From Pipeline Optimization To Problem-oriented Automl: Advancing Clustering Automation

UNIVERSITÀ DEGLI STUDI DI TRIESTE

by

Matheus Camilo da Silva

Submitted for the degree of

Doctor of Philosofy

PROF. SYLVIO BARBON JR AND PROF. ERIC MEDVET
DEPARTMENT OF ENGINEERING AND ARCHITECTURE
UNIVERSITY OF TRIESTE

Academic Year 2024/ 2025

Abstract

Automated Machine Learning (AutoML) aims to lower the entry barrier of machine learning by automating the design of pipelines, including algorithm selection, hyperparameter optimisation, and model composition. While AutoML has matured substantially in supervised learning, its extension to unsupervised tasks, particularly clustering—remains limited. Existing AutoClustering approaches predominantly optimise fixed internal validation indices and treat clustering quality as an objective notion, despite strong evidence that clustering is inherently subjective and task-dependent. Moreover, although meta-learning has been applied to clustering, its role is often restricted to static algorithm recommendation, leaving user intent and problem semantics largely unaddressed.

This thesis addresses these gaps by investigating how pipeline synthesis, meta-learning, and surrogate modelling can be combined to enable problem-oriented AutoML for clustering. First, evolutionary pipeline synthesis is extended to unsupervised learning through TPOT-Clustering. Empirical results across diverse datasets show that optimisation based on individual clustering validity indices—or their ensembles—leads to over-specialised solutions that fail to generalise across problem contexts.

Second, a systematic review and empirical analysis of meta-learning in AutoClustering reveals that a compact subset of statistical and landmarking meta-features dominates predictive performance, while more complex descriptors provide marginal benefit despite higher computational cost. These findings demonstrate that current meta-spaces are often overparameterised and can be simplified without substantial loss in effectiveness.

Finally, these insights are integrated into the Problem-oriented AutoML in Clustering (PoAC) framework, which employs surrogate models and meta-objectives to approximate user-defined notions of clustering quality. Experimental results show that PoAC produces adaptive, algorithm-agnostic clustering pipelines that better align with problem-specific goals, such as visualisation and anomaly detection, while improving interpretability and search efficiency. Overall, this work reframes AutoClustering as a problem-oriented, knowledge-driven process, advancing AutoML toward more human-aligned automation in unsupervised learning.

Summary

Chapter 1: Introduction Motivates the work, defines the problem, presents research questions, and outlines the thesis contributions.

Chapter 2: Background and Related Work Introduces key AutoML principles: algorithm selection, hyperparameter optimization, and pipeline synthesis—and discusses their relevance and challenges for clustering.

Chapter 3: AutoML for Clustering: Pipeline Synthesis and Meta-Objectives Details the evolutionary pipeline synthesis framework for unsupervised learning and presents its experimental evaluation on multiple clustering tasks.

Chapter 4: Building Meta-Spaces and Meta-Objectives for Clustering Provides a systematic taxonomy of meta-features, datasets, and evaluation strategies, highlighting how prior knowledge can guide pipeline selection.

Chapter 5: The Problem-Oriented AutoML in Clustering (PoAC) Framework Integrates surrogate modeling with meta-learning to optimize clustering pipelines based on problem-oriented objectives, demonstrated in visualization and anomaly detection scenarios.

Chapter 6: Discussions Synthesizes findings across the studies, explicitly answers the research questions, and discusses implications for designing AutoML systems for subjective tasks like clustering.

Chapter 7: Conclusion Summarizes the key contributions of the thesis and outlines future research directions.

Appendix A: Publications and Scientific Output During the PhD Lists all publications produced during the doctoral programme, including works related to the thesis and other research projects.

Table of Contents

Figure table

Note: Unless otherwise stated, all figures in this thesis are the author's own work.

List of Tables

Dedication

I dedicate this work to my wife and adventure companion, Victória, whose unwavering love, support, and encouragement sustained me throughout this journey.

Acknowledgements

I am deeply grateful to my supervisor, Prof. Sylvio Barbon, for his guidance, patience, and insightful feedback throughout this journey. His mentorship has been invaluable in shaping both this research and my growth as a scholar.

I also wish to thank my colleagues and friends, whose collaboration, encouragement, and occasional distractions made this process more enjoyable and meaningful. Their support reminded me that even in the most challenging moments, shared laughter and thoughtful discussion can make all the difference.

Preface

The motivation for this work arises from a simple but persistent question: can we teach machines to understand what makes a model *"good"* for a given problem, even when there are no labels to guide them? Among the many challenges in Machine Learning, this question is perhaps most evident in *clustering*, a task where success cannot be measured by accuracy or error rates, but rather by how well the discovered patterns align with human understanding and analytical goals. As data grows more complex and heterogeneous, translating this human intuition into scalable, systematic processes becomes increasingly important.

This thesis explores how automation can move beyond static optimization and begin to reason about problems themselves, how to represent them, how to generalize from experience, and how to adapt to new objectives. By focusing on clustering, one of the most fundamental yet subjective problems in unsupervised learning, this work seeks to bring AutoML closer to problem understanding rather than mere performance optimization.

The research presented here builds on the intersection of AutoML, meta-learning, and surrogate modeling. It aims to bridge the gap between automatic pipeline synthesis and context-aware problem formulation, contributing to the broader vision of a more flexible and interpretable form of machine learning automation.

Chapter 1

Introduction

Contents

Building on the motivations introduced in the Preface, this chapter presents the context, challenges, and objectives of the thesis. It situates the research within the broader field of Automated Machine Learning (AutoML) and outlines how this work contributes to advancing clustering automation through problem-oriented design.

1.1 Towards Problem-Oriented Automation

Machine Learning (ML) has become an essential tool across diverse domains, from bioinformatics to natural language processing and finance. This widespread adoption has given rise to increasingly intricate analytical workflows, where data preprocessing, feature engineering, model selection, and hyperparameter tuning must interact seamlessly. The resulting structure, commonly referred to as an *ML pipeline*, defines a sequence of computational steps that transform raw data into actionable insight (see Figure 1.1).

Designing effective ML pipelines often demands substantial expertise and extensive experimentation (Olson et al., 2016; Hutter et al., 2019). AutoML seeks to alleviate this burden by automatically searching for suitable models and configurations (Guyon et al., 2015; Brazdil et al., 2022).

To tackle this complex automation, AutoML divides it into distinct sub-problems, each progressively expanding the scope and flexibility of the search space. At the foundational level, *Algorithm Selection (AS)* focuses on identifying the most suitable learning

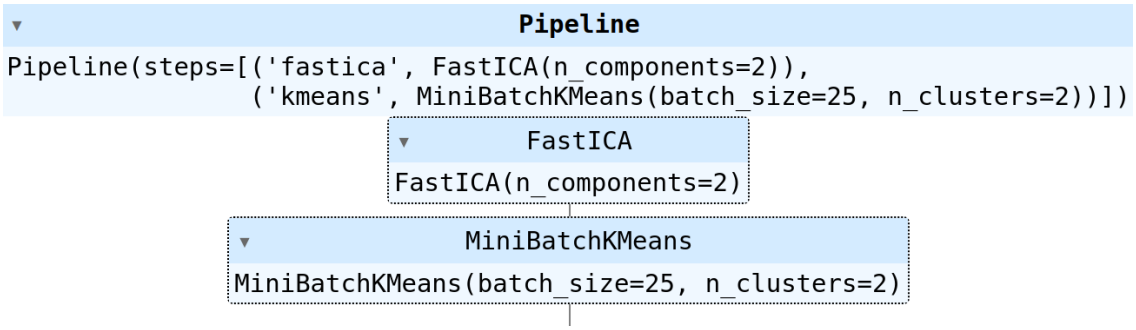


Figure 1.1: An example of a clustering pipeline, illustrating a sequential process where the Fast Independent Component Analysis (FastICA) step reduces the dimensionality of the input data, followed by the MiniBatchKMeans step, which partitions the transformed data into distinct clusters.

algorithm for a given dataset. Building on this, *Hyperparameter Optimization (HPO)* fine-tunes the internal Hyperparameter of a chosen algorithm to improve performance. The *Combined Algorithm Selection and Hyperparameter Optimization (CASH)* formulation unifies these two tasks, searching jointly over algorithm choices and their configurations. Finally, the most general form of automation, *Pipeline Synthesis (PS)*, extends beyond model selection to incorporate data preprocessing, feature engineering, and algorithmic composition, thereby constructing complete end-to-end pipelines automatically. For a more detailed discussion of these tasks, see Section 2.2.

Each step in this hierarchy increases the dimensionality and complexity of the search space. While this complicates optimization, it also unlocks the potential to discover richer and more adaptive solutions, particularly important in heterogeneous or open-ended problem domains such as clustering.

1.2 Bridging Pipeline Synthesis and Meta-Learning

While the automation of supervised learning has benefited from clear optimization targets such as accuracy or loss, unsupervised learning—particularly clustering—lacks universally accepted objectives. In this setting, AutoClustering systems must rely on alternative optimization strategies to evaluate and guide pipeline synthesis.

A common approach is to optimize pipelines using internal *Cluster Validity Indices* (CVIs), which estimate clustering quality based on structural properties of the data, such as compactness and separation (Vendramin et al., 2010; Halkidi et al., 2001). Although this allows fully automated search, it also imposes rigid assumptions about what constitutes a “good” clustering. CVIs capture specific statistical patterns, not the analytical

goals or interpretability needs of the user. As discussed in Chapter 3, such optimization often leads to pipelines that perform well according to these indices but fail to produce meaningful results in practice.

An alternative line of research leverages *meta-learning* to overcome this limitation. Instead of optimizing directly over CVIs, meta-learning draws on prior experience from a collection of datasets to infer relationships between dataset characteristics (meta-features), algorithmic configurations, and observed clustering outcomes (Brazdil et al., 2008). This allows AutoClustering frameworks to transfer knowledge across tasks, effectively learning *meta-objectives* that capture higher-level notions of performance. Chapter 4 explores this direction in detail, presenting a systematic review of meta-learning approaches in unsupervised AutoML.

Building on these foundations, this thesis hypothesizes that meaningful clustering automation requires unifying *pipeline synthesis* and *meta-learning* under a common, problem-oriented framework. Pipeline synthesis provides the mechanism for exploring and constructing candidate workflows, while meta-learning introduces inductive knowledge about how algorithms behave across datasets. Together, they can enable AutoClustering systems to move beyond static, index-based optimization and adapt dynamically to new problems.

In this context, *surrogate models* play a pivotal role. These models approximate the performance of candidate clustering configurations as a function of both meta-features and algorithmic choices. By learning these relationships, surrogate models enable optimization to be guided by *learned objectives* rather than fixed indices, aligning the search process with user intent and contextual priorities. The design and implementation of such surrogate models are discussed in Chapter 5, where they form the backbone of the proposed Problem-oriented AutoML in Clustering (PoAC) framework.

This integration defines the problem-oriented automation paradigm explored throughout this thesis: AutoML systems should not only automate model discovery but also reason about what constitutes meaningful performance given the data, the task, and the user's analytical intent.

1.3 Problem Statement

The core problem addressed in this thesis is the absence of adaptive, user-aware automation in clustering. Existing AutoClustering frameworks rely on fixed objectives,

typically internal validity indices, that inadequately capture the contextual and subjective nature of clustering quality. Moreover, the separation between pipeline synthesis and meta-learning limits their ability to transfer knowledge and adapt to new problems. This thesis seeks to overcome these limitations by unifying both paradigms through surrogate-based, problem-oriented automation.

1.4 Research Objectives and Questions

The overarching goal of this thesis is to advance the automation of clustering by developing methods that combine pipeline synthesis and meta-learning into a unified, problem-oriented AutoML framework. Specifically, this work investigates how surrogate models can be used to guide unsupervised pipeline optimization based on meta-knowledge, user intent, and contextual information.

To achieve this goal, the thesis addresses the following research questions:

- **RQ1:** How can pipeline synthesis methods be adapted to support flexible and data-driven optimization in clustering, beyond fixed internal validation metrics?
- **RQ2:** Which meta-features and meta-learning strategies are most effective for characterizing clustering problems and predicting algorithmic performance?
- **RQ3:** How can surrogate models and meta-objectives be combined to create a problem-oriented AutoML framework capable of generating clustering pipelines aligned with user intent?

Addressing these questions requires integrating principles from AutoML, meta-learning, and surrogate modeling into a cohesive framework that can learn from prior experience while adapting to new and diverse clustering tasks.

1.5 Contributions

To answer these questions, this thesis introduces three core contributions that collectively advance the state of AutoClustering through the integration of meta-learning and pipeline synthesis:

1. **A structured literature review and taxonomy of meta-learning in AutoClustering.** We conduct a comprehensive review of over 20 representative frameworks

spanning 2008–2025, identifying key trends in meta-feature design, dataset usage, and evaluation strategies. This taxonomy provides a structured understanding of how meta-learning has been applied to clustering and highlights open challenges in the construction of meta-spaces and meta-objectives.

2. **TPOT-Clustering: An evolutionary pipeline synthesis framework for unsupervised learning.** Building on the TPOT system, we introduce *TPOT-Clustering*, an extension that automates the design of clustering pipelines through evolutionary optimization. TPOT-Clustering supports both CVI-based and surrogate-based objectives, allowing users to tailor optimization criteria to specific analytical goals.

3. **PoAC: A problem-oriented AutoML framework for clustering.** We propose *PoAC* (Problem-oriented AutoML in Clustering), a unified framework that integrates meta-learning, surrogate modeling, and adaptive optimization. PoAC learns meta-objectives that approximate user-defined quality functions, enabling the synthesis of clustering pipelines tailored to distinct contexts such as visualization and anomaly detection.

Collectively, these contributions demonstrate how surrogate-based optimization and meta-learning can transform AutoClustering from a metric-driven process into a context-aware, problem-oriented paradigm.

1.6 Thesis Structure

The remainder of this thesis is organized as follows:

- **Chapter 2** introduces the theoretical background of AutoML, covering algorithm selection, hyperparameter optimization, and pipeline synthesis, and discusses open challenges in extending these principles to clustering.

- **Chapter 3** presents *TPOT-Clustering*, detailing its architecture, optimization strategy, and experimental evaluation across multiple unsupervised learning tasks.

- **Chapter 4** develops a *taxonomy of meta-learning in AutoClustering*, mapping how meta-features, datasets, and evaluation measures have been employed across existing frameworks.

- **Chapter 5** introduces the *PoAC framework*, integrating surrogate-based optimization with problem-oriented meta-objectives and demonstrating its effectiveness in visualization and anomaly detection tasks.
- **Chapter 6** discusses broader implications and open challenges in problem-oriented AutoML, synthesizing insights from all experimental findings.
- **Chapter 7** concludes the thesis, summarizing key contributions and outlining directions for future research.

Through this progression, the thesis traces a trajectory from conventional pipeline optimization toward a new generation of adaptive, meta-knowledge–driven AutoML systems. By advancing the integration of meta-learning and pipeline synthesis, it contributes to the foundation of a problem-oriented approach to clustering automation.

Chapter 2

Background and Related Work

Contents

2.1 Clustering

Clustering is a fundamental task in unsupervised learning that aims to discover structure within unlabeled data by grouping similar instances into clusters. Unlike supervised learning, there is no ground truth to guide optimization, making the definition of "good" clusters inherently subjective and context-dependent (von Luxburg et al., 2012).

From a practical standpoint, clustering can be viewed as a composite task involving three interdependent components: (i) preprocessing and representation of the data Jain et al. (1999), (ii) selection and configuration of a clustering algorithm Jain et al. (1999); Xu and Wunsch (2008), and (iii) evaluation of the resulting partition (Halkidi et al., 2001; Rokach and Maimon, 2005). Together, these stages define the design space that AutoClustering systems aim to automate.

2.1.1 Types of Clustering Algorithms

Different clustering paradigms capture different notions of similarity and structure in data (Xu and Wunsch, 2005). Understanding their principles is essential for contextualizing AutoClustering, as each algorithm family interacts differently with data characteristics and CVIs.

- **Partitional methods:** Algorithms such as k-means (McQueen, 1967) and k-medoids (Kaufman, 1990) aim to partition the data into a predefined number of clusters by minimizing intra-cluster variance. These methods are efficient but sensitive to initialization and scaling.
- **Hierarchical methods:** These algorithms (e.g., agglomerative clustering, BIRCH) (Mojena, 1977; Zhang et al., 1996) build a hierarchy of clusters represented as a dendrogram. They are flexible and interpretable but scale poorly for large datasets.
- **Density-based methods:** Algorithms such as DBSCAN (Ester et al., 1996) and OPTICS (Ankerst et al., 1999) identify clusters as regions of high density separated by sparse regions. They can discover arbitrarily shaped clusters and handle noise well but depend strongly on density hyperparameters.
- **Model-based methods:** Mixture models, such as Gaussian Mixture Models (GMMs), assume data are generated by probabilistic distributions and use likelihood-based estimation (McLachlan and Peel, 2000). These approaches offer statistical interpretability but are sensitive to assumptions about cluster shape.
- **Spectral and graph-based methods:** These methods (e.g., Spectral Clustering (Von Luxburg, 2007)) use eigen-decomposition of similarity matrices to capture nonlinear structures, often outperforming distance-based approaches for complex manifolds.

Each family of algorithms introduces unique biases and sensitivities—particularly to data preprocessing (e.g., scaling, normalization, noise). These variations complicate the definition of universal clustering quality criteria, motivating the use of CVIs to assess results.

Internal CVIs

Internal CVIs are quantitative measures designed to evaluate the quality of clustering results in the absence of external ground truth labels. These indices assess the intrinsic properties of the partitioning by analysing relationships between data points, typically balancing notions of compactness, how cohesive individual clusters are, and separation, how distinct clusters are from one another. Internal CVIs are essential for unsupervised learning tasks, where performance must be inferred from the data's geometry and density structure rather than predefined class labels (Halkidi et al., 2001; Vendramin et al., 2010).

The Silhouete score (SIL) ranges from -1 to 1: a value close to 1 for a data point indicates that it is well-clustered, meaning it is further away from neighbouring clusters than its own; a value around 0 suggests that the point lies on the boundary between two clusters; and a negative value indicates that the point may be assigned to the wrong cluster, as it is closer to a different cluster than its own. The average SIL across all points in a dataset provides an overall measure of clustering quality (Rousseeuw, 1987). Let the i, j be n-dimensional feature vectors (data points) and $i, j \in C_I$; C as a given cluster of the dataset D; $d(i, j)$ as the average Euclidean distance between data points i and j; $|C_I|$ as the number of data points in cluster C_I and $|N|$ is the total number of data points present in the dataset D. The SIL for i can be calculated using Equation (2.1) if and only if $|C_I| > 1$.

$$\mathscr{SIL}(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \tag{2.1}$$

Where, $a(i)$ is the average distance from the i-th data point to the other data points in the same cluster, given by Equation (2.2).

$$a(i) = \frac{1}{|C_I| - 1} \sum_{j \in C_I, j \neq i} d(i, j) \tag{2.2}$$

And, $b(i)$ is the smallest average distance between the i-th data point of C_I and the k-th data point of C_K, where $C_I \neq C_K$, given by Equation (2.3).

$$b(i) = \min_{K \neq I} \frac{1}{|C_K|} \sum_{k \in C_K} d(i, k) \tag{2.3}$$

Finally, the overall SIL for the entire dataset D is the average of the SIL for all data points, as given by Equation (2.4):

$$\mathscr{SIL}_D = \frac{1}{|N|}\sum_{i=1}^{N}\mathscr{SIL}(i) \tag{2.4}$$

The Davies-Bouldin Score (DBS), as introduced by Davies and Bouldin (1979), is defined as the ratio of compactness within clusters and the separation between clusters. It ranges from 0 to infinite, where a DBS closer to 0 indicates better (more compact) clustering, with well-defined and separated clusters. Let C_I be a cluster of a dataset D; σ the average Euclidean distance from each data point of C_I to its centroid A_I; $d(C_I, C_K)$ be the Euclidean distance between the A_I and A_K; and the data-point $i \in C_I$; $|C|$ be the total number of clusters in D; and $|C_I|$ be the total number of data points in C_I.

Where, the compactness of C_I is given by Equation (2.5):

$$\sigma_I = \frac{1}{|C_I|}\sum_{i=1}^{|C_I|}||i - A_I|| \tag{2.5}$$

The distance between the centroids is given by Equation (2.6):

$$d(C_I, C_K) = ||A_I - A_K|| \tag{2.6}$$

And, the DBS for the entire dataset D is calculated using the Equation (2.7):

$$\mathscr{DBS}_D = \frac{1}{|C|}\sum_{I=1}^{|C|}\max_{K\neq I}\left(\frac{\sigma_I + \sigma_K}{d(C_I, C_K)}\right) \tag{2.7}$$

The Calinski–Harabasz Score (CHS), also known as the Variance Ratio Criterion, was introduced by Caliński and Harabasz (1974). It evaluates clustering quality based on the ratio of between-cluster dispersion to within-cluster dispersion. Higher CHS values indicate better-defined clusters with high separation and compactness.

Let C_I be a cluster of a dataset D; A_I the centroid of cluster C_I; A_D the global centroid of the dataset; $|C_I|$ the number of samples in C_I; and $|C|$ the total number of clusters. The within-cluster dispersion W and the between-cluster dispersion B are defined as:

$$W = \sum_{I=1}^{|C|}\sum_{i\in C_I}||i - A_I||^2 \tag{2.8}$$

$$B = \sum_{I=1}^{|C|}|C_I|\,||A_I - A_D||^2 \tag{2.9}$$

The CHS for dataset D is then given by Equation (2.10):

$$\mathscr{CHS}_D = \frac{B/(|C|-1)}{W/(|N|-|C|)} \tag{2.10}$$

where $|N|$ is the total number of data points in the dataset. Higher $\mathscr{CHS}_D$ values correspond to better-defined and well-separated cluster structures.

The Density-Based Clustering Validation Score (DBCV), proposed by Moulavi et al. (2014), is a validation index specifically designed for density-based clustering algorithms such as DBSCAN. Unlike centroid-based measures, DBCV evaluates clustering quality by considering both the density-connectedness within clusters and the separability between clusters.

Let $\rho(i)$ denote the local density of point i, estimated as the inverse of the average distance to its k nearest neighbors; and $r(i,j)$ the reachability distance between points i and j, defined based on density connectivity. The density separation between two clusters C_I and C_K is given by:

$$\text{Sep}(C_I, C_K) = \min_{i \in C_I, j \in C_K} r(i,j) \tag{2.11}$$

The density within-cluster connectedness for cluster C_I is defined as the average inverse reachability distance between all points within the cluster:

$$\text{Con}(C_I) = \frac{1}{|C_I|(|C_I|-1)} \sum_{i,j \in C_I, i \neq j} \frac{1}{r(i,j)} \tag{2.12}$$

The DBCV for the entire clustering is then calculated as:

$$\mathscr{DBCV}_D = \frac{\sum_{I=1}^{|C|} \text{Validity}(C_I) \cdot |C_I|}{\sum_{I=1}^{|C|} |C_I|} \tag{2.13}$$

where $\text{Validity}(C_I)$ is a function of the ratio between connectedness and separability:

$$\text{Validity}(C_I) = \frac{\text{Con}(C_I) - \max_{K \neq I} \text{Sep}(C_I, C_K)}{\max\left(\text{Con}(C_I), \max_{K \neq I} \text{Sep}(C_I, C_K)\right)} \tag{2.14}$$

The DBCV value ranges from -1 to 1, where higher scores indicate better-defined, density-coherent clusters that are well separated from one another.

External CVIs

External CVIs evaluate the quality of clustering results with respect to a known ground truth or reference partition. Unlike internal indices, which rely solely on data geometry, external CVIs quantify the agreement between the predicted clusters and the true labels by measuring similarity, overlap, or consistency between the two partitions. External CVIs are primarily used for benchmarking and validation purposes, providing an objective means to compare algorithms when true labels are available (Halkidi et al., 2001; Xu and Wunsch, 2008).

The Adjusted Ranom Index (ARI) is a essentially an expansion of the Rand Index (RI), which is a measure of similarity between two partitionings (Hubert and Arabie, 1985). It is an external CVI, meaning it is useful for evaluating the performance of clustering algorithms when the ground truth is known. The RI considers pairs of samples and classifies them as either concordant or discordant based on whether they are placed in the same or different clusters in both partitionings. It ranges from 0 to 1, where 1 indicates perfect agreement between the two partitionings, and 0 indicates no agreement beyond that expected by chance.

Let n be the number of elements in the set $D = \{o_1, \ldots, o_n\}$, and let $U = \{U_1, \ldots, U_I\}$ and $V = \{V_1, \ldots, V_J\}$ represent two different partitionings of D into I and J clusters, respectively.

The RI is calculated using the formula (2.15):

$$RI = \frac{\alpha + \beta}{\binom{n}{2}} \tag{2.15}$$

Where:

- α is the number of pairs of elements (o_i, o_j) that are placed in the **same cluster** in both U and V.
- β is the number of pairs of elements (o_i, o_j) that are placed in **different clusters** in both U and V.

For all $1 \leq i, j \leq n, i \neq j$:

$$\alpha = \left|\{(o_i, o_j) \mid o_i, o_j \in U_k \text{ and } o_i, o_j \in V_l \text{ for some } k, l\}\right| \tag{2.16}$$

$$\beta = \left|\{(o_i, o_j) \mid o_i \in U_{k_1}, o_j \in U_{k_2} \text{ and } o_i \in V_{l_1}, o_j \in V_{l_2}, \text{ for } k_1 \neq k_2 \text{ and } l_1 \neq l_2\}\right| \tag{2.17}$$

In this sense, α counts the number of pairs in the same cluster in both partitionings, while β counts pairs in different clusters in both partitionings. The denominator $\binom{n}{2}$ represents the total number of possible pairs of elements in D.

The ARI accounts for chance agreement, providing a normalized measure that ranges from -1 to 1. A score of 1 indicates perfect agreement, 0 suggests agreement expected by chance, and negative values imply worse-than-chance agreement. The formula for the ARI is given by:

$$\mathscr{ARI} = \frac{\sum_{ij}\binom{n_{ij}}{2} - [\sum_i \binom{\alpha_i}{2} \sum_j \binom{\beta_j}{2}]/\binom{n}{2}}{\frac{1}{2}[\sum_i \binom{\alpha_i}{2} + \sum_j \binom{\beta_j}{2}] - [\sum_i \binom{\alpha_i}{2} \sum_j \binom{\beta_j}{2}]/\binom{n}{2}} \tag{2.18}$$

This formula normalizes the RI by accounting for random chance, making the ARI a valuable tool for assessing clustering accuracy in various fields, including biology, image analysis, and social sciences.

2.2 AutoML

AutoML encompasses a set of techniques aimed at automating the design, configuration, and optimization of ML pipelines (Hutter et al., 2019). In essence, it operates at the intersection between automation and ML (Yao et al., 2019). Given the vast array of algorithms and processing strategies available for each stage of an ML workflow, the resulting configuration space grows exponentially (Hutter et al., 2019). In practical applications, this abundance of alternatives introduces a high degree of complexity, often challenging even experienced practitioners. Traditionally, addressing this complexity relies on iterative, trial-and-error procedures (Zöller and Huber, 2021), which tend to be computationally demanding and time-consuming, frequently yielding suboptimal results due to limited exploration of possible configurations.

To tackle this issue, the AutoML community has proposed various approaches that decompose the overall pipeline optimization into smaller, more tractable subproblems. One of the foundational problems in this context is AS. Given a set of candidate algorithms and a dataset, AS aims to identify the algorithm that achieves the highest performance for

the given data. The concept was originally introduced by Rice (1976), who formalized it as a mapping between a problem space (datasets) and a performance space (evaluation metrics). This relationship can be mediated by a feature space derived from meta-features that characterize the datasets. These meta-features serve as descriptors that connect problem instances to algorithmic behaviours, allowing for a predictive mapping that identifies the algorithm expected to perform best on a new task.

Each algorithm, however, is also governed by a set of hyperparameters that substantially influence its performance, particularly in clustering and unsupervised learning scenarios (Mishra et al., 2022). The process of identifying optimal hyperparameter configurations is known as HPO (Hutter et al., 2019). While conceptually related to AS, HPO differs in that the hyperparameter space is often continuous and high-dimensional, significantly enlarging the search space and complicating the optimization process.

A natural extension of these two ideas leads to the CASH problem, which unifies AS and HPO into a single optimization task (Hutter et al., 2019). Although the resulting search space becomes even more complex, CASH methods often produce superior solutions by jointly considering algorithmic and configurational factors that influence performance. Building upon this concept, PS can be seen as a broader task that not only involves algorithm and hyperparameter choices but also includes other components of the ML pipeline, such as data preprocessing, feature extraction, dimensionality reduction, and selection Vanschoren (2018). In this work, PS is used to denote the overall process of automatically discovering optimal pipeline configurations for clustering problems.

2.3 AutoClustering

AutoClustering has emerged as a subfield of AutoML dedicated to automating the design, configuration, and optimization of clustering pipelines. Unlike supervised learning, where performance can be directly measured through ground-truth labels, clustering requires indirect evaluation. This lack of supervision introduces a fundamental challenge: defining what constitutes a "good" clustering solution. Traditionally, AutoClustering frameworks rely on internal CVIs to assess performance. While these indices effectively capture structural properties such as compactness and separation, they do not necessarily align with the true goals of the analysis, as each CVI reflects a distinct notion of clustering quality (Rokach and Maimon, 2005; von Luxburg et al., 2012). Consequently, frameworks optimized for specific CVIs may generalize poorly across diverse datasets

and problem formulations.

2.3.1 Early Meta-Learning Approaches

The first wave of AutoClustering research explored the use of meta-learning to recommend clustering algorithms based on dataset characteristics. Pioneering works such as de Souto et al. (2008), Nascimento et al. (2009), and Soares et al. (2009) developed meta-databases linking dataset-level meta-features to the performance of candidate algorithms. Subsequent contributions by Ferrari and de Castro (2012) and Ferrari and de Castro (2015) formalized this process as an algorithm ranking problem, while Pimentel and de Carvalho (2018); Pimentel and de Carvalho (2019) proposed improved dataset characterization through statistical and distance-based meta-features. Together, these studies established a foundation for understanding how dataset properties influence algorithm performance, enabling early forms of automated algorithm recommendation for unsupervised learning.

2.3.2 Meta-Learning for Algorithm Selection and CASH

As the field matured, researchers sought to extend meta-learning beyond AS toward the broader CASH problem. Poulakis et al. (2020) introduced *AutoClust*, a meta-learning framework that constructs a meta-database combining dataset characteristics with CVI-based evaluations. The meta-learner recommends an algorithm for a new dataset, while Bayesian optimization refines its hyperparameters. Similarly, Liu et al. (2021) proposed *AutoCluster*, which incorporates a multi-objective optimization function and applies grid search for hyperparameter tuning. However, the reliance on exhaustive search limits scalability and exploration efficiency.

ElShawi et al. (2021) presented *cSmartML*, which leverages meta-learning not only to recommend algorithms but also to select the most suitable CVI according to dataset characteristics. The framework then employs evolutionary algorithms for HPO. Although this approach improves adaptivity, it also introduces inconsistency, as different datasets are optimized toward distinct objectives. Other contributions, such as Cohen-Shapira and Rokach (2021b), explored supervised graph embeddings for clustering algorithm selection, while Tschechlov et al. (2021) proposed *AutoML4Clust*, a general purpose AutoML framework integrating Bayesian optimization, random search, and Hyperband. Despite these advances, such frameworks typically require users to manually define the CVI to be

optimized, which contradicts AutoML's goal of minimizing human intervention.

2.3.3 Recent Trends

Recent efforts in AutoClustering have emphasized transparency, interpretability, and adaptability. Early approaches were primarily algorithm-centric, focusing on recommending clustering algorithms based on dataset similarity. Later frameworks incorporated hyperparameter optimization (HPO) and CASH to jointly select algorithms and their configurations. More recent methods leverage meta-learning, surrogate models, and adaptive evaluation strategies to accelerate optimization and improve generalization across diverse datasets.

These developments can be summarized in three broad methodological approaches, as shown in Table 2.1.

Table 2.1: General Approaches in AutoClustering

Approach	Focus	Optimization	Evaluation
Algorithm-centric	Recommend algorithms based on dataset similarity	Ranking / nearest-neighbor retrieval	Internal CVIs / offline benchmarking
Optimization-centric	Joint algorithm + hyperparameter selection	Bayesian / evolutionary / grid search	Fixed internal CVIs (SIL, CHS)
Meta-learning-centric	Use past knowledge to guide pipeline selection	Surrogate models, meta-feature prediction, learning-to-rank	Adaptive CVIs, dataset-dependent

Table 2.1 illustrates how AutoClustering has evolved from simple algorithm recommendation to complex, adaptive systems that integrate meta-knowledge, surrogate-based optimization, and flexible evaluation. This evolution reflects the shift from metric-centric, static designs toward more general, problem-oriented AutoML approaches.

2.4 Meta-learning for AutoClustering

Meta-learning, often referred to as "learning to learn" (Vanschoren, 2019), is a paradigm in which prior experience from solving multiple learning tasks is leveraged to improve performance on new, unseen ones (Vilalta and Drissi, 2002; Hospedales et al., 2022). In this context, the term *task* refers to a dataset along with its associated learning objective (e.g., classification, regression, or clustering), and should not be confused with AutoML tasks like AS or HPO, for instance. Rather than learning directly from raw data, meta-learning relies on *meta-data*, such as past performance evaluations, dataset characteristics, or model behaviours, to drive generalization across tasks (Brazdil et al., 2008).

At the core of this process is the *meta-learner*, which learns patterns in how pipelines perform across diverse tasks. This often involves training a *meta-model*, that maps properties of tasks (e.g., meta-features) to properties of pipelines (e.g., expected performance). These meta-models can be used to predict performance (Brazdil et al., 2008), rank candidate pipelines (Reif et al., 2012), or recommend pipeline configurations (Fusi et al., 2018), often in combination with different optimization or learning-to-rank strategies (Vanschoren, 2019).

In unsupervised learning, meta-learning can be more complex due to the absence of ground-truth labels and standard evaluation metrics like accuracy or recall. As a result, the meta-learner often rely on CVIs (e.g., Sil, DBS, CHS) (Treder-Tschechlov et al., 2023b; da Silva et al., 2024a) or task-specific heuristics (Bahri et al., 2022; Chan et al., 2024) as proxies for clustering quality. These proxies may be noisy or conflicting, introducing ambiguity into the meta-learning process. Nevertheless, the meta-learning framework remains largely the same from supervised tasks. Based on two main phases, the off line (learning phase) and the online (inference phase).

During the offline phase, a collection of historical datasets $\mathscr{D} = \{D_1, \ldots, D_n\}$ is processed to extract meta-features and evaluate clustering pipelines using internal or external validation measures. This results in a meta-dataset $\mathscr{D}_{\text{meta}}$, used to train a meta-model $\mathscr{M} : \mathscr{F} \mapsto \mathscr{Y}$, where $\mathscr{F} \in \mathbb{R}^d$ represents a vector of meta-features and $\mathscr{Y}$ denotes a clustering pipeline recommendation.

The choice of meta-model formulation, whether classification (Lemke et al., 2015; Treder-Tschechlov et al., 2023b), regression (da Silva et al., 2024c), or learning-to-rank (de Souto et al., 2008; Nascimento et al., 2009; Reif et al., 2014), depends on the nature of the available tasks. Although AutoClustering aims to work in unsupervised settings, many meta-learning approaches rely on labeled datasets during the offline phase to compute external metrics like ARI or NMI, which provide more reliable performance signals. These signals enable classification or regression models. When labels are unavailable and only internal CVIs can be used, the problem is more often cast as a ranking task, where the goal is to order clustering pipelines by their expected relative performance.

The two main modelling approaches commonly used during the offline phase, are:

(1) Ranking-Based Meta-Learning. This approach treats the meta-knowledge base as a repository of prior experience (Gabbay et al., 2021a). The core assumption is that

tasks with similar meta-features benefit from similar clustering pipelines. Given the meta-feature vector x_{new} of a new dataset, the system retrieves the most similar historical datasets $\{x_i\}$ using a similarity function $\text{sim}(\cdot,\cdot)$, and recommends the best-performing pipelines associated with them (De Souto et al., 2008; Pimentel and de Carvalho, 2018; Cohen-Shapira and Rokach, 2021b; Treder-Tschechlov et al., 2023b).

Let $\mathscr{P} = \{\pi_1, \pi_2, \ldots, \pi_n\}$ be the set of pipeline configurations evaluated during the offline phase, where each π_i denotes a clustering algorithm and its hyperparameters.

$$\pi_{\text{rec}} = \arg\max_{\pi_i \in \mathscr{P}} \text{sim}(x_{\text{new}}, x_i), \tag{2.19}$$

The top-K pipelines with the best historical performance among the most similar datasets are selected as:

$$\{\pi_{(1)}, \ldots, \pi_{(K)}\} = \text{TopK}_{\pi_i}\left(\text{perf}(\pi_i \mid \text{sim}(x_{\text{new}}, x_i))\right). \tag{2.20}$$

(2) Performance Prediction. An alternative is to train a meta-model $f_{\text{meta}} : \mathscr{X} \to \mathbb{R}$ that directly estimates the performance of each pipeline based on the meta-features of the dataset (Adam and Blockeel, 2015; Poulakis et al., 2020; da Silva et al., 2024c). Here, $\mathscr{X} \subseteq \mathbb{R}^d$ represents the meta-feature space. For a new dataset $\mathscr{D}_{\text{new}}$, with meta-features x_{new}, the model predicts the performance of each candidate pipeline π_i:

$$\hat{y}_i = f_{\text{meta}}(x_{\text{new}}, \pi_i), \tag{2.21}$$

and recommends the top-K pipelines with the highest predicted scores:

$$\{\pi_{(1)}, \ldots, \pi_{(K)}\} = \text{TopK}_{\pi_i}\left(\hat{y}_i\right). \tag{2.22}$$

In the online phase, a new dataset D_{new} is transformed into its meta-feature representation $\mathscr{F}_{\text{new}}$ and passed to the meta-model $\mathscr{M}$, which returns a clustering recommendation. This can support a variety of AutoML tasks, such as AS, HPO, CASH, or PS.

By leveraging prior knowledge, AutoClustering systems can avoid exhaustive searches and instead focus on the most promising regions of the pipeline space. This allows for efficient search and optimization using strategies such as Bayesian optimization, genetic algorithms (ElShawi and Sakr, 2022b; da Silva et al., 2024c), or regression-based ranking approaches (de Souto et al., 2008), improving both performance and search efficiency in

unsupervised settings.

Overall, the structure of AutoClustering systems underscores a key principle: virtually all decision-making operates in the meta-feature space. Whether through meta-model predictions or similarity-based retrieval, these systems depend entirely on descriptive summaries of the data, without access to raw instances or ground-truth cluster labels. Insight into meta-learning by application of Explainable AI (XAI) techniques enables fairer comparisons across systems, facilitates reproducibility, and can guide efficiency improvements by identifying and eliminating costly or redundant features that contribute little to the final decision.

Surrogate Models for Efficient Meta-Prediction. In many AutoClustering frameworks, evaluating candidate pipelines across multiple datasets can be computationally expensive, especially when internal or external validation measures require repeated clustering runs. To address this challenge, surrogate models are employed as part of the meta-learning workflow to approximate pipeline performance efficiently.

Surrogate modelling is a methodology used to approximate the behaviour of complex and computationally expensive models with simpler, more efficient alternatives Alizadeh et al. (2020). Acting as a stand-in for the original system, surrogate models enable faster evaluations and optimisation Eggensperger et al. (2018). This approach is widely adopted in engineering, optimisation, and machine learning, particularly in situations where repeated evaluations of a high-fidelity model would be computationally prohibitive Cozad et al. (2014); Bliek (2022); Han et al. (2012).

In the context of AutoML, surrogate models are commonly employed within Bayesian optimisation frameworks to estimate the performance of machine learning pipelines across various hyperparameter configurations Lindauer et al. (2022). Rather than training and evaluating the full model for every configuration, the surrogate predicts the expected performance, thereby accelerating the optimisation process Hutter et al. (2019). Techniques frequently used for surrogate modelling include Gaussian processes Gramacy (2020), Random Forests Dasari et al. (2019), and neural networks Hutter et al. (2019).

The effectiveness of a surrogate model depends on its ability to capture the underlying relationships between inputs and outputs of the system it approximates Jiang et al. (2020); Gramacy (2020). Typically, the model is trained on a set of input-output pairs, where inputs represent configurations (such as hyperparameters or algorithms) and outputs correspond to the associated performance metrics Tschechlov et al. (2021); Treder-

Tschechlov et al. (2023a). Once trained, the surrogate can predict the performance of new configurations, guiding the search for optimal solutions. A surrogate model can be formalised as follows:

Let $f : \mathscr{X} \to \mathscr{Y}$ represent the original, potentially expensive or complex function, where $\mathscr{X} \subseteq \mathbb{R}^d$ is the input space and $\mathscr{Y} \subseteq \mathbb{R}$ is the output space. The surrogate model $\hat{f}(x)$, where $\hat{f} : \mathscr{X} \to \mathscr{Y}$, is an approximation of $f(x)$ that is computationally cheaper to evaluate.

To construct $\hat{f}$, one has to train it on a dataset $\mathscr{D} = \{(x_i, y_i)\}_{i=1}^n$, where $x_i \in \mathscr{X}$ are input samples and $y_i = f(x_i) \in \mathscr{Y}$ are the corresponding outputs (evaluations of f at these points). The surrogate model is obtained by minimizing the error between the true function $f(x)$ and the approximation $\hat{f}(x)$:

$$\hat{f} = \arg\min_{\hat{f} \in \mathscr{H}} \sum_{i=1}^{n} \ell(f(x_i), \hat{f}(x_i))$$

where $\mathscr{H}$ denotes the hypothesis space of possible surrogate models (such as linear models, neural networks, or Gaussian processes), and $\ell(f(x_i), \hat{f}(x_i))$ is a loss function that measures the error between $f(x_i)$ and $\hat{f}(x_i)$, for example, the mean squared error $\ell(y, \hat{y}) = (y - \hat{y})^2$.

In optimisation contexts, the surrogate model $\hat{f}(x)$ is often used to approximate an objective function. For a maximization problem:

$$x^* = \arg\max_{x \in \mathscr{X}} f(x),$$

the surrogate-based optimisation problem becomes:

$$x^*_{\text{surrogate}} = \arg\max_{x \in \mathscr{X}} \hat{f}(x).$$

2.5 Discussion and Open Challenges

Despite considerable progress, current AutoClustering frameworks remain largely *metric-centric* and *static* in design. They are constrained by fixed algorithm portfolios, predefined hyperparameter spaces, and rigid sets of evaluation metrics. These design choices limit their ability to adapt to diverse problem domains or user-defined objectives. While meta-learning has improved generalization and reduced search costs, its integra-

tion into more flexible, problem-oriented AutoML systems remains incomplete.

A fundamental methodological gap persists between PS and *meta-learning*. Most existing approaches either (i) treat AutoClustering as a direct optimization problem guided by fixed CVIs, or (ii) use meta-learning primarily for algorithm recommendation or initialization. Yet these paradigms are complementary rather than mutually exclusive: search-based pipeline synthesis enables flexible exploration of high-dimensional design spaces, while meta-learning offers inductive guidance based on accumulated experience. Bridging the two requires mechanisms that enable the optimization process to be informed by meta-knowledge while remaining responsive to new problem contexts.

This integration calls for surrogate models capable of approximating clustering performance as a function of meta-features and hyperparameter configurations. Such models can guide exploration without the need for exhaustive evaluation, enabling adaptive and data-driven search. Equally important is the concept of *meta-objectives*, functions that encode user intent or task-specific notions of clustering quality. By learning to approximate these objectives, AutoClustering systems can move beyond fixed metrics and optimize toward problem-oriented criteria.

The over reliance on predefined CVIs represents another critical limitation. By optimizing a fixed set of indices, existing frameworks implicitly assume a universal definition of clustering "goodness," overlooking the subjectivity inherent in unsupervised learning (von Luxburg et al., 2012; Hennig, 2015; Van Mechelen et al., 2023). In practice, clustering goals are context-dependent: interpretability may matter more in exploratory analysis, whereas compactness or density separation may dominate in anomaly detection. A flexible, problem-oriented AutoML framework must therefore support adaptive optimization guided by the user's intent and the nature of the data.

These observations motivate the development of the PoAC framework introduced in this thesis. PoAC unifies Ps and meta-learning under a surrogate-driven optimization scheme, allowing clustering automation to be guided by learned meta-objectives rather than static CVIs. By dynamically aligning meta-features, optimization strategies, and user-defined goals, PoAC advances AutoClustering toward a new paradigm of context-aware, adaptive unsupervised learning. The following chapters detail its architecture, training procedures, and empirical validation across diverse clustering and anomaly detection tasks.

Chapter 3

AutoML for Clustering: Pipeline Synthesis and Meta-Objectives

Contents

This chapter presents TPOT-Clustering, a framework for the automated synthesis of clustering pipelines in the unsupervised domain. TPOT-Clustering extends the TPOT AutoML system by incorporating two complementary optimisation strategies:

- **CVI-based optimisation**, which leverages standard internal Cluster Validity Indices (e.g., Silhouette, Davies–Bouldin) to guide pipeline synthesis when user objectives align with conventional notions of cluster quality such as compactness or separation; and
- **Surrogate-based optimisation**, which employs a meta-learned regression model that predicts clustering performance based on dataset descriptors (meta-features),

internal CVIs, and optional external indicators, allowing the framework to approximate more complex, domain-specific user goals, including historical partitioning patterns.

This dual approach enables TPOT-Clustering to flexibly accommodate both straightforward and sophisticated analytical objectives. The chapter details the architecture, meta-learning methodology, and evolutionary optimisation mechanisms underpinning the framework.

The work presented in this chapter was published in the *International Journal of Neural Networks*, 2025, demonstrating its contributions to the field of unsupervised AutoML.

Motivation for Using TPOT as the AutoML Engine. While numerous commercial and open-source AutoML systems exist, most are primarily designed for supervised learning and rely on fixed, task-specific objective functions. The objectives of this work require an AutoML engine capable of end-to-end *pipeline synthesis* in the unsupervised setting and, critically, the ability to incorporate *custom optimisation criteria*, including meta-learned surrogate models. TPOT was selected because its genetic programming formulation represents workflows as compositional pipelines and explicitly decouples the optimisation procedure from the fitness function, allowing arbitrary objectives to guide the search. This flexibility is essential for PoAC, where clustering quality cannot be expressed solely through standard internal CVIs but must reflect problem-oriented and transferable goals. In contrast, alternative AutoML frameworks typically provide limited control over pipeline structure and tightly bind optimisation to predefined metrics, making them unsuitable for surrogate-guided AutoClustering. TPOT therefore provides the architectural flexibility necessary to support the proposed framework.

3.1 Optimisation for Pipeline Synthesis

The synthesis of clustering pipelines can be formulated as a complex optimization problem over a structured and high-dimensional search space. Each candidate pipeline represents a combination of data preprocessing, feature extraction, clustering algorithms, and their associated hyperparameters. The combinatorial nature of this space requires optimization strategies capable of efficiently exploring diverse configurations while balancing exploration and exploitation under limited computational budgets (Hutter et al., 2019; Olson et al., 2016).

Foundational AutoML systems such as Auto-WEKA (Thornton et al., 2013) and TPOT (Olson and Moore, 2016) have employed a variety of optimization strategies for supervised pipeline synthesis. *Grid search* exhaustively evaluates all hyperparameter combinations, ensuring comprehensive coverage but with exponential computational cost. *Random search* samples configurations stochastically and can achieve comparable or superior results with fewer evaluations (Bergstra and Bengio, 2012). More sophisticated strategies, such as *Bayesian optimization*, iteratively propose promising configurations based on probabilistic surrogate models. Evolutionary algorithms, as adopted by TPOT, represent pipelines as genetic programs that evolve through mutation, crossover, and selection, which is particularly suited for hierarchical and combinatorial search spaces.

In the context of clustering, frameworks such as AutoML4Clust (Tschechlov et al., 2021) and AutoClust (Cassier et al., 2022) have adapted these optimization paradigms to unsupervised learning. Without ground-truth labels, AutoClustering systems commonly rely on internal CVIs to guide optimization (Bahri et al., 2022; Tschechlov et al., 2021). However, CVIs capture only partial aspects of cluster quality, such as compactness, separation, or density, and may produce divergent evaluations across datasets or structures (Rokach and Maimon, 2005; von Luxburg et al., 2012). Their effectiveness depends on how closely the assumptions encoded by a CVI align with the user's analytical intent.

When the user's notion of a "good" partition coincides with the properties emphasized by a particular CVI, optimization can be effective. Conversely, if objectives prioritize interpretability, pattern discovery, or robustness to noise, reliance on a single CVI may be misleading. For example, pipelines optimized for compactness-based indices (e.g., SIL or DBS) may penalize algorithms capable of uncovering complex or overlapping structures (Dudek, 2019), whereas separation-based CVIs may exaggerate artificial boundaries or overlook small but semantically meaningful clusters (Li et al., 2016; Dudek, 2019).

To address these challenges, this work introduces **TPOT-Clustering**, an extension of the TPOT framework (Olson and Moore, 2016) tailored for unsupervised learning. TPOT-Clustering adapts TPOT's evolutionary optimization engine to clustering, enabling automatic pipeline synthesis under diverse objectives. The framework supports both conventional CVI-based criteria and advanced *surrogate-based* objectives, allowing custom evaluation functions that capture domain-specific notions of clustering quality.

As illustrated in Figure 3.1, TPOT-Clustering retains TPOT's modular architecture, supporting flexible compositions of preprocessing, feature extraction, and clustering al-

gorithms. Practitioners can instantiate the optimization process with single CVIs for targeted goals or surrogate models for more complex criteria, enhancing adaptability across visualization, anomaly detection, and exploratory pattern discovery tasks.

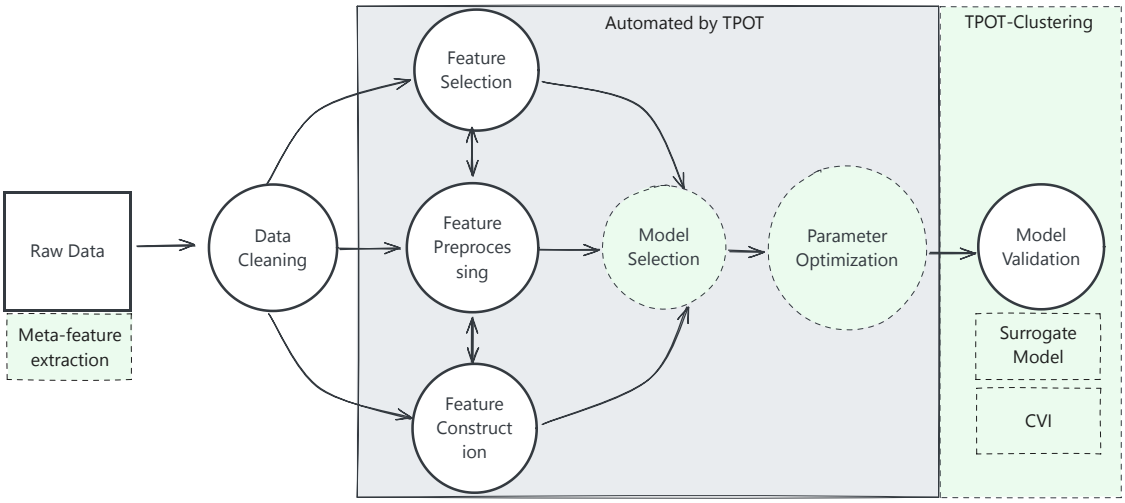


Figure 3.1: Overview of the proposed TPOT-Clustering framework, extending TPOT for unsupervised pipeline synthesis with surrogate-based optimization. Source: Author, reproduced from da Silva et al. (2025).

Through this design, TPOT-Clustering extends evolutionary pipeline optimization to the unsupervised domain, enabling automatic construction of workflows ranging from simple algorithmic configurations to complex multi-stage pipelines integrating diverse preprocessing and feature extraction procedures.

3.2 TPOT-Clustering

TPOT-Clustering comprises two main modules: the *Meta-Learning Module*, which constructs a surrogate model to approximate clustering performance, and the *Optimisation Module*, which integrates either internal CVIs or the surrogate model into an evolutionary search process for pipeline synthesis. This section provides an overview of both modules, as illustrated in Figure 3.2.

3.2.1 Meta-Learning Module

The Meta-Learning Module implements a surrogate modeling approach to approximate clustering performance, providing an inexpensive proxy for evaluating candidate pipelines. Both the dataset descriptors (meta-features) and the evaluation criteria (internal CVIs) are **unsupervised**, meaning they are computed solely from the data without

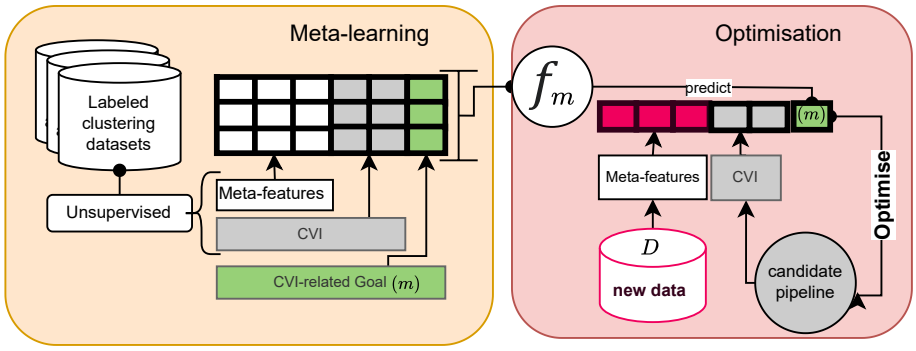


Figure 3.2: Overview of the TPOT-Clustering framework. Source: Author, reproduced from da Silva et al. (2025).

access to ground-truth labels. This allows the surrogate model to predict pipeline quality even when no labeled data is available, enabling the optimisation process to focus on promising configurations without relying on expensive full evaluations.

The module operates through four main stages: *Problem Space Definition*, *Clustering Simulations*, *Feature Extraction*, and *Surrogate Modeling*. Together, these stages produce a comprehensive meta-database and a trained model that captures the relationship between **unsupervised dataset descriptors** and clustering quality.

Problem Space Definition

The first stage defines the problem space from which the meta-database is constructed. Following the approach proposed by Zellinger and Bühlmann (2023), we use the *Repliclust*[1] library to generate a diverse collection of synthetic datasets. By varying structural hyperparameters such as the number of clusters, cluster shapes, densities, and degrees of overlap, we obtain a representative sample of clustering scenarios. This systematic exploration ensures broad coverage of possible dataset archetypes and provides a rich basis for meta-learning.

Clustering Simulations

In the second stage, multiple clustering solutions are generated for each synthetic dataset to simulate a spectrum of partitioning qualities. Gaussian perturbations are progressively introduced into the cluster labels to create partitions of varying fidelity. For each clustering solution, the system computes a set of landmark meta-features—specifically, the

[1] *Repliclust*: A Python package for generating synthetic clustering datasets based on geometric archetypes.

internal CVIs SIL, DBS, and CHS, as well as the external ARI, which serves as a supervised ground truth reference. This process yields a broad distribution of clustering outcomes that reflects how changes in cluster quality manifest in the associated CVI and meta-feature space.

Feature Extraction

The third stage focuses on extracting descriptive meta-features from the datasets and their clustering solutions. These features summarize statistical, structural, and geometric aspects of the data, such as dimensionality, compactness, and density distribution. The default feature set follows the unsupervised meta-feature taxonomy proposed by Alcobaça et al. (2020), though the framework also supports user-defined feature sets. The extracted features are combined with the computed CVIs and ARI scores to form a meta-database capturing both dataset characteristics and clustering performance patterns.

Surrogate Modeling

The final stage involves training a regression model on the constructed meta-database to approximate clustering performance for unseen datasets. By default, TPOT-Clustering employs a Random Forest regressor (Pedregosa et al., 2011) due to its robustness and interpretability, although alternative models can be specified by the user. The trained regressor learns the mapping between meta-features, internal CVIs, and a target external performance indicator (e.g., ARI), enabling it to act as a surrogate objective function during optimisation. Once trained, the surrogate model can predict expected clustering quality.

3.2.2 Optimisation Module

Building upon the meta-learning foundation, the optimisation module adapts TPOT's evolutionary search strategy to the clustering domain. This involved reconfiguring TPOT's internal search space to include clustering algorithms and their corresponding hyperparameters, replacing the original supervised-learning operators. The resulting configuration space—summarized in Table 3.1—enables exploration of pipelines that combine diverse preprocessing steps and clustering algorithms such as K-Means, DBSCAN, Agglomerative Clustering, MiniBatch K-Means, and Spectral Clustering.

The optimisation process can operate in two modes:

Table 3.1: List of clustering algorithms and their hyperparameters.

Algorithm	Hyperparameters
Agglomerative Clustering	**n_clusters**: [2, 23]
DBSCAN	**eps**: [2, 23]; **min_samples**: [1e-3, 1e-2, 1e-1, 1, 10, 100]; **leaf_size**: [3–50]
K-Means	**n_clusters**: [2, 23]; **init**: [k-means++, random]
MiniBatch K-Means	**n_clusters**: [2, 23]; **batch_size**: [2–23]
Spectral Clustering	**n_clusters**: [2, 23]; **eigen_solver**: [arpack, lobpcg, amg]; **affinity**: [nearest_neighbors, rbf, precomputed]

(i) **CVI-based optimisation**, in which standard internal indices (e.g., SIL, DBS) serve as the objective function;

(ii) **Surrogate-based optimisation**, where the surrogate model trained by the meta-learning module predicts clustering quality based on meta-features and CVIs.

The latter mode enables efficient optimisation by approximating clustering performance without fitting every candidate pipeline, significantly reducing computational cost.

To support surrogate-based optimisation, we developed a *Surrogate Scorer*, described in Algorithm 1. This component uses the trained surrogate model to predict expected clustering performance given the extracted meta-features and computed CVIs of a pipeline. The scorer integrates seamlessly into TPOT's evolutionary loop, replacing the standard scoring function and allowing selection pressure to be driven by surrogate predictions.

Algorithm 1: SurrogateScorer

```
Input: model: Trained surrogate model; meta_features: List of meta-features; cvi: List of
       clustering validity indices
Function SurrogateScorer(pipeline, X):
    cluster_labels = pipeline.fit_predict(X)
    foreach score in cvi do
        score_value = score(X, cluster_labels)
        meta_features.append(score_value)
    surrogate_score = model.predict(meta_features)
    if len(unique(cluster_labels)) > 1 then
        return surrogate_score
    else
        return −∞ // Invalid clustering
```

Finally, TPOT-Clustering outputs the synthesized pipelines as executable Python scripts, ensuring reproducibility and ease of integration into downstream workflows. This design preserves TPOT's philosophy of transparent AutoML while extending its applicability to unsupervised learning tasks, enabling practitioners to deploy optimized clustering workflows with minimal manual intervention.

3.3 Experimental Setup

This section details the experimental setup used to evaluate the proposed *TPOT-Clustering* framework[2]. The primary goal of the experiment is to compare how different optimisation strategies leverage CVIs and meta-features during clustering pipeline synthesis. Both elements play complementary roles in AutoML for clustering: while CVIs evaluate the intrinsic quality of cluster partitions, meta-features characterise the structural properties of datasets, helping to guide optimisation toward pipelines that align with the user's analytical objectives.

3.3.1 Baseline Frameworks

Two recent frameworks, *TPE-AutoClust* and *ML2DAC*, were selected as meta-learning-driven baselines for comparison. Both represent state-of-the-art approaches to unsupervised pipeline optimisation:

- **TPE-AutoClust** employs meta-learning to *warm-start* its optimisation process and utilises a multi-objective strategy balancing pipeline length and clustering performance. It integrates multiple CVIs through an ensemble and Pareto-based optimisation procedure, producing robust clustering solutions that trade off different quality criteria.
- **ML2DAC** applies meta-learning to select the most appropriate CVI for each dataset, thereby narrowing the search space and improving the efficiency of the optimisation process. It further identifies effective clustering algorithms and hyperparameter configurations tailored to dataset characteristics.

3.3.2 Evaluated TPOT-Clustering Variants

Within the *TPOT-Clustering* framework, we evaluated several optimisation modes to assess the influence of different objective functions. These include single-CVI-based optimisation strategies and a surrogate-guided variant that leverages meta-learning:

(i) **CVI-based optimisation:** synthesises pipelines with the goal of directly optimising a given internal index, namely, SIL, DBS, or CHS, resulting in three variants:

[2]The complete implementation and experiment scripts are publicly available at TPOT-Clustering GitHub repository.

TPOT-SIL, TPOT-DBS, and TPOT-CHS.

(ii) **Surrogate-based optimisation:** employs a surrogate model trained on meta-features and CVIs to predict clustering quality. This model serves as the optimisation objective, guiding the evolutionary process without the need to compute CVIs directly for each candidate pipeline.

For comparison, we also included *TPE-AutoClust* and *ML2DAC* under equivalent computational constraints. A summary of the evaluated strategies and their characteristics is provided in Table 3.2.

Table 3.2: Summary of optimisation strategies, their use of CVIs, and meta-learning components.

Strategy Name	**Use of CVIs**	**Use of Meta-Learning**
TPOT-SIL	Synthesises pipelines to maximise the SIL.	None.
TPOT-DBS	Synthesises pipelines to minimise the DBS.	None.
TPOT-CHS	Synthesises pipelines to maximise the CHS.	None.
TPOT-Clustering (Surrogate Model)	Uses a surrogate model trained on SIL, DBS, and CHS to guide optimisation.	Employs meta-learning to train surrogate models based on meta-features and CVIs.
TPE-AutoClust	Combines SIL, DBS, and CHS into a multi-objective optimisation using Pareto fronts; final results obtained via ensemble aggregation.	Uses meta-learning to warm-start the search with promising pipeline candidates.
ML2DAC	Selects the most suitable CVI per dataset and tunes clustering algorithms accordingly.	Uses meta-learning to choose CVIs, reduce the search space, and guide hyperparameter selection.

3.3.3 Evaluation Protocol

While TPE-AutoClust and ML2DAC typically generate ranked lists of pipelines through their warm-start mechanisms, our focus is on isolating the effect of the optimisation objective. To ensure fairness, warm-starting was disabled in all comparative runs, and only the best-performing pipeline from each approach was retained for evaluation.

Each optimisation run was assigned a fixed computational budget of 30 minutes. Following the design of prior studies (Tschechlov et al., 2021; Bahri et al., 2022), the external evaluation metric used for comparison was the ARI, which measures the similarity

between the predicted cluster assignments and the known ground-truth labels. ARI is particularly suitable for benchmarking as it provides a consistent, label-independent measure of clustering agreement, ranging from -1 (complete disagreement) to 1 (perfect alignment).

3.3.4 Datasets

The experimental evaluation was conducted using 25 datasets drawn from the *Clustering Benchmark Repository*[3], encompassing both real-world datasets (primarily from the UCI repository (Asuncion et al., 2007)) and synthetic datasets with diverse structural properties. These datasets cover a broad spectrum of clustering challenges, including variations in cluster number, dimensionality, density, and overlap, ensuring that the evaluated optimisation strategies are tested under heterogeneous conditions.

Table 3.3 summarises the datasets used in the evaluation, including the number of clusters (k), instances (N), and dimensions (d).

3.4 Results and Discussion

This section presents the empirical evaluation of the proposed *TPOT-Clustering* framework, comparing it against baseline AutoML approaches across a diverse collection of real-world and synthetic datasets. The analysis focuses on three aspects: (i) the overall clustering performance in terms of ARI, (ii) the structural characteristics of the best-performing pipelines, and (iii) the statistical significance of the observed differences.

3.4.1 Overall Performance Across Datasets

The performance of all optimisation strategies was evaluated using ARI, which quantifies the similarity between the predicted cluster labels and the ground-truth partitions. Table 3.4 reports the ARI values for each method and dataset, with the number of steps in the corresponding best pipeline indicated in parentheses. The bold values highlight the best performance per dataset.

Overall, the results demonstrate clear performance differences between the optimisation strategies. On average, **TPOT-Clustering** achieves the highest ARI score (0.33), outperforming both baseline frameworks and single-CVI variants. This result supports

[3] Clustering Benchmark Repository

Table 3.3: Datasets used in the evaluation, including number of clusters (k), instances (N), and dimensions (d).

No.	Name	k	N	d
Real-world Datasets (UCI)				
1	Arrhythmia	13	452	279
2	Balance-scale	3	625	4
3	Ecoli	8	336	7
4	Glass	6	214	9
5	Ionosphere (Iono)	2	351	34
6	Iris	3	150	4
7	Segment	7	2310	19
8	Sonar	2	608	60
9	TAE	3	151	5
10	Thyroid (Thy)	3	215	5
11	Wine	3	178	13
Synthetic Datasets				
12	3-Spiral	3	312	2
13	Cassini	3	1000	2
14	Cluto-t7-10k	9	9208	2
15	Compound	6	399	2
16	Disk-6000n	2	6000	2
17	Elliptical_10_2	10	500	10
18	Engytime	2	4096	2
19	Flame	3	240	2
20	Fourty	39	1000	2
21	Jain	2	373	2
22	Pathbased	3	300	2
23	Sizes2	15	1500	2
24	Sizes4	15	1500	2
25	Twodiamonds	2	800	2

Table 3.4: Optimisation Strategies for Clustering: ARI Scores Across Benchmark Datasets with Number of Steps in the Pipeline Indicated in Parentheses and Best Performances Highlighted

No.	Dataset	TPOT-CHS	TPOT-DBS	TPOT-SIL	TPOT-Clustering	TPE-Autoclust	ML2DAC
1	Arrhythmia	0.01$^{(2)}$	0.01$^{(2)}$	0.04$^{(2)}$	**0.04**$^{(2)}$	0.00$^{(0)}$	0.03$^{(1)}$
2	Balance-scale	0.08$^{(2)}$	0.10$^{(2)}$	0.10$^{(1)}$	**0.13**$^{(2)}$	0.07$^{(0)}$	0.04$^{(1)}$
3	Ecoli	**0.39**$^{(2)}$	0.22$^{(2)}$	0.00$^{(4)}$	0.04$^{(2)}$	0.13$^{(0)}$	0.15$^{(1)}$
4	Glass	0.13$^{(2)}$	0.18$^{(3)}$	**0.25**$^{(2)}$	0.21$^{(2)}$	0.07$^{(0)}$	0.13$^{(1)}$
5	Iono	0.18$^{(2)}$	**0.21**$^{(2)}$	0.00$^{(3)}$	0.01$^{(2)}$	0.02$^{(0)}$	0.08$^{(1)}$
6	Iris	0.20$^{(1)}$	0.19$^{(2)}$	0.57$^{(1)}$	**0.73**$^{(2)}$	0.13$^{(0)}$	0.09$^{(1)}$
7	Segment	0.48$^{(1)}$	0.10$^{(1)}$	0.10$^{(1)}$	**0.34**$^{(2)}$	0.10$^{(0)}$	0.10$^{(1)}$
8	Sonar	0.00$^{(2)}$	0.04$^{(1)}$	**0.07**$^{(2)}$	**0.07**$^{(2)}$	0.02$^{(0)}$	0.03$^{(1)}$
9	Tae	0.03$^{(2)}$	0.03$^{(1)}$	**0.04**$^{(2)}$	**0.04**$^{(2)}$	0.02$^{(0)}$	0.01$^{(1)}$
10	Thy	0.14$^{(1)}$	0.13$^{(2)}$	**0.34**$^{(2)}$	**0.34**$^{(2)}$	0.04$^{(0)}$	0.00$^{(1)}$
11	Wine	0.37$^{(2)}$	0.21$^{(2)}$	0.46$^{(2)}$	0.46$^{(2)}$	0.13$^{(0)}$	**0.79**$^{(1)}$
12	3-spiral	**0.12**$^{(1)}$	0.11$^{(1)}$	-0.01$^{(1)}$	0.00$^{(2)}$	0.03$^{(0)}$	0.08$^{(1)}$
13	Cassini	0.64$^{(1)}$	**0.68**$^{(1)}$	0.65$^{(1)}$	0.53$^{(1)}$	0.64$^{(0)}$	0.31$^{(1)}$
14	Cluto-t7-10k	0.28$^{(1)}$	0.41$^{(1)}$	0.41$^{(2)}$	**0.57**$^{(1)}$	0.24$^{(0)}$	0.18$^{(1)}$
15	Compound	**0.52**$^{(1)}$	0.48$^{(1)}$	0.47$^{(1)}$	0.47$^{(1)}$	0.40$^{(0)}$	0.21$^{(1)}$
16	Disk-6000n	0.08$^{(1)}$	0.00$^{(2)}$	0.00$^{(1)}$	**0.09**$^{(2)}$	0.08$^{(0)}$	-0.01$^{(1)}$
17	Elliptical_10_2	**0.18**$^{(1)}$	**0.18**$^{(1)}$	**0.18**$^{(1)}$	**0.18**$^{(1)}$	0.13$^{(0)}$	0.00$^{(1)}$
18	Engytime	**0.56**$^{(1)}$	0.00$^{(1)}$	0.00$^{(1)}$	**0.56**$^{(2)}$	0.33$^{(0)}$	0.03$^{(1)}$
19	Flame	0.43$^{(1)}$	0.10$^{(2)}$	0.45$^{(1)}$	**0.46**$^{(1)}$	0.07$^{(0)}$	0.14$^{(1)}$
20	Fourty	0.27$^{(1)}$	0.25$^{(1)}$	0.25$^{(1)}$	**0.38**$^{(1)}$	0.24$^{(0)}$	0.00$^{(1)}$
21	Jain	0.07$^{(1)}$	0.14$^{(1)}$	0.55$^{(2)}$	**0.58**$^{(1)}$	0.05$^{(0)}$	0.57$^{(1)}$
22	Pathbased	0.20$^{(1)}$	0.18$^{(2)}$	0.40$^{(2)}$	**0.46**$^{(1)}$	0.12$^{(0)}$	**0.46**$^{(1)}$
23	Sizes2	0.60$^{(1)}$	0.60$^{(3)}$	0.60$^{(2)}$	0.88$^{(2)}$	0.45$^{(0)}$	**0.93**$^{(1)}$
24	Sizes4	**0.90**$^{(1)}$	0.11$^{(2)}$	0.41$^{(1)}$	0.33$^{(1)}$	0.06$^{(0)}$	0.75$^{(1)}$
25	Twodiamonds	0.10$^{(1)}$	0.09$^{(1)}$	**1.00**$^{(1)}$	**1.00**$^{(1)}$	0.09$^{(0)}$	0.00$^{(1)}$
Average	–	0.27	0.19	0.29	**0.33**	0.16	0.20

the hypothesis that surrogate-based optimisation can better generalise across heterogeneous datasets than approaches relying on a single internal metric.

More specifically, TPOT-Clustering consistently performs well on datasets characterised by complex cluster structures and moderate-to-high dimensionality, such as *Cluto-t7-10k*, *Cassini*, and *Compound*. This robustness stems from the framework's ability to explore composite pipeline configurations that combine preprocessing, feature transformation, and clustering algorithms. In contrast, single-objective CVI-based variants (TPOT-CHS, TPOT-DBS, TPOT-SIL) show higher variability, often excelling only when the dataset's structure aligns with the assumptions of their respective indices.

3.4.2 Performance Trends and Dataset Characteristics

The results reveal interesting trends when relating dataset properties to pipeline performance:

Effect of dimensionality. Datasets with high dimensionality, such as *Ionosphere* (34 features) and *Sonar* (60 features), exhibit reduced ARI scores across all frameworks.

This highlights the challenge of clustering in high-dimensional spaces, where irrelevant or redundant features can obscure meaningful structures. Notably, in these cases, TPOT-generated pipelines that incorporated dimensionality reduction or feature scaling, such as *PCA*, *MinMaxScaler*, or *VarianceThreshold*, tended to outperform simpler, single-step pipelines.

Beyond dimensionality, another critical factor influencing surrogate reliability is the geometric nature of the data manifold.

Limitations under Non-Linear Structures. While the surrogate model exhibited consistent predictive behavior across most datasets, certain cases revealed its limitations under distributional shift. A notable example is the *3-Spiral* benchmark dataset, characterized by intertwined spiral-shaped clusters that differ substantially from the synthetic archetypes used to train the surrogate model—primarily convex and moderately overlapping geometries. Consequently, the surrogate exhibited poor generalization in this setting.

The pipeline predicted by the surrogate as optimal, `KMeans(input_matrix, init=random, n_clusters=4)`, received a predicted score of 0.97516, suggesting excellent expected performance. However, its actual clustering quality was near-random (*ARI* = 0.0), with internal CVIs of *SIL* = 0.354, *DBS* = 0.880, and *CHS* = 245.54, as shown in Figure 3.3. In contrast, a pipeline incorporating manifold learning, which better captures non-linear structures, achieved substantially higher agreement with the ground truth (*ARI* = 0.663) despite receiving a lower surrogate-predicted score of 0.478 and modest internal CVIs (*SIL* = 0.023, *DBS* = 2.756, *CHS* = 26.87), as shown in Figure 3.4.

This observation underscores a key limitation of surrogate-based optimisation: the surrogate's predictive accuracy depends critically on the representativeness of its training meta-database. When the encountered data distribution deviates substantially from those seen during training, surrogate predictions may become unreliable, leading to suboptimal pipeline synthesis. This reflects a broader trade-off between *generalisability* and *specialisation* in meta-learning, highlighting the need for domain-aware meta-bases or adaptive retraining mechanisms when addressing atypical data geometries.

Role of preprocessing. Among the 25 datasets, 13 of the best-performing pipelines included at least one preprocessing step, confirming that appropriate feature transformations can significantly improve clustering quality. Preprocessing techniques such as

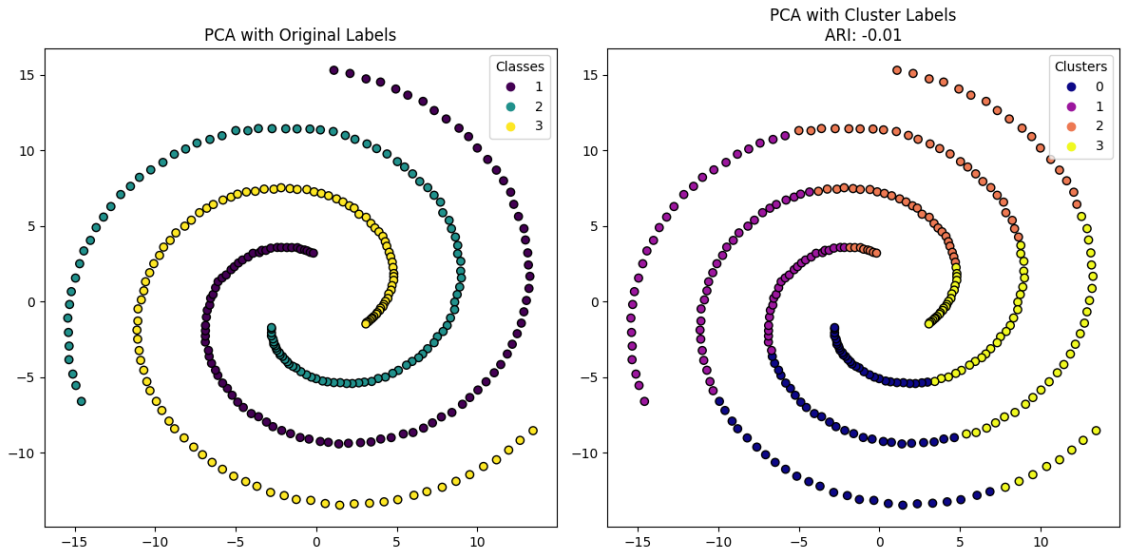


Figure 3.3: Clustering result of the surrogate-predicted best pipeline (`KMeans`, $n_{clusters} = 4$) on the 3-Spiral dataset. The pipeline achieved high predicted performance (0.975) but near-random clustering quality (*ARI*=-0.01).

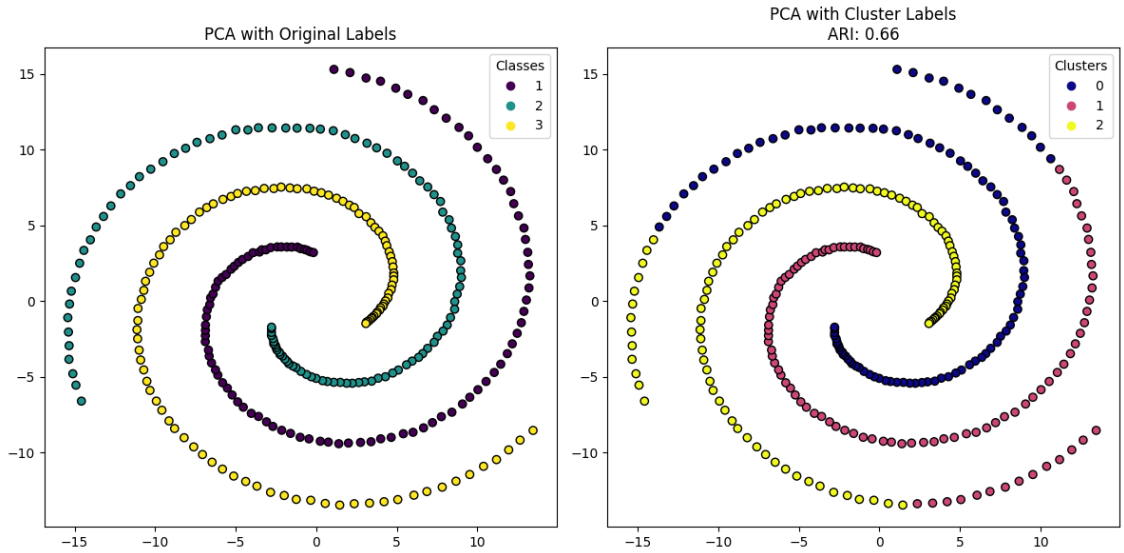


Figure 3.4: Clustering result of a manifold-based pipeline on the 3-Spiral dataset. Despite a lower surrogate-predicted score (0.478), this configuration achieved substantially better alignment with the ground truth (*ARI*=0.663).

StandardScaler and *FeatureUnion* proved particularly beneficial in real-world datasets, where noise, scale disparities, and feature correlations are prevalent.

Pipeline complexity. A comparison between single-step and multi-step pipelines suggests that real-world datasets generally benefit from more complex workflows. For instance, the *Ecoli* dataset achieved an ARI of 0.39 using a two-step pipeline (TPOT-CHS), while *Segment* reached 0.34 with a similar configuration in TPOT-Clustering. Conversely, for synthetic datasets such as *Twodiamonds* and *Sizes2*, simple one-step pipelines

achieved near-perfect results (ARI 1.00 and 0.93, respectively). This indicates that well-structured synthetic data often already conforms to algorithmic assumptions, reducing the need for preprocessing.

3.4.3 Comparative Statistical Analysis

To assess the significance of performance differences, we conducted a Friedman test ($p < 0.05$), followed by a Nemenyi post-hoc analysis. The average ranks of the compared frameworks and the corresponding critical difference (CD = 1.50) are depicted in Figure 3.5.

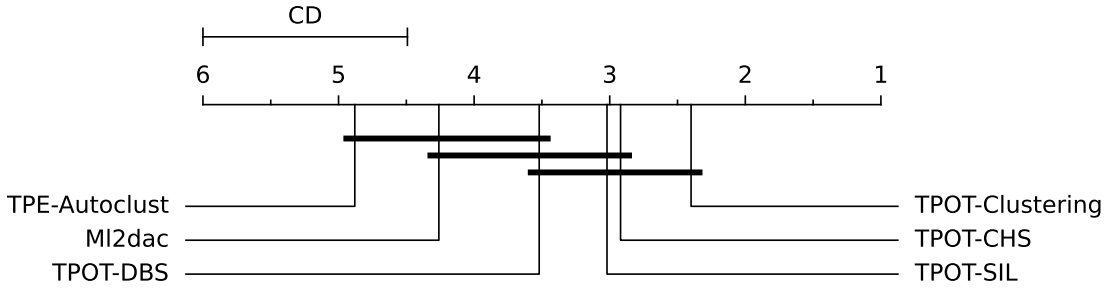


Figure 3.5: Average ranks of clustering optimisation frameworks with corresponding critical difference (CD = 1.50). Lower ranks indicate better performance. Source: Author, reproduced from da Silva et al. (2024c).

The results confirm that **TPOT-Clustering** ranks significantly higher than TPE-AutoClust, ML2DAC, and TPOT-DBS, indicating a consistent improvement in clustering performance. TPOT-SIL and TPOT-CHS obtain comparable ranks to TPOT-Clustering, suggesting that in some cases, single-CVI objectives can perform competitively—particularly when the dataset structure aligns with the CVI's assumptions. Nevertheless, the surrogate-based optimisation in TPOT-Clustering provides a more stable and generalisable performance across diverse datasets.

3.5 Discussion and Limitations

The experimental results provide valuable insights into the behaviour of different AutoML strategies for unsupervised clustering and highlight the limitations of traditional optimisation objectives. Internal CVIs such as the SIL, CHS, and DBS scores capture distinct structural aspects of clustering, compactness, separation, or density, but only partially reflect what constitutes a "good" clustering solution (Rokach and Maimon, 2005;

von Luxburg et al., 2012). As a result, single-CVI optimisation often produces pipelines that perform well according to a specific index but fail to align with user intent or domain-specific notions of cluster quality. For example, compactness-oriented CVIs tend to penalise elongated or overlapping clusters, while separation-based CVIs can overemphasise artificial boundaries in continuous data distributions.

The surrogate-based approach in **TPOT-Clustering** mitigates these limitations by integrating multiple CVIs and meta-features into a unified optimisation signal. Rather than relying on a single structural assumption, it learns a mapping between dataset characteristics and expected clustering outcomes. This allows the framework to approximate higher-level notions of quality, such as robustness, interpretability, or visual separability, that are often more meaningful to practitioners. Consequently, **TPOT-Clustering** achieves more stable and generalisable performance across datasets of varying complexity, outperforming or matching competing frameworks like **TPE-AutoClust** and **ML2DAC**.

Comparatively, **ML2DAC** and **TPE-AutoClust** excel in narrower contexts, such as low-dimensional or well-separated datasets, but their limited integration of preprocessing or reliance on static CVI objectives restricts adaptability. In contrast, TPOT-Clustering's pipeline-level synthesis allows the inclusion of preprocessing and feature-selection stages, which substantially improve results on noisy or high-dimensional data. This effect is especially evident in real-world datasets, where multi-step pipelines often outperform single-step configurations, while synthetic datasets, typically well-structured, linear and noise-free, achieve near-optimal results without additional transformations.

Despite these advantages, the current approach also reveals key limitations of surrogate-guided optimisation. The surrogate model's effectiveness depends strongly on the representativeness of its underlying meta-database: broad models may generalise across diverse datasets but lack precision for specific ones, whereas specialised models can yield higher accuracy but reduced transferability. Moreover, the surrogate objective remains a learned proxy for clustering quality, it can capture complex relationships between data characteristics and expected outcomes, but cannot fully replace human interpretive criteria.

Overall, the findings reinforce a central insight: advancing AutoML for unsupervised learning requires moving beyond fixed internal validation metrics toward richer, data-aware **meta-objectives**. Such objectives should explicitly encode relationships between dataset properties, algorithmic behaviour, and user intent. Building these representations

demands a structured exploration of meta-feature spaces and learned objective functions. The next chapter therefore surveys existing approaches to **meta-learning for clustering**, analysing which datasets, meta-features, and meta-models are most commonly employed in the literature. This analysis provides the empirical and conceptual foundation for the methodological developments introduced in Chapter 3.

Chapter 4

Building Meta-Spaces and Meta-Objectives for Clustering

Contents

The previous chapter demonstrated that the performance of a surrogate-based PS optimisation is closely tied to the composition and quality of its meta-knowledge base. In particular, the surrogate model's ability to generalize across datasets and objectives depends heavily on how clustering problems are represented, through their meta-features, and how these are linked to the chosen evaluation criteria or CVIs. These observations underscore a broader insight: while meta-learning is central to AutoClustering, the design of the meta-space in which learning occurs remains an open and often under-specified challenge.

Constructing a meta-space involves defining which characteristics of a dataset (meta-features) and which measures of clustering quality are most informative for a given objective. Despite the abundance of available meta-features in the literature, ranging from statistical and information-theoretic measures to complexity and model-based descriptors, the optimal combination for representing clustering tasks is not well understood. In practice, AutoClustering frameworks often adopt heterogeneous or ad hoc feature sets, leading to inconsistencies in surrogate performance and limited generalization across domains.

This chapter addresses this gap by investigating the structure, composition, and in-

fluence of meta-spaces and meta-objectives in clustering-oriented AutoML. Specifically, we aim to analyse how different categories of meta-features contribute to the representation of clustering problems, how their relevance varies across different objectives, and how the choice of CVIs shapes the resulting surrogate models. While interpretability techniques are used in our analysis to probe feature importance and surrogate behaviour, explainability is not the primary focus. Rather, it serves as a means to better understand the underlying mechanisms driving meta-learning in clustering.

The study is guided by the following research questions:

- **RQ1:** *Which families of meta-features most strongly influence the performance of surrogate models in AutoClustering frameworks, and how are these features distributed across existing meta-knowledge bases?*
- **RQ2:** *What redundancies, biases, or correlations emerge in the current design of meta-feature spaces, and how do these affect the representational quality of clustering tasks?*
- **RQ3:** *Can more compact or simplified meta-models retain predictive performance without substantially compromising expressiveness, and what insights do interpretability analyses provide toward defining effective meta-objectives for clustering?*

This chapter serves as a preparatory step toward the development of the PoAC framework presented in the following chapter. By systematically analysing meta-feature families, CVI selection strategies, and surrogate modelling behaviour, we gather the methodological foundations and empirical evidence necessary to inform the design of PoAC's meta-space and learning objectives. In doing so, this chapter consolidates the analytical tools and insights required to construct an adaptive, data-aware AutoClustering framework capable of aligning optimization with user intent and problem context.

4.1 Structured Literature Review

We now examine how existing AutoClustering frameworks construct and organize their meta-knowledge bases. The performance of surrogate-based optimisation methods, as shown earlier, is strongly influenced by how clustering tasks are represented within the meta-space, particularly through the selection of base datasets and the meta-features

extracted from them. Yet, despite the central role of these design choices, the literature offers limited guidance on how meta-spaces should be systematically composed or evaluated.

To address this gap, this section presents a structured literature review of AutoClustering frameworks, focusing on how they define, extract, and utilize meta-features. Our review covers 21 representative works published between 2008 and 2025 that explicitly employ meta-learning techniques for clustering. The selected frameworks span diverse application domains and methodological traditions, reflecting the evolution of the field from early domain-specific studies (e.g., biomedical data) to more recent, general-purpose AutoML systems. The full list of reviewed frameworks is provided in Table 4.1.

Table 4.1: Overview of the 21 selected AutoClustering frameworks.

Title	Reference
Ranking and selecting clustering algorithms using a meta-learning approach	de Souto et al. (2008)
Mining Rules for the Automatic Selection Process of Clustering Methods Applied to Cancer Gene Expression Data	Nascimento et al. (2009)
An Analysis of Meta-learning Techniques for Ranking Clustering Algorithms Applied to Artificial Data	Soares et al. (2009)
Clustering Algorithm Recommendation: A Meta-learning Approach	Ferrari and de Castro (2012)
Dealing with overlapping clustering: a constraint-based approach to algorithm selection	Adam and Blockeel (2015)
Clustering algorithm selection by meta-learning systems: A new distance-based problem characterization and ranking combination methods	Ferrari and de Castro (2015)
Extending meta-learning framework for clustering gene expression data with component based algorithm design and internal evaluation measures	Vukicevic et al. (2016)
Constraint-based clustering selection	Van Craenendonck and Blockeel (2017)
Statistical versus Distance-Based Meta-Features for Clustering Algorithm Recommendation Using Meta-Learning	Pimentel and de Carvalho (2018)
A new data characterization for selecting clustering algorithms using meta-learning	Pimentel and de Carvalho (2019)
Unsupervised Meta-Learning for Clustering Algorithm Recommendation	Pimentel and de Carvalho (2019)
AutoClust: A Framework for Automated Clustering based on Cluster Validity Indices	Poulakis et al. (2020)
Automatic selection of clustering algorithms using supervised graph embedding	Cohen-Shapira and Rokach (2021a)
cSmartML: A Meta Learning-Based Framework for Automated Selection and Hyperparameter Tuning for Clustering	ElShawi et al. (2021)
Towards Understanding Clustering Problems and Algorithms: An Instance Space Analysis	Fernandes et al. (2021)
Isolation forests and landmarking-based representations for clustering algorithm recommendation using meta-learning	Gabbay et al. (2021b)
AutoCluster: Meta-learning Based Ensemble Method for Automated Unsupervised Clustering	Liu et al. (2021)
AutoML4Clust: Efficient AutoML for Clustering Analyses	Tschechlov et al. (2021)
cSmartML-Glassbox: Increasing Transparency and Controllability in Automated Clustering	ElShawi and Sakr (2022a)
TPE-AutoClust: A Tree-based Pipeline Ensemble Framework for Automated Clustering	ElShawi and Sakr (2022b)
ML2DAC: Meta-learning to Democratize AutoML for Clustering Analyses	Treder-Tschechlov et al. (2023a)

Rather than aiming for exhaustiveness, our objective is to identify recurring patterns and divergences in how meta-knowledge is built and operationalized. In particular, we focus on two key aspects: (i) the types and sources of datasets used to construct meta-datasets, and (ii) the families of meta-features that characterize clustering problems and guide the learning of surrogate models. Together, these dimensions form the foundation for analysing how meta-spaces are designed and how they influence the generalization capacity of AutoClustering systems.

4.1.1 Dataset Analysis

The construction of a meta-knowledge base begins with the selection of base datasets that reflect the types of clustering problems the framework is intended to address. This stage is particularly crucial in the unsupervised setting, where there is no unique or universally valid clustering solution, and where both the intrinsic characteristics of the data and the intended grouping logic must align with the target application domain (Von Luxburg et al., 2012). Consequently, the datasets used to populate a meta-dataset should not only resemble the tasks that the framework will encounter in practice, but should also encode clustering structures that are meaningful within that intended context.

From these base datasets, meta-features are extracted to characterize the underlying data distributions, providing the foundation upon which meta-models are trained. The properties of these datasets, such as their domain, scale, dimensionality, and number of clusters, therefore directly influence the types of patterns a meta-model can learn and the extent to which it can generalize to new, unseen data. Understanding the origins and composition of the datasets used across different frameworks is thus essential for interpreting their design choices and assessing their expected transferability.

Based on the reviewed literature summarized in Table 4.1, four major groups of datasets can be distinguished according to their provenance and intended purpose:

Cancer Gene Expression Datasets. This category comprises 35 publicly available microarray datasets originally curated by De Souto et al. (2008). These datasets capture various cancer types and are characterized by extremely high dimensionality (often exceeding 1,000 features) and very limited sample sizes (typically fewer than 100 instances per dataset), making them inherently challenging for clustering. The data were collected using two principal microarray technologies—Affymetrix and cDNA—which differ in

measurement methodology.

Such datasets were predominantly used in early AutoClustering frameworks (de Souto et al., 2008; Nascimento et al., 2009; Vukicevic et al., 2016), which focused on biomedical applications where reliable ground-truth labels (e.g., cancer subtypes) were available. Their small sample sizes and high dimensionality provided a natural testbed for evaluating algorithmic robustness to noise and overfitting. However, their strong domain specificity and limited scalability have led to a decline in their use in more recent studies, as the community has shifted toward more diverse and representative benchmarks.

General-Purpose Benchmark Repositories. Datasets from repositories such as the UCI Machine Learning Repository (Frank, 2010) and OpenML (Vanschoren et al., 2013) are widely adopted in AutoClustering research due to their accessibility, domain diversity, and established role in benchmarking machine learning systems. These repositories encompass a broad spectrum of real-world datasets across domains such as healthcare, finance, biology, and sensor data, with substantial variation in size, dimensionality, and inherent cluster structure.

Frameworks such as Ferrari and de Castro (2015); Cohen-Shapira and Rokach (2021b); Pimentel and de Carvalho (2019) have leveraged these repositories to assess generalization performance across heterogeneous settings. The use of publicly recognized benchmark datasets also facilitates reproducibility and comparability with prior work. The growing prevalence of UCI and OpenML datasets reflects a broader community trend toward transparency, accessibility, and the pursuit of general-purpose AutoClustering methods.

Synthetic Data. Synthetic datasets are algorithmically generated to emulate a variety of data morphologies and clustering scenarios. By systematically varying parameters such as the number of clusters, dimensionality, sample size, and noise level, researchers can construct controlled experiments that probe the sensitivity and generalization of AutoClustering methods. Early examples include the generators proposed by Handl and Knowles (2005), which produced Gaussian and ellipsoidal clusters with adjustable overlap and orientation. More recent frameworks often employ the `scikit-learn` utilities (`make_blobs`, `make_circles`, `make_moons`) or advanced packages such as `pymfe`, enabling scalable, high-throughput generation of diverse clustering tasks.

Synthetic datasets are especially useful for constructing large meta-datasets, as they

allow for systematic exploration of algorithmic behaviour under controlled conditions. Frameworks such as Soares et al. (2009); Fernandes et al. (2021); Treder-Tschechlov et al. (2023a) make extensive use of synthetic data to study generalization and performance trends across varied levels of structural complexity.

Clustering Benchmark Suites. Benchmark collections such as SIPU (Fränti and Sieranoja, 2018) and the *clustering-benchmark* repository (Barton, 2015) include synthetic datasets designed specifically to test the limits of clustering algorithms rather than to mimic real-world data. Classic examples include "TwoDiamonds," "Jain," and "Smile," which incorporate geometric structures, overlapping boundaries, or chaining effects that deliberately violate the assumptions of common similarity metrics. For instance, the "TwoDiamonds" dataset (Ultsch, 2003) features adjacent diamond-shaped clusters whose continuity challenges Euclidean-based methods. Such benchmark suites are commonly employed (Van Craenendonck and Blockeel, 2017; Fernandes et al., 2021) to assess algorithmic robustness and failure modes in controlled yet difficult scenarios.

Table 4.2 summarizes dataset usage across the reviewed frameworks, ordered chronologically by publication year. For each work, we report the number of datasets used, together with average values for samples, dimensions, and clusters where available. Dataset groups (A–D) correspond to the categories introduced above.

Table 4.2: Summary of dataset usage across AutoClustering frameworks. Dataset groups: **A** = Cancer gene expression microarray data, **B** = Synthetic datasets, **C** = General-purpose benchmark repositories, **D** = Clustering benchmarking collections. *Note:* Parentheses indicate different dataset settings within the same publication.

Framework	Group	#Datasets	#Samples	#Dimensions	#Clusters
de Souto et al. (2008)	A	32	76.58	1383.38	3.00
Nascimento et al. (2009)	A	35	76.58	1383.38	3.00
Soares et al. (2009)	B	160	309.40	75.00	7.50
Ferrari and de Castro (2012)	C	30	1636.23	36.20	7.50
Adam and Blockeel (2015) (1)	C	14	522.93	11.07	3.92
Adam and Blockeel (2015) (2)	–	22	–	–	–
Ferrari and de Castro (2015)	C	84	369.00	87.22	9.45
Vukicevic et al. (2016)	A	30	89.77	1713.67	3.70
Van Craenendonck and Blockeel (2017) (1)	C	16	804.62	273.12	4.69
Van Craenendonck and Blockeel (2017) (2)	D	5	1954.40	2.00	3.40
Pimentel and de Carvalho (2018)	–	218	–	–	–
Pimentel and de Carvalho (2019)	C	219	4799.85	19.96	5.84
Pimentel and de Carvalho (2019)	–	57	–	–	–
Poulakis et al. (2020)	C	24	810.04	27.42	4.54
Cohen-Shapira and Rokach (2021b)	C	210	726.79	22.29	**165.46**
ElShawi et al. (2021) (1)	C	12	**922857.04**	48.50	5.50
ElShawi et al. (2021) (2)	D	15	1950.13	8.13	13.00
Fernandes et al. (2021) (1)	B	160	1859.25	40.50	18.50
Fernandes et al. (2021) (2)	C	219	275.27	16.73	23.62
Fernandes et al. (2021) (3)	D	220	3466.69	191.21	8.53
Gabbay et al. (2021b)	C	100	2259.55	58.85	**237.07**
Liu et al. (2021)	–	150	–	–	–
Tschechlov et al. (2021)	C	5	8691.80	142.40	12.00
ElShawi and Sakr (2022a)	–	200	–	–	–
ElShawi and Sakr (2022b)	–	118	–	–	–
Treder-Tschechlov et al. (2023a)	B	78	5333.33	21.38	21.38
da Silva et al. (2024c)	B	6130	2081.86	41.55	16.54

The composition of datasets used in AutoClustering frameworks has evolved markedly over time. Early studies emphasized small, domain-specific biomedical datasets, while more recent works increasingly adopt synthetic data generators and large-scale repositories such as UCI and OpenML. This progression reflects both the diversification of available resources and a methodological shift toward scalability and generalization. Nonetheless, reproducibility remains a persistent challenge: several frameworks provide only partial dataset descriptions or do not release their meta-datasets publicly. This lack of transparency complicates cross-framework comparisons and limits the community's ability to rigorously assess methodological progress in meta-learning for clustering.

4.1.2 Meta-Feature Families

Meta-features are quantitative descriptors extracted from base datasets that capture properties relevant to the behaviour of learning algorithms. In the context of AutoClustering, they provide a structured representation of the dataset's statistical, structural, and geometric characteristics, serving as the foundation of the meta-space where surrogate models operate (Castiello et al., 2005). These descriptors enable the comparison of clustering tasks, facilitate the modelling of task similarity, and support the prediction of effective algorithm or pipeline configurations (De Souto et al., 2008; Pimentel and de Carvalho, 2019).

The extraction of meta-features typically involves a multi-step process comprising feature selection, computation, aggregation, and normalization. Many descriptors are derived from individual data attributes (e.g., feature-wise skewness) or their pairwise relationships (e.g., correlation matrices), and are summarized through statistical aggregates such as mean, variance, or quartiles (Vanschoren, 2018). Normalization across datasets is generally required to ensure comparability (Rivolli et al., 2018b, 2022). Importantly, the particular design choices made during this process, ranging from which meta-features are computed to how they are preprocessed, can substantially influence the expressiveness and predictive power of the resulting meta-model (Pinto et al., 2016).

Given the flexibility of meta-feature engineering, a vast number of potential descriptors can be constructed. The most suitable set, however, depends on the objective of the meta-model, whether it is intended to rank algorithms, tune hyperparameters, or recommend complete pipeline configurations. Consequently, the selection of meta-features must be aligned with the downstream task, the domain of interest, and the available com-

putational budget (Rivolli et al., 2018b; Reif et al., 2014; Castiello et al., 2005).

To facilitate interpretability and standardization, meta-features are often grouped into high-level families based on the type of information they encode. While various taxonomies have been proposed (Castiello et al., 2005; Reif et al., 2014; Rivolli et al., 2018b), six broad families encompass most of the descriptors employed across AutoClustering frameworks reviewed in this work.

Simple/General. This family comprises basic univariate statistics such as mean, variance, minimum, and maximum, computed over the raw features of a dataset. These measures provide coarse yet computationally inexpensive summaries of the data distribution. Despite their simplicity, they have proven consistently useful, especially in high-dimensional or resource-constrained settings, and remain widely adopted from early studies (de Souto et al., 2008; Nascimento et al., 2009) to recent frameworks (Gabbay et al., 2021b; Treder-Tschechlov et al., 2023a; da Silva et al., 2024c; Liu et al., 2021). Their generality makes them a natural baseline in most meta-feature sets.

Statistical. Statistical meta-features capture relationships among attributes and summarize the distributional structure of the dataset. Common examples include skewness, kurtosis, correlation coefficients, and covariance. These measures help identify redundancy, symmetry, and dependencies between features, offering a more nuanced view of the data's geometry. Owing to their interpretability and theoretical grounding (Castiello et al., 2005), statistical descriptors constitute a core component of nearly all AutoClustering frameworks.

Information-Theoretical. This family quantifies the amount of information, redundancy, or uncertainty present in the dataset. Typical descriptors include entropy, mutual information, and signal-to-noise ratios (Castiello et al., 2005). Originally popularized in supervised meta-learning (e.g., StatLog (Michie et al., 1995), METAL (Kalousis and Hilario, 2000)), they were later adapted to AutoClustering by frameworks such as (Ferrari and de Castro, 2012) and (Pimentel and de Carvalho, 2019). Information-theoretical features are especially suited to categorical data but can also describe continuous variables. Their continued adoption in works such as (Vukicevic et al., 2016; da Silva et al., 2024c) underscores their value in characterizing dataset complexity and feature informativeness in unsupervised contexts.

Landmarker. Landmarking meta-features were originally introduced by Pfahringer et al. (2000) for supervised learning, using the performance of simple, fast learners (e.g., decision stumps, naive Bayes) as indicators of dataset characteristics. In the unsupervised setting, where accuracy-based metrics are unavailable, this idea has been adapted to clustering by applying lightweight algorithms and assessing their partitions using internal validity indices (e.g., SIL, DBS, or CHS). These meta-features capture how different algorithms perceive the data's structure, providing indirect but powerful signals of algorithm–data compatibility. Landmarker features have become increasingly prominent in AutoClustering frameworks such as (Pimentel and de Carvalho, 2018; Liu et al., 2021; ElShawi and Sakr, 2022b; Treder-Tschechlov et al., 2023a), ranking among the most widely used families after statistical descriptors.

Model-Based. Model-based meta-features extend the idea of landmarking by describing properties of models trained on the data rather than merely their performance. In supervised settings, these may include model size, tree depth, or number of rules (Vanschoren, 2018). In AutoClustering, they encode information about the structure or complexity of the partitions produced by different clustering algorithms. For example, the MARCO-GE framework (Cohen-Shapira and Rokach, 2021b) employs a graph convolutional neural network (GCNN) to derive topological descriptors from learned graph representations, while AutoCluster (Liu et al., 2021) extracts structural statistics from the outputs of KMeans, Agglomerative, and OPTICS algorithms (e.g., cluster compactness, number of dendrogram leaves, or reachability distances). Such descriptors encapsulate the interaction between algorithms and data, enabling surrogates to reason about clustering behaviour at a higher level of abstraction.

Complexity. Complexity-based meta-features quantify the intrinsic difficulty of clustering a dataset. They measure aspects such as class overlap, separability, boundary ambiguity, and density discontinuities (Ho and Basu, 2002; Lorena et al., 2019b; Vanschoren, 2018). Although computationally more demanding, these descriptors provide valuable insights into how challenging a dataset is for clustering algorithms. Their use in AutoClustering remains comparatively limited, but several studies have demonstrated their potential. For instance, Adam and Blockeel (2015) introduced the Constraint-Based Overlapping (CBO) value to assess ambiguity between clusters, while (Pimentel and de Carvalho, 2019; Treder-Tschechlov et al., 2023a; da Silva et al., 2024c) have explored

related measures as indicators of structural complexity.

Overall, these meta-feature families provide a conceptual framework for organizing and comparing meta-spaces across AutoClustering frameworks. The relative emphasis placed on each family varies depending on the system's goals—some prioritize scalability and generalization, while others emphasize structural fidelity or interpretability. As discussed in the following section, these design choices have a direct impact on the surrogate's capacity to model task similarity and to guide effective clustering pipeline synthesis.

4.1.3 Meta-Feature Usage Across AutoClustering Frameworks

Figure 4.1 summarizes the distribution of meta-feature families across the surveyed AutoClustering frameworks. Each cell represents the number of meta-features belonging to a given family, providing a comparative overview of how different systems construct their meta-spaces.

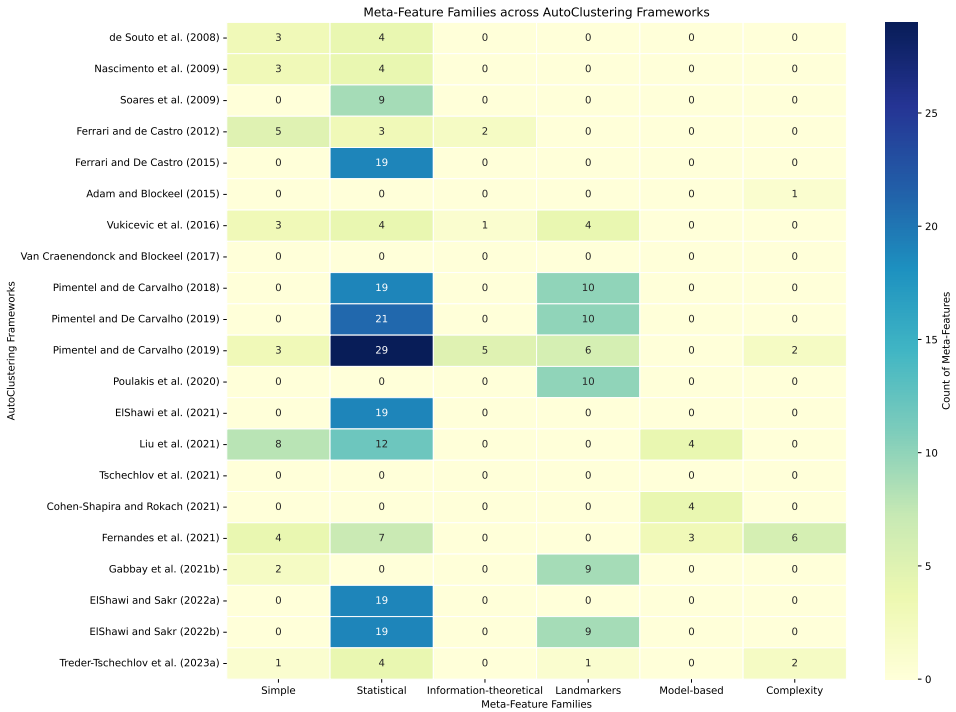


Figure 4.1: Distribution of meta-feature families across AutoClustering frameworks. Color intensity reflects the count of meta-features per family within each framework.

Early studies such as de Souto et al. (2008) and Nascimento et al. (2009) concentrated primarily on biological datasets characterized by high dimensionality and heterogeneous feature scales. These works relied predominantly on simple and statistical meta-features to capture global distributional properties while remaining computationally tractable. As

AutoClustering research evolved, subsequent frameworks began to incorporate landmarkers and information-theoretical descriptors to better represent structural relationships and latent dependencies within datasets. Around 2015, the use of standardized benchmarks such as UCI datasets became widespread, enabling broader empirical comparisons and accelerating methodological convergence. More recent frameworks exhibit a clear shift towards hybrid meta-spaces that integrate multiple feature families, aiming to improve both predictive accuracy and interpretability of surrogate models.

A cross-framework analysis reveals several consistent trends. Statistical meta-features are by far the most ubiquitous across all systems. Their broad applicability, low computational cost, and ease of extraction make them a natural baseline for characterizing dataset distributions. Landmarker features, in turn, have gained substantial traction since approximately 2018, coinciding with the growing interest in algorithm selection and surrogate-based optimization. Their appeal lies in their ability to encode algorithm–data interactions through lightweight proxies, providing informative signals without requiring supervision.

In contrast, model-based and complexity-oriented meta-features remain comparatively under-represented. Their limited adoption can be attributed to their higher computational demands, dependence on auxiliary model training, and the absence of standardized implementations in common meta-learning toolkits. Nonetheless, these families capture complementary aspects of the data, such as geometric structure or intrinsic clustering difficulty, and may hold untapped potential for advancing meta-space expressiveness.

Finally, simple and information-theoretical features appear in moderate proportions, typically complementing other families rather than serving as the core of the meta-space. While these features offer compact and interpretable summaries of data characteristics, their descriptive capacity is often limited when used in isolation.

Taken together, the observed diversity in meta-feature usage across frameworks highlights a lack of consensus regarding optimal meta-space composition. Most existing systems rely on empirically assembled feature sets, often driven by computational convenience rather than systematic evaluation. This underscores an important open question for AutoClustering research: how different combinations of meta-feature families influence surrogate generalization, interpretability, and robustness across domains. Addressing this question is central to developing principled guidelines for constructing effective

and transparent meta-spaces in unsupervised AutoML.

4.2 Explainability of Meta-Models

4.2.1 Global Explanation

Having established the taxonomy of meta-features and datasets used to train meta-models in AutoClustering systems, we now examine the extent to which these meta-features influence model behaviour. Specifically, this section investigates the global explainability of representative AutoClustering frameworks, focusing on how different meta-feature categories contribute to their decision-making processes.

To this end, we employ the Decision Predicate Graphs (DPG) method (Arrighi et al., 2024), a technique that captures predicate-based explanations at a global level. Unlike local feature attribution methods such as SHAP or LIME, which interpret individual predictions, DPG derives a symbolic representation of the decision logic learned by the model. This allows us to quantify and interpret the influence of different meta-feature families across the entire decision space. Our objectives are twofold: (1) to identify which types of meta-features (e.g., statistical, complexity-based, landmarking) consistently shape the clustering pipeline recommendations, and (2) to compare patterns of feature importance across frameworks with differing design choices, dataset compositions, and meta-model architectures. Together, these analyses provide insight into the drivers of generalization in current AutoClustering systems.

Framework Selection and Experimental Setup

For this analysis, we selected three representative frameworks: **AutoClust** (Poulakis et al., 2020), **AutoCluster** (Liu et al., 2021), and **ML2DAC** (Treder-Tschechlov et al., 2023b). These frameworks were chosen based on two key criteria: (1) their reliance on explicit meta-modeling, as opposed to meta-embedding or end-to-end neural approaches (see Section 2.4), and (2) their reproducibility, specifically the availability of their meta-feature spaces and trained meta-models or the data necessary to reconstruct them.

While several other AutoClustering frameworks employ meta-learning components, many do not disclose sufficient details about their meta-feature construction or provide reproducible artifacts, which limited their inclusion in this comparative study.

Discriminative Power of Meta-Features

To assess the discriminative contribution of each meta-feature, we analysed the ten most and least central predicates identified by DPG in the decision graphs of each framework. Predicates were ranked according to their *Logical Relevance Coefficient* (LRC), which quantifies the degree to which each predicate participates in the overall information flow of the model.

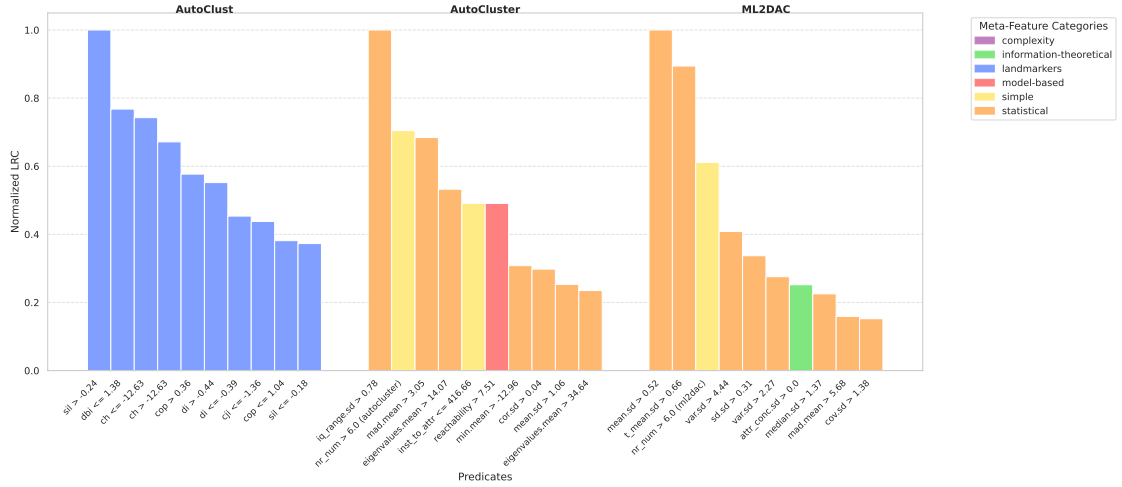
AutoClust. By design, AutoClust relies exclusively on a compact set of landmarker-based meta-features derived from internal CVIs, including `sil`, `dbs`, `ch`, `dunn`, and `cop`. Despite the narrow scope of this feature set (ten CVIs in total), a clear hierarchy emerges when ranked by LRC. The top-ranked predicates are dominated by `sil`, `dbs`, and `ch`, indicating their stronger influence on the meta-model's decision logic. In contrast, indices such as `dunn` and `cop` appear among the lowest-ranked predicates, suggesting limited discriminative utility.

Interestingly, even though all features belong to the same meta-feature family, the LRC distribution is not uniform. The dominance of a small subset suggests redundancy within the CVI family, implying that a reduced and better-calibrated subset could yield equivalent predictive power while improving interpretability and computational efficiency.

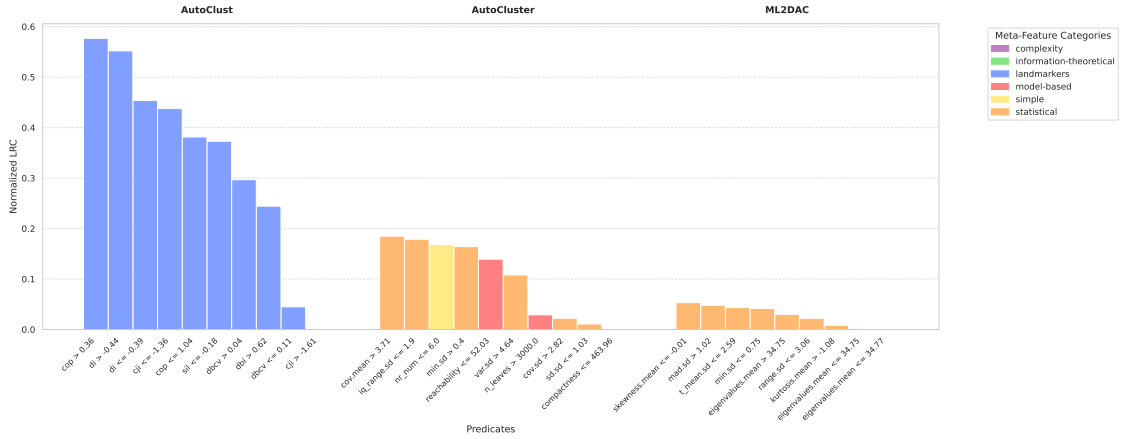
AutoCluster. In contrast, **AutoCluster** primarily employs *statistical* and *simple* meta-features, such as `mean.sd`, `iq_range.sd`, and `nr_num`. These descriptors are inexpensive to compute and consistently favoured by the decision graph, highlighting their discriminative strength. Although the framework's original design also includes *model-based* meta-features, most exhibit limited relevance. One exception is `reachability` $>$ `7.51`, which ranks highly and indicates that datasets with moderate reachability values are particularly informative for the meta-model. Conversely, predicates such as `reachability` $\leq$ `52.03` or `n_leaves` $>$ `3000` appear in the bottom group, suggesting that overly specific or extreme thresholds provide little discriminative signal.

Overall, the DPG analysis reveals that AutoCluster's decision logic is concentrated on a small, stable subset of low-cost meta-features. More complex or noisy descriptors, particularly model-based ones, appear to contribute minimally, raising questions about the trade-off between feature extraction cost and explanatory value.

ML2DAC. The **ML2DAC** framework follows a mixed strategy, integrating statistical, information-theoretical, and landmarker features. However, its top-ranked predicates are again dominated by statistical measures such as `std.mean` and `t_mean.sd`. Notably, while ML2DAC incorporates two complexity-oriented meta-features, none appear among the most relevant predicates. This absence suggests that complexity descriptors, though theoretically informative, may suffer from limited stability or generalizability across datasets. Their extraction cost further discourages their inclusion in large-scale meta-modelling workflows.



(a) Top 10 most relevant meta-feature predicates per AutoClustering framework, ranked by LRC.



(b) Bottom 10 least relevant meta-feature predicates per framework, ranked by LRC.

Figure 4.2: Comparison of the most and least influential meta-feature predicates across AutoClustering frameworks (*AutoClust*, *AutoCluster*, *ML2DAC*). Bars are color-coded by meta-feature category. Higher LRC values indicate stronger participation in the model's global decision logic.

Overall, the analysis shows that although AutoClustering frameworks often rely on large and heterogeneous meta-feature spaces, their predictive behaviour is typically gov-

erned by a small subset of stable and inexpensive descriptors. The limited influence of more complex families—particularly complexity and model-based features—suggests a need for more systematic feature selection and redundancy reduction strategies. These findings motivate further exploration into adaptive meta-feature design guided by explainability insights, which we examine in the next chapter.

4.3 Findings

This chapter investigated how meta-features are employed across AutoClustering frameworks and analysed their relative influence on the decision logic of representative meta-models. The goal was to address the research questions introduced at the beggining of this chapter, which focused on: (1) identifying which meta-feature families contribute most consistently to the generalization of AutoClustering systems, (2) understanding how biases in meta-space design affect explainability and model efficiency, and (3) assessing whether simpler or reduced meta-feature sets could retain comparable predictive power.

RQ1: Which meta-feature families are most influential in AutoClustering meta–models? Our analysis of published frameworks and subsequent DPG-based explainability study revealed a clear hierarchy of meta-feature influence. *Statistical* and *landmarking* features consistently emerge as the most discriminative and stable across systems. Statistical descriptors capture fundamental aspects of data distribution and dispersion with minimal computational cost, while landmarking features, particularly those based on internal CVIs, encode structural information that is both interpretable and robust across datasets. In contrast, *complexity* and *model-based* meta-features, although conceptually appealing, show limited practical impact on model inference. Their extraction cost and instability across heterogeneous datasets reduce their utility in large-scale AutoClustering workflows.

RQ2: What biases or redundancies exist in current meta-feature design? A consistent pattern of redundancy was observed within families of related features, most notably among CVI-based landmarkers. Despite using broad and diverse meta-feature sets, many frameworks rely on only a small subset of features for the majority of their decision logic. This suggests that current AutoClustering systems often over-parametrize their meta-space without proportional performance gains. Furthermore, a dataset-level

bias persists in the literature, with many frameworks evaluated primarily on UCI or biological datasets. This concentration limits the diversity of meta-knowledge available to generalize across new domains.

RQ3: Can simpler meta-models retain performance without major loss of expressiveness? The global explainability analysis indicated that only a few features drive the majority of model behaviour, implying that simpler meta-models—trained on reduced or carefully selected meta-feature subsets—could achieve comparable effectiveness. This opens the possibility of more efficient and interpretable AutoClustering systems, where meta-space design is guided by explainability rather than exhaustive inclusion.

Overall, the findings underscore that current AutoClustering systems tend to favour stability and interpretability over complexity. However, the process by which meta-features and evaluation criteria are selected remains largely heuristic. This lack of principled alignment between feature choice, evaluation objectives, and user intent motivates a more systematic approach to constructing the meta-space, which we explore next.

4.4 Meta-objectives for User Intent

The analyses presented in this and the previous chapter collectively highlight two fundamental insights. First, meta-learning plays a central role in enabling AutoClustering systems to generalize across datasets and evaluation criteria. Second, the composition of the meta-space, specifically the choice of meta-features and CVIs, exerts a critical influence on model performance, explainability, and adaptability. Despite this importance, current frameworks provide little guidance on how to tailor the meta-objective to reflect the user's analytical intent.

Different users may prioritize distinct clustering qualities: compactness, separability, robustness, or interpretability. Yet existing AutoClustering systems typically rely on fixed sets of internal indices or predefined meta-feature families that implicitly encode one particular notion of "good" clustering. This mismatch between system-level objectives and user-level intent limits both flexibility and transparency.

To address this gap, the next chapter introduces the **PoAC** framework, which explicitly incorporates user intent into the construction of the meta-objective. PoAC enables dynamic adaptation of the meta-space by weighting or selecting meta-features and CVIs according to user-defined clustering preferences. In doing so, it moves beyond static

meta-learning formulations and towards a more interactive, intent-aware paradigm of AutoClustering.

Chapter 5

The Problem-Oriented AutoML in Clustering (PoAC) Framework

Contents

The previous chapters explored how AutoML techniques can be extended to unsupervised learning through pipeline synthesis and meta-learning. We demonstrated that while optimizing for internal CVIs can yield reasonable clustering pipelines, such approaches often fail to reflect the underlying goals of the clustering task. These findings highlight the need for more flexible and context-aware optimization strategies. Meta-learning emerged

as a promising alternative, enabling the transfer of knowledge across tasks and allowing models to leverage prior experience to guide new clustering problems.

Building upon these insights, recent AutoClustering approaches have incorporated meta-learning to improve generalization and reduce search costs (Poulakis et al., 2020; Liu et al., 2021; ElShawi et al., 2021; Treder-Tschechlov et al., 2023a). These methods rely on meta-features and surrogate models to capture relationships between datasets and clustering performance, effectively learning how to recommend or predict suitable pipelines (Brazdil et al., 2022). However, most existing frameworks remain constrained by static design choices, such as fixed algorithm sets, hyperparameter spaces, and evaluation metrics that limit their adaptability and hinder truly problem-oriented optimization.

The definition of clustering "quality" is inherently subjective and problem-dependent, what constitutes a good partition for visualization may differ entirely from what matters in noise reduction or anomaly detection (Hennig, 2015; Van Mechelen et al., 2023). Therefore, effective AutoClustering requires not only automation but also problem orientation, the ability to align the optimization process with the user's intent and the characteristics of the data.

To address these limitations, this chapter introduces the **PoAC** framework. PoAC establishes a flexible connection between clustering problems, CVIs, and meta-features, enabling adaptive optimization guided by user-specified objectives. At its core, PoAC leverages a surrogate model trained on a large meta-knowledge base to infer the expected quality of candidate clustering pipelines. This design allows PoAC to synthesize new pipelines dynamically for unseen datasets while incorporating user-defined meta-objectives that reflect diverse problem formulations.

Unlike existing AutoML frameworks, typically bound to fixed CVIs and static optimization strategies, PoAC decouples the optimization process from specific metrics or algorithms. It supports flexible adaptation across problem domains by learning meta-objectives that capture user intent. In doing so, PoAC bridges the gap between traditional AutoML's automation and the contextual understanding required in unsupervised learning.

We hypothesize that a surrogate-based, problem-oriented AutoML approach can outperform state-of-the-art unsupervised AutoML frameworks in both generalization and adaptability. The remainder of this chapter details PoAC's architecture, its training and inference mechanisms, and its empirical evaluation across clustering and anomaly detec-

tion tasks, demonstrating the framework's capacity to tailor solutions to diverse unsupervised learning problems.

The contents of this chapter are part of the manuscript *"PoAC: Problem-Oriented AutoML in Clustering"*, currently under review (round 2) at *Springer Machine Learning*.

5.1 PoAC

An overview of the proposed approach is presented in Figure 5.1. The framework is composed of four main stages: Problem Space Design, Feature Space Mapping, Surrogate Modeling, and Function Optimization. Each of these stages is described in detail below.

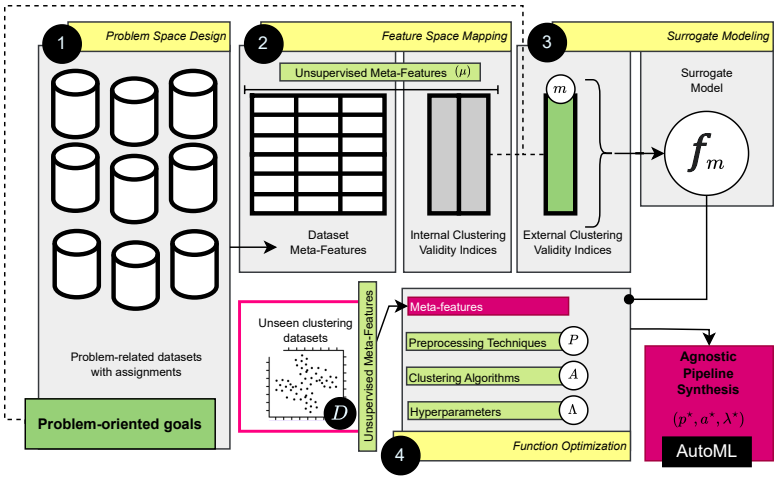


Figure 5.1: Overview of the proposed PoAC framework. The system consists of four stages: (1) **Problem Space Design** , assembling labelled clustering datasets for specific goals (e.g., visualization, anomaly detection); (2) **Feature Space Mapping**, extracting unsupervised meta-features and internal CVIs; (3) **Surrogate Modeling**, training a predictive model to estimate external CVIs; and (4) **Function Optimization**, using the surrogate model as an objective function to synthesize optimal clustering pipelines. The process allows user-defined customization of CVIs and meta-features, ensuring problem-oriented adaptability. Source: Author, reproduced from da Silva et al. (2024c).

The first stage, *Problem Space Design*, involves constructing a problem space composed of labelled clustering datasets associated with specific objectives, such as visualization, denoising, or dimensionality reduction. This foundational step aims to assemble a diverse collection of clustering problems that capture a wide range of structures, complexities, and data distributions. Collectively, these datasets establish the empirical basis for learning how dataset characteristics relate to clustering performance across different analytical goals.

In the second stage, *Feature Space Mapping*, each dataset is projected into a meta-feature space, enabling PoAC to construct an enriched representation suitable for surrogate model induction. Datasets are described by a combination of unsupervised meta-features—capturing statistical, structural, and complexity-related properties—and internal CVIs. To generate diverse CVI values without rerunning clustering algorithms, incremental noise is introduced into the true cluster labels, thereby simulating alternative partitions and enhancing the robustness of the learning process.

The third stage, *Surrogate Modeling*, involves training a predictive model to estimate an external CVI from the extracted meta-features and internal CVIs. Various regression algorithms or ensembles can be employed for this task. The surrogate model learns the complex relationships among dataset characteristics, clustering validation measures, and expected external performance, producing a task-specific predictive function that serves as a learned performance estimator.

Finally, the fourth stage, *Function Optimization*, represents the application phase. Here, an optimization algorithm leverages the trained surrogate model as its objective function to synthesize complete clustering pipelines for new datasets. While the first three stages constitute the *learning phase* of PoAC, performed once to construct the meta-knowledge base, the fourth stage corresponds to the *optimization phase*, executed for each new clustering problem.

PoAC supports problem-specific instantiations through the selection of meta-features and CVIs that best reflect the user's analytical objectives (e.g., compactness, separability, robustness). This customization relies on domain understanding rather than automatic metric selection, ensuring flexibility while maintaining interpretability. Furthermore, PoAC is algorithm-agnostic, as it assesses clustering quality through CVIs rather than through the algorithms themselves. Consequently, any clustering algorithm, or combination thereof, can be incorporated into the search space without requiring retraining of the surrogate model. Additionally, PoAC is decoupled from the optimization procedure: the surrogate model defines a learned objective function that can be optimized by any search strategy (e.g., evolutionary, Bayesian, or reinforcement learning). This separation allows the framework to adapt to different computational budgets and optimization paradigms while preserving methodological consistency.

5.1.1 Problem Statement: Surrogate-based Pipeline Synthesis for Clustering

Let $D = x^{(i)}_{i=1}^{n} \in \mathscr{P}(X)$ denote a dataset consisting of n observations $x^{(i)} \in X$, where $\mathscr{P}(X)$ represents the power set of X.

A set of meta-features is defined as a function $\mu : \mathscr{P}(X) \to \mathbb{R}^l$ that maps a dataset D to a point $\mu(D)$ in an l-dimensional numerical space.

A preprocessing technique is a function $p : \mathscr{P}(X) \to \mathscr{P}(X')$ that maps a dataset D to another dataset $p(D, \lambda_p) \in \mathscr{P}(X')$, where $\lambda_p \in \Lambda_p$ denotes the hyperparameters associated with p. The set of all preprocessing techniques compatible with X and X' is denoted $P_{X,X'}$.

A clustering algorithm is defined as a function $a : \mathscr{P}(X') \to \Pi(X')$, mapping a dataset $D \in \mathscr{P}(X')$ to a partition $a(D, \lambda_a) \in \Pi(X')$, where $\Pi(X')$ denotes the set of all possible partitions of D. Each algorithm is parameterized by hyperparameters $\lambda_a \in \Lambda_a$.

A CVI is a function $m : \Pi(X') \to \mathbb{R}$ that measures the quality of a partition, with higher values corresponding to better performance. The set of all CVIs compatible with X is denoted by M_X.

A clustering pipeline over X is defined as a tuple $((p_1, \lambda_{p_1}), \ldots, (p_k, \lambda_{p_k}), (a, \lambda_a))$, where $(p_1, \ldots, p_k)$ is a sequence of preprocessing techniques and a is a clustering algorithm, each associated with its hyperparameters. For simplicity and without loss of generality, we assume all components operate on the same feature space X. Under this assumption, the set of all possible clustering pipelines is $P^* \times A \times \Lambda_{P^*} \times \Lambda_A$, where P^* denotes the set of all finite sequences of elements from P. A pipeline can thus be represented as $(\vec{p}, a, \lambda)$, where $\vec{p} = (p_1, \ldots, p_k)$ and $\lambda \in \Lambda = \Lambda_{P^*} \times \Lambda_A$ denotes the joint parameterization.

Given a CVI $m \in M$ and a dataset $D \in \mathscr{P}(X)$, the objective is to find the optimal pipeline $(\vec{p}^\star, a^\star, \lambda^\star)$ that maximizes the clustering quality:

$$(\vec{p}^\star, a^\star, \lambda^\star) = \underset{(\vec{p},a,\lambda) \in P^* \times A \times \Lambda}{\arg\max} (m \circ a \circ \vec{p})(D, \lambda) \tag{5.1}$$

, where $\circ$ denotes function composition.

Exhaustively evaluating all candidate pipelines would require executing each on D, which is computationally infeasible. To mitigate this, we introduce, for each m, a surrogate function:

$$f_m((\vec{p}, a, \lambda), \mu(D)) \approxeq (m \circ a \circ \vec{p})(D, \lambda) \tag{5.2}$$

, that approximates the CVI value produced by applying a pipeline to D:

$$f_m((\vec{p}, a, \lambda), \mu(D)) \approx (m \circ a \circ \vec{p})(D, \lambda) \tag{5.3}$$

Thus, the optimization problem can be reformulated as:

$$\underset{(\vec{p},a,\lambda) \in P^* \times A \times \Lambda}{\arg\max} f_m((\vec{p}, a, \lambda), \mu(D)) \tag{5.4}$$

Intuitively, the pair (m, f_m), that is, a CVI and its corresponding surrogate function, encapsulates a specific intended use of clustering. For instance, one may define m as the SIL for clustering tasks aimed at visualization, while another m may correspond to the DBCV index for anomaly detection. The surrogate models trained under these two objectives will differ in the meta-features they leverage most, thereby tailoring the learned optimization landscape to the underlying analytical goal.

5.2 PoAC for Visualization

In this work, we consider the visualization as the goal of a PoAC instance. Recently, the relevance of visualization for the interpretation of ML methods is increasing (Chatzimparmpas et al., 2020), specially in unsupervised learning, where it provides a unique lens through which complex data structures can be interpreted. The objective of clustering for visualization is to uncover meaningful patterns or groupings within datasets and represent them in a visually comprehensible manner (Ezugwu et al., 2022; Al-Jabery et al., 2019). The aim is not only to identify clusters but also to convey their inherent relationships and structures graphically. This task finds applications across various domains, from exploratory data analysis to pattern recognition, where understanding the inherent organization of data fosters insights and informed decision-making (von Luxburg et al., 2012). Effective visualization-driven clustering strategies contribute to the development of interpretable and actionable representations (Assent, 2012), facilitating a deeper understanding of intricate dataset structures, even in high-dimensional data (Strehl and Ghosh, 2003).

5.2.1 Problem Space Design

We have synthesized 6130 datasets to represent a vast range of clustering problems to compose the $\mathscr{P}(X)$. These datasets were created by combining different ranges of important clustering characteristics (Zellinger and Bühlmann, 2023), namely: number of dimensions, number of clusters, quantity of data points, imbalance ratio as well as the shapes of the clusters, described in Table 5.1.

Table 5.1: Dataset synthesized for the PoAC's *problem space* described in section 5.2.

Feature	**Range**
Dimensions	2 - 100
Clusters	2 - 35
Samples	150 - 5000
Overlap	1e-6 - 1e-5
Aspect Ref	1.5 - 5
Aspect Max Min	1 - 5
Radius Max Min	1 - 5
Distributions	normal - exponential - gumbel
Imbalance Ratio	1 - 3

To compose the observations X, we conducted a data augmentation process to generate a range of different partitionings of each dataset in $\mathscr{P}(X)$. This augmentation process consisted of inserting Gaussian noise (Lopes et al., 2019) one hundred times into the cluster labels of the $\mathscr{P}(X)$ datasets. The observations X are mapped in terms of μ, which consists of (i) the unsupervised meta-features extracted from the original datasets and CVIs calculated on the augmented partitioned data of $\mathscr{P}(X)$, and (ii) the external CVI (m), used as the target for the surrogate model.

5.2.2 Feature Space Mapping

Dataset Meta-features

The meta-features selected for this work were based on the taxonomy proposed by Lorena et al. (2019a), chosen for their demonstrated effectiveness and interpretability in characterizing data distributions in unsupervised learning settings (Rivolli et al., 2018a; Vanschoren, 2018; Rivolli et al., 2022). In total, 37 meta-features were employed. While Table 5.2 summarizes the core feature definitions, several statistical descriptors (e.g., mean, standard deviation) were computed for multiple meta-feature groups, such as `sd`, `var`, `mad`, `eigenvalues`, and `cohesiveness`, to better capture variability across attributes.

This expansion from the base set to 37 features enhances the representational richness of the meta-space by incorporating measures of both central tendency and dispersion.

Table 5.2: List of meta-features extracted from datasets to compose the surrogate model.

Meta-Features	Group	Description
attr_to_inst	general	Ratio between the number of attributes.
inst_to_attr	general	Ratio between the number of instances and attributes.
nr_attr	general	Total number of attributes.
nr_inst	general	Number of instances (rows) in the dataset.
attr_conc	info-theory	Concentration coefficient of each pair of distinct attributes.
attr_ent	info-theory	Shannon's entropy for each predictive attribute.
wg_dist	concept	Weighted distance, that captures how dense or sparse is the example distribution.
cohesiveness	concept	Improved version of the weighted distance, that captures how dense or sparse is the example distribution.
one_itemset	itemset	One itemset meta-feature.
two_itemset	itemset	Two itemset meta-feature.
t2	complexity	Average number of features per dimension.
t3	complexity	Average number of PCA dimensions per points.
t4	complexity	Ratio of the PCA dimension to the original dimension.
cov	statistical	Absolute value of the covariance of distinct dataset attribute pairs.
eigenvalues	statistical	Eigenvalues of covariance matrix from dataset.
iq_range	statistical	Interquartile range (IQR) of each attribute.
mad	statistical	Median Absolute Deviation (MAD) adjusted by a factor.
median	statistical	Median value from each attribute.
nr_cor_attr	statistical	Number of distinct highly correlated pair of attributes.
sd	statistical	Standard deviation of each attribute.
sparsity	statistical	Calculates the (possibly normalized) sparsity metric for each attribute.
t_mean	statistical	Trimmed mean of each attribute.
var	statistical	Variance of each attribute.
SIL	landmarker	Measures how well an object fits within its cluster (cohesion) versus its separation from other clusters.
DBS	landmarker	Evaluates clustering compactness and separation.
ARI	landmarker	Quantifies clustering similarity by comparing the overlap between two clusterings while correcting for chance.

The selected descriptors collectively capture complementary aspects of the data, encompassing: (i) general structural properties (e.g., instance-to-attribute ratios), (ii) statis-

tical moments and correlation structure, (iii) information-theoretic complexity, (iv) density and distance-based characteristics, and (v) landmarking measures related to clustering validation indices. This combination enables a comprehensive characterization of dataset behaviour relevant to clustering and anomaly detection performance.

Internal CVI

We opted to use SIL (Rousseeuw, 1987) and DBS (Davies and Bouldin, 1979) as CVIs. Their selection as visualization descriptors for clustering problems is grounded in their ability to provide comprehensive insights into the quality and distinctiveness of clustering solutions, additionally they have been proved to be efficient CVI for AutoML tasks (McCrory and Thomas, 2025). A high SIL suggests visually distinct and well-separated clusters, enhancing the interpretability of visual representations (Shahapure and Nicholas, 2020; Bagirov et al., 2023). Complementing this, a low DBS reinforces the notion of visually cohesive and distinguishable clusters, contributing to an enhanced visual understanding (Maulik and Bandyopadhyay, 2002; Thomas et al., 2013). The two CVIs assess complementary aspects in regards to visualization, their combination allows for a more reliable optimization function, i.e., providing more sound clustering solutions. We chose not to include other internal CVIs due to (i) not being complementary (i.e., measuring the same facets of the problem) and (ii) including many dimensions to optimize makes the optimization problem more challenging and resource costly.

5.2.3 Surrogate Modeling

External CVI

For m, we chose the ARI, as defined in Equation 2.2, to quantify the difference between the augmented partitions and the ground truth according to the original dataset. The intent for the composition of μ and m in this way is to create a range of possible partitionings that represents how visually different they are to the original clustering. By jointly employing these metrics with the meta-features, we aim to capture a nuanced understanding of the visual separability and coherence of clusters in our evaluation, ensuring a robust assessment of the visibility of clustering patterns in diverse datasets.

Surrogate Model

To select an appropriate surrogate model f_m, we evaluated several regressors, including Random Forest, XGBoost, Gradient Boosting, MLP, SVR, Ridge, Lasso, and Gaussian Process Regressor (GPR). Using a 10-fold cross-validation to train different models on the visualization meta-data, Random Forest and XGBoost achieved the highest predictive performances with low RMSE and MAE, while the other models performed noticeably worse (see Table 5.3). Considering both predictive accuracy and interpretability, we selected Random Forest, which captures complex relationships while providing feature importance scores that help understand the influence of individual meta-features and clustering validation indices.

Table 5.3: Comparison of surrogate model regressors on the visualization meta-dataset. Metrics are averaged over 10-fold cross-validation, sorted by RMSE mean.

Regressor	R^2 mean	RMSE mean	MAE mean
RandomForest	0.909	0.069	0.052
XGBoost	0.910	0.069	0.051
GradientBoosting	0.855	0.087	0.067
MLP	0.866	0.084	0.062
SVR	0.288	0.193	0.155
Ridge	0.489	0.163	0.131
Lasso	0.195	0.205	0.167
GPR	0.039	0.224	0.185

The Random Forest regressor was trained using the observations X (meta-features and clustering validation indices) as independent variables and the ARI as the dependent variable. The model provides a reliable fitness function for optimization by effectively capturing the relationships between dataset characteristics and clustering performance. Training was performed with 10-fold cross-validation, yielding strong predictive performance (R^2 $\sim$0.91, RMSE $\sim$0.069, MAE $\sim$0.052), indicating neither overfitting nor underfitting.

Feature importance analysis of the trained Random Forest, shown in Figure 5.2, reveals that the most influential feature is the SIL score (importance = 0.467), followed by the DBS score (importance = 0.22). These results indicate that the model primarily relies on cluster compactness and separation measures when predicting clustering performance.

In contrast, statistical and information-theoretic features such as attr_ent.sd, sparsity.sd, cov.mean, and var.mean have lower importance (0.02–0.03), and measures like median.mean, t_mean.sd, and attr_to_inst contribute minimally (below 0.01). This sug-

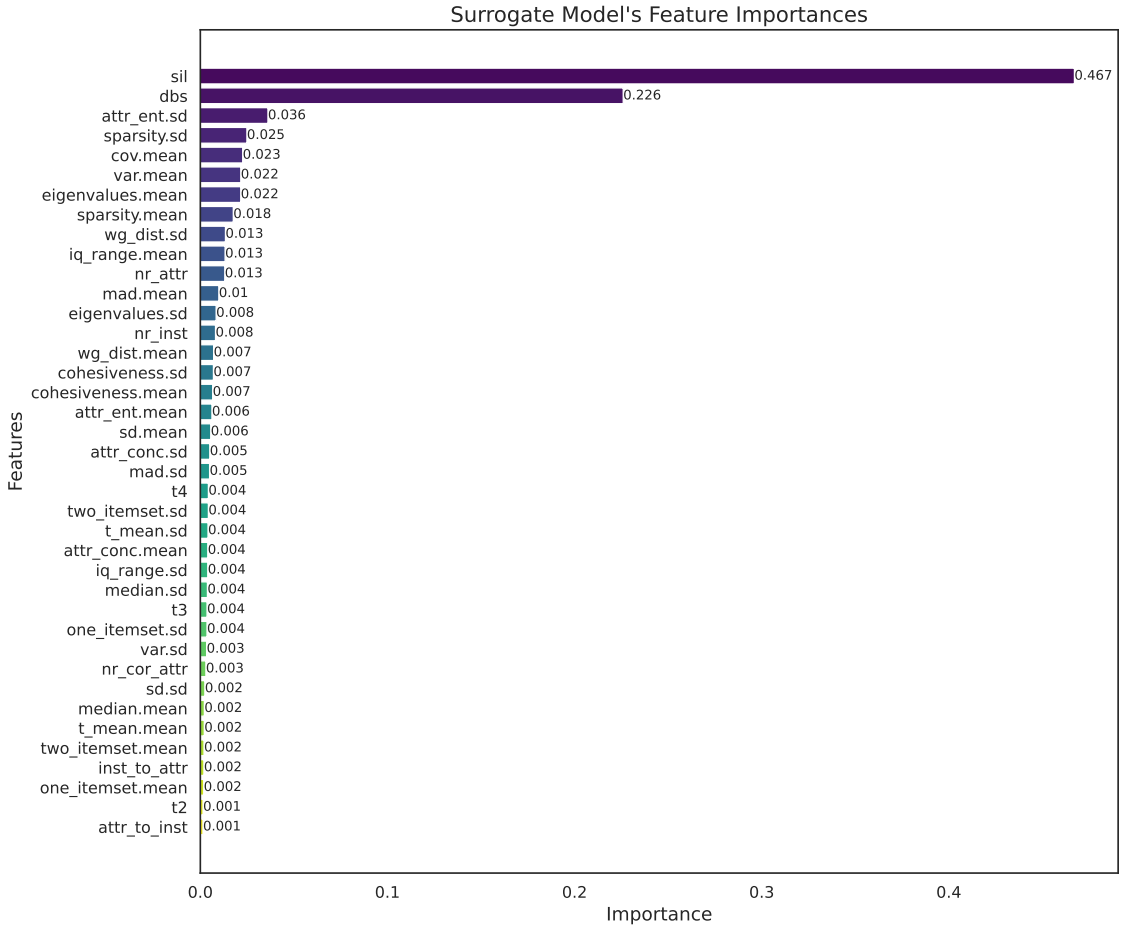


Figure 5.2: Feature importance scores of the Random Forest surrogate model trained on the visualization meta-dataset. The model prioritizes internal CVIs, SIL and DBS, as the most influential predictors, followed by statistical and information-theoretic meta-features such as entropy, sparsity, and covariance descriptors. Importance values are averaged across all decision trees.

gests that the model prioritizes clustering validation indices over intrinsic dataset statistics in assessing clustering quality.

5.2.4 Function Optimization

We extended the capabilities of the AutoML framework TPOT, presented by Olson and Moore (2016), to allow it to synthesize pipelines for clustering problems. This extension consists of a set of important modifications. Mainly, the incorporation of the surrogate model to serve as a fitness function for the evolutionary optimization process. This entails the extraction of meta-features on new unseen clustering datasets, and the measuring of both SIL and DBS for each synthesized clustering pipeline solution, as displayed in Figure 5.3. TPOT was chosen due to its robustness, high level of maintenance, and widespread use within the machine learning community, making it a reliable and well-supported tool for extending into new domains. These qualities make TPOT particularly suitable for our purposes, as it provides a stable and flexible framework for automating complex machine learning tasks.

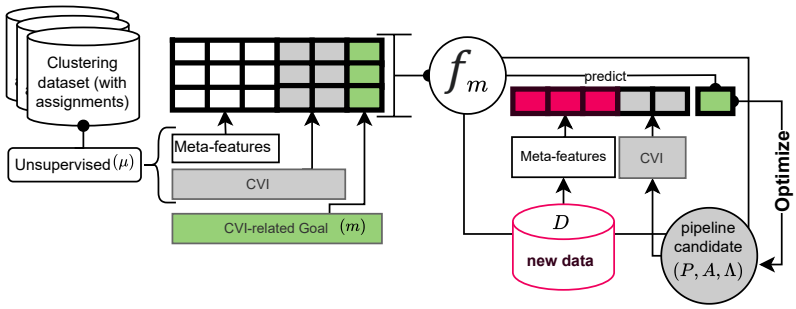


Figure 5.3: Pipeline optimization process for clustering problems within the PoAC framework. The process begins by extracting meta-features (μ) and CVI from clustering datasets. This CVI-related goal (m) serves as the target for the surrogate model f_m, which is used to predict the quality for pipeline candidates (P, A, Λ) on new, unseen data (D). The surrogate model then optimizes the pipeline, evaluating potential solutions based on their predicted (m), using extracted meta-features from the new data and CVI from pipelines to inform the optimization process. Source: Author, reproduced from da Silva et al. (2024c).

Furthermore, we have replaced the conventional TPOT's classification operators with clustering counterparts, as displayed in Table 5.4. Through these modifications, we effectively extended TPOT's utility to the realm of clustering problems, providing a versatile and adaptive AutoML tool [1].

[1] We created a code repository for this version of TPOT, called Tpot Clustering.

Table 5.4: List of operators added to the TPOT configuration.

Operator	Type	Hyperparameters
Agglomerative Clustering	cluster	• **n_clusters**: range(2, 23) • **metric**: euclidean • **linkage**: ward
DBSCAN	cluster	• **eps**: range(2, 23) • **min_samples**: [1e-3, 1e-2, 1e-1, 1., 10., 100.] • **metric**: [10, 25, 50] • **leaf_size**: [3, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50]
KMeans	cluster	• **n_clusters**: range(2, 23) • **init**: [k-means++, random]
Mini Batch KMeans	cluster	• **n_clusters**: range(2, 23) • **eps**: range(2, 23)
Spectral Clustering	cluster	• **n_clusters**: range(2, 23) • **eigen_solver**: [arpack, lobpcg, amg] • **affinity**: [nearest_neighbors, rbf, precomputed, precomputed_nearest_neighbors]
MinMax Scaler	preprocessing	• **-**
Normalizer	preprocessing	• **norm**: [l1, l2]
Standard Scaler	preprocessing	• **-**
Variance Threshold	feature selection	• **threshold**: [0.1, 0.25]
PCA	decomposition	• **n_components**: [2, 3, 5, 10]
Fast ICA	decomposition	• **n_components**: [2, 3, 5, 10]

5.2.5 Baselines

We designed and performed experiments to evaluate the quality of PoAC when compared to the reproducible state-of-the-art frameworks that focus on CASH, namely *ML2DAC* from Treder-Tschechlov et al. (2023a), *Autocluster* proposed by Liu et al. (2021), *cSmartML* as presented in ElShawi et al. (2021) and, finally, *AutoML4Clust* presented in Tschechlov et al. (2021). As for PoAC, we specifically created an instance of the PoAC framework for visualization as a clustering problem (Ezugwu et al., 2022) and assessed the performance of pipeline optimizations. In the context of the proposed *problem-oriented* methodology, prioritizing visualization as a clustering problem underscores its significance as a user-defined goal, catering to scenarios where the interpretability and visual clarity of clustering solutions are paramount.

It is important to clarify that the datasets employed in this evaluation are not restricted to visualization-oriented contexts. Both the synthetic and real-world datasets encompass a wide range of data distributions, dimensionalities, and cluster structures representative of general clustering problems. In this setup, PoAC's "visualization" instance does not rely on visual data but rather defines an interpretable clustering objective, favoring partitions that align well with human-understandable structure and separability. This ensures that the results reflect general unsupervised clustering performance rather than being limited to a visualization-specific domain.

We considered two distinct groups of datasets to comprehensively evaluate the frameworks in the context of clustering. The first group is composed of datasets used in the benchmark work done by da Silva et al. (2024b), it consists of one hundred synthetic clustering datasets with ranging degrees of instances distributions, clusters separations and densities, among others, as described in Table 5.5.

Table 5.5: Dataset validation group for experiments in subsection 5.2.5.

Feature	**Range**
Dimensions	2 - 100
Clusters	2 - 35
Samples	150 - 5000
Overlap	1e-6 - 1e-5
Aspect Ref	1 - 10
Aspect Max Min	1 - 10
Radius Max Min	1 - 10
Imbalance Ratio	1 - 3

The second group (Table 5.6), encompassed twenty two datasets sourced from the UCI repository (Markelle Kelly, 2017), providing a real-world dimension to our analysis, they were chosen for their usage in the validation of related works, namely: Ml2dac and AutoCluster.

Table 5.6: Real-world datasets validation group.

Dataset	Instances	Num Features	Number of Clusters
arrhythmia	452	262	13
balance-scale	625	4	3
dermatology	366	34	6
ecoli	336	7	8
german	1000	7	2
glass	214	9	6
haberman	306	3	2
heart-statlog	270	13	2
iono	351	34	2
iris	150	4	3
letter	20000	16	26
segment	2310	19	7
sonar	208	60	2
tae	151	5	3
thy	215	5	3
vehicle	846	18	4
vowel	990	10	11
wdbc	569	30	2
wine	178	13	3
wisc	699	9	2
yeast	1484	8	10
zoo	101	16	7

By incorporating both synthetic and real-world datasets, our experimental design aimed to provide a holistic assessment of the PoAC's performance across a spectrum of clustering challenges, thereby enhancing the generalizability and applicability of our findings.

5.2.6 Results and Discussion

We applied each one of the baseline frameworks, as well as the proposed PoAC, on the group of one hundred validation datasets. The results are presented in Table 5.7. The obtained results pertaining ARI will be discussed in subsubsection 5.2.6.

PoAC exhibited outstanding results, achieving an average SIL of 0.54, which indicates well-defined and clearly separated clusters, outperforming all competing frame-

Table 5.7: Frameworks performance regarding ARI, SIL, and DBS for the validation datasets (mean, median, and standard deviation).

Framework	ARI			SIL			DBS		
	Mean	**Median**	**SD**	**Mean**	**Median**	**SD**	**Mean**	**Median**	**SD**
PoAC	**0.70**	**0.87**	**0.27**	**0.63**	**0.63**	**0.07**	**0.52**	**0.52**	**0.18**
ML2DAC	0.68	0.83	0.32	0.40	0.41	0.19	1.68	1.34	1.31
AutoML4Clust	0.59	0.64	0.29	0.37	0.35	0.16	1.39	1.26	0.76
Autocluster	0.59	0.61	0.22	0.23	0.19	0.19	2.43	2.11	1.63
cSmartML	0.39	0.39	0.29	0.25	0.25	0.20	1.37	1.28	0.60

works in this regard. ML2DAC and AutoML4Clust followed with mean SIL values of 0.39 and 0.36, respectively, while cSmartML and Autocluster obtained lower scores of 0.24 and 0.23. In terms of DBS, PoAC again achieved the best result with a score of 0.76, reflecting compact and well-separated clusters. The next best-performing frameworks were cSmartML (1.37), AutoML4Clust (1.39), ML2DAC (1.67), and Autocluster (2.42).

These results highlight PoAC's ability to consistently balance cluster compactness and separation across multiple evaluation dimensions. In contrast, the relative ranking of the other frameworks varies depending on the metric considered, some perform well on a single index (e.g., SIL or DBS) but degrade on others, indicating that they tend to optimize specific aspects of clustering rather than achieving globally coherent solutions. PoAC's superiority in generating pipelines that yield visually cohesive and structurally meaningful clusters further demonstrates its effectiveness for visualization-oriented clustering tasks. This robustness likely arises from PoAC's unique optimization strategy, which jointly considers multiple cluster quality measures aligned with the intended visualization goal.

As illustrated in Figure 5.4, PoAC's results form a compact cluster of points concentrated within the region of optimal SIL and DBS values, high SIL (close to 1) and low DBS (close to 0), indicating that its clustering solutions are consistently well-structured and well-separated. The color coding (ARI) further shows that PoAC's points are predominantly green, confirming that these solutions also achieve higher agreement with the ground truth compared to the other frameworks. In contrast, the other methods produce more dispersed patterns across the SIL-DBS space, with lower or more variable ARI values (warmer colors). This dispersion suggests that those frameworks are more flexible in terms of the CVIs, but less stable and less reliable in jointly optimizing them. Therefore, Figure 5.4 highlights that PoAC not only maintains coherence across both CVIs but also

achieves the highest and most consistent ARI performance.

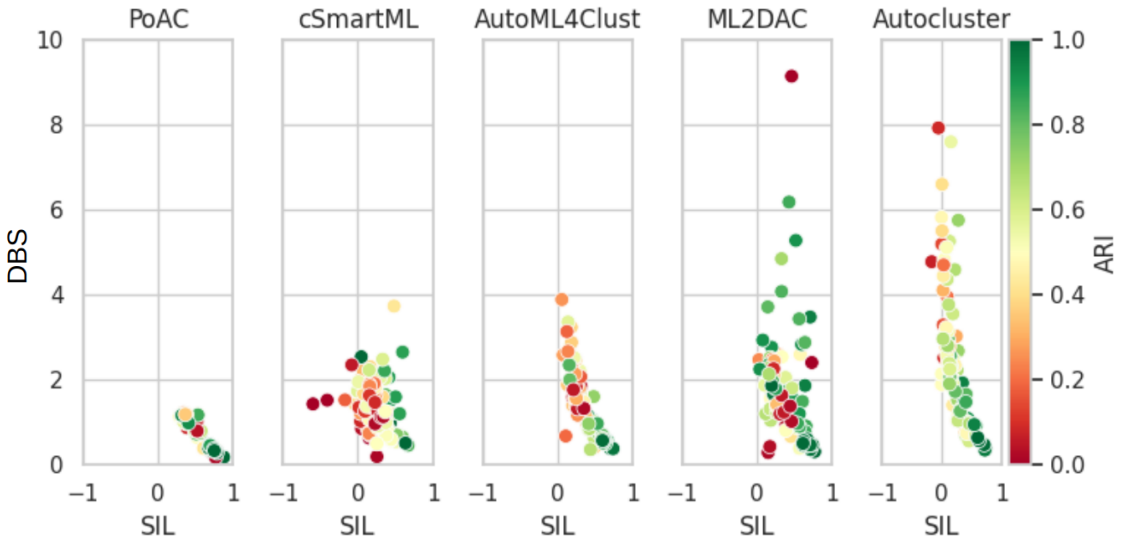


Figure 5.4: Comparison of clustering quality across frameworks (PoAC, ML2DAC, AutoML4Clust, Autocluster, cSmartML) on validation datasets. Each subplot shows the relationship between SIL and DBS, with color intensity indicating ARI. PoAC achieves clusters concentrated in the optimal region (high SIL, low DBS), indicating better separation and compactness compared to competing AutoML methods. Source: Author, reproduced from da Silva et al. (2024c).

Patterns in Predictive Performance. Beyond absolute performance, several systematic patterns emerge across the validation datasets. First, PoAC demonstrates consistently high performance across all three metrics (ARI, SIL, and DBS), suggesting that its surrogate-guided optimization effectively balances compactness, separation, and alignment with ground truth. In contrast, other frameworks show metric-specific tendencies: ML2DAC often achieves high ARI values but lower SIL and higher DBS, indicating that it prioritizes alignment with labels over structural cluster quality; AutoML4Clust and Autocluster frequently achieve moderate SIL or DBS but display higher variability, reflecting less consistent pipeline performance across datasets. cSmartML generally underperforms in all metrics, highlighting its sensitivity to dataset characteristics and its reliance on fixed CVI optimization.

Second, variability (SD) across datasets reveals insights into robustness. PoAC's low standard deviations for SIL and DBS indicate stable structural quality, whereas high SDs in competing frameworks suggest that they are more sensitive to dataset heterogeneity. This pattern is particularly noticeable in Autocluster, whose ARI varies widely depending on the dataset, implying overfitting to certain cluster configurations or instability in pipeline recommendations.

Finally, correlations between metrics show that high ARI does not always coincide with high SIL or low DBS. For instance, ML2DAC sometimes achieves ARI comparable to PoAC but at the cost of suboptimal SIL/DBS, suggesting that frameworks optimizing a single objective (e.g., label alignment or a fixed CVI) may sacrifice overall clustering coherence. PoAC's joint consideration of multiple internal validation measures allows it to maintain consistency across dimensions, highlighting the benefits of multi-objective surrogate-guided optimization in predictive pipeline performance.

Clustering Performance Analysis

To evaluate the similarity between the clustering solutions produced by each framework and the original cluster labels for the validation datasets, we compared their ARI distributions. The proposed PoAC achieved the highest mean ARI of 0.70, followed by ML2DAC (0.67), AutoML4Clust (0.59), Autocluster (0.58), and cSmartML (0.38).

A statistical comparison was performed over five populations, each consisting of 100 paired ARI samples. Preliminary tests indicated that some populations (AutoML4Clust, Autocluster, and PoAC) deviated from normality. Consequently, we applied the non-parametric Friedman test with a 95% confidence level, which confirmed significant differences among the medians of the populations. A post-hoc Nemenyi test (Nemenyi, 1963), with a critical distance (CD) of 0.61, was then used to identify pairwise differences. As shown in Figure 5.5, no significant differences were found within the following groups: (i) AutoML4Clust, Autocluster, and ML2DAC; and (ii) ML2DAC and PoAC. All other comparisons revealed significant differences, particularly between cSmartML and any other framework, and between PoAC and both AutoML4Clust and Autocluster.

Figure 5.5 summarizes these results by ranking the frameworks according to their mean ranks (lower is better). The horizontal bars indicate groups of methods whose performances are not significantly different at the 95% confidence level. The proximity between PoAC and ML2DAC confirms that both achieve similarly high ARI values, whereas the remaining frameworks form a distinctly lower-performing group. Therefore, while ML2DAC performs competitively, PoAC attains the lowest mean rank (2.22) and stands at the upper boundary of the top statistical group, consolidating its position as the leading framework across the evaluated datasets.

To ensure a fair and directly comparable evaluation between the two top-performing frameworks, PoAC and ML2DAC, we replicated the real-world experiment from the

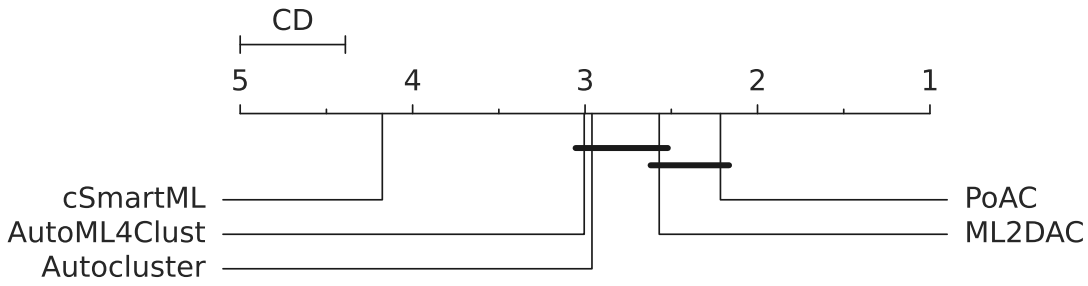


Figure 5.5: Frameworks ranked by mean rank according to the Nemenyi test on ARI (lower values indicate better performance). From last to first: cSmartML (MR=4.18), AutoML4Clust (MR=3.01), Autocluster (MR=2.96), ML2DAC (MR=2.57), and PoAC (MR=2.22). Source: Author, reproduced from da Silva et al. (2024c).

ML2DAC study using the same set of UCI datasets (referred to as the second validation group). Because ML2DAC does not include automated preprocessing, we conducted an ablation study of PoAC by removing all preprocessing operators from its search space. Under these matched conditions, PoAC achieved a mean ARI of 0.26 (median = 0.18, SD = 0.28), closely aligning with ML2DAC's reported mean ARI of 0.27 for its full configuration and 0.19 for its simplified variant (Treder-Tschechlov et al., 2023a). These results demonstrate that even without preprocessing, PoAC achieves comparable performance to ML2DAC while retaining the benefits of a meta-learned, transferable optimization strategy.

Overall, the comparison highlights PoAC's distinct advantages. In the first experiment, it delivered the best overall clustering quality, confirming the effectiveness of using a surrogate model as a fitness function for pipeline search. Furthermore, the surrogate model trained by PoAC, originally designed for visualization-oriented clustering, proved adaptable to datasets beyond that scope, achieving comparable results on the UCI benchmarks without fine-tuning or reconfiguration. This robustness underscores the generalizability of PoAC's design, emphasizing its potential across diverse clustering scenarios without requiring complex hyperparameter adjustments.

Runtime Analysis and Computational Cost

Constructing the meta-space constitutes the primary computational cost in both PoAC and ML2DAC, as it involves extracting descriptive meta-features from large numbers of datasets. For PoAC, computing around 40 meta-features and 3 CVIs across 6,000 synthetic datasets required approximately 4 hours and 52 minutes on a workstation with

an Intel i9 CPU and 64 GB of RAM, corresponding to an average of 2.9 seconds per dataset.

In comparison, ML2DAC (Treder-Tschechlov et al., 2023a) reports per-dataset meta-feature extraction times ranging from 0.1 to 82.8 seconds, depending on the selected meta-feature set. Its most comprehensive configuration (`MF_AutoCluster`) averages 35.7 seconds per dataset, while PoAC provides competitive meta-feature coverage with substantially reduced extraction time, facilitating large-scale meta-space construction.

Optimization and Inference Time. ML2DAC constructs its meta-space from a predefined set of 78 datasets, each optimized independently with a two-hour time budget, totaling approximately 156 hours (Treder-Tschechlov et al., 2023a). Once this meta-space is built, inference on new datasets takes on average 74 seconds per dataset when using the full meta-feature configuration, enabling rapid deployment on unseen problems.

For PoAC, meta-feature extraction and meta-space construction required less than five hours. Under the original setup, full optimization across both sets of benchmark datasets required approximately 306.7 hours, corresponding to 100 generations with a population of 100 individuals per dataset and no early stopping. To align with ML2DAC's experimental protocol, we also evaluated PoAC under a two-hour time budget per dataset, resulting in a total optimization time of 44 hours for the 22 datasets in the benchmark.

Once the PoAC surrogate is trained, inference on new datasets is efficient and fully configurable, depending on user-defined hyperparameters such as population size, number of generations, or optional time budgets. Overall, PoAC achieves a favorable balance between meta-space construction and optimization, offering competitive computational efficiency and strong scalability across diverse downstream tasks.

Generated Pipeline Complexity

An analysis of the mean pipeline complexity (i.e., the number of pipeline steps) across generations reveals a clear trend within the PoAC framework. As shown in Figure 5.6, PoAC consistently evolves towards pipelines with a mean complexity of at least two steps. This stabilization suggests an optimal balance between essential preprocessing operations and the clustering algorithm itself. The trend highlights PoAC's ability to autonomously converge toward compact yet effective configurations, minimizing unnecessary complexity while maintaining strong clustering performance.

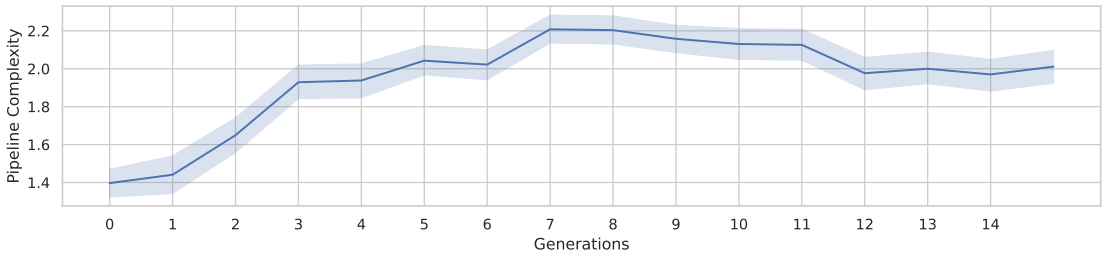


Figure 5.6: Evolution of pipeline complexity during the PoAC optimization process. Complexity is measured as the average number of components (preprocessing + clustering steps) per pipeline generation. The model converges toward compact pipelines with approximately two steps, suggesting an optimal trade-off between performance and simplicity. Source: Author, reproduced from da Silva et al. (2024c).

As depicted in Figure 5.7, the distribution of pipeline complexities reveals a clear association with ARI values. Pipelines yielding lower ARI scores display a broader spread of complexities, reflecting an exploratory search behavior. In contrast, higher-performing pipelines (with higher ARI) are concentrated around one or two steps, consistent with the evolutionary convergence observed earlier. Nonetheless, a small number of high-performing pipelines with three or four steps remain, indicating that PoAC retains flexibility to adapt its structure when additional preprocessing improves performance. Overall, the framework tends to favor simpler, well-balanced pipelines that maintain high effectiveness.

The influence of pipeline composition on clustering quality is illustrated in Figure 5.8 using the UCI *dermatology* dataset (Ilter and Guvenir, 1998). In this example, ML2DAC recommends a simple CASH-based solution, *KMeans* with two clusters, whereas PoAC proposes a multi-step pipeline including (i) *MinMaxScaler* for normalization and (ii) *MiniBatchKMeans* as the clustering algorithm, with batch size 10 and six clusters. The resulting pipeline produces a clustering more faithful to the ground truth, demonstrating the advantage of solving the full pipeline search (PS) problem instead of limiting the optimization to algorithm and hyperparameter selection (CASH). This example illustrates how PoAC leverages preprocessing choices to enhance clustering outcomes.

An important factor underlying this flexibility is PoAC's algorithm-agnostic design, which directly relates to the diversity of effective pipeline configurations reported above. Unlike other AutoML approaches that build a meta-database tied to specific algorithms and hyperparameter settings, PoAC introduces a label-augmentation process during meta-learning, adding controlled noise to the synthetic datasets' cluster labels. This allows the surrogate model to learn algorithm-independent partitioning patterns, enabling it to

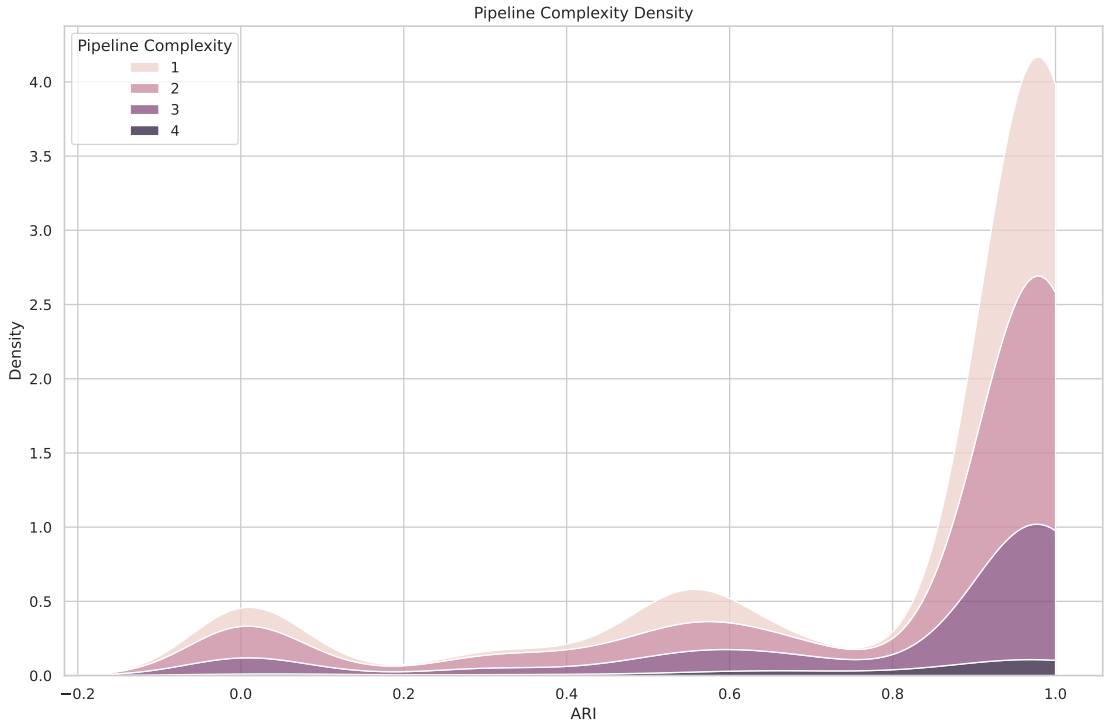


Figure 5.7: Relationship between pipeline complexity and ARI performance across generations. Higher-performing pipelines (with higher ARI) tend to be simpler, typically involving one or two processing steps. This trend reflects PoAC's capacity to discover minimal yet effective clustering pipelines while preserving adaptability for complex datasets. Source: Author, reproduced from da Silva et al. (2024c).

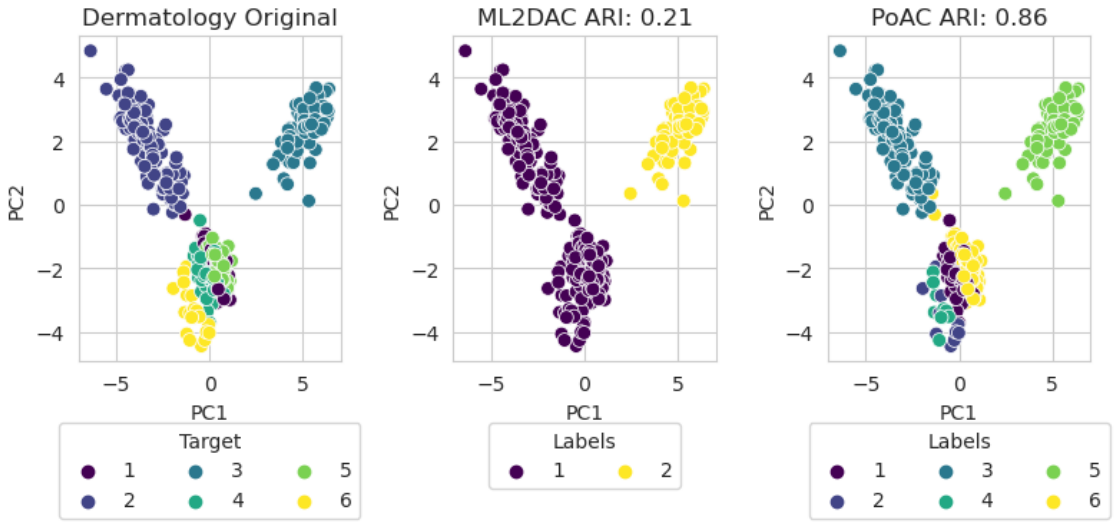


Figure 5.8: Example of clustering pipeline recommendations for the Dermatology dataset. PoAC proposes a two-step pipeline combining MinMaxScaler normalization and MiniBatchKMeans, achieving a clustering more faithful to the ground truth than ML2DAC's single-step KMeans solution. This demonstrates PoAC's advantage in full pipeline synthesis and problem-oriented optimization. Source: Author, reproduced from da Silva et al. (2024c).

generalize across a broad range of clustering strategies. Consequently, PoAC can recommend different, but equivalently performant, pipeline configurations without retraining the surrogate model when the search space changes. This capability explains the observed adaptability in pipeline composition and reinforces PoAC's robustness as a truly algorithm-agnostic AutoML framework.

Ablation Study

It is crucial to ensure that the performance/complexity trade-off is justified. To this end, we conduct an ablation study to evaluate the efficacy of the complete PoAC method against sub-configurations of its components in the task of clustering PS. We named the different optimization strategies considered in this study respectively:

1. **PoAC SIL** optimizes pipelines based solely on their SIL score, favoring those that demonstrate higher clustering quality as indicated by this metric.

2. **PoAC DBS** focuses exclusively on the DBS score, selecting pipelines that achieve superior performance according to this criterion.

3. **PoAC CVI** adopts a different approach by creating a surrogate model that combines SIL and DBS scores, specifically examining their non-linear correlation without incorporating the meta-features group. This surrogate model, trained using a Random Forest on the same dataset as the complete PoAC method, provides a more nuanced understanding of the relationship between SIL and DBS.

We ran each of the four optimization strategies ten times for every dataset in the first validation group (Table 5.5), and recorded the mean ARI per dataset for each of the strategies.

The boxplot analysis in Figure 5.9 reveals distinct performance characteristics across the four aforementioned optimization strategies, based on their ARI scores. PoAC presents a relatively narrow interquartile range (IQR), indicating less variability in clustering performance. The upper quartile of the data extends close to the maximum possible ARI score of 1.0, and numerous individual data points are clustered near this upper bound. The median ARI is significantly higher than the other strategies, around 0.8. This suggests that PoAC frequently achieves highly accurate clustering results. While the lower whisker does extend down to approximately 0.1, highlighting some instances of lower performance, PoAC generally offers robust and dependable clustering outcomes.

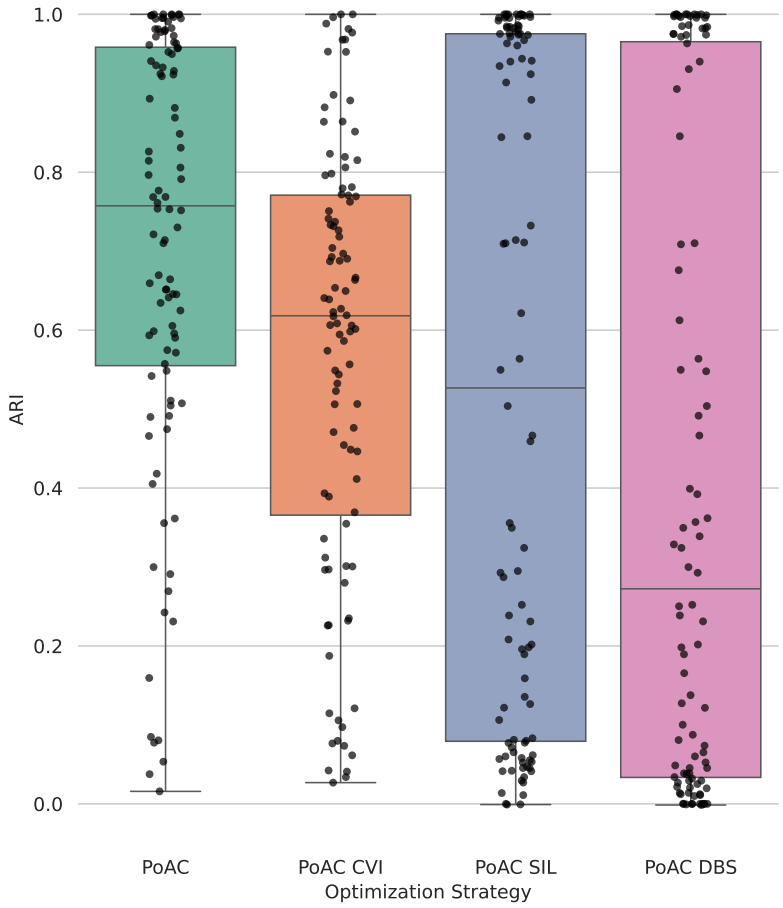


Figure 5.9: Distribution of ARI scores for different optimization strategies: complete PoAC (meta-features + CVIs + surrogate), PoAC-CVI (CVI + surrogate), PoAC-SIL, and PoAC-DBS. The complete PoAC configuration shows the highest median ARI and lowest variance, indicating superior and more consistent clustering performance across datasets. Source: Author, reproduced from da Silva et al. (2024c).

PoAC CVI, while somewhat similar to PoAC, exhibits a lower median ARI score, around 0.6, and a slightly wider IQR. This wider range might indicate more variability in performance. The lower whisker reaches down to 0.0, meaning that PoAC CVI occasionally produces very poor clustering results. Nevertheless, it also achieves a significant number of high ARI scores, demonstrating that while it may be less consistent than PoAC, it can still be effective in certain cases.

PoAC SIL, shows a wide spread of ARI values. The median ARI is around 0.5, which suggests moderate clustering performance. The IQR is substantial, indicating variability in the results. While it can achieve results around the maximum value, suggesting some instances of excellent performance, the overall inconsistency is notable.

Lastly, the ARI values for PoAC DBS are also widely distributed, covering the full range from 0.0 to 1.0. However, the median ARI is lower than PoAC SIL, and the larger IQR reflects even greater variability in clustering performance, it is in fact the largest in the whole group of strategies. This indicates that while PoAC DBS can occasionally achieve high ARI values, its performance is highly inconsistent, leading to a broad range of outcomes.

As shown in Figure 5.10, the optimization strategies that rely on a single CVI tend to have a higher concentration of lower ARI values when compared to the complete and the ablated PoAC. The green line, representing PoAC, has the steepest and most direct rise, indicating that this strategy achieves higher ARI scores more consistently. Most of the density for PoAC is concentrated above the 0.8 ARI mark, suggesting that it tends to produce highly accurate clustering outcomes. The orange line, representing PoAC CVI, also shows a steady rise, though it lags behind PoAC. This indicates that while PoAC CVI is generally effective, it produces slightly lower ARI scores on average compared to PoAC.

The blue (PoAC SIL) and pink (PoAC DBS) lines display more gradual curves, indicating a broader range of ARI scores. PoAC SIL starts to rise steeply around the 0.2 ARI mark, while PoAC DBS has a more pronounced curve starting from 0.1. These curves suggest that these strategies have a more variable performance, with a significant proportion of their density at lower ARI scores, indicating that they are less likely to consistently produce high-quality clustering results.

Both of these analysis reveal that the complete PoAC method is the most effective and consistent optimization strategy based on ARI values, showing high clustering quality and

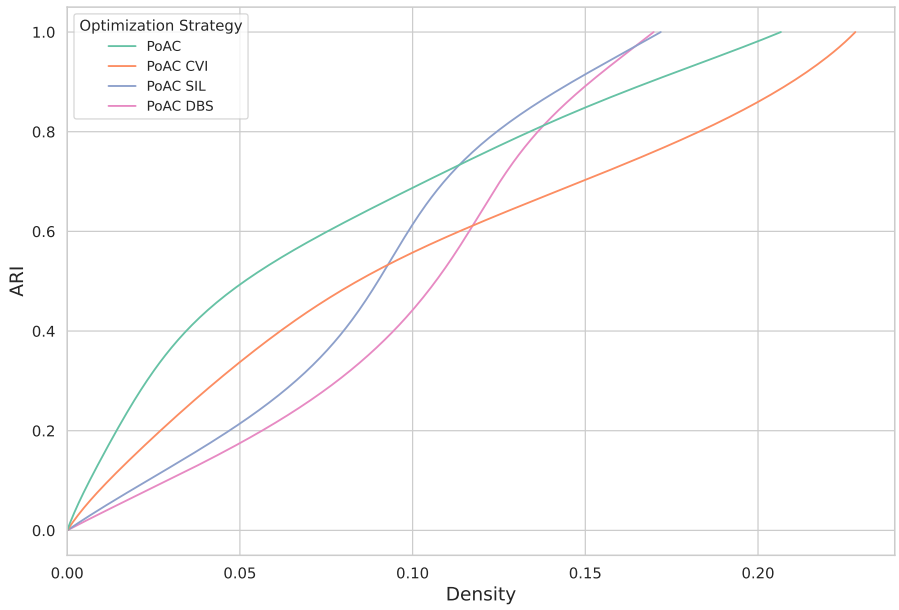


Figure 5.10: Density curves of ARI across four optimization strategies. The PoAC CVI rises sharply near ARI = 0.8, reflecting consistent high-quality clustering, whereas simpler variants (SIL-only or DBS-only) display broader, flatter curves, indicating more variable and less reliable results. Source: Author, reproduced from da Silva et al. (2024c).

minimal variability. PoAC SIL shows moderate performance with significant variability, while PoAC DBS exhibits greater variability and less consistent performance. PoAC CVI shows relatively consistent performance but is not as reliable or efficient as PoAC. These findings highlight the importance of selecting the appropriate optimization strategies in relation to the specific clustering problem.

To understand the specific cases in which the optimization strategies can be at least as suited to perform as the complete PoAC method, we conducted further analysis to compare the performance of the four optimization strategies side by side, while identifying the features that define the archetype of the validation datasets, as shown in Figure 5.11. This analysis aimed to correlate the performance of the strategies with the dataset features described in Table 5.5, e.g. number of clusters, number of dimensions, and number of features. By visualizing the data in a heatmap, we were able to observe patterns and relationships that highlight how different optimization strategies perform across datasets with varying characteristics. Each cell in the heatmap represents the mean ARI for a given dataset-optimization strategy combination, with darker colors indicating higher ARI values.

For instance, the analysis revealed that strategies such as the PoAC method consistently performed well across datasets with a high number of dimensions, whereas PoAC SIL showed variable performance that was influenced more by the number of clusters.

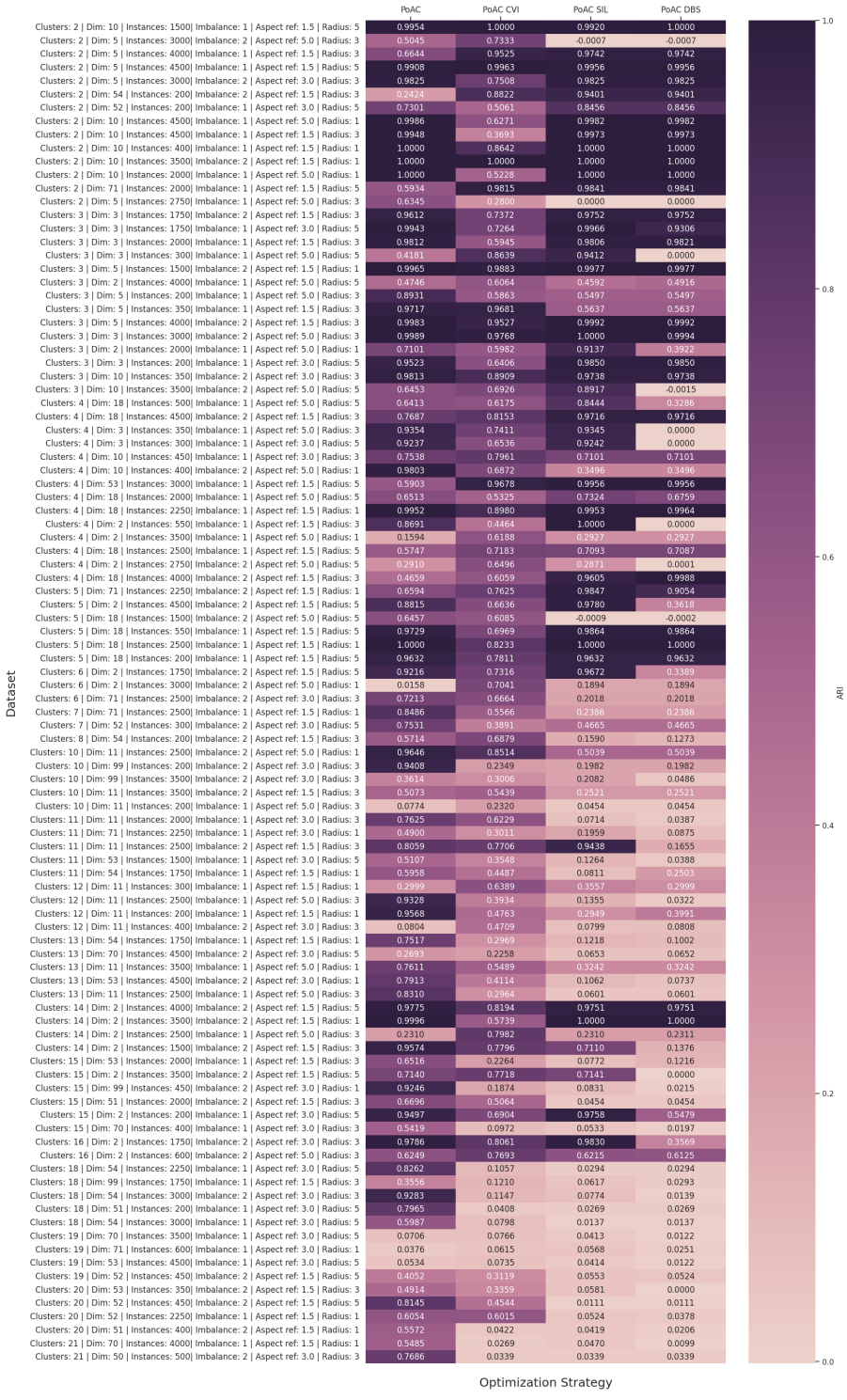


Figure 5.11: Heatmap showing mean ARI per dataset and optimization strategy. Each row represents a dataset characterized by its number of clusters, dimensions, instances, imbalance ratio, and geometric features. Darker cells correspond to higher ARI scores. PoAC consistently achieves high ARI across diverse data profiles, highlighting its robustness and generalization across clustering scenarios. Source: Author, reproduced from da Silva et al. (2024c).

This analysis underscores the importance of considering dataset-specific features when selecting an optimization strategy for clustering, as the effectiveness of each strategy can vary significantly depending on these characteristics. This nuanced understanding enables more informed decisions when applying clustering techniques to diverse datasets, ensuring better alignment between the chosen strategy and the inherent properties of the data.

The heatmap reveals that PoAC consistently achieves high ARI values across a wide range of datasets, indicating robust performance regardless of the dataset characteristics. This strategy particularly excels in datasets with high dimensionality, a larger number of clusters, and varying levels of imbalance. PoAC maintains superior clustering quality even in datasets with higher aspect ratios (elongated clusters), and varying ratios between the maximum and minimum cluster sizes.

In contrast, the PoAC Sil strategy shows moderate performance, with variability influenced by the number of clusters and dimensions. Notably, PoAC Sil performs better in datasets with fewer clusters and lower dimensions, as well as those with lower aspect ratios and more spherical clusters. However, its performance diminishes in datasets with high aspect ratios and greater elongation. That happens because SIL suffers from increasing dimensions since it is based on euclidean distances. So it does not handle well large dimensionalities.

PoAC DBS demonstrates wide variability in performance, struggling particularly with higher dimensional datasets and those with an increased number of clusters. The inconsistent performance of PoAC DBS is evident from the lighter color cells dispersed throughout the heatmap. PoAC DBS also performs poorly in datasets with high aspect ratios and significant elongation of clusters, as well as those with high radius values.

Lastly, PoAC CVI, while generally outperforming PoAC SIL and PoAC DBS, still shows significant variability and is less reliable compared to PoAC. Its performance is more consistent in datasets with moderate to high dimensions but falls short in achieving high ARI values across all datasets. This strategy also exhibits variability in datasets with different levels of imbalance, aspect ratios, and radii, indicating a less robust performance in comparison to PoAC.

In conclusion, while sub-configurations of PoAC such as PoAC SIL and PoAC DBS are capable of achieving high values for their respective CVIs, it is important to balance them with a surrogate model. The data indicate that there are numerous instances where

the highest SIL and DBS do not necessary correspond to the best partitioning results. For example, as shown in Table 5.8, PoAC SIL achieves the highest mean SIL (0.75) and PoAC DBS achieves the lowest mean DBS (0.27), yet their mean ARI values (0.53 and 0.41, respectively) are significantly lower than that of the complete PoAC method (0.70). This discrepancy underscores the need for a balanced approach that incorporates surrogate modeling to better represent the true clustering quality according to a particular training dataset. Moreover, the results for PoAC CVI, which operates without the meta-features group, further reinforce the necessity of meta-features to accurately represent the datasets and their optimal SIL and DBS values. The mean ARI for PoAC CVI (0.57) is higher than that for PoAC SIL and PoAC DBS but still lower than the complete PoAC method, demonstrating the critical role of meta-features in achieving superior clustering performance. Thus, the complete PoAC method, with its integration of surrogate modeling and meta-features, provides a more robust and effective approach to clustering optimization.

Table 5.8: Optimization Strategies performance regarding ARI, SIL, and DBS for the validation datasets (mean, median, and standard deviation).

Strategy	**ARI**			**SIL**			**DBS**		
	Mean	**Median**	**SD**	**Mean**	**Median**	**SD**	**Mean**	**Median**	**SD**
PoAC	**0.70**	**0.87**	**0.27**	**0.63**	**0.63**	**0.07**	**0.52**	**0.52**	**0.18**
PoAC CVI	0.58	0.57	0.25	0.47	0.46	0.05	0.79	0.83	0.15
PoAC SIL	0.63	0.98	0.45	0.76	0.77	0.07	0.30	0.27	0.18
PoAC DBS	0.11	0.00	0.29	0.61	0.58	0.09	0.30	0.31	0.06

5.3 PoAC for Anomaly Detection

This section explores the application of PoAC to anomaly detection tasks, building upon the methodology outlined in section 5.2 for clustering problems. While the overarching process remains consistent, we adapt specific components to address the unique challenges of anomaly detection: (i) we synthesize datasets and inject them with controlled levels of anomalies, (ii) we adopt a tailored set of meta-features and CVIs for feature-space mapping, (iii) we train the surrogate model on a meta-dataset specifically designed for anomaly detection, and (iv) we use this surrogate as a cheap objective to guide an optimizer toward anomaly-detection pipelines. The overall process remains the same, but the design choices reflect the specific requirements of the anomaly detection

task. We keep the description concise and emphasize the experimental choices and practical issues encountered.

5.3.1 Problem Space Design

In anomaly detection, particularly within unsupervised settings, the challenge lies in identifying data points that deviate significantly from the majority without prior labeling. This approach is prevalent due to the scarcity of labeled anomaly data in many real-world applications (Chandola et al., 2009).

To model such scenarios, we focus on *global anomalies*, also known as point anomalies (Chandola et al., 2009), which are data points that lie far outside the support of the original distribution. These are the most common targets in unsupervised anomaly detection benchmarks (Mejri et al., 2024). In contrast, *local anomalies* refer to points that deviate from their immediate neighborhood but not globally, and are often more challenging to detect without domain-specific knowledge (Agyemang, 2024).

As in the visualization case, the problem space for anomaly detection is constructed by generating a large and diverse collection of synthetic datasets. We use the `repliclust` library to synthesize multi-cluster distributions with varying dimensionality, density, overlap, and imbalance. To model anomaly detection scenarios and mimic the challenges of uncurated real-world datasets (Aqeel et al., 2025), we inject anomalies at three controlled contamination rates $1\%, 3\%, 5\%$. This three-level approach simulates varying degrees of anomaly prevalence, allowing the meta-space to capture both subtle and extreme deviations from normality.

The resulting problem space comprises thousands of synthetic datasets, each labeled with a ground-truth anomaly indicator. These datasets form the basis for extracting meta-features, computing internal validity indices, and training the surrogate model for anomaly detection pipeline optimization.

5.3.2 Feature Space Mapping

Each dataset in the problem space is mapped into a vector of descriptive statistics to enable transfer of knowledge across tasks. For anomaly detection, we adopt a tailored set of 20 meta-features extracted with `pymfe`, focusing on properties that are particularly relevant for distinguishing anomalies from normal structure. These include: measures of distributional spread and dispersion (`iq_range.sd`, `sd.mean`, `var.mean`,

kurtosis.mean), density- and distance-based descriptors (wg_dist.mean, wg_dist.sd, cohesiveness.mean), dimensionality and attribute ratios (inst_to_attr, attr_to_inst, nr_attr, nr_inst, nr_cor_attr), as well as structural statistics such as eigenvalue spectra (eigenvalues.mean, eigenvalues.sd, cov.mean, cov.sd), correlation measures (cor.mean, cor.sd), and pattern-based descriptors (t2, t3, t4, one_itemset.mean, two_itemset.mean, attr_ent.mean, attr_conc.mean). Together, these features capture density, variance structure, redundancy, and higher-order interactions that influence anomaly detectability.

In addition to meta-features, we also compute CVIs. We use SIL to measure cluster cohesion and separation. Additionally, the CHS (Caliński and Harabasz, 1974) quantifies the ratio of between-cluster variance to within-cluster variance, rewarding compact and well-separated clusters. Finally, the DBCV (Moulavi et al., 2014) evaluates the relative density connectedness within clusters against density separability between clusters, yielding values in $[-1, 1]$, where values closer to 1 indicate well-formed, density-consistent clusters and negative values suggest poor structure or excessive noise, which may reflect the presence of global anomalies (Ester et al., 1996; Chicco et al., 2025).

As in the visualization setup, ARI serves as the external target, treating anomalies as a separate class. The final meta-dataset thus combines meta-features, CVIs, and ARI scores, comprising around 4,600 instances with approximately 30 descriptors per dataset.

5.3.3 Surrogate Modeling

We train a surrogate model that predicts clustering performance given a dataset's meta-features and internal CVIs. For this purpose, we also employ a Random Forest Regressor (Breiman, 2001).

The surrogate is trained on the meta-dataset comprising 40,000 instances with approximately 30 descriptors, where the target variable is the ARI computed with respect to ground-truth anomaly labels. Model performance is assessed using cross-validation, with metrics including the coefficient of determination (R^2), RMSE, and MAE. Our experiments achieved strong predictive performance (e.g., $R^2 \approx 0.81$, RMSE ≈ 0.095, MAE ≈ 0.071), indicating that the surrogate captures the relationship between dataset characteristics and achievable ARI effectively.

Feature importance analysis indicates that both meta-features and CVIs contribute meaningfully to the surrogate's predictions. The most influential descriptors include SIL,

number of attributes, DBCV, within-group distance (wg_dist.sd), correlation measures (cor.mean, cor.sd), kurtosis, CHS, interquartile range (iq_range.sd), and attribute concentration (attr_conc.mean). The top ten features by importance are: SIL (0.3169), nr_attr (0.1237), DBCV (0.0912), wg_dist.sd (0.0641), cor.mean (0.0544), cor.sd (0.0388), kurtosis.mean (0.0358), CHS (0.0319), iq_range.sd (0.0314), and attr_conc.mean (0.0280).

Once trained, the surrogate serves as a accurate proxy for pipeline evaluation, enabling efficient search over the defined problem space and guiding the optimizer toward pipelines likely to yield high-quality anomaly detection results.

5.3.4 Function Optimization

The optimization stage follows the same pipeline search methodology described in Section 5.2, but adapted to the anomaly detection setting. We employ `TPOTClustering` as the optimizer, using the surrogate model as the scoring function to efficiently guide the search toward promising pipelines. The surrogate integrates both meta-features and CVIs, as described earlier, and predicts the expected ARI of candidate anomaly detection pipelines.

The search space is designed to include density-based clustering methods commonly used for anomaly detection, as well as standard preprocessing and dimensionality-reduction operators. Table 5.9 summarises the search space explored in our experiments.

Table 5.9: Search space for anomaly detection pipelines in PoAC.

Operator	**Hyperparameters**
`DBSCAN`	`eps` $\in \{0.1, 0.5, 1.0, 5.0\}$; `min_samples` $\in \{5, 10, 20\}$; metric = `euclidean`
`OPTICS`	`min_samples` $\in \{5, 10, 20\}$; $\xi \in \{0.01, 0.05, 0.1\}$; `min_cluster_size` $\in \{0.05, 0.1, 0.2\}$
`StandardScaler`	–
`MinMaxScaler`	–
`PCA`	`n_components` $\in \{2, 3, 5\}$

In practice, the optimizer runs for 10 generations with a population size of 30, balancing exploration of the pipeline space with computational efficiency.

For the baseline configuration, we used the `IsolationForest` implementation from `scikit-learn` with the following hyperparameters: `n_estimators=100`, `max_samples='auto'`, `max_features=1.0`, and `contamination='auto'`. We deliberately retained the default

unsupervised setting (`contamination='auto'`) to ensure methodological parity with PoAC, which likewise receives no prior information about the proportion of anomalies. This configuration allows both approaches to operate under the same information constraints, enabling a fair comparison of their intrinsic anomaly detection capabilities.

PoAC and IForest were evaluated under identical preprocessing and data-splitting conditions, using the same injected contamination levels (1%, 3%, and 5%) and evaluation metrics (Precision, Recall, F1, and ROC AUC). Each experiment was repeated ten times with different random seeds, and the reported results correspond to the mean performance across runs.

5.3.5 Results and Discussion

On the task of anomaly detection, we have evaluated PoAC against the classical IForest across eight representative datasets, chosen to capture diverse domains and anomaly characteristics. The datasets include medical and health records (*haberman, heart-statlog*), industrial monitoring (*vehicle*), image or segment recognition tasks (*segment, dermatology*), and classical benchmarks (*balance-scale, wdbc, yeast*). Figure 5.12 presents the comparative F1 scores and ROC AUC for each dataset.

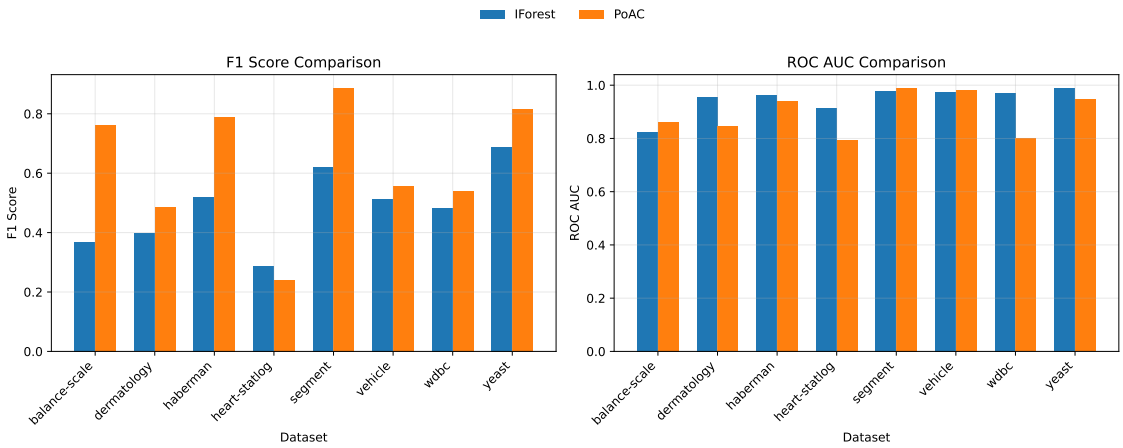


Figure 5.12: Comparison between PoAC and IForest for anomaly detection on eight representative UCI datasets. Left: F1-score comparison; Right: ROC AUC comparison. PoAC achieves higher F1 scores on most datasets — especially balance-scale, haberman, and yeast — due to improved precision–recall balance, while maintaining comparable AUC values, confirming effective generalization to anomaly detection tasks. Source: Author, reproduced from da Silva et al. (2024c).

Across these datasets, PoAC frequently achieves higher F1 scores than IForest, notably on *balance-scale, haberman, segment*, and *yeast*. This indicates that PoAC can more precisely identify anomalies in contexts where both missed anomalies and false

alarms carry significant consequences, such as in medical diagnostics or industrial quality control. In contrast, ROC AUC values are broadly similar between the two methods, suggesting comparable ability to distinguish anomalous from normal samples. PoAC's F1 advantage is most relevant when exact detection is prioritized over overall separability.

Table 5.10: Precision, Recall, and F1 score for PoAC and IForest across the eight representative datasets. PoAC generally achieves better balance between precision and recall, resulting in higher F1.

Dataset	Prec. IF	Rec. IF	F1 IF	Prec. PoAC	Rec. PoAC	F1 PoAC
balance-scale	0.302	1.000	0.369	1.000	0.722	0.762
dermatology	0.273	1.000	0.397	0.392	0.963	0.485
haberman	0.429	1.000	0.521	0.711	0.889	0.789
heart-statlog	0.187	1.000	0.286	0.173	0.897	0.241
segment	0.521	0.999	0.621	0.811	0.986	0.887
vehicle	0.377	1.000	0.514	0.402	1.000	0.556
wdbc	0.349	1.000	0.482	0.491	0.608	0.539
yeast	0.576	1.000	0.687	0.789	0.905	0.816

Analysing precision and recall explains why PoAC achieves higher F1 scores. IForest generally maintains very high recall but suffers from low precision, flagging many normal samples as anomalies. PoAC balances precision and recall more effectively, reducing false positives while retaining strong detection of true anomalies. For instance, on *balance-scale*, IForest achieves recall of 1.0 but precision of 0.30 (F1 = 0.37), whereas PoAC improves precision to 1.0 with only a moderate drop in recall (0.72), resulting in F1 = 0.76. Similar patterns are observed across other datasets, highlighting PoAC's practical advantage in applications such as medical diagnostics, industrial quality control, and fraud detection, where accurately identifying anomalies while minimizing false alarms is critical.

5.4 Limitations

The principal limitations of this study are associated with the selection of CVIs and the composition of the datasets employed during the training phase. The choice of CVIs is pivotal to the quality of clustering outcomes, as their effectiveness directly determines how accurately the framework can evaluate and guide the partitioning process. Consequently, the alignment between the selected CVIs and the characteristics of the clustering problem is of utmost importance. A higher degree of correlation between the CVIs and the intrinsic properties of the data, such as its structure or visual separability, tends to

yield more meaningful and effective optimization results.

Moreover, the performance of PoAC is influenced by the representativeness and diversity of the datasets used to train the surrogate model. As evidenced by the experimental results, PoAC exhibits improved performance when it learns clustering strategies from datasets that share structural or statistical similarities with the target dataset. Thus, the careful curation of training datasets is crucial to ensure adequate coverage of the problem space and to promote generalization across heterogeneous clustering tasks.

Another limitation concerns the trade-off between optimization time and solution quality. Longer optimization horizons can potentially yield better-performing pipelines but at the cost of increased computational effort. Additionally, the informativeness of the meta-features significantly impacts the predictive accuracy of the surrogate model; insufficiently descriptive or poorly chosen features may lead to suboptimal pipeline recommendations. Since PoAC allows users to define problem-specific CVIs and meta-features, a certain degree of domain expertise is beneficial to appropriately tailor the optimization objective to the intended analytical goal.

Finally, while the present work evaluated PoAC in the contexts of visualization and anomaly detection, its modular architecture suggests that the framework can be extended to a broader range of unsupervised learning scenarios. Such flexibility represents both an opportunity for further development and a limitation of the current experimental scope.

5.4.1 Conclusion

The PoAC framework introduces a modular and extensible paradigm for automating clustering tasks, addressing key challenges inherent to conventional unsupervised AutoML systems. By leveraging a meta-knowledge base constructed from prior clustering datasets and their corresponding solutions, PoAC employs a surrogate model capable of inferring the quality of novel clustering pipelines without the need for retraining or additional data ingestion. This design enables the framework to generalize across distinct unsupervised learning scenarios while maintaining a unified methodological foundation.

A defining characteristic of PoAC is its capacity to formalize dynamic relationships among clustering problems, CVIs, and meta-features at the framework level. This capability allows for task-specific instantiations of PoAC, wherein users can define suitable meta-features, optimization objectives, and evaluation metrics to address applications such as visualization or anomaly detection. Once instantiated, each configuration oper-

ates with fixed internal components, ensuring both reproducibility and adaptability across problem domains. Furthermore, PoAC remains algorithm-agnostic and independent of specific optimization strategies, facilitating seamless integration with diverse clustering algorithms and preprocessing techniques.

Empirical evaluations demonstrated that PoAC achieves superior performance relative to state-of-the-art AutoML frameworks across a variety of datasets. In visualization-oriented experiments, PoAC consistently produced higher clustering validity scores, while in the anomaly detection scenario, it outperformed the IForest baseline in terms of F1 score and achieved competitive ROC AUC results. These outcomes validate the robustness and generality of PoAC as a unified AutoML solution for unsupervised learning.

Future research will focus on extending PoAC's applicability to additional problem domains and enhancing its scalability and resilience. In particular, forthcoming work aims to improve computational efficiency for large-scale and high-dimensional datasets and to integrate mechanisms for adaptive feature selection and robust noise handling. Such advancements will further strengthen PoAC's potential as a versatile and interpretable framework for context-aware clustering across diverse real-world applications.

Chapter 6

Discussions

Contents

6.1 Synthesis of Findings Across Studies

This thesis explored how the principles of AutoML can be extended to unsupervised learning through the joint lenses of pipeline synthesis, meta-learning, and surrogate-based optimization. Across its three main contributions, *TPOT-Clustering*, the meta-learning analysis, and the *PoAC* framework, a recurring theme emerged: effective automation in clustering depends not merely on searching the design space of algorithms and hyperparameters, but on understanding the relationship between problems, objectives, and representations.

In the first stage, **TPOT-Clustering** demonstrated that evolutionary pipeline synthesis can be successfully adapted to unsupervised settings. The experiments revealed that internal validation indices CVIs provide limited but consistent guidance, especially when their assumptions align with the underlying data structure. However, they also highlighted that single-CVI optimization tends to overfit to particular structural notions of cluster quality, compactness, separation, or density, yielding pipelines that perform well numerically but may fail to reflect the analytical intent of the user.

The second stage, a systematic review and empirical analysis of **meta-learning in AutoClustering**, extended this understanding by examining how clustering problems can be represented and generalized through meta-features. The findings showed that a relatively small subset of statistical and landmarking meta-features dominates model performance, while more complex descriptors contribute marginally despite higher computational cost. This result suggests that current AutoClustering systems overparameterize their meta-spaces, often without proportional gains in generalization. At the same time, the study confirmed that meta-learning offers a powerful means to transfer inductive knowledge across tasks, an ability essential for scaling unsupervised AutoML beyond fixed metrics.

Building on these insights, the **PoAC** framework integrated pipeline synthesis and meta-learning into a unified surrogate-driven architecture. PoAC reframes clustering automation as a problem-oriented process: rather than optimizing a single internal metric, it learns meta-objectives that approximate user-defined notions of quality. The framework demonstrated that surrogate-based optimization can generalize across diverse unsupervised scenarios, including visualization and anomaly detection, producing competitive or superior results to existing AutoClustering systems while offering greater interpretability and flexibility.

Together, these studies illustrate a gradual methodological shift—from static, metric-centric AutoML toward adaptive, knowledge-informed systems capable of reasoning about the problem itself. This synthesis underscores the importance of meta-knowledge not as an auxiliary component but as a core mechanism for guiding automated decision-making in unsupervised learning.

6.2 Answers to the Research Questions

This section explicitly revisits the research questions introduced in Chapter 1, summarizing how each was addressed and what conclusions can be drawn.

RQ1: How can pipeline synthesis methods be adapted to support flexible and data-driven optimization in clustering, beyond fixed internal validation metrics?

This question was primarily addressed through the development and evaluation of **TPOT-Clustering** (Chapter 3). The experiments demonstrated that evolutionary pipeline synthesis can effectively be extended to unsupervised learning by incorporating inter-

nal CVIs or surrogate objectives as optimization criteria. While single-CVI optimization offers limited guidance, the integration of surrogate-based objectives, trained on meta-knowledge of prior clustering performance, enables more flexible and data-driven search behaviour. This adaptation allows AutoML systems to explore richer pipeline configurations aligned with varying data structures and problem contexts, rather than being constrained by static notions of clustering quality.

RQ2: Which meta-features and meta-learning strategies are most effective for characterizing clustering problems and predicting algorithmic performance?

This question was investigated in the systematic review and empirical study on **meta-learning in AutoClustering** (Chapter 4). The analysis revealed that a compact set of statistical and landmarking meta-features provide the strongest predictive power for surrogate models, while complex or highly correlated descriptors contribute little additional value. Moreover, meta-learning strategies that combine feature relevance analysis and surrogate interpretability techniques were found to yield the most consistent generalization across unseen datasets. These findings indicate that effective meta-representations in AutoClustering should prioritize parsimony, robustness, and interpretability over sheer dimensionality.

RQ3: How can surrogate models and meta-objectives be combined to create a problem-oriented AutoML framework capable of generating clustering pipelines aligned with user intent?

This question was addressed through the design and evaluation of the **PoAC framework** (Chapter 5). PoAC integrates surrogate models with meta-objectives that approximate user-defined quality functions, allowing clustering pipelines to be optimized according to contextual goals such as visualization clarity or anomaly detection accuracy. Empirical results confirmed that this surrogate-guided, problem-oriented approach improves both adaptability and interpretability relative to conventional AutoClustering systems. Thus, combining meta-learning with surrogate optimization provides a viable path toward AutoML systems that align pipeline synthesis with user intent and problem semantics.

6.3 Theoretical and Practical Implications

From a theoretical standpoint, this research advances the view of AutoML as a two-level learning process: one that operates not only on model hyperparameters but also on the representations of problems and objectives. By formalizing the relationships between datasets, meta-features, and clustering outcomes, the thesis contributes to a conceptual framework for *problem representation in AutoML*. In this view, optimization is no longer bound to explicit metrics but guided by learned surrogates that encode prior experience and user-defined goals.

Practically, this shift has several implications. First, it enables AutoML systems to move closer to human-like reasoning, leveraging experience from past tasks to guide new ones with limited feedback. Second, it introduces a level of transparency and control absent from purely data-driven optimization: users can specify or learn what "good performance" means for their context, leading to more interpretable and task-aligned results. Finally, the modular and surrogate-based design of PoAC offers a blueprint for future AutoML systems that must operate in complex, label-scarce environments, such as exploratory data analysis, anomaly detection, or representation learning.

6.4 AutoML Design for Subjective Tasks

One of the central insights emerging from this work is that automation in subjective tasks, such as clustering, cannot rely on objective metrics alone. In supervised learning, optimization targets are clear, but in clustering, these targets are ill-defined and context-dependent. The notion of "good clustering" varies with the purpose: visualization, segmentation, noise reduction, or pattern discovery may each demand different structural properties.

The results of TPOT-Clustering and PoAC show that AutoML systems must therefore incorporate mechanisms for *subjective adaptability*—the ability to interpret, approximate, or learn user intent. Surrogate models and meta-objectives provide one pathway toward this goal, enabling the system to learn mappings between meta-features and quality assessments that implicitly encode human or domain-driven preferences. This does not fully eliminate the need for human judgment but reframes it as part of a collaborative, interpretable optimization loop.

Such adaptability also raises design questions for future AutoML systems: how should

user objectives be specified, learned, or adjusted over time? How can surrogate models remain interpretable and robust while representing subjective notions of quality? Addressing these questions requires blending algorithmic efficiency with principles from human-computer interaction and explainable AI.

6.5 The Role of User Intent and Problem Orientation

A recurring theme across all chapters is the importance of aligning automation with the user's analytical intent. In traditional AutoML, the user defines a task (e.g., classification) and the system optimizes a fixed metric (e.g., accuracy). In unsupervised learning, however, the "task" itself must be inferred or co-defined. The concept of *problem orientation* introduced in this thesis captures this distinction: an AutoML system should adapt not only to data but to the *purpose* of analysis.

PoAC operationalizes this principle through its meta-objective mechanism, which links clustering problems, meta-features, and validation criteria under a unified optimization framework. By learning to predict outcomes that align with user-specified goals—whether interpretability, visual separability, or robustness—PoAC represents a step toward AutoML systems that reason about *why* a model is good, not only *how* to find it. This orientation transforms AutoML from a purely algorithmic process into a semi-cognitive one, where optimization and understanding evolve together.

6.6 Lessons for Future AutoClustering Research

Several lessons emerge from this research for the design and evaluation of future AutoClustering frameworks. First, effective automation in unsupervised learning requires the integration of complementary paradigms: search-based pipeline synthesis ensures flexibility, while meta-learning and surrogate modeling provide efficiency and guidance. Systems that combine these elements can balance generalization and specialization more effectively than those relying on fixed metrics or isolated optimization routines.

Second, the construction of meta-knowledge bases remains a critical bottleneck. The quality, diversity, and representativeness of the datasets used to train surrogate models directly determine the adaptability of AutoClustering frameworks. Standardized repositories and benchmarks for unsupervised meta-learning would therefore play an important role in advancing reproducibility and comparability across systems.

Third, the evaluation of AutoClustering methods should expand beyond internal CVIs toward problem-oriented metrics that better capture user intent. The inclusion of qualitative or task-specific evaluation—such as alignment with human perception or downstream utility—can provide a more holistic assessment of clustering automation.

6.7 Emerging Opportunities and Open Research Directions

Beyond the limitations discussed above, the findings of this thesis open several promising avenues for further investigation in AutoClustering and problem-oriented AutoML more broadly. These opportunities extend across methodological, representational, and human-centered dimensions.

Learning Problem Representations End-to-End. A central assumption throughout this work is that clustering problems can be characterized through hand-crafted meta-features. While effective, this approach relies on predefined statistical and structural descriptors. An important direction for future research lies in learning problem representations automatically, for instance through graph-based encodings, neural embeddings of datasets, or representation learning over distance matrices. Such approaches could reduce dependence on manual feature engineering and allow surrogate models to capture richer, task-specific structure in clustering problems.

Dynamic and Continual Meta-Learning. Most existing AutoClustering frameworks, including PoAC, rely on static meta-knowledge bases constructed offline. Future systems could instead adopt continual or lifelong meta-learning paradigms, where surrogate models are incrementally updated as new datasets, user feedback, or objectives become available. This would enable AutoML systems to adapt over time, improving robustness and relevance in evolving analytical environments.

Interactive and Human-in-the-Loop AutoClustering. While this thesis emphasizes alignment with user intent, the interaction between users and AutoML systems remains largely implicit. A promising research direction involves integrating explicit human-in-the-loop mechanisms, such as interactive refinement of meta-objectives, active querying for constraints or preferences, or visual analytics interfaces for steering optimization. Combining surrogate-based optimization with user feedback could further bridge the gap

between automated search and exploratory data analysis.

Evaluation Beyond Internal Validation Indices. The reliance on CVIs, even when mediated through meta-objectives, remains a fundamental limitation of unsupervised learning. Future research could explore evaluation frameworks that incorporate downstream utility, perceptual metrics, or task-specific outcomes more directly. For example, clustering quality could be assessed through its impact on subsequent learning tasks, decision-making processes, or human interpretability, enabling more holistic and application-aware AutoClustering systems.

Cross-Domain and Multi-Objective AutoClustering. Finally, problem-oriented AutoML naturally extends to multi-objective and cross-domain settings, where clustering pipelines must balance competing criteria such as interpretability, stability, and computational cost. Investigating surrogate models capable of handling such trade-offs, as well as transferring knowledge across domains with different data modalities, represents a key challenge for future AutoClustering research.

Together, these directions suggest that the next generation of AutoClustering systems will move beyond static automation toward adaptive, interactive, and continuously learning frameworks. By grounding optimization in problem representations and user intent, future research can further transform AutoML from a tool for algorithm selection into a partner for exploratory and subjective data analysis.

6.8 Limitations

Despite its contributions, this work has several limitations that point to opportunities for refinement. First, the scalability of surrogate-based optimization remains constrained by the computational cost of building and maintaining large meta-knowledge bases. While PoAC demonstrates that these costs can be amortized over time, they still represent a barrier to real-time or large-scale applications.

Second, the performance of the surrogate model depends heavily on the quality and diversity of the selected meta-features. As shown in the meta-learning study, not all features contribute equally to predictive power, and redundancy or dataset bias can reduce generalization. Automatic or learned meta-feature extraction could provide a promising path forward, but remains an open challenge.

Finally, interpretability remains a delicate balance. Although surrogate-based AutoML allows for more transparent reasoning than black-box search, the mapping between meta-features, objectives, and clustering performance can still be difficult to explain. Developing clearer visualization and interaction mechanisms for these mappings would enhance the usability of problem-oriented AutoML in practical settings.

Overall, these limitations reflect the complexity of automating subjective, label-scarce learning tasks. Yet they also highlight the central insight of this thesis: advancing AutoML requires not only better optimization algorithms but deeper representations of problems, objectives, and user intent. By reframing automation as a problem-oriented process, this work lays the groundwork for more adaptive, interpretable, and human-aligned systems in unsupervised machine learning.

Chapter 7

Conclusion

Contents

7.1 Summary of Contributions

This thesis has investigated the automation of unsupervised learning, with a particular focus on clustering, through the integration of pipeline synthesis, meta-learning, and surrogate-based optimization. The key contributions can be summarized as follows:

1. **Meta-learning in AutoClustering:** We conducted a comprehensive review and empirical analysis of existing AutoClustering frameworks, revealing patterns in meta-feature usage, dataset biases, and evaluation strategies. This study clarified which meta-features contribute most to predictive performance and highlighted the importance of principled meta-space construction for effective knowledge transfer across clustering tasks.

2. **TPOT-Clustering:** We extended the TPOT framework to unsupervised learning, demonstrating that evolutionary pipeline synthesis can be adapted to clustering tasks. TPOT-Clustering supports both conventional CVI-based and surrogate-based objectives, enabling automated pipeline generation tailored to user-defined or domain-specific evaluation criteria.

3. **PoAC: Problem-oriented AutoML for Clustering:** Building on insights from meta-learning and pipeline synthesis, we proposed the PoAC framework, which unifies surrogate-based optimization with problem-oriented meta-objectives. PoAC allows AutoClustering systems to align pipeline synthesis with user intent, adapt-

ing automatically to diverse unsupervised tasks such as visualization and anomaly detection.

7.2 Key Insights

Across these contributions, several overarching insights emerged:

- Effective AutoML for clustering requires moving beyond fixed metrics toward context-aware, problem-oriented optimization.
- Meta-learning plays a critical role in enabling transfer of knowledge across datasets, but careful design of the meta-space is essential to ensure stability, interpretability, and generalization.
- Surrogate models provide a practical mechanism for approximating complex clustering objectives, allowing the system to adapt dynamically to new tasks without exhaustive evaluation.
- Combining pipeline synthesis with meta-learning enables a more flexible, adaptive, and interpretable approach to unsupervised automation, bridging the gap between algorithmic search and human intent.

7.3 Concluding Remarks

By integrating these principles, this thesis advances the state of AutoClustering from a metric-driven process toward a knowledge-informed, problem-oriented paradigm. The proposed frameworks demonstrate that unsupervised automation can be both flexible and interpretable, capable of generating pipelines that not only perform well according to numerical criteria but also reflect the objectives and priorities of the user.

Ultimately, this work contributes to the broader vision of AutoML as a tool for reasoning about problems themselves—learning not just how to optimize models, but how to understand, represent, and adapt to the structure and intent of the task at hand. The combination of pipeline synthesis, meta-learning, and surrogate modeling provides a foundation for future systems that are more adaptive, human-aligned, and capable of addressing the intrinsic subjectivity of unsupervised learning.

Appendix A

Appendix: Publications and Scientific Output During the PhD

A.1 Publications and Scientific Output During the PhD

During the PhD programme, several research works were developed in collaboration with colleagues and supervisors, resulting in publications in international journals and conferences. The following list summarizes the main scientific outputs produced throughout the doctoral period:

1. **Close to Reality: Interpretable and Feasible Data Augmentation for Imbalanced Learning.**
   M.C. da Silva, G.G. Costanzo, A. De Lorenzo, S. Barbon Junior.
   Under review at Transactions of Machine Learning, 2025.

2. **TPOT-Clustering.**
   M.C. da Silva, G.M. Tavares, S. Barbon Junior.
   International Journal of Neural Networks (IJCNN), 2025.

3. **Meta-learning Approach for Variational Autoencoder Hyperparameter Tuning.**
   M. Berti, M.C. da Silva, S. Saccani, S. Barbon Junior.
   Journal of Universal Computer Science, Vol. 31, No. 7, pp. 668–678, 2025.

4. **Problem-oriented AutoML in Clustering.**
   M.C. da Silva, G.M. Tavares, E. Medvet, S. Barbon Junior.
   arXiv preprint arXiv:2409.16218, under review at *Springer Machine Learning*, 2024.

5. **Benchmarking AutoML Clustering Frameworks.**
   M.C. da Silva, B. Licari, G. Marques Tavares, S. Barbon Junior.
   AutoML Conference 2024 (ABCD Track), Paris, September 2024.

6. **Automated Trace Clustering Pipeline Synthesis in Process Mining.**

I.M. Grigore, G.M. Tavares, M.C. da Silva, P. Ceravolo, S. Barbon Junior. *Information*, Vol. 15, No. 4, p. 241, 2024.

7. **Using Process Mining to Reduce Fraud in Digital Onboarding.**
   M.C. da Silva, G.M. Tavares, M.C. Gritti, P. Ceravolo, S. Barbon Junior. *FinTech*, Vol. 2, No. 1, pp. 120–137, 2023.

Bibliography

Antoine Adam and Hendrik Blockeel. Dealing with overlapping clustering: A constraint-based approach to algorithm selection. In *Meta-learning and Algorithm Selection workshop-ECMLPKDD2015*, volume 1, pages 43–54. CEUR Workshop proceedings, 2015.

Edmund Fosu Agyemang. Anomaly detection using unsupervised machine learning algorithms: A simulation study. *Scientific African*, 26:e02386, 2024. ISSN 2468-2276. doi: https://doi.org/10.1016/j.sciaf.2024.e02386. URL https://www.sciencedirect.com/science/article/pii/S2468227624003284.

Khalid Al-Jabery, Tayo Obafemi-Ajayi, Gayla Olbricht, and Donald Wunsch. Computational learning approaches to data analytics in biomedical applications, 2019.

Edesio Alcobaça, Felipe Siqueira, Adriano Rivolli, Luís P. F. Garcia, Jefferson T. Oliva, and André C. P. L. F. de Carvalho. Mfe: Towards reproducible meta-feature extraction. *Journal of Machine Learning Research*, 21(111):1–5, 2020. URL http://jmlr.org/papers/v21/19-348.html.

Reza Alizadeh, Janet K Allen, and Farrokh Mistree. Managing computational complexity using surrogate models: a critical review. *Research in Engineering Design*, 31(3):275–298, 2020.

Mihael Ankerst, Markus M Breunig, Hans-Peter Kriegel, and Jörg Sander. Optics: Ordering points to identify the clustering structure. *ACM Sigmod record*, 28(2):49–60, 1999.

Muhammad Aqeel, Shakiba Sharifi, Marco Cristani, and Francesco Setti. Towards real unsupervised anomaly detection via confident meta-learning. *arXiv preprint arXiv:2508.02293*, 2025.

Leonardo Arrighi, Luca Pennella, Gabriel Marques Tavares, and Sylvio Barbon Junior. Decision predicate graphs: Enhancing interpretability in tree ensembles. In *World Conference on Explainable Artificial Intelligence*, pages 311–332. Springer, 2024.

Ira Assent. Clustering high dimensional data. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2(4):340–350, 2012.

Arthur Asuncion, David Newman, et al. Uci machine learning repository, 2007.

Adil M. Bagirov, Ramiz M. Aliguliyev, and Nargiz Sultanova. Finding compact and well-separated clusters: Clustering using silhouette coefficients. *Pattern Recognition*, 135:109144, 2023. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2022.109144. URL https://www.sciencedirect.com/science/article/pii/S0031320322006239.

Maroua Bahri, Flavia Salutari, Andrian Putina, and Mauro Sozio. Automl: state of the art with a focus on anomaly detection, challenges, and research directions. *International Journal of Data Science and Analytics*, 14(2):113–126, 2022.

Tomas Barton. Clustering benchmarks, 2015.

James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *The journal of machine learning research*, 13(1):281–305, 2012.

Laurens Bliek. A survey on sustainable surrogate-based optimisation. *Sustainability*, 14 (7), 2022. ISSN 2071-1050. doi: 10.3390/su14073867. URL https://www.mdpi.com/2071-1050/14/7/3867.

Pavel Brazdil, Christophe Giraud-Carrier, Carlos Soares, and Ricardo Vilalta. *Metalearning: Applications to Data Mining*. Springer, 2008.

Pavel Brazdil, Jan N. van Rijn, Carlos Soares, and Joaquin Vanschoren. Metalearning: Applications to automated machine learning and data mining, 2022. URL https://doi.org/10.1007/978-3-030-67024-5.

Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, Oct 2001. ISSN 1573-0565. doi: 10.1023/A:1010933404324. URL https://doi.org/10.1023/A:1010933404324.

Tadeusz Caliński and Jerzy Harabasz. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1):1–27, 1974.

Romain Cassier, Isabelle Guyon, Patryk Orzechowski, and Ben Wood. Autoclust: Automated pipeline design for unsupervised learning. In *NeurIPS AutoML Workshop*, 2022.

Ciro Castiello, Giovanna Castellano, and Anna Maria Fanelli. Meta-data: Characterization of input features for meta-learning. In *International conference on modeling decisions for artificial intelligence*, pages 457–468. Springer, 2005.

Gerlise Chan, Tom Claassen, Holger Hoos, Tom Heskes, and Mitra Baratchi. Autocd: Automated machine learning for causal discovery algorithms. In *AutoML 2024 Methods Track*, 2024.

Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):1–58, 2009.

Angelos Chatzimparmpas, Rafael M Martins, Ilir Jusufi, and Andreas Kerren. A survey of surveys on the use of visualization for interpreting machine learning models. *Information Visualization*, 19(3):207–233, 2020.

Davide Chicco, Giuseppe Sabino, Luca Oneto, and Giuseppe Jurman. The dbcv index is more informative than dcsi, cdbw, and viasckde indices for unsupervised clustering internal assessment of concave-shaped and density-based clusters. *PeerJ Computer Science*, 11:e3095, 2025.

Noy Cohen-Shapira and Lior Rokach. Automatic selection of clustering algorithms using supervised graph embedding. *Information Sciences*, 577:824–851, 2021a. ISSN 0020-0255. doi: https://doi.org/10.1016/j.ins.2021.08.028. URL https://www.sciencedirect.com/science/article/pii/S0020025521008288.

Noy Cohen-Shapira and Lior Rokach. Automatic selection of clustering algorithms using supervised graph embedding. *Information Sciences*, 577:824–851, 2021b. ISSN 0020-0255. doi: https://doi.org/10.1016/j.ins.2021.08.028. URL https://www.sciencedirect.com/science/article/pii/S0020025521008288.

Alison Cozad, Nikolaos V Sahinidis, and David C Miller. Learning surrogate models for simulation-based optimization. *AIChE Journal*, 60(6):2211–2227, 2014.

Matheus Camilo da Silva, Biagio Licari, Gabriel Marques Tavares, and Sylvio Barbon Junior. Benchmarking automl clustering frameworks. In *AutoML Conference 2024 (ABCD Track)*, 2024a.

Matheus Camilo da Silva, Biagio Licari, Gabriel Marques Tavares, and Sylvio Barbon Junior. Benchmarking autoML clustering frameworks. In *AutoML Conference 2024 (ABCD Track)*, 2024b. URL https://openreview.net/forum?id=RzUKJnph1g.

Matheus Camilo da Silva, Gabriel Marques Tavares, Eric Medvet, and Sylvio Barbon Junior. Problem-oriented automl in clustering. *arXiv preprint arXiv:2409.16218*, 2024c.

Matheus Camilo da Silva, Gabriel Marques Tavares, and Sylvio Barbon. Tpot-clustering. In *2025 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2025.

Siva Krishna Dasari, Abbas Cheddad, and Petter Andersson. Random forest surrogate models to support design space exploration in aerospace use-case. In *Artificial Intelligence Applications and Innovations: 15th IFIP WG 12.5 International Conference, AIAI 2019, Hersonissos, Crete, Greece, May 24–26, 2019, Proceedings 15*, pages 532–544. Springer, 2019.

David L Davies and Donald W Bouldin. A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence*, 1(2):224–227, 1979.

Marcilio C. P. de Souto, Ricardo B. C. Prudencio, Rodrigo G. F. Soares, Daniel S. A. de Araujo, Ivan G. Costa, Teresa B. Ludermir, and Alexander Schliep. Ranking and selecting clustering algorithms using a meta-learning approach. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pages 3729–3735, 2008. doi: 10.1109/IJCNN.2008.4634333.

Marcilio CP De Souto, Ivan G Costa, Daniel Sa De Araujo, Teresa B Ludermir, and Alexander Schliep. Clustering cancer gene expression data: a comparative study. *BMC bioinformatics*, 9:1–14, 2008.

Andrzej Dudek. Silhouette index as clustering evaluation tool. In *Conference of the section on classification and data analysis of the polish statistical association*, pages 19–33. Springer, 2019.

Katharina Eggensperger, Marius Lindauer, Holger H Hoos, Frank Hutter, and Kevin Leyton-Brown. Efficient benchmarking of algorithm configurators via model-based surrogates. *Machine Learning*, 107:15–41, 2018.

Radwa ElShawi and Sherif Sakr. csmartml-glassbox: Increasing transparency and controllability in automated clustering. In *2022 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 47–54. IEEE, 2022a.

Radwa ElShawi and Sherif Sakr. Tpe-autoclust: A tree-based pipline ensemble framework for automated clustering. In *2022 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 1144–1153, 2022b. doi: 10.1109/ICDMW58026.2022.00149.

Radwa ElShawi, Hudson Lekunze, and Sherif Sakr. csmartml: A meta learning-based framework for automated selection and hyperparameter tuning for clustering. In *2021 IEEE International Conference on Big Data (Big Data)*, pages 1119–1126, 2021. doi: 10.1109/BigData52589.2021.9671542.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231, 1996.

Absalom E Ezugwu, Abiodun M Ikotun, Olaide O Oyelade, Laith Abualigah, Jeffery O Agushaka, Christopher I Eke, and Andronicus A Akinyelu. A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects. *Engineering Applications of Artificial Intelligence*, 110:104743, 2022.

Luiz Henrique dos Santos Fernandes, Ana Carolina Lorena, and Kate Smith-Miles. Towards understanding clustering problems and algorithms: An instance space analysis. *Algorithms*, 14(3), 2021. ISSN 1999-4893. doi: 10.3390/a14030095. URL https://www.mdpi.com/1999-4893/14/3/95.

Daniel G. Ferrari and Leandro Nunes de Castro. Clustering algorithm recommendation: A meta-learning approach. In Bijaya Ketan Panigrahi, Swagatam Das, Ponnuthurai Nagaratnam Suganthan, and Pradipta Kumar Nanda, editors, *Swarm, Evolutionary, and Memetic Computing*, pages 143–150, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. ISBN 978-3-642-35380-2.

Daniel Gomes Ferrari and Leandro Nunes de Castro. Clustering algorithm selection by meta-learning systems: A new distance-based problem characterization and ranking combination methods. *Information Sciences*, 301:181–194, 2015. ISSN 0020-0255. doi: https://doi.org/10.1016/j.ins.2014.12.044. URL https://www.sciencedirect.com/science/article/pii/S0020025514011967.

Andrew Frank. Uci machine learning repository. *http://archive. ics. uci. edu/ml*, 2010.

Pasi Fränti and Sami Sieranoja. K-means properties on six clustering benchmark datasets, 2018. URL http://cs.uef.fi/sipu/datasets/.

Nicolo Fusi, Rishit Sheth, and Nello Cristianini. Probabilistic matrix factorization for automated machine learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Itay Gabbay, Bracha Shapira, and Lior Rokach. Isolation forests and landmarking-based representations for clustering algorithm recommendation using meta-learning. *Information Sciences*, 574:473–489, 2021a. ISSN 0020-0255. doi: https://doi.org/10.1016/j.ins.2021.06.033. URL https://www.sciencedirect.com/science/article/pii/S0020025521006241.

Itay Gabbay, Bracha Shapira, and Lior Rokach. Isolation forests and landmarking-based representations for clustering algorithm recommendation using meta-learning. *Information Sciences*, 574:473–489, 2021b. ISSN 0020-0255. doi: https://doi.org/10.1016/j.ins.2021.06.033. URL https://www.sciencedirect.com/science/article/pii/S0020025521006241.

Robert B Gramacy. *Surrogates: Gaussian process modeling, design, and optimization for the applied sciences*. Chapman and Hall/CRC, 2020.

Isabelle Guyon, Kristin Bennett, Gavin Cawley, Hugo Jair Escalante, Sergio Escalera, Tin Kam Ho, Núria Macià, Bisakha Ray, Mehreen Saeed, Alexander Statnikov, and Evelyne Viegas. Design of the 2015 chalearn automl challenge. In *2015 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2015. doi: 10.1109/IJCNN.2015.7280767.

Maria Halkidi, Yannis Batistakis, and Michalis Vazirgiannis. On clustering validation techniques. *Journal of intelligent information systems*, 17:107–145, 2001.

Zhong-Hua Han, Ke-Shi Zhang, et al. Surrogate-based optimization. *Real-world applications of genetic algorithms*, 343:343–362, 2012.

Julia Handl and Joshua Knowles. Cluster generators for large high-dimensional data sets with large numbers of clusters. *Dimension*, 2:20, 2005.

Christian Hennig. What are the true clusters? *Pattern Recognition Letters*, 64:53–62, 2015. ISSN 0167-8655. doi: https://doi.org/10.1016/j.patrec.2015.04.009. URL https://www.sciencedirect.com/science/article/pii/S0167865515001269. Philosophical Aspects of Pattern Recognition.

Tin Kam Ho and M. Basu. Complexity measures of supervised classification problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3):289–300, 2002. doi: 10.1109/34.990132.

Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. Meta-learning in neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):5149–5169, 2022.

Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of classification*, 2: 193–218, 1985.

Frank Hutter, Lars Kotthoff, and Joaquin Vanschoren. Automated machine learning: methods, systems, challenges, 2019.

Nilsel Ilter and H. Guvenir. Dermatology. UCI Machine Learning Repository, 1998. DOI: https://doi.org/10.24432/C5FK5P.

Anil K Jain, M Narasimha Murty, and Patrick J Flynn. Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323, 1999.

Ping Jiang, Qi Zhou, Xinyu Shao, Ping Jiang, Qi Zhou, and Xinyu Shao. *Surrogate-model-based design and optimization*. Springer, 2020.

Alexandros Kalousis and Melanie Hilario. Model selection via meta-learning: a comparative study. In *Proceedings 12th IEEE Internationals Conference on Tools with Artificial Intelligence. ICTAI 2000*, pages 406–413. IEEE, 2000.

Leonard Kaufman. Partitioning around medoids (program pam). *Wiley series in probability and statistics*, 344:68–125, 1990.

Christian Lemke, Bogdan Gabrys, and Joachim M Buhmann. Metalearning: a survey of trends and technologies. *Artificial Intelligence Review*, 44(1):117–130, 2015.

Huapeng Li, Shuqing Zhang, Xiaohui Ding, Ce Zhang, and Patricia Dale. Performance evaluation of cluster validity indices (cvis) on multi/hyperspectral remote sensing datasets. *Remote Sensing*, 8(4):295, 2016.

Marius Lindauer, Katharina Eggensperger, Matthias Feurer, André Biedenkapp, Difan Deng, Carolin Benjamins, Tim Ruhkopf, René Sass, and Frank Hutter. Smac3: A versatile bayesian optimization package for hyperparameter optimization. *Journal of Machine Learning Research*, 23(54):1–9, 2022.

Yue Liu, Shuang Li, and Wenjie Tian. Autocluster: Meta-learning based ensemble method for automated unsupervised clustering. In Kamal Karlapalem, Hong Cheng, Naren Ramakrishnan, R. K. Agrawal, P. Krishna Reddy, Jaideep Srivastava, and Tanmoy Chakraborty, editors, *Advances in Knowledge Discovery and Data Mining*, pages 246–258, Cham, 2021. Springer International Publishing. ISBN 978-3-030-75768-7.

Raphael Gontijo Lopes, Dong Yin, Ben Poole, Justin Gilmer, and Ekin D Cubuk. Improving robustness without sacrificing accuracy with patch gaussian augmentation. *arXiv preprint arXiv:1906.02611*, 2019.

Ana C. Lorena, Luís P. F. Garcia, Jens Lehmann, Marcilio C. P. Souto, and Tin Kam Ho. How complex is your classification problem? a survey on measuring classification complexity. *ACM Comput. Surv.*, 52(5), sep 2019a. ISSN 0360-0300. doi: 10.1145/3347711. URL https://doi.org/10.1145/3347711.

Ana C Lorena, Luís PF Garcia, Jens Lehmann, Marcilio CP Souto, and Tin Kam Ho. How complex is your classification problem? a survey on measuring classification complexity. *ACM Computing Surveys (CSUR)*, 52(5):1–34, 2019b.

Kolby Nottingham Markelle Kelly, Rachel Longjohn. Uci machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

Ujjwal Maulik and Sanghamitra Bandyopadhyay. Performance evaluation of some clustering algorithms and validity indices. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24:1650–1654, 2002. URL https://api.semanticscholar.org/CorpusID:2453418.

Miles NF McCrory and Spencer A Thomas. Cluster metric sensitivity to irrelevant features. In *Computational Problems in Science and Engineering II*, pages 85–95. Springer, 2025.

Geoffrey J McLachlan and David Peel. *Finite mixture models*. John Wiley & Sons, 2000.

James B McQueen. Some methods of classification and analysis of multivariate observations. In *Proc. of 5th Berkeley Symposium on Math. Stat. and Prob.*, pages 281–297, 1967.

Nesryne Mejri, Laura Lopez-Fuentes, Kankana Roy, Pavel Chernakov, Enjie Ghorbel, and Djamila Aouada. Unsupervised anomaly detection in time-series: An extensive evaluation and analysis of state-of-the-art methods. *Expert Systems with Applications*, 256:124922, 2024.

Donald Michie, David J Spiegelhalter, Charles C Taylor, and John Campbell. *Machine learning, neural and statistical classification*. Ellis Horwood, 1995.

Siddhartha Mishra, Nicholas Monath, Michael Boratko, Ariel Kobren, and Andrew McCallum. An evaluative measure of clustering methods incorporating hyperparameter sensitivity. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(7):7788–7796, Jun. 2022. doi: 10.1609/aaai.v36i7.20747. URL https://ojs.aaai.org/index.php/AAAI/article/view/20747.

Richard Mojena. Hierarchical grouping methods and stopping rules: an evaluation. *The Computer Journal*, 20(4):359–363, 1977.

Davoud Moulavi, Pablo A Jaskowiak, Ricardo JGB Campello, Arthur Zimek, and Jörg Sander. Density-based clustering validation. In *Proceedings of the 2014 SIAM international conference on data mining*, pages 839–847. SIAM, 2014.

André C. A. Nascimento, Ricardo B. C. Prudêncio, Marcilio C. P. de Souto, and Ivan G. Costa. Mining rules for the automatic selection process of clustering methods applied to cancer gene expression data. In Cesare Alippi, Marios Polycarpou, Christos Panayiotou, and Georgios Ellinas, editors, *Artificial Neural Networks – ICANN 2009*, pages 20–29, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

Peter Bjorn Nemenyi. Distribution-free multiple comparisons., 1963.

Randal S Olson and Jason H Moore. Tpot: A tree-based pipeline optimization tool for automating machine learning. In *Workshop on automatic machine learning*, pages 66–74. PMLR, 2016.

Randal S. Olson, Ryan J. Urbanowicz, Peter C. Andrews, Nicole A. Lavender, La Creis Kidd, and Jason H. Moore. Automating biomedical data science through tree-based pipeline optimization. In *EvoApplications*, 2016. URL https://api.semanticscholar.org/CorpusID:9709316.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Bernhard Pfahringer, Hilan Bensusan, and Christophe G Giraud-Carrier. Meta-learning by landmarking various learning algorithms. In *ICML*, pages 743–750, 2000.

Bruno Almeida Pimentel and André C. P. L. F. de Carvalho. Statistical versus distance-based meta-features for clustering algorithm recommendation using meta-learning. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2018. doi: 10.1109/IJCNN.2018.8489182.

Bruno Almeida Pimentel and André C. P. L. F. de Carvalho. Unsupervised meta-learning for clustering algorithm recommendation. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2019. doi: 10.1109/IJCNN.2019.8851989.

Bruno Almeida Pimentel and André C.P.L.F. de Carvalho. A new data characterization for selecting clustering algorithms using meta-learning. *Information Sciences*, 477:203–219, 2019. ISSN 0020-0255. doi: https://doi.org/10.1016/j.ins.2018.10.043. URL https://www.sciencedirect.com/science/article/pii/S0020025518308624.

Fábio Pinto, Carlos Soares, and Joao Mendes-Moreira. Towards automatic generation of metafeatures. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 215–226. Springer, 2016.

Yannis Poulakis, Christos Doulkeridis, and Dimosthenis Kyriazis. Autoclust: A frame-

work for automated clustering based on cluster validity indices. In *2020 IEEE International Conference on Data Mining (ICDM)*, pages 1220–1225. IEEE, 2020.

Matthias Reif, Faisal Shafait, and Andreas Dengel. Meta-learning for evolutionary parameter optimization of classifiers. *Machine Learning*, 87(3):357–380, 2012.

Matthias Reif, Faisal Shafait, and Andreas Dengel. Automatic classifier selection for non-experts. *Pattern Analysis and Applications*, 17:83–96, 2014.

John R Rice. The algorithm selection problem, 1976.

Adriano Rivolli, Luís PF Garcia, Carlos Soares, Joaquin Vanschoren, and André CPLF de Carvalho. Characterizing classification datasets: a study of meta-features for meta-learning. *arXiv preprint arXiv:1808.10406*, 2018a.

Adriano Rivolli, Luıs PF Garcia, Carlos Soares, Joaquin Vanschoren, and André CPLF de Carvalho. Towards reproducible empirical research in meta-learning. *arXiv preprint arXiv:1808.10406*, pages 32–52, 2018b.

Adriano Rivolli, Luís PF Garcia, Carlos Soares, Joaquin Vanschoren, and André CPLF de Carvalho. Meta-features for meta-learning. *Knowledge-Based Systems*, 240: 108101, 2022.

Lior Rokach and Oded Maimon. Clustering methods. *Data mining and knowledge discovery handbook*, pages 321–352, 2005.

Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

Ketan Rajshekhar Shahapure and Charles K. Nicholas. Cluster quality analysis using silhouette score. *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 747–748, 2020. URL https://api.semanticscholar.org/CorpusID:227122930.

Rodrigo G. F. Soares, Teresa B. Ludermir, and Francisco A. T. De Carvalho. An analysis of meta-learning techniques for ranking clustering algorithms applied to artificial data. In Cesare Alippi, Marios Polycarpou, Christos Panayiotou, and Georgios Ellinas, editors, *Artificial Neural Networks – ICANN 2009*, pages 131–140, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg. ISBN 978-3-642-04274-4.

Alexander Strehl and Joydeep Ghosh. Relationship-based clustering and visualization for high-dimensional data mining. *INFORMS Journal on Computing*, 15(2):208–230, 2003.

Juan Carlos Rojas Thomas, Matilde Santos Peñas, and Marco Mora. New version of davies-bouldin index for clustering validation based on cylindrical distance. *2013 32nd International Conference of the Chilean Computer Science Society (SCCC)*, pages 49–53, 2013. URL https://api.semanticscholar.org/CorpusID:13035201.

Chris Thornton, Frank Hutter, Holger H Hoos, and Kevin Leyton-Brown. Auto-weka: Combined selection and hyperparameter optimization of classification algorithms. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 847–855, 2013.

Dennis Treder-Tschechlov, Manuel Fritz, Holger Schwarz, and Bernhard Mitschang. Ml2dac: Meta-learning to democratize automl for clustering analysis. *Proceedings of the ACM on Management of Data*, 1(2):1–26, 2023a.

Dennis Treder-Tschechlov, Manuel Fritz, Holger Schwarz, and Bernhard Mitschang. Ml2dac: Meta-learning to democratize automl for clustering analysis. *Proc. ACM Manag. Data*, 1(2), jun 2023b. doi: 10.1145/3589289. URL https://doi.org/10.1145/3589289.

Dennis Tschechlov, Manuel Fritz, and Holger Schwarz. Automl4clust: Efficient automl for clustering analyses, 2021. URL https://openproceedings.org/2021/conf/edbt/p87.pdf.

Alfred Ultsch. U*-matrix: a tool to visualize clusters in high dimensional data. Technical report, Univ., Fachbereich Mathematik und Informatik, 2003.

Toon Van Craenendonck and Hendrik Blockeel. Constraint-based clustering selection. *Machine Learning*, 106(9):1497–1521, 2017.

Iven Van Mechelen, Anne-Laure Boulesteix, Rainer Dangl, Nema Dean, Christian Hennig, Friedrich Leisch, Douglas Steinley, and Matthijs J. Warrens. A white paper on good research practices in benchmarking: The case of cluster analysis. *WIREs Data Mining and Knowledge Discovery*, 13(6):e1511, 2023. doi: https://doi.org/

10.1002/widm.1511. URL https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/widm.1511.

J Vanschoren. Meta-learning: A survey. *arXiv preprint arXiv:1810.03548*, 2018.

Joaquin Vanschoren. Meta-learning. In *Automated machine learning: methods, systems, challenges*, pages 35–61. Springer International Publishing Cham, 2019.

Joaquin Vanschoren, Jan N. van Rijn, Bernd Bischl, and Luis Torgo. Openml: Networked science in machine learning. *SIGKDD Explorations*, 15(2):49–60, 2013. doi: 10.1145/2641190.2641198. URL http://doi.acm.org/10.1145/2641190.2641198.

Lucas Vendramin, Ricardo JGB Campello, and Eduardo R Hruschka. Relative clustering validity criteria: A comparative overview. *Statistical analysis and data mining: the ASA data science journal*, 3(4):209–235, 2010.

Ricardo Vilalta and Youssef Drissi. A perspective view and survey of meta-learning. *Artificial Intelligence Review*, 18(2):77–95, 2002.

Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4): 395–416, 2007.

Ulrike von Luxburg, Robert C. Williamson, and Isabelle Guyon. Clustering: Science or art? In Isabelle Guyon, Gideon Dror, Vincent Lemaire, Graham Taylor, and Daniel Silver, editors, *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, volume 27 of *Proceedings of Machine Learning Research*, pages 65–79, Bellevue, Washington, USA, 02 Jul 2012. PMLR. URL https://proceedings.mlr.press/v27/luxburg12a.html.

Ulrike Von Luxburg, Robert C Williamson, and Isabelle Guyon. Clustering: Science or art? In *Proceedings of ICML workshop on unsupervised and transfer learning*, pages 65–79. JMLR Workshop and Conference Proceedings, 2012.

Milan Vukicevic, Sandro Radovanovic, Boris Delibasic, and Milija Suknovic. Extending meta-learning framework for clustering gene expression data with component-based algorithm design and internal evaluation measures. *International Journal of Data Mining and Bioinformatics*, 14(2):101–119, 2016.

Rui Xu and Don Wunsch. *Clustering*. John Wiley & Sons, 2008.

Rui Xu and Donald Wunsch. Survey of clustering algorithms. *IEEE Transactions on neural networks*, 16(3):645–678, 2005.

Quanming Yao, Mengshuo Wang, Yuqiang Chen, Wenyuan Dai, Yu-Feng Li, Wei-Wei Tu, Qiang Yang, and Yang Yu. Taking human out of learning applications: A survey on automated machine learning. 2019.

Michael J. Zellinger and Peter Bühlmann. repliclust: Synthetic data for cluster analysis, 2023.

Tian Zhang, Raghu Ramakrishnan, and Miron Livny. Birch: an efficient data clustering method for very large databases. *ACM sigmod record*, 25(2):103–114, 1996.

Marc-André Zöller and Marco F. Huber. Benchmark and survey of automated machine learning frameworks. *J. Artif. Int. Res.*, 70:409–472, may 2021. ISSN 1076-9757. doi: 10.1613/jair.1.11854. URL https://doi.org/10.1613/jair.1.11854.