. 35–45. [Online]. Available: https://doi.org/10.1109/ICT4S55073.2022.00015 [Cited on pages 1, 8, 15, 27, and 33.]

[5] S. Salehi and A. Schmeink, “Data-centric green artificial intelligence: A survey,” *IEEE Transactions on Artificial Intelligence*, vol. 5, no. 5, pp. 1973–1989, 2024. [Online]. Available: https://doi.org/10.1109/TAI.2023.3315272 [Cited on pages 1, 4, and 8.]

[6] P. Branco, L. Torgo, and R. P. Ribeiro, “A survey of predictive modeling on imbalanced domains,” *ACM Computing Surveys*, vol. 49, no. 2, pp. 1–50, 2016. [Online]. Available: https://doi.org/10.1145/2907070 [Cited on pages 1 and 5.]

[7] A. Fernández, S. García, M. Galar, R. C. Prati, B. Krawczyk, and F. Herrera, *Learning from Imbalanced Data Sets*. Springer International Publishing, 2018. [Online]. Available: https://doi.org/10.1007/978-3-319-98074-4 [Cited on pages 1, 5, and 22.]

[8] V. Bolón-Canedo, L. Morán-Fernández, B. Cancela, and A. Alonso-Betanzos, “A review of green artificial intelligence: Towards a more sustainable future,”

*Neurocomputing*, vol. 599, p. 128096, 2024. [Online]. Available: https://doi.org/10.1016/j.neucom.2024.128096 [Cited on pages 3 and 4.]

[9] R. Verdecchia, J. Sallou, and L. Cruz, "A systematic review of green ai," *WIREs Data Mining and Knowledge Discovery*, vol. 13, no. 4, 2023. [Online]. Available: https://doi.org/10.1002/widm.1507 [Cited on page 3.]

[10] F. Kamalov, S. Elnaffar, Z. E. Khatib, A. K. Cherukuri, and A. Jonnalagadda, "Undersampled random forest: A green approach to imbalanced learning," in *2024 Third International Conference on Sustainable Mobility Applications, Renewables and Technology (SMART)*. IEEE, 2024, pp. 1–7. [Online]. Available: https://doi.org/10.1109/SMART63170.2024.10815385 [Cited on pages 12 and 14.]

[11] D. Sanderson and T. Kalganova, "Identifying suitability for data reduction in imbalanced time-series datasets," *AI*, vol. 6, no. 5, p. 98, 2025. [Online]. Available: https://doi.org/10.3390/ai6050098 [Cited on pages 12 and 15.]

[12] S. S. Meriem Amina, B. Abdolkhalegh, N. K. Khoa, and C. Mohamed, "Featuring real-time imbalanced network traffic classification," in *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. IEEE, 2018, pp. 840–846. [Online]. Available: https://doi.org/10.1109/Cybermatics_2018.2018.00163 [Cited on pages 4, 11, and 14.]

[13] D. Zha, Z. P. Bhat, K.-H. Lai, F. Yang, and X. Hu, *Data-centric AI: Perspectives and Challenges*. Society for Industrial and Applied Mathematics, 2023, pp. 945–948. [Online]. Available: https://doi.org/10.1137/1.9781611977653.ch106 [Cited on page 4.]

[14] S. Das, S. S. Mullick, and I. Zelinka, "On supervised class-imbalanced learning: An updated perspective and some key challenges," *IEEE Transactions on Artificial Intelligence*, vol. 3, no. 6, pp. 973–993, 2022. [Online]. Available: https://doi.org/10.1109/TAI.2022.3160658 [Cited on page 6.]

[15] W. Chen, K. Yang, Z. Yu, Y. Shi, and C. L. P. Chen, "A survey on imbalanced learning: latest research, applications and future directions," *Artificial Intelligence Review*, vol. 57, no. 6, 2024. [Online]. Available: https://doi.org/10.1007/s10462-024-10759-6 [Cited on page 6.]

[16] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “Smote: synthetic minority over-sampling technique,” *J. Artif. Int. Res.*, vol. 16, no. 1, p. 321–357, 2002. [Online]. Available: https://dl.acm.org/doi/10.5555/1622407.1622416 [Cited on pages 6, 9, and 22.]

[17] H. Han, W.-Y. Wang, and B.-H. Mao, *Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning*. Springer Berlin Heidelberg, 2005, pp. 878–887. [Online]. Available: https://doi.org/10.1007/11538059_91 [Cited on pages 6 and 23.]

[18] H. He, Y. Bai, E. A. Garcia, and S. Li, “Adasyn: Adaptive synthetic sampling approach for imbalanced learning,” in *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*. IEEE, 2008, pp. 1322–1328. [Online]. Available: https://doi.org/10.1109/IJCNN.2008.4633969 [Cited on page 6.]

[19] J. Xu, W. Zhou, Z. Fu, H. Zhou, and L. Li, “A survey on green deep learning,” 2021, arXiv preprint. [Online]. Available: https://arxiv.org/abs/2111.05193 [Cited on page 8.]

[20] B. R. Bartoldson, B. Kailkhura, and D. Blalock, “Compute-efficient deep learning: algorithmic trends and opportunities,” *J. Mach. Learn. Res.*, vol. 24, no. 1, 2023. [Online]. Available: https://dl.acm.org/doi/abs/10.5555/3648699.3648821 [Cited on page 8.]

[21] M. Alswaitti, R. Verdecchia, G. Danoy, P. Bouvry, and J. Pecero, “Training green ai models using elite samples,” *IEEE Transactions on Sustainable Computing*, pp. 1–16, 2025. [Online]. Available: https://doi.org/10.1109/TSUSC.2025.3544430 [Cited on page 8.]

[22] P. Dhabe, P. Mirani, R. Chugwani, and S. Gandewar, *Data Set Reduction to Improve Computing Efficiency and Energy Consumption in Healthcare Domain*. Springer International Publishing, 2021, pp. 53–64. [Online]. Available: https://doi.org/10.1007/978-3-030-61089-0_4 [Cited on page 8.]

[23] M. Anselmo and M. Vitali, *A Data-Centric Approach for Reducing Carbon Emissions in Deep Learning*. Springer Nature Switzerland, 2023, pp. 123–138. [Online]. Available: https://doi.org/10.1007/978-3-031-34560-9_8 [Cited on page 8.]

[24] B. Yousefimehr, M. Ghatee, M. A. Seifi, J. Fazli, S. Tavakoli, Z. Rafei, S. Ghaffari, A. Nikahd, M. R. Gandomani, A. Orouji, R. M. Kashani, S. Heshmati, and N. S. Mousavi, "Data balancing strategies: A survey of resampling and augmentation methods," 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2505.13518 [Cited on page 10.]

[25] A. Zhang, H. Yu, S. Zhou, Z. Huan, and X. Yang, "Instance weighted smote by indirectly exploring the data distribution," *Knowledge-Based Systems*, vol. 249, p. 108919, 2022. [Online]. Available: https://doi.org/10.1016/j.knosys.2022.108919 [Cited on page 10.]

[26] S. Maldonado, C. Vairetti, A. Fernandez, and F. Herrera, "Fw-smote: A feature-weighted oversampling approach for imbalanced classification," *Pattern Recognition*, vol. 124, p. 108511, 2022. [Online]. Available: https://doi.org/10.1016/j.patcog.2021.108511 [Cited on page 10.]

[27] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, ser. NIPS'14. MIT Press, 2014, p. 2672–2680. [Online]. Available: https://dl.acm.org/doi/10.5555/2969033.2969125 [Cited on page 10.]

[28] L. Xu, M. Skoularidou, A. Cuesta-Infante, and K. Veeramachaneni, *Modeling tabular data using conditional GAN*. Curran Associates Inc., 2019. [Online]. Available: https://dl.acm.org/doi/10.5555/3454287.3454946 [Cited on pages 10, 25, and 40.]

[29] Y. Dong, H. Xiao, and Y. Dong, "Sa-cgan: An oversampling method based on single attribute guided conditional gan for multi-class imbalanced learning," *Neurocomputing*, vol. 472, pp. 326–337, 2022. [Online]. Available: https://doi.org/10.1016/j.neucom.2021.04.135 [Cited on page 10.]

[30] D. K. K. Reddy, B. K. Rao, and T. A. Rashid, *Intelligent Under Sampling Based Ensemble Techniques for Cyber-Physical Systems in Smart Cities*. Springer Nature Switzerland, 2024, pp. 219–244. [Online]. Available: https://doi.org/10.1007/978-3-031-54038-7_8 [Cited on pages 11, 12, and 14.]

[31] L. C. M. Liaw, S. C. Tan, P. Y. Goh, and C. P. Lim, "A histogram smote-based sampling algorithm with incremental learning for imbalanced data classification,"

*Information Sciences*, vol. 686, p. 121193, 2025. [Online]. Available: https://doi.org/10.1016/j.ins.2024.121193 [Cited on pages 13, 14, and 15.]

[32] Y. Yan, Y. Zhu, R. Liu, Y. Zhang, Y. Zhang, and L. Zhang, "Spatial distribution-based imbalanced undersampling," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–1, 2022. [Online]. Available: https://doi.org/10.1109/TKDE.2022.3161537 [Cited on pages 11, 12, 14, and 15.]

[33] I. Tae, H. Kong, J. Lee, and Y. Lee, "Machine learning for disease-specific prediction of high-cost patients," *Engineering Applications of Artificial Intelligence*, vol. 161, p. 112200, 2025. [Online]. Available: https://doi.org/10.1016/j.engappai.2025.112200 [Cited on pages 11 and 14.]

[34] Z. Guo, S. Liu, L. Pan, and Q. He, "An ego network embedding model via neighbors sampling and self-attention mechanism," in *2020 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/-SocialCom/SustainCom)*. IEEE, 2020, pp. 425–432. [Online]. Available: https://doi.org/10.1109/ISPA-BDCloud-SocialCom-SustainCom51426.2020.00079 [Cited on page 13.]

[35] Y. B. Sanap and P. G. Aher, "Seagull-optimized deep convolutional neural network for detecting distributed denial of service attack in cloud computing," in *2024 2nd International Conference on Self Sustainable Artificial Intelligence Systems (ICSSAS)*. IEEE, 2024, pp. 35–42. [Online]. Available: https://doi.org/10.1109/ICSSAS64001.2024.10760748 [Cited on pages 13 and 14.]

[36] N. Bannour, S. Ghannay, A. Névéol, and A.-L. Ligozat, "Evaluating the carbon footprint of nlp methods: a survey and analysis of existing tools," in *Proceedings of the Second Workshop on Simple and Efficient Natural Language Processing*. Association for Computational Linguistics, 2021. [Online]. Available: https://doi.org/10.18653/v1/2021.sustainlp-1.2 [Cited on page 15.]

[37] R. Zhou, T. Zheng, X. Wang, L. Wang, and E. Sirvent-Hien, "Capability and applicability of measurement tools for ai model's environmental impact," *International Journal On Advances in Systems and Measurements*, vol. 17, no. 1-2, pp. 25–35, 2024. [Online]. Available: https://personales.upv.es/thinkmind/SysMea/SysMea_v17_n12_2024/sysmea_v17_n12_2024_3.html [Cited on page 15.]

[38] S. Rincé, “Awesome green ai,” Github repository, 2025, Accessed: Dec. 1, 2025. [Online]. Available: https://github.com/samuelrince/awesome-green-ai [Cited on page 15.]

[39] L. Lannelongue, J. Grealey, and M. Inouye, “Green algorithms: Quantifying the carbon footprint of computation,” *Advanced Science*, vol. 8, no. 12, 2021. [Online]. Available: https://doi.org/10.1002/advs.202100707 [Cited on page 15.]

[40] A. Lacoste, A. Luccioni, V. Schmidt, and T. Dandres, “Quantifying the carbon emissions of machine learning,” 2019. [Online]. Available: https://doi.org/10.48550/arXiv.1910.09700 [Cited on page 15.]

[41] *CodeCarbon*, Code Carbon, May 2024, version 2.4.1. [Online]. Available: https://docs.codecarbon.io/latest/ [Cited on page 15.]

[42] L. F. W. Anthony, B. Kanding, and R. Selvan, “Carbontracker: Tracking and predicting the carbon footprint of training deep learning models,” ICML Workshop on Challenges in Deploying and monitoring Machine Learning Systems, 2020, arXiv:2007.03051. [Online]. Available: https://doi.org/10.48550/arXiv.2007.03051 [Cited on page 15.]

[43] J. You, J.-W. Chung, and M. Chowdhury, “Zeus: Understanding and optimizing GPU energy consumption of DNN training,” in *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)*. Boston, MA: USENIX Association, 2023, pp. 119–139. [Online]. Available: https://www.usenix.org/conference/nsdi23/presentation/you [Cited on page 15.]

[44] *pyJoules*, Spirals Research Group. University of Lille and Inria, Oct. 2021, version 0.5.1. [Online]. Available: https://pyjoules.readthedocs.io/en/latest/index.html [Cited on page 15.]

[45] R. Fischer, “Ground-truthing ai energy consumption: Validating codecarbon against external measurements,” 2025, arXiv preprint. [Online]. Available: https://arxiv.org/abs/2509.22092 [Cited on page 15.]

[46] J. A. Garcia, “Exploration of energy consumption using the intel running average power limit interface,” in *2019 IEEE Space Computing Conference (SCC)*. IEEE, 2019, pp. 1–10. [Online]. Available: https://doi.org/10.1109/SpaceComp.2019.00005 [Cited on page 16.]

[47] "Running average power limit (rapl)," 2025, Accessed: Dec. 1, 2025. [Online]. Available: https://projectexigence.eu/green-ict-digest/running-average-power-limit-rapl/ [Cited on page 16.]

[48] J. Alcala-Fdez, A. Fernández, J. Luengo, J. Derrac, S. Garc'ia, L. Sanchez, and F. Herrera, "Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework," *Journal of Multiple-Valued Logic and Soft Computing*, vol. 17, pp. 255–287, 2010. [Cited on page 18.]

[49] M. S. Santos, P. H. Abreu, N. Japkowicz, A. Fernández, C. Soares, S. Wilk, and J. Santos, "On the joint-effect of class imbalance and overlap: a critical review," *Artificial Intelligence Review*, vol. 55, no. 8, pp. 6207–6275, 2022. [Online]. Available: https://doi.org/10.1007/s10462-022-10150-3 [Cited on page 19.]

[50] A. C. Lorena, L. P. F. Garcia, J. Lehmann, M. C. P. Souto, and T. K. Ho, "How complex is your classification problem?: A survey on measuring classification complexity," *ACM Computing Surveys*, vol. 52, no. 5, pp. 1–34, 2019. [Online]. Available: https://doi.org/10.1145/3347711 [Cited on pages 19, 20, and 53.]

[51] J. Komorniczak and P. Ksieniewicz, "problexity—an open-source python library for supervised learning problem complexity assessment," *Neurocomputing*, vol. 521, pp. 126–136, 2023. [Online]. Available: https://doi.org/10.1016/j.neucom.2022.11.056 [Cited on page 19.]

[52] R. A. Fisher, "Iris dataset," 1936. [Online]. Available: https://archive.ics.uci.edu/dataset/53/iris [Cited on page 20.]

[53] G. Lemaître, F. Nogueira, and C. K. Aridas, "Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning," *Journal of Machine Learning Research*, vol. 18, no. 17, pp. 1–5, 2017. [Online]. Available: http://jmlr.org/papers/v18/16-365.html [Cited on page 21.]

[54] G. Kovács, "smote-variants: a python implementation of 85 minority oversampling techniques," *Neurocomputing*, vol. 366, pp. 352–354, 2019. [Online]. Available: https://doi.org/10.1016/j.neucom.2019.06.100 [Cited on page 21.]

[55] N. Patki, R. Wedge, and K. Veeramachaneni, "The synthetic data vault," in *IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, 2016, pp. 399–410. [Online]. Available: https://doi.org/10.1109/DSAA.2016.49 [Cited on page 21.]

[56] I. Tomek, “Two modifications of cnn,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-6, no. 11, pp. 769–772, 1976. [Online]. Available: https://doi.org/10.1109/tsmc.1976.4309452 [Cited on page 22.]

[57] D. L. Wilson, “Asymptotic properties of nearest neighbor rules using edited data,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-2, no. 3, pp. 408–421, 1972. [Online]. Available: https://doi.org/10.1109/tsmc.1972.4309137 [Cited on page 22.]

[58] C. Bunkhumpornpat, K. Sinapiromsaran, and C. Lursinsap, *Safe-Level-SMOTE: Safe-Level-Synthetic Minority Over-Sampling TEchnique for Handling the Class Imbalanced Problem*. Springer Berlin Heidelberg, 2009, pp. 475–482. [Online]. Available: https://doi.org/10.1007/978-3-642-01307-2_43 [Cited on page 24.]

[59] D. Cieslak, N. Chawla, and A. Striegel, “Combating imbalance in network intrusion datasets,” in *2006 IEEE International Conference on Granular Computing*, 2006, pp. 732–737. [Online]. Available: https://doi.org/10.1109/GRC.2006.1635905 [Cited on page 24.]

[60] G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard, “A study of the behavior of several methods for balancing machine learning training data,” *ACM SIGKDD Explorations Newsletter*, vol. 6, no. 1, pp. 20–29, 2004. [Online]. Available: https://doi.org/10.1145/1007730.1007735 [Cited on page 25.]

[61] B. Ardiansyah, R. Yuana, and D. Maryono, “Effective ml techniques for small, imbalanced, and multi-class data: A comparative evaluation,” *International Journal of Computing and Digital Systems*, vol. 18, no. 1, pp. 1–11, 2025. [Online]. Available: https://doi.org/10.12785/ijcds/1571145625 [Cited on page 26.]

[62] G. Eom and H. Byeon, “Searching for optimal oversampling to process imbalanced data: Generative adversarial networks and synthetic minority over-sampling technique,” *Mathematics*, vol. 11, no. 16, p. 3605, 2023. [Online]. Available: https://doi.org/10.3390/math11163605

[63] D.-H. Min, Y. Kim, S. Kim, and H.-K. Yoon, “Strategy of oversampling geotechnical parameters through geostatistical, smote, and ctgan methods for assessing susceptibility of landslide,” *Landslides*, vol. 21, no. 2, pp. 291–307, 2023. [Online]. Available: https://doi.org/10.1007/s10346-023-02166-9

[64] A. F. Gündüz and C. B. Şahin, “Synthetic data augmentation for imbalanced tabular protein subcellular localization: A comparative study of smote, ctgan, tvae, and tabddpm methods,” *Applied Sciences*, vol. 16, no. 8, p. 3694, 2026. [Online]. Available: https://doi.org/10.3390/app16083694 [Cited on page 26.]

[65] J. Demšar, “Statistical comparisons of classifiers over multiple data sets,” *Journal of Machine learning research*, vol. 7, pp. 1–30, 2006. [Cited on page 30.]

[66] *Synthetic Data Metrics*, DataCebo, Inc., Mar. 2026, version 0.28.0. [Online]. Available: https://docs.sdv.dev/sdmetrics/ [Cited on page 36.]

[67] Y. Liu and Q. Liu, “Smote oversampling algorithm based on generative adversarial network,” *Cluster Computing*, vol. 28, no. 4, 2025. [Online]. Available: https://doi.org/10.1007/s10586-024-04980-9 [Cited on page 40.]

[68] E. Choi, S. Biswal, B. Malin, J. Duke, W. F. Stewart, and J. Sun, “Generating multi-label discrete patient records using generative adversarial networks,” 2017. [Online]. Available: https://doi.org/10.48550/arXiv.1703.06490 [Cited on page 40.]

# A. Dataset Characteristics

Table A.1: Description of the low-complexity datasets.

| dataset | n_features | n_samples | n_class0 | n_class1 | IR | score |
|---|---|---|---|---|---|---|
| iris0 | 4 | 150 | 100 | 50 | 2.000 | 0.123 |
| kddcup-land_vs_satan | 41 | 1610 | 1589 | 21 | 75.667 | 0.128 |
| kddcup-guess_passwd_vs_satan | 41 | 1642 | 1589 | 53 | 29.981 | 0.128 |
| kddcup-rootkit-imap_vs_back | 41 | 2225 | 2203 | 22 | 100.136 | 0.128 |
| kddcup-land_vs_portsweep | 41 | 1061 | 1040 | 21 | 49.524 | 0.130 |
| kddcup-buffer_overflow_vs_back | 41 | 2233 | 2203 | 30 | 73.433 | 0.135 |
| shuttle-c0-vs-c4 | 9 | 1829 | 1706 | 123 | 13.870 | 0.138 |
| shuttle-c2-vs-c4 | 9 | 129 | 123 | 6 | 20.500 | 0.166 |
| shuttle-2_vs_5 | 9 | 3316 | 3267 | 49 | 66.673 | 0.167 |
| abalone-3_vs_11 | 8 | 502 | 487 | 15 | 32.467 | 0.174 |
| ecoli-0_vs_1 | 7 | 220 | 77 | 143 | 0.538 | 0.179 |
| new-thyroid1 | 5 | 215 | 180 | 35 | 5.143 | 0.184 |
| new-thyroid2 | 5 | 215 | 180 | 35 | 5.143 | 0.184 |
| shuttle-6_vs_2-3 | 9 | 230 | 220 | 10 | 22.000 | 0.196 |
| glass-0-1-2-3_vs_4-5-6 | 9 | 214 | 163 | 51 | 3.196 | 0.211 |
| glass6 | 9 | 214 | 185 | 29 | 6.379 | 0.222 |
| page-blocks-1-3_vs_4 | 10 | 472 | 444 | 28 | 15.857 | 0.226 |
| segment0 | 19 | 2308 | 1979 | 329 | 6.015 | 0.230 |
| dermatology-6 | 34 | 358 | 338 | 20 | 16.900 | 0.234 |
| glass-0-4_vs_5 | 9 | 92 | 83 | 9 | 9.222 | 0.236 |
| wisconsin | 9 | 683 | 444 | 239 | 1.858 | 0.250 |
| glass-0-1-6_vs_5 | 9 | 184 | 175 | 9 | 19.444 | 0.254 |
| ecoli-0-1-3-7_vs_2-6 | 7 | 281 | 274 | 7 | 39.143 | 0.255 |
| glass4 | 9 | 214 | 201 | 13 | 15.462 | 0.256 |
| kr-vs-k-one_vs_fifteen | 6 | 2244 | 2166 | 78 | 27.769 | 0.257 |
| kr-vs-k-zero_vs_fifteen | 6 | 2193 | 2166 | 27 | 80.222 | 0.258 |
| glass5 | 9 | 214 | 205 | 9 | 22.778 | 0.260 |
| yeast5 | 8 | 1484 | 1440 | 44 | 32.727 | 0.261 |
| glass-0-6_vs_5 | 9 | 108 | 99 | 9 | 11.000 | 0.266 |
| ecoli4 | 7 | 336 | 316 | 20 | 15.800 | 0.266 |
| vehicle0 | 18 | 846 | 647 | 199 | 3.251 | 0.272 |
| abalone-21_vs_8 | 8 | 581 | 567 | 14 | 40.500 | 0.272 |
| vowel0 | 13 | 988 | 898 | 90 | 9.978 | 0.277 |

Table A.2: Description of the medium-complexity datasets.

| dataset | n_features | n_samples | n_class0 | n_class1 | IR | score |
|---|---|---|---|---|---|---|
| lymphography-normal-fibrosis | 18 | 148 | 142 | 6 | 23.667 | 0.282 |
| vehicle2 | 18 | 846 | 628 | 218 | 2.881 | 0.284 |
| ecoli1 | 7 | 336 | 259 | 77 | 3.364 | 0.291 |
| zoo-3 | 16 | 101 | 96 | 5 | 19.200 | 0.297 |
| yeast3 | 8 | 1484 | 1321 | 163 | 8.104 | 0.301 |
| kr-vs-k-zero-one_vs_draw | 6 | 2901 | 2796 | 105 | 26.629 | 0.301 |
| ecoli-0-2-3-4_vs_5 | 7 | 202 | 182 | 20 | 9.100 | 0.302 |
| ecoli-0-3-4-6_vs_5 | 7 | 205 | 185 | 20 | 9.250 | 0.303 |
| ecoli3 | 7 | 336 | 301 | 35 | 8.600 | 0.303 |
| ecoli-0-3-4_vs_5 | 7 | 200 | 180 | 20 | 9.000 | 0.303 |
| ecoli-0-1_vs_5 | 6 | 240 | 220 | 20 | 11.000 | 0.305 |
| kr-vs-k-zero_vs_eight | 6 | 1460 | 1433 | 27 | 53.074 | 0.306 |
| ecoli-0-4-6_vs_5 | 6 | 203 | 183 | 20 | 9.150 | 0.307 |
| ecoli-0-1-4-6_vs_5 | 6 | 280 | 260 | 20 | 13.000 | 0.309 |
| kr-vs-k-three_vs_eleven | 6 | 2935 | 2854 | 81 | 35.235 | 0.310 |
| ecoli2 | 7 | 336 | 284 | 52 | 5.462 | 0.311 |
| page-blocks0 | 10 | 5472 | 4913 | 559 | 8.789 | 0.311 |
| abalone-20_vs_8-9-10 | 8 | 1916 | 1890 | 26 | 72.692 | 0.314 |
| winequality-white-9_vs_4 | 11 | 168 | 163 | 5 | 32.600 | 0.315 |
| abalone-17_vs_7-8-9-10 | 8 | 2338 | 2280 | 58 | 39.310 | 0.315 |
| yeast-2_vs_8 | 8 | 482 | 462 | 20 | 23.100 | 0.323 |
| ecoli-0-3-4-7_vs_5-6 | 7 | 257 | 232 | 25 | 9.280 | 0.327 |
| ecoli-0-6-7_vs_5 | 6 | 220 | 200 | 20 | 10.000 | 0.328 |
| ecoli-0-1_vs_2-3-5 | 7 | 244 | 220 | 24 | 9.167 | 0.329 |
| yeast-2_vs_4 | 8 | 514 | 463 | 51 | 9.078 | 0.330 |
| winequality-red-3_vs_5 | 11 | 691 | 681 | 10 | 68.100 | 0.331 |
| ecoli-0-1-4-7_vs_5-6 | 6 | 332 | 307 | 25 | 12.280 | 0.332 |
| abalone19 | 8 | 4174 | 4142 | 32 | 129.438 | 0.333 |
| car-vgood | 6 | 1728 | 1663 | 65 | 25.585 | 0.335 |
| glass0 | 9 | 214 | 144 | 70 | 2.057 | 0.335 |
| ecoli-0-6-7_vs_3-5 | 7 | 222 | 200 | 22 | 9.091 | 0.337 |
| yeast-0-2-5-7-9_vs_3-6-8 | 8 | 1004 | 905 | 99 | 9.141 | 0.337 |
| yeast6 | 8 | 1484 | 1449 | 35 | 41.400 | 0.337 |
| glass2 | 9 | 214 | 197 | 17 | 11.588 | 0.337 |

Table A.3: Description of the high-complexity datasets.

| dataset | n_features | n_samples | n_class0 | n_class1 | IR | score |
|---|---|---|---|---|---|---|
| glass-0-1-5_vs_2 | 9 | 172 | 155 | 17 | 9.118 | 0.338 |
| glass-0-1-4-6_vs_2 | 9 | 205 | 188 | 17 | 11.059 | 0.339 |
| ecoli-0-2-6-7_vs_3-5 | 7 | 224 | 202 | 22 | 9.182 | 0.339 |
| cleveland-0_vs_4 | 13 | 177 | 164 | 13 | 12.615 | 0.342 |
| glass-0-1-6_vs_2 | 9 | 192 | 175 | 17 | 10.294 | 0.344 |
| ecoli-0-1-4-7_vs_2-3-5-6 | 7 | 336 | 307 | 29 | 10.586 | 0.347 |
| abalone9-18 | 8 | 731 | 689 | 42 | 16.405 | 0.347 |
| flare-F | 11 | 1066 | 1023 | 43 | 23.791 | 0.347 |
| winequality-white-3_vs_7 | 11 | 900 | 880 | 20 | 44.000 | 0.348 |
| winequality-red-8_vs_6 | 11 | 656 | 638 | 18 | 35.444 | 0.350 |
| car-good | 6 | 1728 | 1659 | 69 | 24.043 | 0.359 |
| winequality-red-8_vs_6-7 | 11 | 855 | 837 | 18 | 46.500 | 0.361 |
| abalone-19_vs_10-11-12-13 | 8 | 1622 | 1590 | 32 | 49.688 | 0.366 |
| yeast4 | 8 | 1484 | 1433 | 51 | 28.098 | 0.368 |
| yeast-1-2-8-9_vs_7 | 8 | 947 | 917 | 30 | 30.567 | 0.371 |
| yeast-0-2-5-6_vs_3-7-8-9 | 8 | 1004 | 905 | 99 | 9.141 | 0.374 |
| winequality-white-3-9_vs_5 | 11 | 1482 | 1457 | 25 | 58.280 | 0.381 |
| glass1 | 9 | 214 | 138 | 76 | 1.816 | 0.381 |
| yeast-0-5-6-7-9_vs_4 | 8 | 528 | 477 | 51 | 9.353 | 0.383 |
| vehicle1 | 18 | 846 | 629 | 217 | 2.899 | 0.387 |
| vehicle3 | 18 | 846 | 634 | 212 | 2.991 | 0.390 |
| yeast-1_vs_7 | 7 | 459 | 429 | 30 | 14.300 | 0.392 |
| yeast-0-3-5-9_vs_7-8 | 8 | 506 | 456 | 50 | 9.120 | 0.397 |
| winequality-red-4 | 11 | 1599 | 1546 | 53 | 29.170 | 0.409 |
| led7digit-0-2-4-5-6-7-8-9_vs_1 | 7 | 443 | 406 | 37 | 10.973 | 0.419 |
| yeast-1-4-5-8_vs_7 | 8 | 693 | 663 | 30 | 22.100 | 0.421 |
| yeast1 | 8 | 1484 | 1055 | 429 | 2.459 | 0.424 |
| pima | 8 | 768 | 500 | 268 | 1.866 | 0.435 |
| poker-9_vs_7 | 10 | 244 | 236 | 8 | 29.500 | 0.451 |
| poker-8-9_vs_5 | 10 | 2075 | 2050 | 25 | 82.000 | 0.491 |
| haberman | 3 | 306 | 225 | 81 | 2.778 | 0.495 |
| poker-8_vs_6 | 10 | 1477 | 1460 | 17 | 85.882 | 0.501 |
| poker-8-9_vs_6 | 10 | 1485 | 1460 | 25 | 58.400 | 0.548 |

# B. Complexity Measures

Table B.1: Description of the data complexity measures defined by Lorena et al. [50]. For all measures, higher values correspond to higher complexity.

| Category | Acronym | Name | Description |
|---|---|---|---|
| Feature-based | F1 | Maximum Fisher's discriminant ratio | Measures how well the best single feature separates classes using between-class vs within-class variance. A high value indicates no single feature is discriminative. |
| | F1v | Directional vector maximum Fisher's discriminant ratio | A complement to F1, measures class separability using the best discriminant direction. A high value indicates that even the best linear projection provides strong overlap between classes. |
| | F2 | Volume of overlap region | Calculates the overlap of the distributions of the feature values within the classes. A high value indicates a higher amount of overlap between the classes. |
| | F3 | Maximum individual feature efficiency | Estimates the individual efficiency of features on separating the classes. A high value indicates that even the most informative feature provides poor class separation. |
| | F4 | Collective feature efficiency | Measures how well combinations of features jointly separate classes. A high value indicates that even multi-feature combinations fail to clearly separate classes. |
| Linearity | L1 | Sum of the error distance by linear programming | Measures the linear separability of classes by computing the sum of the distances of incorrectly classified examples to an optimal linear boundary obtained using a linear SVM. A high value indicates that the data is far from being linearly separable. |
| | L2 | Error rate of linear classifier | Computes the error rate of a linear SVM classifier. A high value indicates that a linear decision boundary misclassifies many instances, reflecting poor linear separability. |
| | L3 | Non-linearity of linear classifier | Measures non-linearity by generating synthetic points via interpolation between same-class samples and evaluating the error of a linear classifier trained on the original data. A high value indicates complex structure, including concavities and overlap of class regions. |
| Neighborhood | N1 | Fraction of borderline points | Based on a minimum spanning tree, counts points connected to other-class neighbors. A high value indicates strong class mixing in local neighborhoods, which directly increases the complexity of the required decision boundary. |

*Continued on next page*

| Category | Acronym | Name | Description |
|---|---|---|---|
| | N2 | Ratio of intra/extra class nearest neighbor distance | Compares within-class distances to between-class distances. A high value indicates that examples from different classes are, on average, as close or closer than examples from the same class, reflecting strong class overlap. |
| | N3 | Error rate of nearest neighbor classifier | Corresponds to the error rate of a 1NN classifier estimated using a leave-one-out procedure. A high value indicates that many examples are close to examples of other classes. |
| | N4 | Non-linearity of nearest neighbor classifier | Measures robustness of nearest-neighbor structure under small changes. Similar to L3, but uses a NN classifier instead of a linear classifier. A high value indicates complex, unstable neighborhood structure. |
| | T1 | Fraction of hyperspheres covering data | Measures how many local hyperspheres are needed to cover class regions. A high value indicates fragmented and irregular class distributions. |
| | LSC | Local set average cardinality | Measures the average cardinality of local sets defined relative to the nearest opposite-class point. A high value indicates that points are typically closer to other-class examples than same-class ones, implying narrow margins and a more complex boundary. |
| Network | Density | Average density of the network | For all network measures, a graph connecting nearby instances within the same class is used. This measure considers the number of edges that are retained in the graph normalized by the maximum number of edges. A high value indicates a low density dataset and/or a high overlap between classes. |
| | ClsCoef | Clustering coefficient | Measures the grouping tendency of the graph vertexes, based on the ratio of the number of edges between the neighbors of a vertex and the maximum number of edges that could exist between them. A high value indicates sparse connections among examples from the same class. |
| | Hubs | Hub score | Considers the number of connections of each node to other nodes, weighted by the number of connections these neighbors have. A high value indicates strong vertexes tend to be less connected to strong neighbors, which may correspond to a high overlap between classes. |
| Dimensionality | T2 | Average number of features per dimension | Relates feature space dimensionality to sample size. A high value indicates sparse data relative to dimensionality. |

*Continued on next page*

| Category | Acronym | Name | Description |
|---|---|---|---|
| | T3 | Average number of PCA dimensions per points | Relates the number of PCA components needed to represent 95% of data variability to sample size. A high value indicates sparse data. |
| | T4 | Ratio of PCA dimension to original dimension | Provides an approximate measure of the proportion of relevant dimensions for the dataset, measured according to the PCA components representing 95% of data variability. A high value indicates the original features are highly needed to describe data variability. |
| Class imbalance | C1 | Entropy of class proportions | Measures how evenly distributed classes are using entropy. A high value indicates that the class proportions are unequal, meaning the dataset is imbalanced. |
| | C2 | Imbalance ratio | Measures the imbalance ratio of the data based on the number of samples of each class. Differently than the traditional imbalance ratio (IR), this version is bounded between 0 and 1. A high value indicates a highly imbalanced dataset. |

# C. Additional Results

Tables C.2-C.5 present pairwise rank differences between the combinations of classification and traditional resampling methods, according to the Nemenyi post-hoc test following the Friedman test. The lower triangle reports F1-score rank differences and the upper triangle reports energy consumption rank differences. Highlighted cells indicate statistically significant differences, and the values represent the difference between the rank of the column method and the rank of the row method. Therefore, positive values indicate that the column method ranked higher, while negative values indicate that the row method ranked higher. Green represents positive energy differences, orange represents negative energy differences, yellow represents positive F1 differences and purple represents negative F1 differences. All tables preserve the same method order, which is independent from performance or energy rankings. Table C.1 shows the mapping between the method names used throughout this dissertation and the abbreviations used in Tables C.2-C.5.

Figures C.1-C.3 present additional critical difference diagrams grouped by dataset complexity for energy consumption, F1-score, and recall. The last two diagrams include CTGAN results.

Table C.1: Mapping between methods and the corresponding abbreviations used in Tables C.2-C.5.

| Method | Abbreviation |
|---|---|
| *logreg_none* | LR-N |
| *logreg_rus* | LR-RU |
| *logreg_ros* | LR-RO |
| *logreg_smote* | LR-S |
| *logreg_borderline1* | LR-B |
| *logreg_tomek* | LR-T |
| *logreg_smote-tomek* | LR-S-T |
| *logreg_safe-level* | LR-SF |
| *logreg_cluster* | LR-C |
| *logreg_enn* | LR-E |
| *logreg_smote-enn* | LR-S-E |
| *rf_none* | RF-N |
| *rf_rus* | RF-RU |
| *rf_smote* | RF-S |
| *rf_ros* | RF-RO |
| *rf_borderline1* | RF-B |
| *rf_tomek* | RF-T |
| *rf_smote-tomek* | RF-S-T |
| *rf_safe-level* | RF-SF |
| *rf_cluster* | RF-C |
| *rf_enn* | RF-E |
| *rf_smote-enn* | RF-S-E |

Table C.2: Pairwise rank differences of all datasets (100 datasets). CD: 3.2995.

| method | RF-S-E | RF-E | RF-C | RF-SF | RF-S-T | RF-T | RF-B | RF-RO | RF-S | RF-RU | RF-N | LR-S-E | LR-E | LR-C | LR-SF | LR-S-T | LR-T | LR-B | LR-S | LR-RO | LR-RU | LR-N |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RF-S-E | - | 5.86 | 1.63 | 1.60 | −2.31 | 4.39 | −0.49 | 2.50 | −1.60 | 6.80 | 4.92 | 9.44 | 13.53 | 14.49 | 14.18 | 8.43 | 13.52 | 12.24 | 11.69 | 12.42 | 16.90 | 17.28 |
| RF-E | 0.84 | - | −4.23 | −4.26 | −8.17 | −1.47 | −6.35 | −3.36 | −7.46 | 0.94 | −0.94 | 3.58 | 7.67 | 8.63 | 8.32 | 2.57 | 7.66 | 6.38 | 5.83 | 6.56 | 11.04 | 11.42 |
| RF-C | −0.17 | −1.01 | - | −0.03 | −3.94 | 2.76 | −2.12 | 0.87 | −3.23 | 5.17 | 3.29 | 7.81 | 11.90 | 12.86 | 12.55 | 6.80 | 11.89 | 10.61 | 10.06 | 10.79 | 15.27 | 15.65 |
| RF-SF | −0.38 | −1.22 | −0.20 | - | −3.91 | 2.79 | −2.09 | 0.90 | −3.20 | 5.20 | 3.32 | 7.84 | 11.93 | 12.89 | 12.58 | 6.83 | 11.92 | 10.64 | 10.09 | 10.82 | 15.30 | 15.68 |
| RF-S-T | −1.00 | −1.84 | −0.83 | −0.63 | - | 6.70 | 1.82 | 4.81 | 0.71 | 9.11 | 7.23 | 11.75 | 15.84 | 16.80 | 16.49 | 10.74 | 15.83 | 14.55 | 14.00 | 14.73 | 19.21 | 19.59 |
| RF-T | 1.50 | 0.67 | 1.68 | 1.88 | 2.51 | - | −4.88 | −1.89 | −5.99 | 2.41 | 0.53 | 5.05 | 9.14 | 10.10 | 9.79 | 4.04 | 9.13 | 7.85 | 7.30 | 8.03 | 12.51 | 12.89 |
| RF-B | 0.85 | 0.01 | 1.02 | 1.23 | 1.85 | −0.66 | - | 2.99 | −1.11 | 7.29 | 5.41 | 9.93 | 14.02 | 14.98 | 14.67 | 8.92 | 14.01 | 12.73 | 12.18 | 12.91 | 17.39 | 17.77 |
| RF-RO | 0.89 | 0.04 | 1.06 | 1.26 | 1.89 | −0.62 | 0.04 | - | −4.10 | 4.30 | 2.42 | 6.94 | 11.03 | 11.99 | 11.68 | 5.93 | 11.02 | 9.74 | 9.19 | 9.92 | 14.40 | 14.78 |
| RF-S | −1.03 | −1.88 | −0.86 | −0.66 | −0.03 | −2.54 | −1.89 | −1.92 | - | 8.40 | 6.52 | 11.04 | 15.13 | 16.09 | 15.78 | 10.03 | 15.12 | 13.84 | 13.29 | 14.02 | 18.50 | 18.88 |
| RF-RU | 7.24 | 6.39 | 7.41 | 7.61 | 8.24 | 5.73 | 6.38 | 6.35 | 8.27 | - | −1.88 | 2.64 | 6.73 | 7.69 | 7.38 | 1.63 | 6.72 | 5.44 | 4.89 | 5.62 | 10.10 | 10.48 |
| RF-N | 1.79 | 0.95 | 1.97 | 2.17 | 2.80 | 0.29 | 0.94 | 0.91 | 2.83 | −5.44 | - | 4.52 | 8.61 | 9.57 | 9.26 | 3.51 | 8.60 | 7.32 | 6.77 | 7.50 | 11.98 | 12.36 |
| LR-S-E | 3.29 | 2.46 | 3.47 | 3.67 | 4.30 | 1.79 | 2.44 | 2.41 | 4.33 | −3.94 | 1.50 | - | 4.09 | 5.05 | 4.74 | −1.01 | 4.08 | 2.80 | 2.25 | 2.98 | 7.46 | 7.84 |
| LR-E | 4.45 | 3.60 | 4.62 | 4.82 | 5.45 | 2.94 | 3.60 | 3.56 | 5.48 | −2.79 | 2.65 | 1.15 | - | 0.96 | 0.65 | −5.10 | −0.01 | −1.29 | −1.84 | −1.11 | 3.37 | 3.75 |
| LR-C | 4.20 | 3.35 | 4.37 | 4.57 | 5.20 | 2.69 | 3.35 | 3.31 | 5.23 | −3.04 | 2.40 | 0.90 | −0.25 | - | −0.31 | −6.06 | −0.97 | −2.25 | −2.80 | −2.07 | 2.41 | 2.79 |
| LR-SF | 5.37 | 4.53 | 5.54 | 5.75 | 6.38 | 3.87 | 4.52 | 4.49 | 6.41 | −1.86 | 3.58 | 2.08 | 0.93 | 1.18 | - | −5.75 | −0.66 | −1.94 | −2.49 | −1.76 | 2.72 | 3.10 |
| LR-S-T | 3.71 | 2.87 | 3.88 | 4.08 | 4.71 | 2.20 | 2.85 | 2.82 | 4.74 | −3.53 | 1.91 | 0.41 | −0.74 | −0.49 | −1.67 | - | 5.09 | 3.81 | 3.26 | 3.99 | 8.47 | 8.85 |
| LR-T | 6.22 | 5.38 | 6.40 | 6.60 | 7.23 | 4.72 | 5.38 | 5.34 | 7.26 | −1.01 | 4.43 | 2.93 | 1.78 | 2.03 | 0.85 | 2.52 | - | −1.28 | −1.83 | −1.10 | 3.38 | 3.76 |
| LR-B | 3.64 | 2.80 | 3.81 | 4.01 | 4.64 | 2.13 | 2.79 | 2.75 | 4.67 | −3.60 | 1.84 | 0.34 | −0.81 | −0.56 | −1.73 | −0.07 | −2.58 | - | −0.55 | 0.18 | 4.66 | 5.04 |
| LR-S | 3.87 | 3.03 | 4.04 | 4.25 | 4.88 | 2.37 | 3.02 | 2.98 | 4.91 | −3.37 | 2.08 | 0.57 | −0.57 | −0.33 | −1.50 | 0.17 | −2.35 | 0.23 | - | 0.73 | 5.21 | 5.59 |
| LR-RO | 4.91 | 4.07 | 5.08 | 5.29 | 5.92 | 3.40 | 4.06 | 4.03 | 5.95 | −2.33 | 3.12 | 1.61 | 0.47 | 0.71 | −0.46 | 1.21 | −1.31 | 1.27 | 1.04 | - | 4.48 | 4.86 |
| LR-RU | 8.99 | 8.15 | 9.16 | 9.37 | 9.99 | 7.49 | 8.14 | 8.11 | 10.03 | 1.75 | 7.20 | 5.70 | 4.54 | 4.79 | 3.62 | 5.29 | 2.77 | 5.35 | 5.12 | 4.08 | - | 0.38 |
| LR-N | 6.72 | 5.88 | 6.90 | 7.10 | 7.73 | 5.22 | 5.88 | 5.84 | 7.76 | −0.51 | 4.93 | 3.43 | 2.28 | 2.53 | 1.35 | 3.02 | 0.50 | 3.08 | 2.85 | 1.81 | −2.27 | - |

Table C.3: Pairwise rank differences of low-complexity datasets (33 datasets). CD: 5.7437.

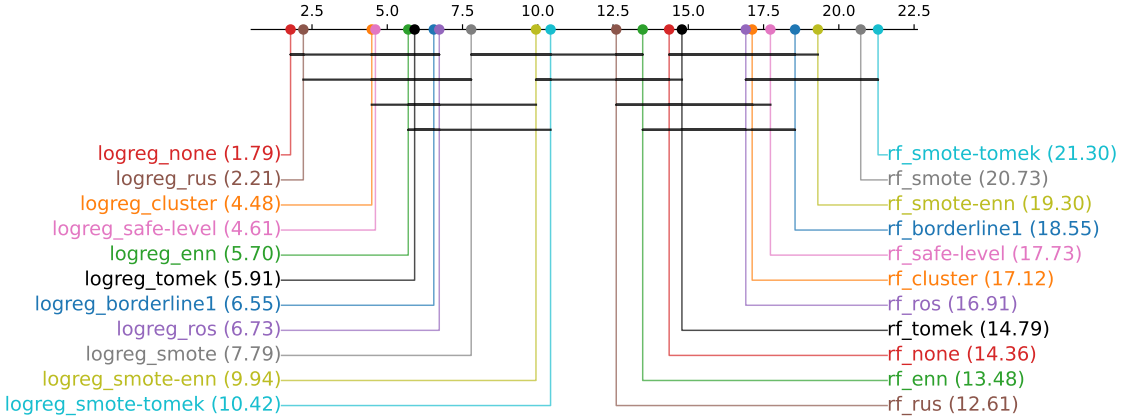
| method | RF-S-E | RF-E | RF-C | RF-SF | RF-S-T | RF-T | RF-B | RF-RO | RF-S | RF-RU | RF-N | LR-S-E | LR-E | LR-C | LR-SF | LR-S-T | LR-T | LR-B | LR-S | LR-RO | LR-RU | LR-N |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RF-S-E | - | 5.82 | 2.18 | 1.58 | −2.00 | 4.51 | 0.76 | 2.39 | −1.42 | 6.70 | 4.94 | 9.36 | 13.61 | 14.82 | 14.70 | 8.88 | 13.39 | 12.76 | 11.52 | 12.58 | 17.09 | 17.52 |
| RF-E | −0.09 | - | −3.64 | −4.24 | −7.82 | −1.30 | −5.06 | −3.42 | −7.24 | 0.88 | −0.88 | 3.54 | 7.79 | 9.00 | 8.88 | 3.06 | 7.58 | 6.94 | 5.70 | 6.76 | 11.27 | 11.70 |
| RF-C | −0.61 | −0.52 | - | −0.61 | −4.18 | 2.33 | −1.42 | 0.21 | −3.61 | 4.51 | 2.76 | 7.18 | 11.42 | 12.64 | 12.52 | 6.70 | 11.21 | 10.58 | 9.33 | 10.39 | 14.91 | 15.33 |
| RF-SF | −1.20 | −1.11 | −0.59 | - | −3.58 | 2.94 | −0.82 | 0.82 | −3.00 | 5.12 | 3.36 | 7.79 | 12.03 | 13.24 | 13.12 | 7.30 | 11.82 | 11.18 | 9.94 | 11.00 | 15.52 | 15.94 |
| RF-S-T | −0.68 | −0.59 | −0.08 | 0.52 | - | 6.51 | 2.76 | 4.39 | 0.58 | 8.70 | 6.94 | 11.36 | 15.61 | 16.82 | 16.70 | 10.88 | 15.39 | 14.76 | 13.52 | 14.58 | 19.09 | 19.52 |
| RF-T | −1.21 | −1.12 | −0.61 | −0.01 | −0.53 | - | −3.76 | −2.12 | −5.94 | 2.18 | 0.42 | 4.85 | 9.09 | 10.30 | 10.18 | 4.36 | 8.88 | 8.24 | 7.00 | 8.06 | 12.58 | 13.00 |
| RF-B | 0.61 | 0.70 | 1.21 | 1.80 | 1.29 | 1.82 | - | 1.64 | −2.18 | 5.94 | 4.18 | 8.61 | 12.85 | 14.06 | 13.94 | 8.12 | 12.64 | 12.00 | 10.76 | 11.82 | 16.33 | 16.76 |
| RF-RO | −0.21 | −0.12 | 0.39 | 0.98 | 0.47 | 1.00 | −0.82 | - | −3.82 | 4.30 | 2.54 | 6.97 | 11.21 | 12.42 | 12.30 | 6.49 | 11.00 | 10.36 | 9.12 | 10.18 | 14.70 | 15.12 |
| RF-S | −1.14 | −1.04 | −0.53 | 0.06 | −0.46 | 0.08 | −1.74 | −0.92 | - | 8.12 | 6.36 | 10.79 | 15.03 | 16.24 | 16.12 | 10.30 | 14.82 | 14.18 | 12.94 | 14.00 | 18.52 | 18.94 |
| RF-RU | 8.41 | 8.50 | 9.02 | 9.61 | 9.09 | 9.62 | 7.80 | 8.62 | 9.54 | - | −1.76 | 2.67 | 6.91 | 8.12 | 8.00 | 2.18 | 6.70 | 6.06 | 4.82 | 5.88 | 10.39 | 10.82 |
| RF-N | −1.04 | −0.95 | −0.44 | 0.15 | −0.36 | 0.17 | −1.65 | −0.83 | 0.09 | −9.46 | - | 4.42 | 8.67 | 9.88 | 9.76 | 3.94 | 8.46 | 7.82 | 6.58 | 7.64 | 12.15 | 12.58 |
| LR-S-E | 3.79 | 3.88 | 4.39 | 4.99 | 4.47 | 5.00 | 3.18 | 4.00 | 4.92 | −4.62 | 4.83 | - | 4.24 | 5.46 | 5.33 | −0.48 | 4.03 | 3.39 | 2.15 | 3.21 | 7.73 | 8.15 |
| LR-E | 6.29 | 6.38 | 6.89 | 7.49 | 6.97 | 7.50 | 5.68 | 6.50 | 7.42 | −2.12 | 7.33 | 2.50 | - | 1.21 | 1.09 | −4.73 | −0.21 | −0.85 | −2.09 | −1.03 | 3.48 | 3.91 |
| LR-C | 3.98 | 4.08 | 4.59 | 5.18 | 4.67 | 5.20 | 3.38 | 4.20 | 5.12 | −4.42 | 5.03 | 0.20 | −2.30 | - | −0.12 | −5.94 | −1.42 | −2.06 | −3.30 | −2.24 | 2.27 | 2.70 |
| LR-SF | 5.67 | 5.76 | 6.27 | 6.86 | 6.35 | 6.88 | 5.06 | 5.88 | 6.80 | −2.74 | 6.71 | 1.88 | −0.62 | 1.68 | - | −5.82 | −1.30 | −1.94 | −3.18 | −2.12 | 2.39 | 2.82 |
| LR-S-T | 3.67 | 3.76 | 4.27 | 4.86 | 4.35 | 4.88 | 3.06 | 3.88 | 4.80 | −4.74 | 4.71 | −0.12 | −2.62 | −0.32 | −2.00 | - | 4.51 | 3.88 | 2.64 | 3.70 | 8.21 | 8.64 |
| LR-T | 6.00 | 6.09 | 6.61 | 7.20 | 6.68 | 7.21 | 5.39 | 6.21 | 7.14 | −2.41 | 7.04 | 2.21 | −0.29 | 2.02 | 0.33 | 2.33 | - | −0.64 | −1.88 | −0.82 | 3.70 | 4.12 |
| LR-B | 4.36 | 4.46 | 4.97 | 5.56 | 5.04 | 5.58 | 3.76 | 4.58 | 5.50 | −4.04 | 5.41 | 0.58 | −1.92 | 0.38 | −1.30 | 0.70 | −1.64 | - | −1.24 | −0.18 | 4.33 | 4.76 |
| LR-S | 3.82 | 3.91 | 4.42 | 5.01 | 4.50 | 5.03 | 3.21 | 4.03 | 4.96 | −4.59 | 4.86 | 0.03 | −2.47 | −0.17 | −1.85 | 0.15 | −2.18 | −0.55 | - | 1.06 | 5.58 | 6.00 |
| LR-RO | 4.56 | 4.65 | 5.17 | 5.76 | 5.24 | 5.77 | 3.96 | 4.77 | 5.70 | −3.85 | 5.61 | 0.77 | −1.73 | 0.58 | −1.11 | 0.89 | −1.44 | 0.20 | 0.74 | - | 4.51 | 4.94 |
| LR-RU | 9.54 | 9.64 | 10.15 | 10.74 | 10.23 | 10.76 | 8.94 | 9.76 | 10.68 | 1.14 | 10.59 | 5.76 | 3.26 | 5.56 | 3.88 | 5.88 | 3.54 | 5.18 | 5.73 | 4.99 | - | 0.42 |
| LR-N | 6.15 | 6.24 | 6.76 | 7.35 | 6.83 | 7.36 | 5.54 | 6.36 | 7.29 | −2.26 | 7.20 | 2.36 | −0.14 | 2.17 | 0.48 | 2.48 | 0.15 | 1.79 | 2.33 | 1.59 | −3.39 | - |

Table C.4: Pairwise rank differences of medium-complexity datasets (34 datasets). CD: 5.6586.

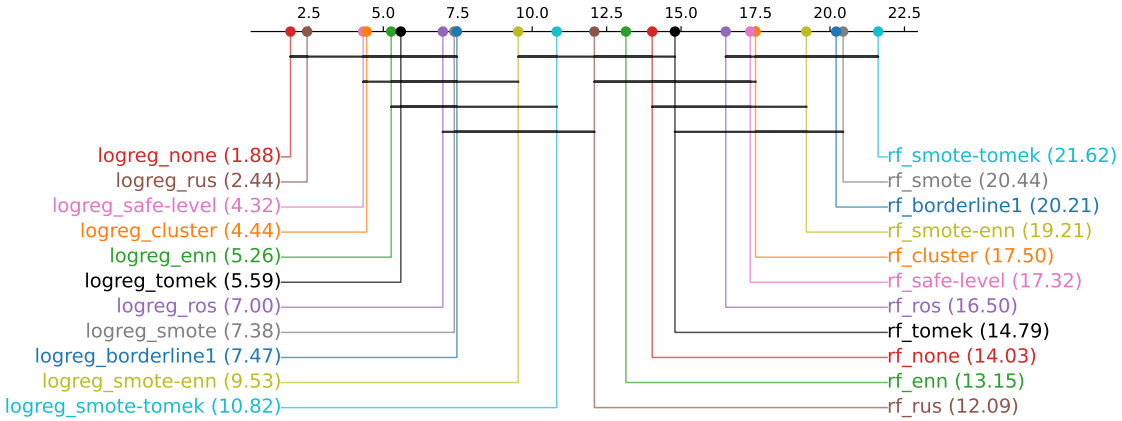
| method | RF-S-E | RF-E | RF-C | RF-SF | RF-S-T | RF-T | RF-B | RF-RO | RF-S | RF-RU | RF-N | LR-S-E | LR-E | LR-C | LR-SF | LR-S-T | LR-T | LR-B | LR-S | LR-RO | LR-RU | LR-N |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RF-S-E | - | 6.06 | 1.71 | 1.88 | −2.41 | 4.41 | −1.00 | 2.71 | −1.24 | 7.12 | 5.18 | 9.68 | 13.94 | 14.77 | 14.88 | 8.38 | 13.62 | 11.73 | 11.82 | 12.21 | 16.77 | 17.32 |
| RF-E | 2.00 | - | −4.35 | −4.18 | −8.47 | −1.65 | −7.06 | −3.35 | −7.29 | 1.06 | −0.88 | 3.62 | 7.88 | 8.71 | 8.82 | 2.32 | 7.56 | 5.68 | 5.76 | 6.15 | 10.71 | 11.27 |
| RF-C | 0.87 | −1.13 | - | 0.18 | −4.12 | 2.71 | −2.71 | 1.00 | −2.94 | 5.41 | 3.47 | 7.97 | 12.23 | 13.06 | 13.18 | 6.68 | 11.91 | 10.03 | 10.12 | 10.50 | 15.06 | 15.62 |
| RF-SF | 0.91 | −1.09 | 0.04 | - | −4.29 | 2.53 | −2.88 | 0.82 | −3.12 | 5.24 | 3.29 | 7.79 | 12.06 | 12.88 | 13.00 | 6.50 | 11.73 | 9.85 | 9.94 | 10.32 | 14.88 | 15.44 |
| RF-S-T | −0.35 | −2.35 | −1.22 | −1.26 | - | 6.82 | 1.41 | 5.12 | 1.18 | 9.53 | 7.59 | 12.09 | 16.35 | 17.18 | 17.29 | 10.79 | 16.03 | 14.15 | 14.23 | 14.62 | 19.18 | 19.73 |
| RF-T | 2.29 | 0.29 | 1.43 | 1.38 | 2.65 | - | −5.41 | −1.71 | −5.65 | 2.71 | 0.77 | 5.26 | 9.53 | 10.35 | 10.47 | 3.97 | 9.21 | 7.32 | 7.41 | 7.79 | 12.35 | 12.91 |
| RF-B | 2.27 | 0.27 | 1.40 | 1.35 | 2.62 | −0.03 | - | 3.71 | −0.23 | 8.12 | 6.18 | 10.68 | 14.94 | 15.77 | 15.88 | 9.38 | 14.62 | 12.73 | 12.82 | 13.21 | 17.77 | 18.32 |
| RF-RO | 0.48 | −1.51 | −0.38 | −0.43 | 0.84 | −1.81 | −1.78 | - | −3.94 | 4.41 | 2.47 | 6.97 | 11.23 | 12.06 | 12.18 | 5.68 | 10.91 | 9.03 | 9.12 | 9.50 | 14.06 | 14.62 |
| RF-S | −0.47 | −2.47 | −1.34 | −1.38 | −0.12 | −2.77 | −2.73 | −0.96 | - | 8.35 | 6.41 | 10.91 | 15.18 | 16.00 | 16.12 | 9.62 | 14.85 | 12.97 | 13.06 | 13.44 | 18.00 | 18.56 |
| RF-RU | 9.65 | 7.65 | 8.78 | 8.73 | 10.00 | 7.35 | 7.38 | 9.16 | 10.12 | - | −1.94 | 2.56 | 6.82 | 7.65 | 7.76 | 1.26 | 6.50 | 4.62 | 4.71 | 5.09 | 9.65 | 10.21 |
| RF-N | 1.53 | −0.47 | 0.66 | 0.62 | 1.88 | −0.77 | −0.73 | 1.04 | 2.00 | −8.12 | - | 4.50 | 8.77 | 9.59 | 9.71 | 3.21 | 8.44 | 6.56 | 6.65 | 7.03 | 11.59 | 12.15 |
| LR-S-E | 4.99 | 2.98 | 4.12 | 4.07 | 5.34 | 2.69 | 2.72 | 4.50 | 5.46 | −4.66 | 3.46 | - | 4.26 | 5.09 | 5.21 | −1.29 | 3.94 | 2.06 | 2.15 | 2.53 | 7.09 | 7.65 |
| LR-E | 4.03 | 2.03 | 3.16 | 3.12 | 4.38 | 1.74 | 1.76 | 3.54 | 4.50 | −5.62 | 2.50 | −0.96 | - | 0.82 | 0.94 | −5.56 | −0.32 | −2.21 | −2.12 | −1.74 | 2.82 | 3.38 |
| LR-C | 7.79 | 5.79 | 6.93 | 6.88 | 8.15 | 5.50 | 5.53 | 7.31 | 8.27 | −1.85 | 6.26 | 2.81 | 3.77 | - | 0.12 | −6.38 | −1.15 | −3.03 | −2.94 | −2.56 | 2.00 | 2.56 |
| LR-SF | 8.32 | 6.32 | 7.46 | 7.41 | 8.68 | 6.03 | 6.06 | 7.84 | 8.79 | −1.32 | 6.79 | 3.34 | 4.29 | 0.53 | - | −6.50 | −1.26 | −3.15 | −3.06 | −2.68 | 1.88 | 2.44 |
| LR-S-T | 6.28 | 4.28 | 5.41 | 5.37 | 6.63 | 3.98 | 4.01 | 5.79 | 6.75 | −3.37 | 4.75 | 1.29 | 2.25 | −1.51 | −2.04 | - | 5.24 | 3.35 | 3.44 | 3.82 | 8.38 | 8.94 |
| LR-T | 5.49 | 3.48 | 4.62 | 4.57 | 5.84 | 3.19 | 3.22 | 5.00 | 5.96 | −4.16 | 3.96 | 0.50 | 1.46 | −2.31 | −2.84 | −0.79 | - | −1.88 | −1.79 | −1.41 | 3.15 | 3.71 |
| LR-B | 6.68 | 4.68 | 5.81 | 5.76 | 7.03 | 4.38 | 4.41 | 6.19 | 7.15 | −2.97 | 5.15 | 1.69 | 2.65 | −1.12 | −1.65 | 0.40 | 1.19 | - | 0.09 | 0.47 | 5.03 | 5.59 |
| LR-S | 6.44 | 4.44 | 5.57 | 5.53 | 6.79 | 4.15 | 4.18 | 5.96 | 6.91 | −3.21 | 4.91 | 1.46 | 2.41 | −1.35 | −1.88 | 0.16 | 0.96 | −0.23 | - | 0.38 | 4.94 | 5.50 |
| LR-RO | 8.60 | 6.60 | 7.74 | 7.69 | 8.96 | 6.31 | 6.34 | 8.12 | 9.07 | −1.04 | 7.07 | 3.62 | 4.57 | 0.81 | 0.28 | 2.32 | 3.12 | 1.93 | 2.16 | - | 4.56 | 5.12 |
| LR-RU | 12.27 | 10.27 | 11.40 | 11.35 | 12.62 | 9.97 | 10.00 | 11.78 | 12.73 | 2.62 | 10.73 | 7.28 | 8.23 | 4.47 | 3.94 | 5.99 | 6.78 | 5.59 | 5.82 | 3.66 | - | 0.56 |
| LR-N | 5.71 | 3.71 | 4.84 | 4.79 | 6.06 | 3.41 | 3.44 | 5.22 | 6.18 | −3.94 | 4.18 | 0.72 | 1.68 | −2.09 | −2.62 | −0.57 | 0.22 | −0.97 | −0.73 | −2.90 | −6.56 | - |

Table C.5: Pairwise rank differences of high-complexity datasets (33 datasets). CD: 5.7437.

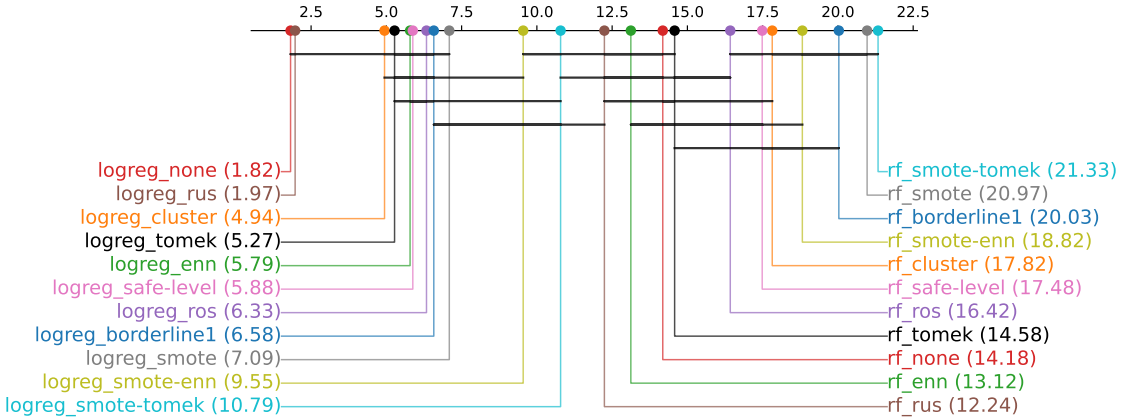
| method | RF-S-E | RF-E | RF-C | RF-SF | RF-S-T | RF-T | RF-B | RF-RO | RF-S | RF-RU | RF-N | LR-S-E | LR-E | LR-C | LR-SF | LR-S-T | LR-T | LR-B | LR-S | LR-RO | LR-RU | LR-N |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RF-S-E | - | 5.70 | 1.00 | 1.33 | −2.52 | 4.24 | −1.21 | 2.39 | −2.15 | 6.58 | 4.64 | 9.27 | 13.03 | 13.88 | 12.94 | 8.03 | 13.54 | 12.24 | 11.73 | 12.48 | 16.85 | 17.00 |
| RF-E | 0.58 | - | −4.70 | −4.36 | −8.21 | −1.46 | −6.91 | −3.30 | −7.85 | 0.88 | −1.06 | 3.58 | 7.33 | 8.18 | 7.24 | 2.33 | 7.85 | 6.54 | 6.03 | 6.79 | 11.15 | 11.30 |
| RF-C | −0.82 | −1.39 | - | 0.33 | −3.52 | 3.24 | −2.21 | 1.39 | −3.15 | 5.58 | 3.64 | 8.27 | 12.03 | 12.88 | 11.94 | 7.03 | 12.54 | 11.24 | 10.73 | 11.48 | 15.85 | 16.00 |
| RF-SF | −0.88 | −1.46 | −0.06 | - | −3.85 | 2.91 | −2.54 | 1.06 | −3.48 | 5.24 | 3.30 | 7.94 | 11.70 | 12.54 | 11.61 | 6.70 | 12.21 | 10.91 | 10.39 | 11.15 | 15.52 | 15.67 |
| RF-S-T | −2.00 | −2.58 | −1.18 | −1.12 | - | 6.76 | 1.30 | 4.91 | 0.36 | 9.09 | 7.15 | 11.79 | 15.54 | 16.39 | 15.46 | 10.54 | 16.06 | 14.76 | 14.24 | 15.00 | 19.36 | 19.52 |
| RF-T | 3.41 | 2.83 | 4.23 | 4.29 | 5.41 | - | −5.46 | −1.85 | −6.39 | 2.33 | 0.39 | 5.03 | 8.79 | 9.64 | 8.70 | 3.79 | 9.30 | 8.00 | 7.49 | 8.24 | 12.61 | 12.76 |
| RF-B | −0.36 | −0.94 | 0.46 | 0.52 | 1.64 | −3.77 | - | 3.61 | −0.94 | 7.79 | 5.85 | 10.48 | 14.24 | 15.09 | 14.15 | 9.24 | 14.76 | 13.46 | 12.94 | 13.70 | 18.06 | 18.21 |
| RF-RO | 2.39 | 1.82 | 3.21 | 3.27 | 4.39 | −1.01 | 2.76 | - | −4.54 | 4.18 | 2.24 | 6.88 | 10.64 | 11.48 | 10.54 | 5.64 | 11.15 | 9.85 | 9.33 | 10.09 | 14.46 | 14.61 |
| RF-S | −1.51 | −2.09 | −0.70 | −0.64 | 0.48 | −4.92 | −1.15 | −3.91 | - | 8.73 | 6.79 | 11.42 | 15.18 | 16.03 | 15.09 | 10.18 | 15.70 | 14.39 | 13.88 | 14.64 | 19.00 | 19.15 |
| RF-RU | 3.58 | 3.00 | 4.39 | 4.46 | 5.58 | 0.17 | 3.94 | 1.18 | 5.09 | - | −1.94 | 2.70 | 6.46 | 7.30 | 6.36 | 1.46 | 6.97 | 5.67 | 5.15 | 5.91 | 10.27 | 10.42 |
| RF-N | 4.91 | 4.33 | 5.73 | 5.79 | 6.91 | 1.50 | 5.27 | 2.52 | 6.42 | 1.33 | - | 4.64 | 8.39 | 9.24 | 8.30 | 3.39 | 8.91 | 7.61 | 7.09 | 7.85 | 12.21 | 12.36 |
| LR-S-E | 1.06 | 0.48 | 1.88 | 1.94 | 3.06 | −2.35 | 1.42 | −1.33 | 2.58 | −2.52 | −3.85 | - | 3.76 | 4.61 | 3.67 | −1.24 | 4.27 | 2.97 | 2.46 | 3.21 | 7.58 | 7.73 |
| LR-E | 3.03 | 2.46 | 3.85 | 3.91 | 5.03 | −0.38 | 3.39 | 0.64 | 4.54 | −0.55 | −1.88 | 1.97 | - | 0.85 | −0.09 | −5.00 | 0.52 | −0.79 | −1.30 | −0.55 | 3.82 | 3.97 |
| LR-C | 0.70 | 0.12 | 1.51 | 1.58 | 2.70 | −2.71 | 1.06 | −1.70 | 2.21 | −2.88 | −4.21 | −0.36 | −2.33 | - | −0.94 | −5.85 | −0.33 | −1.64 | −2.15 | −1.39 | 2.97 | 3.12 |
| LR-SF | 2.03 | 1.46 | 2.85 | 2.91 | 4.03 | −1.38 | 2.39 | −0.36 | 3.54 | −1.54 | −2.88 | 0.97 | −1.00 | 1.33 | - | −4.91 | 0.61 | −0.70 | −1.21 | −0.46 | 3.91 | 4.06 |
| LR-S-T | 1.09 | 0.52 | 1.91 | 1.97 | 3.09 | −2.32 | 1.46 | −1.30 | 2.61 | −2.48 | −3.82 | 0.03 | −1.94 | 0.39 | −0.94 | - | 5.51 | 4.21 | 3.70 | 4.46 | 8.82 | 8.97 |
| LR-T | 7.21 | 6.64 | 8.03 | 8.09 | 9.21 | 3.80 | 7.58 | 4.82 | 8.73 | 3.64 | 2.30 | 6.15 | 4.18 | 6.51 | 5.18 | 6.12 | - | −1.30 | −1.82 | −1.06 | 3.30 | 3.46 |
| LR-B | −0.21 | −0.79 | 0.61 | 0.67 | 1.79 | −3.62 | 0.15 | −2.61 | 1.30 | −3.79 | −5.12 | −1.27 | −3.24 | −0.91 | −2.24 | −1.30 | −7.42 | - | −0.52 | 0.24 | 4.61 | 4.76 |
| LR-S | 1.27 | 0.70 | 2.09 | 2.15 | 3.27 | −2.14 | 1.64 | −1.12 | 2.79 | −2.30 | −3.64 | 0.21 | −1.76 | 0.58 | −0.76 | 0.18 | −5.94 | 1.49 | - | 0.76 | 5.12 | 5.27 |
| LR-RO | 1.46 | 0.88 | 2.27 | 2.33 | 3.46 | −1.96 | 1.82 | −0.94 | 2.97 | −2.12 | −3.46 | 0.39 | −1.58 | 0.76 | −0.58 | 0.36 | −5.76 | 1.67 | 0.18 | - | 4.36 | 4.51 |
| LR-RU | 5.06 | 4.49 | 5.88 | 5.94 | 7.06 | 1.65 | 5.42 | 2.67 | 6.58 | 1.49 | 0.15 | 4.00 | 2.03 | 4.36 | 3.03 | 3.97 | −2.15 | 5.27 | 3.79 | 3.61 | - | 0.15 |
| LR-N | 8.35 | 7.77 | 9.17 | 9.23 | 10.35 | 4.94 | 8.71 | 5.96 | 9.86 | 4.77 | 3.44 | 7.29 | 5.32 | 7.65 | 6.32 | 7.26 | 1.14 | 8.56 | 7.08 | 6.89 | 3.29 | - |




(a) Total energy CD diagram for low-complexity datasets (33 datasets). CD: 5.7437.

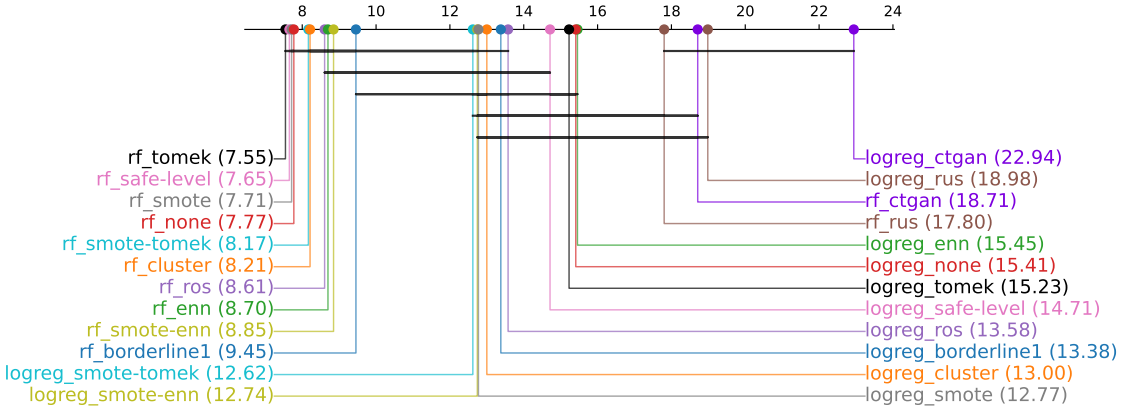



(b) Total energy CD diagram for medium-complexity datasets (34 datasets). CD: 5.6586.

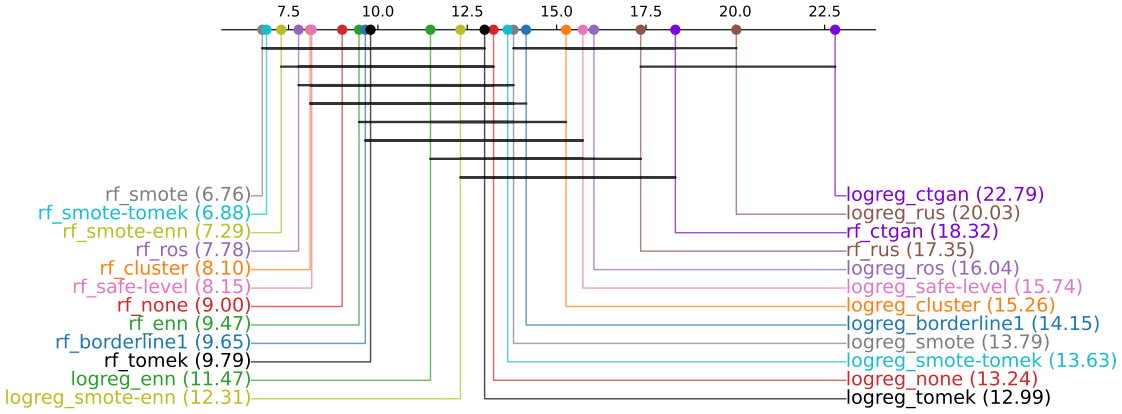



(c) Total energy CD diagram for high-complexity datasets (33 datasets). CD: 5.7437.

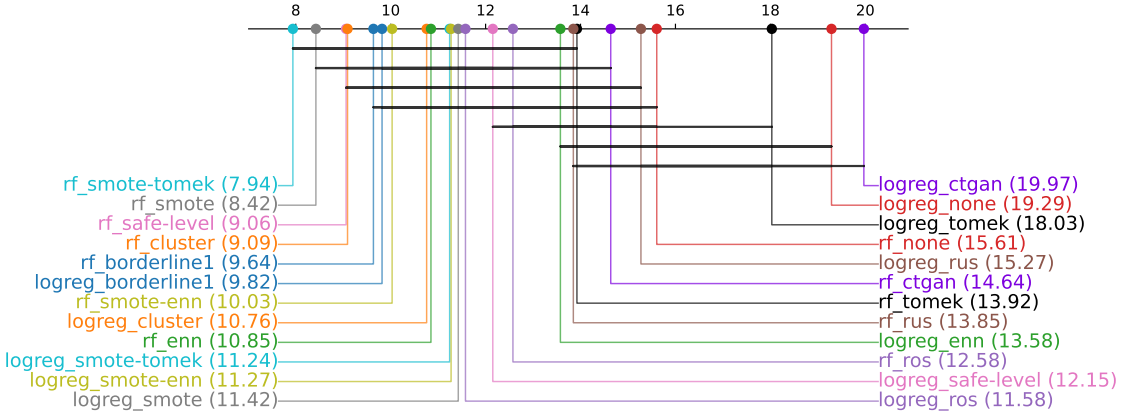
Figure C.1: CD diagrams for total energy by dataset complexity groups. For connections that are harder to visualize, refer to tables C.3-C.5.




(a) F1-score CD diagram for low-complexity datasets (33 datasets). CD: 6.3316.

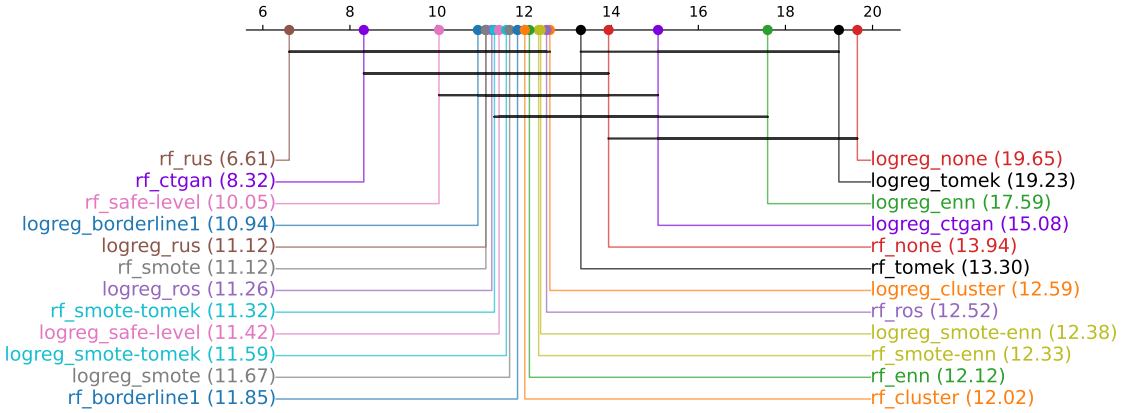



(b) F1-score CD diagram for medium-complexity datasets (34 datasets). CD: 6.2378.

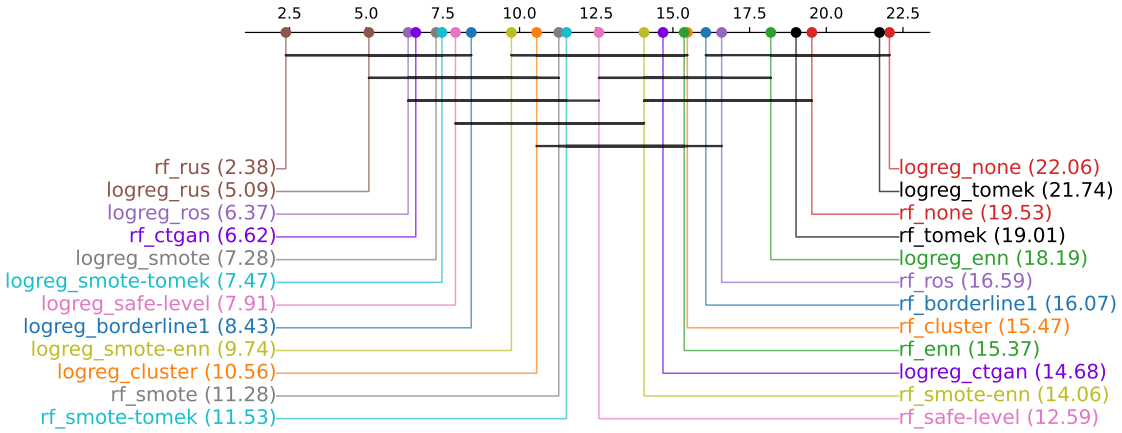



(c) F1-score CD diagram for high-complexity datasets (33 datasets). CD: 6.3316.

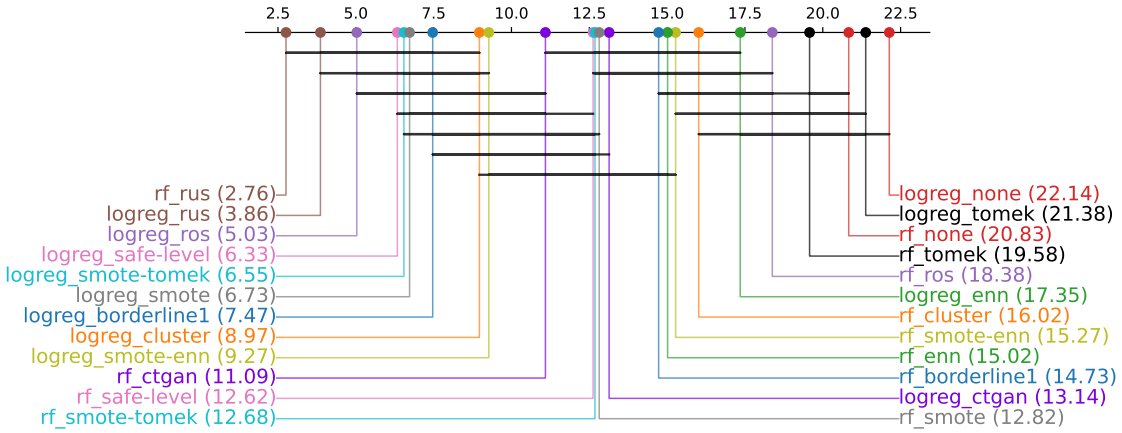
Figure C.2: CD diagrams for F1-score by dataset complexity groups with CTGAN results included.




(a) Recall CD diagram for low-complexity datasets (33 datasets). CD: 6.3316.




(b) Recall CD diagram for medium-complexity datasets (34 datasets). CD: 6.2378.




(c) Recall CD diagram for high-complexity datasets (33 datasets). CD: 6.3316.

Figure C.3: CD diagrams for recall by dataset complexity groups with CTGAN results included.