UNIVERSITY OF TARTU
Institute of Computer Science
Computer Science Curriculum

Gregor Rehand

A Framework for Automated Clustering using TabPFN-Based Meta-Learner

Master's Thesis (30 ECTS)

Supervisor:
Radwa El Shawi, PhD

Tartu 2025

A Framework for Automated Clustering using TabPFN-Based Meta-Learner

Abstract:

The ever-increasing amount of data collected is also creating an increased demand and opportunities for applying machine learning solutions. With more computational resources and data available than ever, the bottleneck of creating new machine learning solutions has arguably shifted to the human element in form of machine learning practitioners. Automated Machine Learning is a discipline that seeks to automate decisions one is faced with when building a machine learning solution, such as the selection of the algorithm and its hyperparameter values. The goal of many Automated Machine Learning frameworks is to recommend the best possible algorithm and hyperparameter configuration for any given dataset, that the practitioner can apply in their machine learning solution.

The goal of this thesis is to present a novel Automated Machine Learning framework for clustering, using meta-learning-based approach with a transformer-based foundation model called TabPFN as the meta-learner.

Keywords: AutoML, automated clustering, CASH Optimization, TabPFN

CERCS: P170 Computer science, numerical analysis, systems, control

TabPFN meta-õppuril põhinev raamistik automatiseeritud klasterdamiseks

Lühikokkuvõte:

Andmete hulk maailmas on pidevas kasvutrendis, mis on tekitanud ka suurenenud nõudlust ja võimalusi masinõppe mudelite rakendamiseks. Kuigi traditsiooniliselt on masinõppe puhul piiravaks sisendiks olnud madal andmete hulk või puuduvad arvutusvõimsused, on üha enam pudelikaelaks muutumas vajaminev tööjõud andmeteadlaste näol. Automatiseeritud masinõppe on teadusharu, mille eesmärk on automatiseerida otsuseid, mida inimesed peavad langetama ehitades masinõppe lahendusi, nagu näiteks sobiva algoritmi valimine ning sellele algoritmile parimate hüperparameetrite väärtuste leidmine. Mitmete automatiseeritud masinõppe raamistike eesmärk on sisendiks antud andmestikule soovitada parim võimalik masinõppe algoritm ja selle hüperparameetrid, mida kasutaja saaks rakendada oma masinõppe lahenduses.

Antud magistritöö eesmärk on luua uus raamistik automatiseeritud klasterdamiseks. Välja pakutud raamistik põhineb meta-õppimisel ning kasutab meta-õppurina transformeritel põhinevat alusmudelit TabPFN.

Võtmesõnad: AutoML, automatiseeritud klasterdamine CASH-optimeerimine, TabPFN

CERCS: P170 Arvutiteadus, arvutusmeetodid, süsteemid, juhtimine (automaatjuhtimisteooria)

Table of Contents

1. Introduction

Ever since the birth of the first computers, researchers and users alike have been captivated by the idea of whether we could teach the computers to learn, to improve their performance with experience [1]. While great achievements in machine learning have been made in a wide variety of domains (computer vision, language processing, recommendation algorithms), most of these solutions are not fully automated i.e. automatically improving with experience [2].

According to No Free Lunch theorems [3], there does not exist an algorithm that can achieve a good performance on all possible problems with equal importance [2]. Therefore, to achieve the best possible result, careful consideration is needed for all aspects of machine learning, including the selection of the algorithm and its hyperparameter configuration. These decisions usually require the input of human machine learning experts, greatly increasing the costs of machine learning solutions.

However, living in the era of big data in which the amount of data generated each day keeps significantly increasing [4], there is tremendous potential applications in analysing this data. As the number of data scientists has not been able to scale with the potential application, there is a need to find way to automate the process of building well performing machine learning pipelines.

The goal of Automated Machine Learning (AutoML) is to automate these decisions and outperform the human machine learning engineers, improving the performance of the models, reducing the cost of their adaptation, therefore lowering the barrier of entry for utilizing machine learning to solve problems [4].

In this thesis, we will take a step toward better Automated Machine Learning by putting forth a novel framework for Automated Clustering.

1.1 Motivation

Clustering is an area often overlooked by the majority of AutoML research [5]. One of the reasons for this is that creating AutoML systems for unsupervised learning comes with additional challenges as it is more difficult to objectively choose the best configuration for the given problem. However, unsupervised learning tasks are widespread as gathering a labelled dataset can be difficult or impractical in real-world situations.

As can be seen from a recent survey paper [6], there is not yet a clear consensus on the best approaches and techniques to solve the algorithm selection and hyperparameter optimization problem for the clustering task, nor is there framework that lies clearly above the rest or comes close to perfect solutions [7]. The field is rapidly evolving, with several state-of-the-art frameworks published only last year [6]. This means that there is still ample opportunity to contribute to the advancement of automated machine learning for clustering.

The aim of this paper is to put forth a novel approach for automated clustering, expanding on previous work done in AutoML. The key difference being the use of a new transformer-based model for tabular data, called TabPFN, as the meta-learner for the AutoML framework. The proposed approach will be also experimentally evaluated and compared against four state-of-the-art AutoML frameworks for clustering.

1.2 Problem Statement

The main problem that AutoML is designed to solve is called *the Combined Algorithm Selection and Hyperparameter Optimization*, or the CASH problem [8], the formulation of which can be seen in Formula (1) [9].

$$A^*, \lambda_* \in \underset{Ai \in \mathcal{A}, \lambda \in \Lambda i}{\operatorname{argmin}} \left(L(A_i, D)\right) \tag{1}$$

In Formula (1), $A = \{A_1, \ldots, A_m\}$ is a set of algorithms with Λ_i denoting the domain of hyper-parameters of algorithm A_i. Finally, $L(A_i(\lambda), D)$ denotes the loss of A_i with hyperparameters $\lambda \in \Lambda_i$ on dataset D.

However, as the clustering task does not have clear impartial metric that can be used to measure the quality of the clustering result, solving the CASH problem is not that simple – its formulation has to be adjusted to fit automated clustering.

In this thesis, the decision was made to use Adjusted Rand Index (ARI) as the optimization goal, so for our purposes the formulation of the CASH problem can be seen in Formula (2).

$$A^*, \lambda_* \in \underset{Ai \in \mathcal{A}, \lambda \in \Lambda i}{\operatorname{argmax}} \left(ARI(A_i, \lambda_i, D) \right) \quad (2)$$

Here, we are looking for clustering algorithm A_i with hyperparameters λ_i that on dataset D would result in maximum possible ARI value.

This means that in its most pure form, an AutoML framework that solves the CASH problem takes a dataset (or a subset of the dataset) as input and returns best performing algorithm with its best performing hyperparameter configuration based on the evaluation metric, in this case the Adjusted Rand Index [10].

1.3 Thesis Organization

The thesis is organized as follows: first, there is a discussion of theoretical background covering the key concept of clustering and automated machine learning. This is followed by an examination of previous solutions for automated clustering, with a closer look taken at four state-of-the-art AutoML for clustering frameworks, all of which are also included in the evaluation phase of the thesis as a benchmark. The next section describes the proposed method, divided into offline and online phases. The fifth chapter covers the experiments done. This includes the description of the experiment setup and the analysis of the results. This is followed by discussion, including limitations and potential future work on the topic. Finally, in the conclusion, the main results and insights from the thesis are summarized.

2. Theoretical Background

The following section gives a brief overview of some theoretical concepts that are relevant to the work presented here or are referenced in the later sections of the thesis. Firstly, an overview of the clustering task is given, along with some key metrics that will be used to describe the performance of the clustering in the experimental phase of the thesis. Secondly, some highly relevant concepts from the paradigm of automated machine learning are introduced. Lastly, a brief overview of the TabPFN model is presented, given that this model plays a key role in the method presented in the thesis.

2.1 Clustering

Unsupervised learning is a kind of machine learning where the learner agent receives inputs but does not obtain supervised target outputs nor rewards from its environment [11]. Therefore, unsupervised learning can be thought as finding patterns in the data [12], a fundamentally descriptive task [13], an example of which is the clustering task.

The goal of data clustering is to identify groupings (clusters) within multidimensional data based on a similarity measure [6, 14]. In other words, the goal of clustering is to form categories of entities and assign individual entities into one of the formed groups [13].

It is important to note here that in the scope of this thesis, only "crisp" clustering is considered, meaning that the clustering algorithms assign each point of data to one and only one cluster [14].

2.2 Cluster Validity Indices

Cluster validity indices (CVIs) are quantifiable metrics that measure the quality of the clustering result. Often CVIs measure how compact and well-separated the resulting clusters are, balancing between intra-cluster similarity (how close data points are within the same cluster) and inter-cluster dissimilarity (how far apart different clusters are) [6]. Cluster validity indices can be split into two major groups – external and internal.

As the name suggests, external CVIs rely on the availability of external data, for example ground truth labels, which consists of the cluster label for each instance in the dataset. As the availability of ground truth is not always given in unsupervised learning, internal CVIs are

most frequently used in real-world applications. However, determining which internal CVI to use to assess clustering quality remains a challenge among practitioners [6].

Since clustering is a descriptive task, it is difficult to objectively compare different clusterings to each other as depending on the task, different outcomes are expected. It is completely reasonable to sometimes prefer a clustering with worse CVI scores. Nevertheless, a quantifiable metric to compare the results of the clustering is useful. This is why there are many different CVIs and there is not one that can be used as a fundamental ground truth (for example, the role that the accuracy metric fills for the classification task).

In the following sub-sections three CVIs described in depth, as these CVIs will also be used as a benchmark to compare the performance of the proposed method against other state-of-the-art AutoML frameworks.

2.2.1 Adjusted Rand Index

The Rand index measures the similarity between two data clusterings. In practice this means that if we have a dataset with known ground truth labels, we can use the Rand index to compare the results of a clustering to the ground truth. This is a very useful metric to optimize the performance of the clustering for, but the drawback of using Rand index in this context is that we need to know the ground truth labels of the data, meaning that this is an external cluster validity index. The Rand index is formulated in Formula (3).

$$RI = \frac{a+b}{\binom{n}{2}} \tag{3}$$

where a denotes the number of pairs of elements that are in the same cluster in both data partitions and b denotes the number of pairs of elements that are in different clusters in both data partitions [15].

While the Rand index gives us a measure of similarity, it does not take into consideration elements that randomly happened to be assigned to the same clusters. That is why a corrected-for-chance version of the Rand Index called Adjusted Rand Index (ARI) is commonly used [10]. The definition of ARI[1] can be found in Formula (4).

[1] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.adjusted_rand_score

$$\text{ARI} = \frac{\sum_{ij}\binom{n_{ij}}{2} - \sum_i\binom{a_i}{2}\sum_j\binom{b_j}{2}\,]\,/\,\binom{n}{2}}{\frac{1}{2}[\,\sum_i\binom{a_i}{2}+\sum_j\binom{b_j}{2}\,] - [\,\sum_i\binom{a_i}{2}\sum_j\binom{b_j}{2}\,]/\binom{n}{2}} \tag{4}$$

ARI value ranges from -1 to 1, with ARI of 1 meaning a perfect clustering and ARI of below 0 meaning that the clustering is worse than random chance.

2.2.2 Silhouette Coefficient

Silhouette coefficient is an internal cluster validity index that measures how similar the object is to the cluster it belongs to (cohesion) compared to the other clusters (separation). The silhouette score[2] for a single point of data *i* is defined in Formula (5).

$$S(i) = \frac{b(i)-a(i)}{\max\{a(i),b(i)\}} \tag{5}$$

Here, *a(i)* denotes the average distance between the point *i* and all the other points of the same cluster and *b(i)* denotes the smallest average distance between the point *i* and all the points in any other cluster.

The overall silhouette coefficient is found by taking the mean of the silhouette scores of all the data points. It ranges from -1 to 1, where a value close to 1 indicates good clustering [16].

2.2.3 Davies-Bouldin Index

Davies-Bouldin index is another internal cluster validity index. It measures the ratio between compactness and separation. The Davies-Bouldin index[3] is defined in Formula (6).

$$DBI = \frac{1}{n}\sum_{i=1}^{n}\max_{j\neq i}\left(\frac{S_i+S_j}{M_{ij}}\right) \tag{6}$$

Where n is the number of clusters, S denotes the average distance of all the points in cluster *i* to its centroid, in other words the cluster compactness. Finally, M_{ij} denotes distance between the centroids of the clusters *i* and *j*, in other words separation of clusters *i* and *j* [17].
It is important to note that for Davies-Boldin index, the lower the value of the index, the better the clustering result.

[2] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouette_score
[3] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.davies_bouldin_score

2.3 Automated Machine Learning

When examining the current landscape of data science, a comparison can be drawn with the software crisis [18], where the industry is heading towards a 'data science crisis' due to a pressing need for a larger number of data scientists to manage the ever-increasing data volumes [19].

While the general populace has been mostly captivated by the applications powered by recent advances in deep learning and transformers technology, the demand for classical machine learning solutions on tabular data keeps increasing as the amount of data being captured has been rapidly increasing.

The first problem AutoML frameworks tackle is the algorithm selection problem. The classic definition of the algorithm selection is that given a space of problems and a space of algorithms, the algorithm selection tries to determine a mapping from problems to algorithms that are best suited to solve them [20].

The second fundamental problem that AutoML aims to automate is hyperparameter optimization. This is the process of identifying a set of hyperparameters for a learning algorithm that results in the best possible performance based on the evaluation criteria [21].

Together these two problems form the CASH-problem, which was defined in chapter 1.2

2.3.1 Meta-learning

Meta-learning, meaning "learning to learn", is the science of systematically observing how different machine learning approaches perform on a wide range of learning tasks, and then learning from this experience. In order to learn from past experience in a data-driven way, we need to gather the meta-data that would describe past learning. This includes but is not limited to machine learning algorithms, their hyperparameter values, model evaluations (for example Adjusted Rand Index) and characteristics of the datasets that were used for learning also known as meta-features [22].

There are many ways to utilize the collected meta-data in an AutoML pipeline. One of the approaches is to use the meta-data to train a meta-model. The goal of the meta-model is to

recommend the most useful configuration to solve the task (in this case clustering) on a dataset with given meta-features. One of the ways to achieve this is to predict the performance of a model on the dataset with given meta-features for several different configurations and then selecting the best configuration.

2.3.2 Bayesian Optimization

Bayesian optimization (BO) is an iterative algorithm, very commonly used to optimize expensive blackbox functions, including hyperparameter optimization for AutoML. The algorithm has two parts: a probabilistic surrogate model and an acquisition function responsible for selecting which point to evaluate in the next iteration, balancing between exploration and exploitation. For each iteration, the surrogate model is fitted to all observation that the target function has made so far. Based on that, the acquisition function evaluates the utility of different candidate points for next evaluation. The core idea behind Bayesian optimization is that the acquisition function is much cheaper to compute that the blackbox function and therefore can be more thoroughly optimized within the same budget [22, 23].

2.3.3 Automated clustering

While the landscape AutoML systems for supervised learning can be considered mature, same cannot be said about AutoML systems for clustering [6]. The main obstacle for AutoML systems for clustering that does not exist in the supervised setting, is the lack of one true objective metric such as accuracy.

Another challenge is the amount of different clustering algorithms available and how much their performance depends on dataset characteristics, such as geometrical properties, density, dimensionality or other statistical properties. Even when one is able to select the correct algorithm, the choice of hyperparameter values can make or break the clustering result, as for many algorithms the practitioner needs to define characteristics of the resulting clustering up-front and give it to the algorithm as a parameter (such as the number of centres or the minimum number of points in a cluster). This is a stark contrast with supervised learning, where running the correct algorithm with default parameters offered by the library can already result in an adequate performance. Therefore, in addition to algorithm selection, it is crucial to correctly set the values of necessary hyperparameters for each clustering algorithm [6]

One of the more common ways to mitigate the problem of not having one single metric (such as accuracy) to determine the best performing algorithm in the context of clustering, is to use the Adjusted Rand Index [24]. This does come with the limitation that all the datasets used for meta-learning must have a true label, but in the context of offline learning this is easier to overcome [6].

2.4 TabPFN

In most of the machine learning domains, end-to-end learned solutions have gradually replaced the hand-crafted statistical models. In computer vision, classical approaches that relied on feature engineering have been made largely obsolete by convolutional neural networks. In natural language processing a similar process has happened where grammar-based approaches carefully crafted by data scientists have been replaced by transformer-based architectures. In game playing learning end-to-end strategies [25].

In stark contrast to this, statistical approaches such as gradient boosted decision trees have still been the most popular, cheapest and best-performing choice for tabular data [26]. However, a recently published model called TabPFN has been shown to offer a considerable boost in performance compared to the popular statistical models [25].

TabPFN[4] stands for Tabular Prior-data Fitted Network. It is a state-of-the-art foundation model designed for small to medium-sized tabular data, which has been demonstrated to significantly outperform gradient-boosted decision trees on datasets with up to 10,000 samples and 500 features. TabPFN is a transformer-based neural network trained on a large corpus of synthetic datasets. Similarly to large language models, TabPFN leverages in-context learning, where the model receives the train and test data in a single pass and performs the training and prediction at once. Furthermore, TabPFN utilizes a two-way attention mechanism, where each cell attends to the other features of the same row and also attends to all the cells in the column. This way TabPFN overcomes the inherent limitation of using transformers (designed for sequential data) on a tabular data structure [25].

[4] https://github.com/PriorLabs/TabPFN

3. Existing work

In this chapter an overview of the existing AutoML frameworks for clustering is given, with a deeper look into four state-of-the-art AutoML frameworks, which will also be used later in the thesis to evaluate the competitiveness of the proposed methodology.

3.1 Algorithm selection

The problem of algorithm selection is commonly solved by training a meta-learner model. The task of the meta-learner model is either [6]:

- **Classification** – returning the best ranking algorithm A (or a list of best ranking algorithms [A_1, A_2, …, A_n])
- **Regression** – returning the value of an evaluation metric

Most works tackling the combined problem of algorithm selection and hyperparameter optimization use the classification-based approaches with the most common meta-learner model architecture being K-nearest neighbours [9, 27, 28 29]. One of the frameworks for solving the CASH problem [30] does use a regressor meta-learner instead of a classifier, but in their approach the evaluation metric does not describe the quality of the resulting cluster, but the time-budget necessary for the framework to find adequate results.
However, when also exploring works that deal with just the algorithm selection part of the CASH problem, there are several examples of using a regressor.

As mentioned in the previous chapter, there is no universally accepted metric to use for the optimization of a clustering task. Therefore, another distinction can be made between the existing works on whether they rely on internal clustering indices [29], known ground truth labels [9] or both [27, 28 30]. The approach proposed in this thesis exploits only the ground truth labels.

3.2 Existing AutoML solutions for clustering

In the following subsections we will take a deeper look into 4 state-of-the-art AutoML frameworks for the clustering task, all of which will be used to comparatively evaluate the performance of the method proposed in this thesis. The characteristics of the frameworks have been brought out for comparison in Table 1.

Table 1. Characteristics of existing AutoML frameworks for clustering

	ML2DAC	**AutoML4Clust**	**Autocluster**	**cSmartML**
Meta-learner	Random Forest	-	KNN	KNN
Uses External CVIs	Yes	No	Yes	Yes
Optimizer	Bayesian	Random / Bayesian / Hyperband / BOHB	Grid Search + Ensembling	Multi-Objective Genetic Optimization
Objective Function	1 Internal CVI (chosen by classifier)	1 Internal CVI (chosen by user)	3 Internal CVIs	3 Internal CVIs
Algorithms supported	10	4	10	8

3.2.1 ML2DAC

The first AutoML framework introduced here is ML2DAC [31]. As is common, the framework is split into two phases – learning (offline) and application (online) phase. In the learning phase, a wide variety of meta-features is extracted for each dataset. Then different configurations are evaluated, using Bayesian optimisation to navigate the search space. For the next step, the most suitable CVI is selected for each dataset. The best CVI is determined by calculating the correlation between the CVI and ARI, using Spearman rank correlation coefficient. Finally, a classifier model is trained to predict the best CVI [31].

In the application phase, the meta-features are extracted, based on which the most similar datasets from the knowledgebase are identified. Then the classifier is used to choose which CVI to use. Next, warmstart configurations are fetched from the meta-knowledge repository, greatly reducing the search space. Finally, Bayesian optimization is used to find the best performing configuration [31].

3.2.2 AutoML4Clust

AutoML4Clust [32] is an AutoML system for clustering, that is split into 3 steps. In the first step, three inputs are given to the system: a dataset, an internal metric and a budget. For the internal metric, user can choose between Calinski-Harabasz index (CH), Davies-Bouldin index (DBI) or Silhouette (SIL). The budget denotes the number of iterations that the optimizer should run for [32].

In the second step, AutoML4Clust uses one of the three optimizers supported (Bayesian Optimization, Hyperband and a combination of the two). Each optimization loop consists of selecting and evaluating a configuration [32]. Notably, it is also possible to run multiple optimization tasks concurrently.

Finally, after exhausting the allocated optimization budget, the configuration that achieves the best result on the metric chosen by the user from all considered configurations is chosen [32]. Out of all the frameworks described in this section, AutoML4Clust supports the least amount of clustering algorithms with the search space composed of only three partitioning-based clustering algorithms and a Gaussian Mixture Model [7, 32].

3.2.3 Autocluster

Unlike the two AutoML frameworks described previously, Autocluster [27] splits the CASH problem into two sequential parts. The authors behind Autocluster have chosen to mitigate the lack of a general CVI for model optimization by combining the clustering results optimized for each CVI used (DBI, CHI, SIL and ARI) into an ensemble.

Firstly, the best clustering algorithms are recommended. This is done using a distance-based approach, comparing the meta-features extracted from the given dataset to the ones from the meta knowledge-base to find the most promising algorithms for which the hyperparameter optimization process is started.

For the hyperparameter optimization grid search is used and the results of the ensemble are combined using Majority Voting [27].

3.2.4 cSmartML

Another framework adhering to the paradigm of separating the algorithm selection and hyperparameter tuning into two phases, is cSmartML [28]. One key difference between cSmartML and other frameworks introduced in this chapter, is the fact that cSmartML does not solely rely on a meta-knowledge repository built in the offline phase, but continuously learns from every task it completes, as the knowledgebase gets updated with the results of the task after the framework makes a recommendation. The meta-learning part of the framework mainly serves to select the most suitable algorithm for a given dataset. After that, the hyperparameter optimization is done by using cluster validity indices for a multi-objective evolutionary algorithm [28]. The fact that cSmartML does not use a meta-learner for hyperparameter optimization is also something that sets it apart from the other frameworks discussed in this chapter.

4. Method

In this chapter, the method of the proposed AutoML framework will be covered. It relies on using a meta-learning to solve the CASH problem. This is achieved by processing multiple datasets with a variety of algorithm configurations, analysing the results. Based on the characteristics of the dataset and the results of clustering the data with various algorithm and hyperparameter configurations, a surrogate model is trained. Following the paradigm of meta-learning, the model learns from past clustering results to predict the best combination of algorithm and its hyperparameters for any given and previously unseen dataset.

As is common in meta-learning based automated machine learning solutions, the proposed method has two distinct phases: offline meta-learning phase and online phase.

4.1 Offline Meta-Learning Phase

The goal of the offline learning phase is to build a meta-knowledge repository that we can learn from in the online phase. This is done by recording meta-data and ARI from a large number of clustering tasks, allowing us to train a meta-model based on past evaluations.

4.1.1 Dataset selection

Since the proposed method relies on calculating the Adjusted Rand Index, we need to know the ground truth of the datasets. This means that all the datasets used in building the meta-knowledge repository are required to have known labels.

Another important aspect to consider when gathering datasets to be used in the offline learning phase is the diversity of the datasets. Since the goal of this method is to build a general pipeline that can recommend a well-performing clustering configuration for any given data, a broad knowledge base is needed as the model cannot greatly generalize beyond the bounds of the learned data [33].

In the end, dataset selection remains a highly subjective set of decisions where one is presented with a classic trade-off where increased computing costs can result in a better performing model.

In the experimental phase of this thesis, a mixture of real-world and synthetic datasets was used. 15 commonly used real-world datasets were fetched from the UCI Machine Learning Library[5], and additionally 80 synthetic datasets were created with a wide variety of generation strategies, creating a comprehensive knowledgebase.

4.1.2 Meta – feature selection

For any framework that relies on meta-learning, the choice of meta-features is a critical step. However, there is no strong consensus on the topic amongst the state-of-the-art autoML clustering frameworks.

In ML2DAC paper, Treder-Tschechlov et al. [31], in addition to building a novel autoML framework, defined a series of categories for meta-features building upon existing works and conducted extensive testing of their framework with different sets of meta-features. In their benchmark, the best results came when using a combined set of meta-features of 3 different sets: General, Statistical and Information-theoretic.

Following these findings, the author also chose to use meta-features from these 3 categories for the TabPFN meta-learner-based framework proposed in this thesis. This has an additional benefit of being computationally much faster than the landmarking techniques used to gather meta-features in other state-of-the-art AutoML frameworks such as cSmartML and Autocluster.

Nevertheless, to further alleviate the "very costly" offline phase of ML2DAC [7, 31], the author decided to reduce the overall number of meta-features used. The complete list of meta-features used for the TabPFN meta-learner can be seen in Table 2.

[5] https://archive.ics.uci.edu/

Table 2. List of meta-features chosen to train the TabPFN meta-learner along their categories

Feature	Category
Number of instances	General
Number of features	General
Feature to instance ratio	General
Missing values ratio	General
Mean variance	Statistical
Mean skewness	Statistical
Mean kurtosis	Statistical
Mean correlation	Statistical
PCA explained variance ratio	Statistical
Nearest neighbour sparsity	Statistical
Density	Statistical
MST mean edge length	Statistical
MST standard edge length	Statistical
Cluster compactness	Statistical
Local density variance	Statistical
Mean mutual information	Information-theoretic
Mean entropy	Information-theoretic

4.1.3 Building the Meta-Knowledge Repository

The goal of building the meta-knowledge repository is to gather a dataset that can be used to train the model to solve the CASH problem. Each record in the meta-knowledge repository has 4 principal components:

1. Meta-features of the dataset
2. The algorithm used
3. The hyperparameters used
4. Resulting ARI score of the clustering

This means that for every dataset chosen for meta-learning, there is a record for every algorithm in every hyperparameter configuration. The number of rows in the repository (denoted as N) has been described in Formula (7).

$$N = D * A * H_A * \Lambda_H \quad (7)$$

Where D describes the number of datasets, A the number of different algorithms evaluated, H the different hyperparameters evaluated for the algorithm A, and finally Λ denotes the number of unique values of hyperparameter H evaluated. The resulting structure of the knowledgebase is depicted in Figure 1.

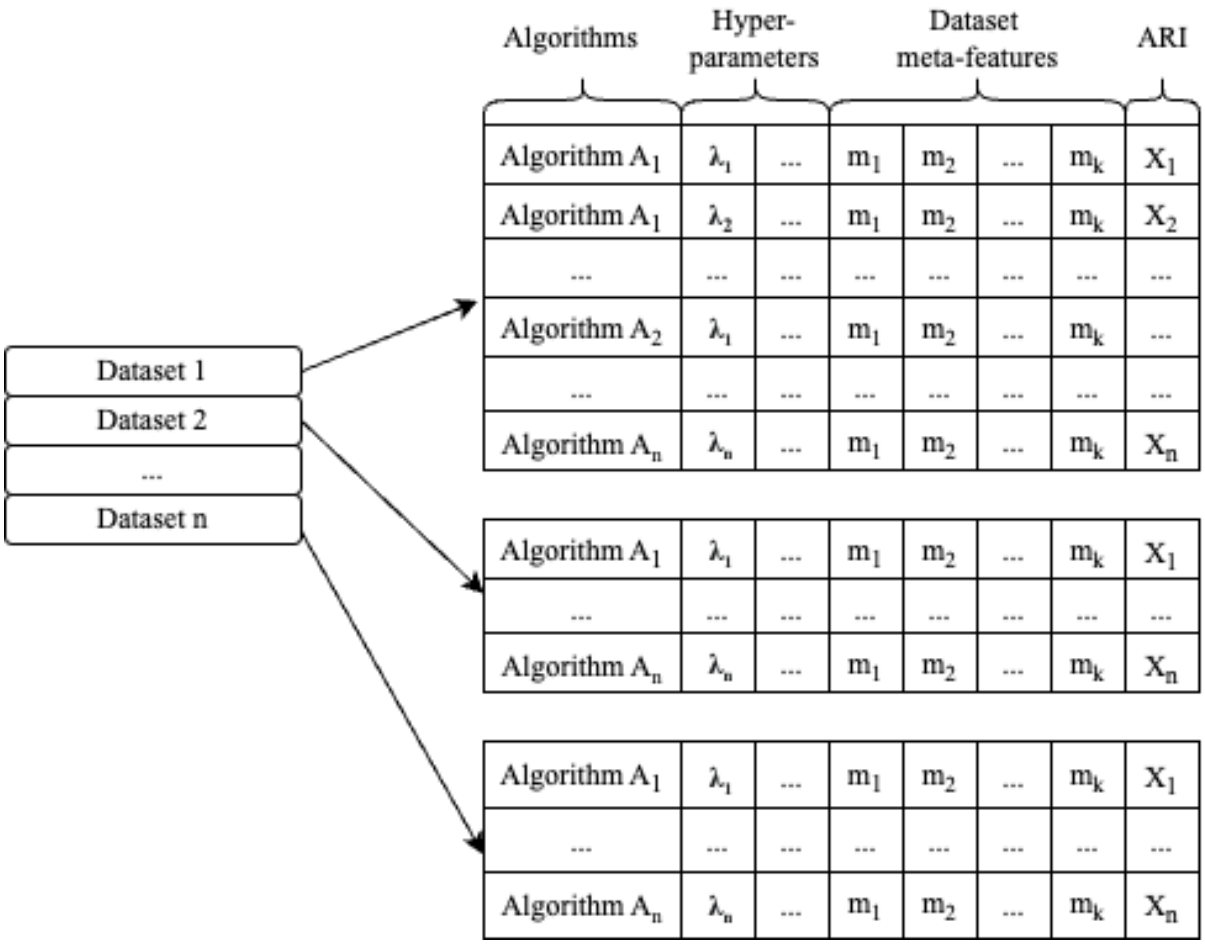

Figure 1. The structure of the meta-knowledge repository built in the offline phase

As we can see, the size of the meta-knowledge repository and the computational costs of building it (since every record in the repository represents one execution of the clustering algorithm on the dataset) can increase drastically when increasing any of the variables.

However, as discussed in section 4.1.1, we do need to include a high number of datasets to build a knowledgebase that would sufficiently cover the wide variety of characteristics the data might have. Additionally, since we are aiming to solve the problem of algorithm selection and hyperparameter optimization, it is difficult to justify reducing the number of algorithms or hyperparameters considered, as this would defeat the purpose of the solution.

That leaves us with only one domain, with which we can greatly reduce the costs of the task: hyperparameter values. For comparison, an existing state-of-the-art solution AutoClust, can have the number of different values evaluated per hyperparameter as high as 20 [9]. During earlier iterations of this thesis in the online learning context, the author tried to run a similar

approach on the HPC cluster in Tartu University. After finishing multiple optimization iteration on the code and seeing the computational costs on the initial results, it was interpolated that building the knowledgebase in this fashion would require several months of non-stop work on the HPC. It is important to note here, that in a non-online setting the costs will be smaller but nevertheless cost remains a significant barrier for building the model from scratch. The chosen hyperparameter configurations for building the meta-knowledgebase can be seen in Table 3.

Table 3. Algorithms and hyperparameter configurations used to build the meta-knowledge repository for the TabPFN meta-learner.

Algorithm	Parameter 1	Values	Parameter 2	Values
KMeans	Number of clusters (n_clusters)	[2, 3, 5, 7, 10, 15, 20, 25, 40]	-	-
DBSCAN	Epsilon (eps)	[0.2, 0.5, 1, 2, 3, 5, 8]	Minimum number of samples (min_samples)	[3, 5, 10, 20, 50, 75, 100]
GMM	Number of components (n_components)	[2, 3, 5, 10, 15, 25]	Covariance type (covariance_type)	[“full”, “tied”, “diag”, “spherical”]

The recent breakthrough by TabPFN provided us with a foundation model for tabular data. This means that instead of training a model from scratch, we can fine-tune the foundation model, thus greatly reducing the necessary size of the dataset to build a model that would be able to sufficiently solve the CASH problem.

4.2 Online Phase

The online phase of the AutoML pipeline is where the algorithm selection and hyperparameter optimization takes place. Commonly this takes place sequentially – first the best algorithm is selected and then starts the hyperparameter optimization for this algorithm.

In the method proposed in this thesis, there is no distinction between the two steps of a CASH pipeline. Since we use a regressor which predicts the performance of an algorithm with a specific hyperparameter combination, there is no need split the phases.

A major downside of this combined approach is that we can spend a lot of resource evaluating unnecessary configurations for algorithms that are not well suited for the task at hand.

We chose to mitigate this problem by using a simple multi-armed bandit approach to stop evaluating low-performing algorithms early.

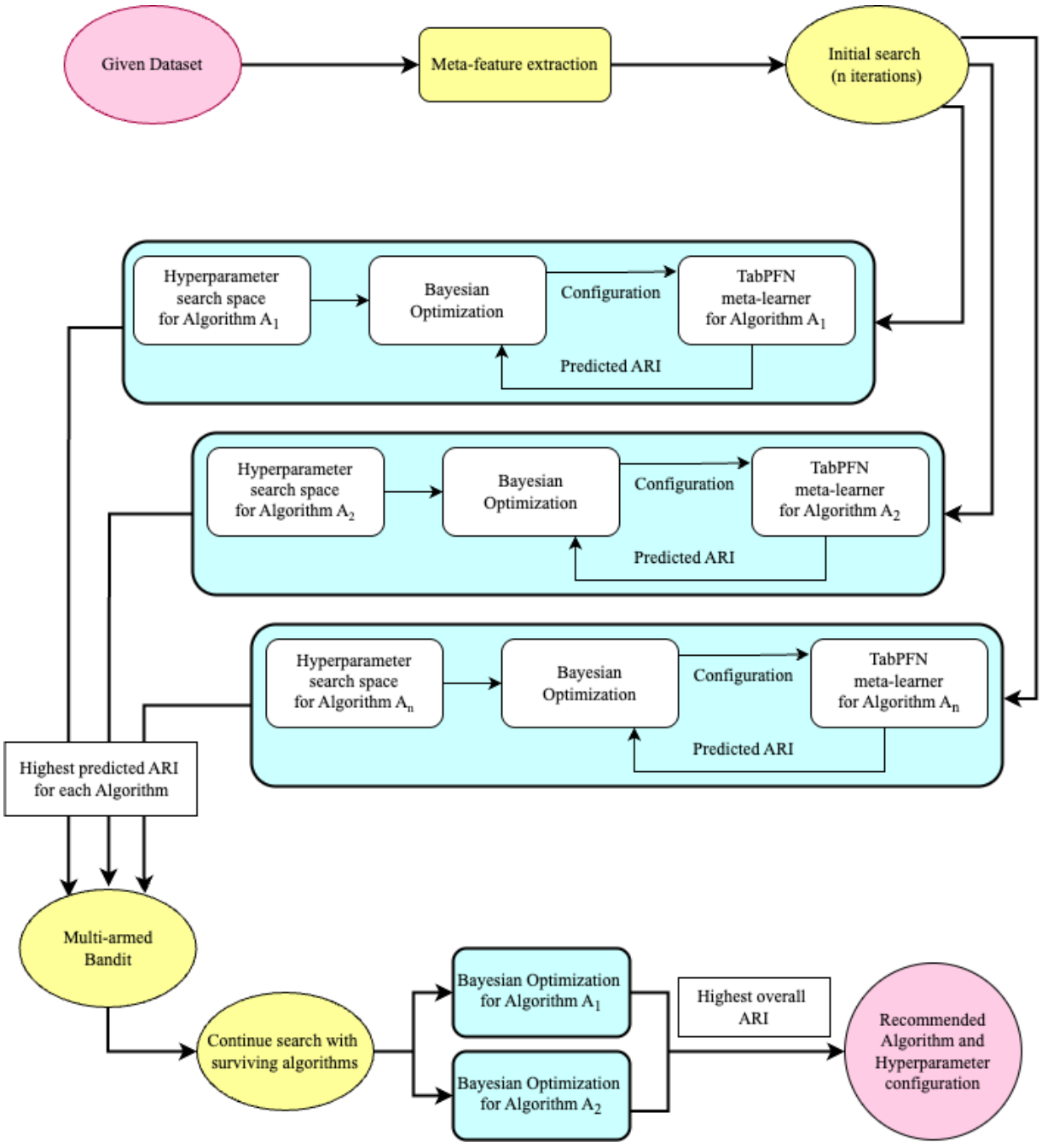

Figure 2. Online Phase

For each algorithm, a TabPFN regressor built in the offline phase for this specific algorithm is used to predict the performance with a specific set of hyperparameter values.

To determine which parameter configurations to evaluate, Bayesian optimization is used. After the allocated budget of the Bayesian optimization is expended, the algorithm and the hyperparameter configuration with the best predicted ARI is recommended by the pipeline. The general flow of the online phase can be seen in Figure 2.

5. Experimental Evaluation

In this chapter we will give an overview of the structure of the experiment that was used to evaluate the performance and viability of the proposed framework.
In 2024, da Silva et al. [7] compiled a comprehensive benchmark of various state-of-the-art AutoML solutions for clustering. Since this work is still very recent and the datasets used have been published, we will evaluate the proposed methodology in a similar way. This gives us the opportunity to easily assess the viability of the proposed methodology.

In the benchmark, da Silva et al. considered 4 different AutoML solutions: Autocluster [27], cSmartML [28], ML2DAC [31] and AutoML4Clust [32]. Out of those 4 frameworks, only Autocluster has the three clustering algorithms considered in this paper and has also detailed records for each dataset, showing which algorithm was chosen.

The frameworks used in the benchmark are fully fledged frameworks supporting up to 10 different algorithms. To fairly and accurately showcase the potential of the method proposed, the author compares it only on the datasets where Autocluster also chose one of the three clustering algorithms considered in this paper. This ensures that any differences in the results stem from the solutions ability to choose the best algorithm and hyperparameter configuration, not from the clustering algorithms considered by the AutoML pipeline.

5.1 Data

For the benchmark, da Silva et al. [7] created 100 synthetic datasets with labels by sampling data from probabilistic mixture models based on dataset archetypes defined by features.
Out of these 100 datasets, Autocluster chose KMeans, DBSCAN or GMM as the best performing algorithm on 28 datasets. These datasets are the ones that will be used to evaluate and compare the performance of the proposed methodology against Autocluster.
The specific archetypes of the datasets used in the experiment can be found in the Appendix 2.

5.2 Algorithms

Three algorithms were chosen for this experiment:

1. **KMeans** [34] – a centroid-based, iterative clustering algorithm that works by partitioning the data into K clusters by minimizing the variance within a cluster
2. **DBSCAN** (Density-Based Spatial Clustering of Applications with Noise) [35] – a density-based clustering algorithm that works by grouping together points with high density while designating point in low-density regions as noise.
3. **GMM** (Gaussian mixture model) [36] – a probabilistic model that uses iterative Expectation-Maximization algorithm to optimize the model parameters. The points are being assigned probabilities of belonging to a Gaussian curve.

These algorithms were chosen to cover a different clustering approach (centroid-based, density-based, model-based). The chosen algorithms are also widely used, making it easier to draw comparisons to earlier works.

5.3 Search Space

The proposed framework takes the range of the potential values for each hyperparameter as an input parameter along the algorithm. These values were chosen to cover a wide search space, so that the algorithm would offer adequate performance even for the extreme cases.

The exact search space for each of the hyperparameters can be found in Table 4. Covariance type of GMM is a choice between 4 constants, epsilon of DBSCAN is a continuous value within the specified range. All the other hyperparameters use discrete values within the specified range.

Table 4. Hyperparameter search space used by the TabPFN meta-learner in the experiment.

Algorithm	Parameter 1	Range	Parameter 2	Range
KMeans[6]	Number of clusters (n_clusters)	[2, 40]	-	-
DBSCAN[7]	Epsilon (eps)	[0.1, 9]	Minimum number of samples (min_samples)	[2, 100]
GMM[8]	Number of components (n_components)	[2, 30]	Covariance type (covariance_type)	[“full”, “tied”, “diag”, “spherical”]

[6] https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans
[7] https://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN
[8] https://scikit-learn.org/stable/modules/generated/sklearn.mixture.GaussianMixture

5.4 Evaluation

First, the Bayesian optimisation[9] was run for 30 iterations, after which low performing algorithms were dropped with the drop threshold of 0.7. This means that any algorithm, for which the best ARI prediction so far was less than 70% of the overall highest ARI prediction, was dropped. After which, the Bayesian optimisation was run for another 60 iterations for the remaining algorithms.

During testing it was noted that for this test setup there is no considerable improvement after increasing the number of iterations above the afromentioned number.

[9] https://github.com/bayesian-optimization/BayesianOptimization

6. Results & Discussion

In this chapter we present the results of the experiment, comparing the performance of the proposed framework to current state-of-the-art. Lastly, we discuss the potential of this approach and any future work that can be done to build upon this thesis.

6.1 Comparisons to the Benchmark

The proposed framework was evaluated on the 28 datasets taken from the benchmark by da Silva et al. [7] as described in section 5.1.1. This enables us to directly compare the performance of the TabPFN meta-learner-based framework to four state-of-the-art frameworks.

The North Star metric chosen for this thesis is ARI, since it directly relates to the ground truth. However, the benchmark by da Silva et al. [7] also measured two internal cluster validity indeces – Silhouette and DBI. While the main objective of the framework is to achieve as good of an ARI score as possible, it can be insightful to also compare Silhouette and DBI scores as these are all widely used metrics in the industry for describing the quality of the clustering.

6.1.1 ARI comparisons

In Table 5, we can see the average ARI scores achieved on the test data set by each of the algorithms and the difference in performance to the proposed TabPFN meta-learner. The resulting ARI scores are also visible as a boxplot in Figure 3 and the detailed result are available in Appendix I.

Table 5. Average ARI and the difference compared to the TabPFN meta-learner for each framework.

	ML2DAC	**AutoML4Clust**	**Autocluster**	**cSmartML**	**TabPFN meta-learner**
Average ARI	0.754	0.596	0.644	0.383	0.694
TabPFN meta-learner ARI Δ	0.059	-0.099	-0.05	-0.312	-

The proposed framework outperformed Autocluster in 16 out of 28 cases, or 57.1% of the time. It also had the better average performance with the average ARI of 0.69 compared to Autocluster's average ARI of 0.64.

Additionally, we can draw indirect comparisons to two other frameworks considered in the benchmark: ML2DAC, AutoML4Clust and cSmartML. Unfortunately, there was no specific data available for these frameworks, meaning that we do not know which clustering algorithm they chose for each dataset. Therefore, any difference in performance cannot be solely attributed to the ability of the framework, but some other clustering algorithm might just be inherently better for the dataset. Nevertheless, the proposed methodology manages to outperform AutoML4Clust's average ARI by 0.1, performing better in 17 out of 28 cases, or 60.7% of the time.

ML2DAC [31] is the only framework of the four that manages to outperform the proposed methodology, with the average ARI higher by 0.06 and outperforming the TabPFN meta-learner-based approach on 19 out of 28 datasets, or 67.9% of the time.
For the fourth framework, cSmartML, the benchmark was missing data on one of the datasets. Still, for 27 datasets where the performance data was available, TabPFN meta-learner outperformed cSmartML 23 times, or on 85.2% of the cases. The average ARI for the TabPFN meta-learner was also higher by 0.31 than the average ARI of cSmartML.

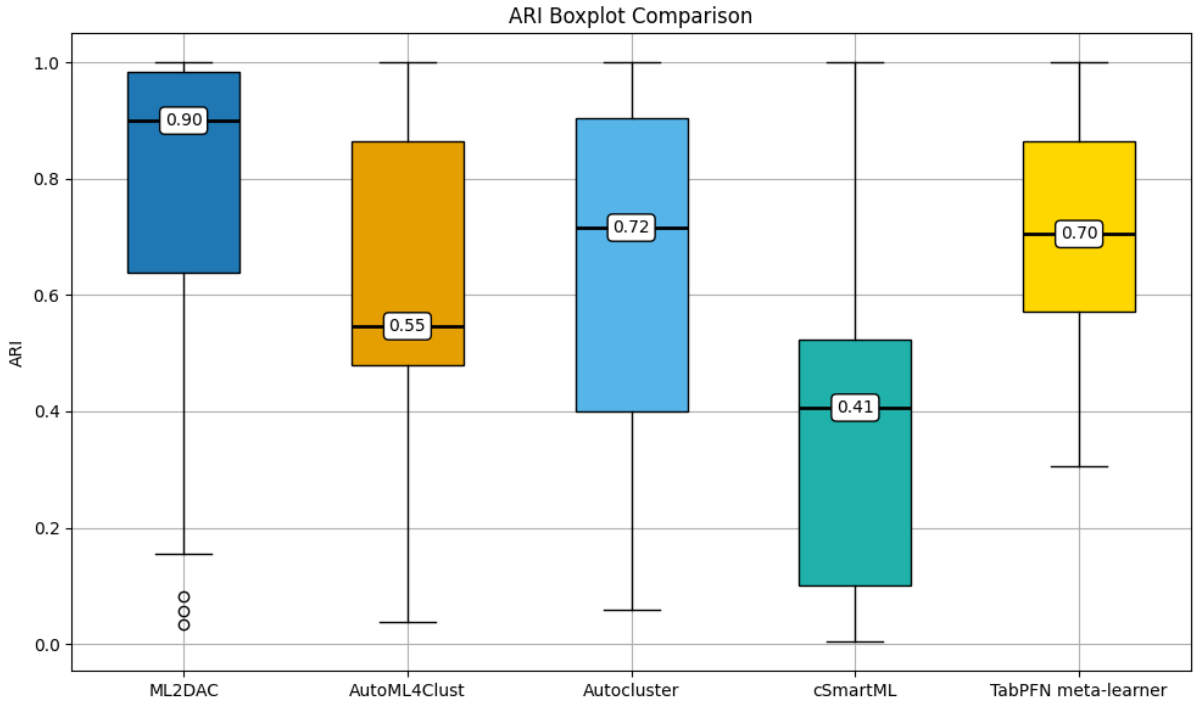

Figure 3. Boxplot of the resulting ARI scores. The colored box represents the middle 50% of the data, while the black line on the box marks the median value. The whiskers show the range of value not considered outliers, while outliers are marked as dots.

From Figure 3 we can observe another notable takeaway – the TabPFN meta-learner performs considerably more consistently, without any outliers or ARI scores below 0.3, unlike the other frameworks which all recommended a very weak configuration for at least a few datasets.

6.1.2 Other metrics

For the Davies-Bouldin index comparison (in Table 5, Figure 4 and Appendix III), we can see that the TabPFN meta-learner outperforms Autocluster, mainly thanks to having less outliers. However, the other three frameworks show slightly better results.

Table 6. Average Davies-Bouldin index and the difference compared to the TabPFN meta-learner for each framework. Imporant note that for DBI, the lower the value, the better.

	ML2DAC	**AutoML4Clust**	**Autocluster**	**cSmartML**	**TabPFN meta-learner**
Average DBI	1.457	1.581	2.984	1.402	2.124
TabPFN meta-learner DBI Δ	-0.667	-0.543	0.86	-0.722	-

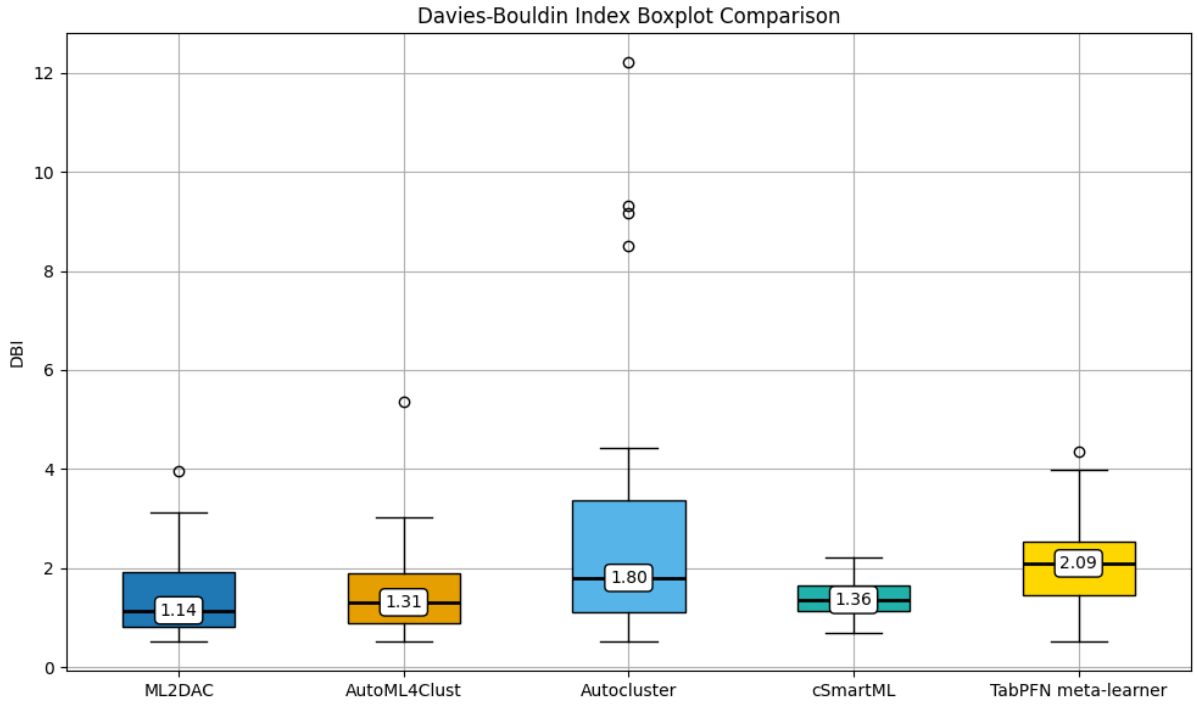

Figure 4. Boxplot of the resulting DBI scores (the lower the better).

When comparing the silhouette coefficient (in Table 6, Figure 5 and Appendix II), we can see that the proposed method outperforms cSmartML, but is outperformed very narrowly by both Autocluster and AutoML4Clust. And ML2DAC is again slightly above the rest of the pack.

Table 7. Average SIL and the difference compared to the TabPFN meta-learner for each framework.

	ML2DAC	**AutoML4Clust**	**Autocluster**	**cSmartML**	**TabPFN meta-learner**
Average SIL	0.415	0.358	0.374	0.198	0.313
TabPFN meta-learner SIL Δ	0.102	0.044	0.061	-0.112	-

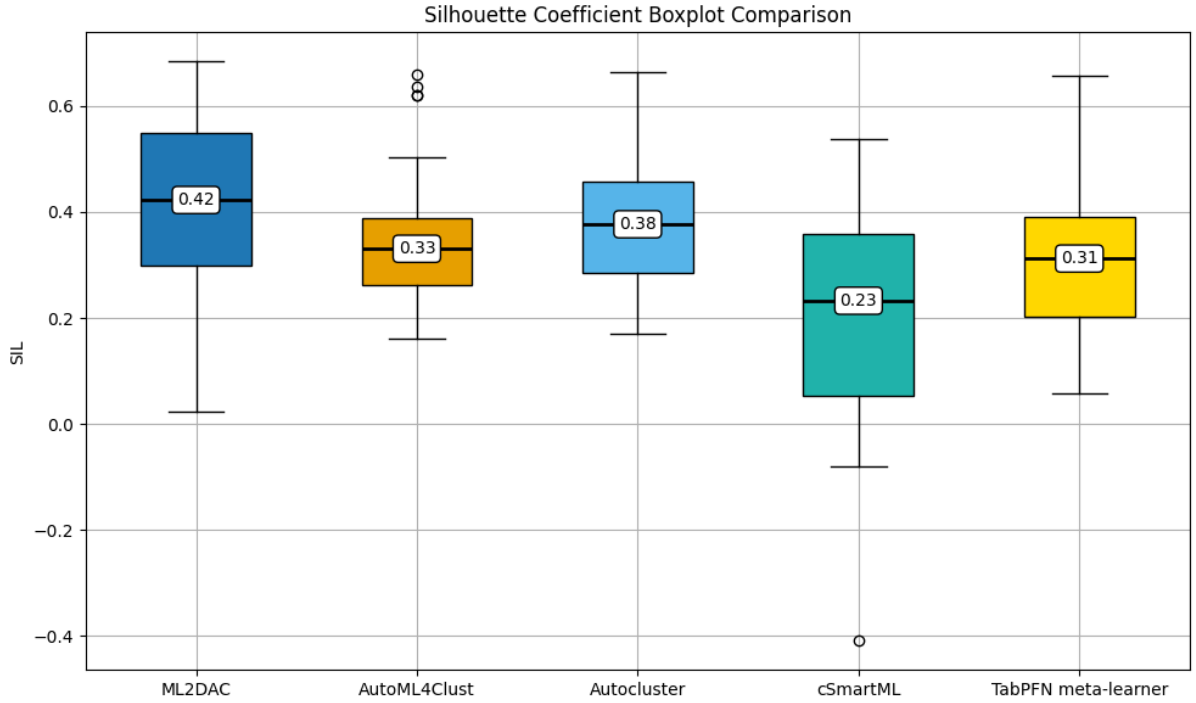

Figure 5. Boxplot of the resulting SIL scores.

The result of the experiment show that the AutoML framework proposed in this thesis fits right into the state-of-the-art, even outperforming most of the frameworks in the metric the model was optimized for (ARI). Another key achievement is the consistency of the results, hinting at the robustness of the proposed TabPFN-based framework.

6.2 Discussion & Future Work

In its current form, with supporting only 3 different clustering algorithms and having a relatively small meta-knowledgebase, the proposed framework performs exceedingly well when compared to the current state-of-the-art frameworks, only being surpassed by ML2DAC. This indicates a huge potential for using TabPFN as a meta-learner in the context of automated machine learning.

The main future efforts would be to expand the scope of the work, bringing it from a mere proof-of-concept to a fully-fledged AutoML framework. There are 5 main areas where the scope of the work can be expanded upon:

1. The number of meta-features used
2. The number of datasets used to build the meta-knowledgebase
3. The number of algorithms supported
4. The number of configurations evaluated per algorithm in the offline phase
5. Ensembling and/or using other target metrics (for example, Silhouette)
6. Publishing the online part of the framework as an open python library, also containing the weights of the models

The authors of TabPFN claim that the framework gives a performance boost compared to other existing regressors on datasets with up to 10000 rows. That still leaves room for expansion as currently depending on the algorithm the author used 900 – 5000 rows for the meta-knowledgebase.

When adding support for additional algorithms, more sophisticated optimization techniques should be also considered such as Hyperband or BOHB, which were not deemed necessary when dealing with only a three-armed bandit.

7. Summary

In this thesis, the author introduced a novel AutoML framework that utilizes TabPFN foundation model as the meta-learner. The framework was built following the paradigm of meta-learning, whereby many different clustering setups were evaluated, the results of which were compiled into a comprehensive meta-knowledge repository. Based on this repository, a TabPFN regressor was trained to predict the best performing algorithm and hyperparameter configuration for any given dataset in terms of Adjusted Rand Index. The proposed framework was then evaluated against four state-of-the-art AutoML frameworks for clustering based on a recently published benchmark. The proposed approach shows great promise, as the framework managed to outperform three out of the four frameworks in terms of the Adjusted Rand Index, while showing the most consistent results.

References

[1] Mitchell T.M., Carbonell J.G., Michalski R.S. (eds.) Machine learning: a guide to current research. Springer Science & Business Media. 1986.

[2] Yao Q., Wang M., Chen Y., Dai W., Li Y.F., Tu W.W., Yang Q., Yu Y. Taking human out of learning applications: A survey on automated machine learning. *arXiv preprint*, arXiv:1810.13306, 2018.

[3] Wolpert D.H., Macready W.G. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1997, Vol. 1, No. 1, pp. 67–82.

[4] Salehin I., Islam M.S., Saha P., Noman S.M., Tuni A., Hasan M.M., Baten M.A. AutoML: A systematic review on automated machine learning with neural architecture search. *Journal of Information and Intelligence*, 2024, Vol. 2, No. 1, pp. 52–81.

[5] Gomes H.M., Read J., Bifet A., Barddal J.P., Gama J. Machine learning for streaming data: state of the art, challenges, and opportunities. *ACM SIGKDD Explorations Newsletter*, 2019, Vol. 21, No. 2, pp. 6–22.

[6] Poulakis, Y., Doulkeridis, C. and Kyriazis, D. A survey on automl methods and systems for clustering. *ACM Transactions on Knowledge Discovery from Data*, 2024, Vol. 18, No. 5, pp.1-30.

[7] da Silva, M.C., Licari, B., Tavares, G.M., Junior, S.B. Benchmarking AutoML Clustering Frameworks. *AutoML Conference 2024*, 2024.

[8] Feurer M., Klein A., Eggensperger K., Springenberg J.T., Blum M., Hutter F. Efficient and robust automated machine learning. *Advances in Neural Information Processing Systems (NeurIPS'15).* 2015, pp. 2962–2970.

[9] Poulakis Y., Doulkeridis C., Kyriazis D. Autoclust: A framework for automated clustering based on cluster validity indices. *Proceedings of the 2020 IEEE International Conference on Data Mining (ICDM)*. 2020, pp. 1220–1225.

[10] Vinh, N.X., Epps, J. and Bailey, J. Information theoretic measures for clusterings comparison: is a correction for chance necessary?, *Proceedings of the 26th annual international conference on machine learning*, 2009, pp. 1073-1080.

[11] Ghahramani, Z. Unsupervised learning. *Summer school on machine learning.* 2003, pp. 72-112.

[12] Hoi, S.C., Sahoo, D., Lu, J. and Zhao, P. Online learning: A comprehensive survey. *Neurocomputing*. 2021, Vol. 459, pp.249-289.

[13] Rokach, L. and Maimon, O. Clustering methods. *Data mining and knowledge discovery handbook*, 2005, pp.321-352.

[14] Omran, M.G., Engelbrecht, A.P. and Salman, A. An overview of clustering methods. *Intelligent Data Analysis*, 2007, Vol. 11, No. 6, pp.583-605.

[15] Rand, W.M. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association,* 1971, Vol. 66, No.336, pp.846-850.

[16] Rousseeuw, P.J. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 1987, Vol. 20, pp.53-65.

[17] Davies, D.L., Bouldin, D.W. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1979, Vol. 1, No. 2, pp. 224-227.

[18] Fitzgerald, B. Software Crisis 2.0. *2018 Software Technology: 10 Years of Innovation in IEEE Computer: 10 Years of Innovation,* 2018, pp.1-16.

[19] Eldeeb, H., El Shawi, R. Empowering Machine Learning Pipelines with Automated Feature Engineering. *IEEE Transactions on Artificial Intelligence.* 2024.

[20] Rice, J.R. The algorithm selection problem. *Advances in computers,* 1976, Vol. 15, pp. 65-118, Elsevier.

[21] Bergstra, J. and Bengio, Y. Random search for hyper-parameter optimization. *The journal of machine learning research*, 2012, Vol. 13, No. 1, pp.281-305.

[22] Hutter, F., Kotthoff, L. and Vanschoren, J. Automated machine learning: methods, systems, challenges. Springer Nature. 2019.

[23] Močkus, J. On Bayesian methods for seeking the extremum. *Optimization Techniques IFIP Technical Conference Novosibirsk, July 1–7, 1974 6,* pp. 400-404. Springer Berlin Heidelberg.

[24] Warrens, M.J., van der Hoef, H. Understanding the adjusted rand index and other partition comparison indices based on counting object pairs. *Journal of Classification*, 2022, Vol. 39, No. 3, pp.487-509.

[25] Hollmann, N., Müller, S., Purucker, L. *et al.* Accurate predictions on small data with a tabular foundation model. *Nature.* 2025, Vol. 637, pp. 319–326.

[26] McElfresh, D.C., Khandagale, S., Valverde, J., Prasad, V., Ramakrishnan, G., Goldblum, M. and White, C., January. When Do Neural Nets Outperform Boosted Trees on Tabular Data? *NeurIPS*, 2023.

[27] Liu, Y., Li, S. and Tian, W. Autocluster: Meta-learning based ensemble method for automated unsupervised clustering. *Pacific-Asia Conference on Knowledge Discovery and Data Mining,* 2021, pp. 246-258, Springer International Publishing.

[28] El Shawi, R., Lekunze, H. and Sakr, S. csmartml: A meta learning-based framework for automated selection and hyperparameter tuning for clustering. *2021 IEEE International Conference on Big Data (Big Data),* 2021, pp. 1119-1126, IEEE.

[29] Pimentel, B., Carvalho, A. A new data characterization for selecting clustering algorithms using meta-learning. *Inf. Sci.,* 2019, Vol. 477, pp. 203–219.

[30] El Shawi, R. and Sakr, S. cSmartML-Glassbox: Increasing transparency and controllability in automated clustering. *2022 IEEE International Conference on Data Mining Workshops (ICDMW),* 2022, pp. 47-54, IEEE.

[31] Treder-Tschechlov, D., Fritz, M., Schwarz, H. and Mitschang, B. Ml2dac: Meta-learning to democratize automl for clustering analysis. *Proceedings of the ACM on Management of Data*, 2023, Vol.1, No.2, pp.1-26.

[32] Tschechlov, D., Fritz, M., and Schwarz, H. AutoML4Clust: Efficient autoML for Clustering Analyses, 2021.

[33] Vapnik, V.N. An overview of statistical learning theory. *IEEE transactions on neural networks,* 1999, Vol.10 No. 5, pp.988-999.

[34] MacQueen, J., January. Some methods for classification and analysis of multivariate observations. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics*

and Probability, Volume 1: Statistics, 1967, Vol. 5, pp. 281-298, University of California press.

[35] Ester, M., Krigel, H. P., Sander, J., and Xu, X. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. *kdd,* 1996, Vol. 96, No. 34, pp. 226-231.

[36] Reynolds DA. Gaussian mixture models. *Encyclopedia of biometrics*, 2009, Vol. 741, pp.659-663.

Appendix

I. Detailed Benchmark Comparisons for ARI

Dataset Number	ML2DAC ARI	AutoML4-Clust ARI	Autocluster ARI	cSmartML ARI	TabPFN meta-learner ARI	Wins
1	0.984	0.536	0.864	0.406	0.567	2
2	1	0.534	0.943	1	0.318	0
3	0.985	0.504	0.892	0.885	0.584	1
4	0.822	0.637	0.284	0.003	0.416	2
5	1	1	0.313	0.54	0.655	2
6	0.88	0.906	0.059	0.483	0.653	2
7	0.989	0.783	0.998	0.058	0.892	2
8	0.894	0.556	0.995	0.634	0.763	2
9	0.514	0.516	0.157	0.081	0.715	4
10	0.998	0.426	0.986	0.33	0.695	2
11	0.89	0.645	0.669	0.597	0.863	3
12	0.289	0.264	0.128	0.074	0.536	4
13	0.154	0.922	0.991	0.47	0.905	2
14	0.939	0.888	0.62	0.486	0.92	3
15	0.893	0.889	0.658	0.104	0.735	2
16	1	0.856	0.996	0.469	1	3
17	0.679	0.471	0.428	0.114	0.572	3
18	0.985	0.056	0.792	0.386	0.432	2
19	0.953	0.49	0.868	0.509	0.306	0
20	0.932	0.506	0.887	0.861	0.586	1
21	0.906	0.991	0.662	0.554	0.983	3
22	0.944	0.481	0.1	0.289	0.767	3
23	0.035	0.038	0.816	0.069	0.86	4
24	0.883	0.685	0.521	Missing a value	0.692	2
25	0.081	0.419	0.762	0.255	0.869	4
26	0.057	0.961	0.504	0.096	0.92	3
27	0.504	0.636	1	0.501	0.526	2
28	0.915	0.082	0.127	0.083	0.714	3
Average	**0.754**	**0.596**	**0.644**	**0.383**	**0.694**	**2.357**
TabPFN Δ	**0.059**	**-0.099**	**-0.05**	**-0.312**	**-**	

II. Detailed Benchmark Comparisons for Silhouette

Dataset Number	ML2DAC SIL	AutoML4-Clust SIL	Autocluster SIL	cSmartML SIL	TabPFN meta-learner SIL
1	0.544	0.309	0.497	-0.035	0.417
2	0.537	0.311	0.411	0.537	0.1
3	0.576	0.325	0.472	0.503	0.389
4	0.252	0.354	0.336	-0.409	0.22
5	0.502	0.502	0.296	0.244	0.267
6	0.478	0.476	0.402	0.435	0.373
7	0.434	0.402	0.438	-0.08	0.358
8	0.411	0.226	0.391	0.23	0.327
9	0.369	0.369	0.349	0.038	0.234
10	0.383	0.189	0.327	0.061	0.242
11	0.375	0.383	0.337	0.365	0.363
12	0.283	0.28	0.235	0.26	0.117
13	0.022	0.359	0.363	0.233	0.324
14	0.684	0.619	0.489	0.153	0.658
15	0.632	0.637	0.511	0.015	0.587
16	0.679	0.659	0.664	0.361	0.632
17	0.49	0.336	0.452	0.096	0.401
18	0.477	0.162	0.4	0.33	0.374
19	0.619	0.38	0.536	0.377	0.058
20	0.585	0.371	0.548	0.356	0.479
21	0.564	0.621	0.426	0.274	0.559
22	0.221	0.224	0.251	0.044	0.184
23	0.246	0.265	0.191	0.141	0.208
24	0.319	0.269	0.222	0	0.267
25	0.269	0.25	0.226	0.222	0.122
26	0.305	0.226	0.239	0.251	0.154
27	0.332	0.303	0.296	0.366	0.3
28	0.035	0.21	0.169	-0.021	0.063
Average	**0.415**	**0.358**	**0.374**	**0.198**	**0.313**
TabPFN Δ	**0.102**	**0.044**	**0.061**	**-0.116**	**-**

III. Detailed Benchmark Comparisons for Davies-Bouldin Index

Dataset Number	ML2DAC DBI	AutoML4-Clust DBI	Autocluster DBI	cSmartML DBI	TabPFN meta-learner DBI
1	0.755	1.232	0.89	1.421	2.268
2	0.726	1.284	1.369	0.726	2.453
3	0.666	1.116	8.506	1.599	2.227
4	1.618	1.684	3.536	1.524	2.284
5	0.824	0.824	3.316	1.334	2.002
6	0.972	0.906	0.816	0.94	1.113
7	1.089	1.111	0.933	1.688	1.19
8	1.935	2.251	1.909	1.88	2.249
9	1.344	1.342	1.775	1.029	2.877
10	0.983	2.343	2.078	1.605	2.756
11	1.646	1.789	1.921	2.034	1.748
12	2.6	2.603	2.985	1.177	2.464
13	0.869	1.897	1.124	1.502	1.882
14	3.123	0.711	1.1	1.074	0.51
15	1.925	0.512	0.622	1.355	0.557
16	0.513	0.684	0.506	0.998	0.56
17	0.988	1.173	1.008	1.261	1.512
18	0.733	1.33	9.162	1.599	1.304
19	0.573	0.825	9.328	0.693	3.837
20	0.672	0.867	12.212	2.217	1.516
21	0.893	0.526	4.42	0.921	0.671
22	2.41	2.452	1.639	2.037	2.185
23	1.728	1.254	1.621	1.733	1.814
24	3.947	3.018	1.685	0	4.353
25	1.311	1.558	1.724	1.731	3.993
26	1.199	1.712	1.826	1.265	3.331
27	2.537	5.35	2.037	1.255	3.86
28	2.22	1.925	3.514	1.267	1.953
Average	**1.457**	**1.581**	**2.984**	**1.402**	**2.124**
TabPFN Δ	**-0.667**	**-0.543**	**0.86**	**-0.722**	**-**

IV. Experiment Dataset Archetypes

In this Appendix are the archetypes of the datasets used for evaluation. The datasets were initially created for the AutoML clustering benchmark by da Silva et al. [7].

Dataset Number	Dimensions	Clusters	Instances	Aspect Ref	Aspect Max Min	Radius Max Min	Imbalance Ratio
1	10	2	1500	1.5	1	5	1
2	10	2	400	1.5	1	1	1
3	10	2	4500	1.5	3	3	1
4	10	3	3500	5	1	5	2
5	10	4	450	3	1	3	1
6	11	12	400	3	3	3	2
7	11	13	2500	5	1	3	1
8	18	4	2000	5	1	5	1
9	18	4	2500	1.5	1	5	1
10	18	4	4000	1.5	3	3	2
11	18	4	500	5	5	5	1
12	18	5	1500	5	5	5	2
13	18	5	550	1.5	1	5	1
14	2	14	4000	1.5	1	5	2
15	2	16	600	5	1	3	2
16	3	3	200	3	1	5	1
17	5	2	2750	5	5	3	1
18	5	2	3000	3	1	3	2
19	5	2	4000	1.5	1	3	1
20	5	2	4500	1.5	5	3	1
21	5	3	350	1.5	5	3	1
22	51	15	2000	1.5	3	3	2
23	52	20	2250	1.5	3	1	1
24	53	4	3000	1.5	1	5	1
25	54	11	1750	1.5	3	1	1
26	54	13	1750	1.5	3	1	1
27	71	2	2000	1.5	3	5	1
28	99	10	200	3	3	3	2

V. Glossary

All the code used in the writing of this thesis is available in Github at https://github.com/gregorrehand/automl-clustering-tabpfn

License

Non-exclusive licence to reproduce the thesis and make the thesis public

I. **Gregor Rehand.**

grant the University of Tartu a free permit (non-exclusive licence) to

reproduce. for the purpose of preservation. including for adding to the digital archives of the University of Tartu until the expiry of the term of copyright. my thesis

A Framework for Automated Clustering using TabPFN-Based Meta-Learner
supervised by **Radwa El Shawi**

2. grant the University of Tartu a permit to make the thesis specified in point 1 available to the public via the web environment of the University of Tartu. including via the digital archives, under the Creative Commons licence CC BY NC ND 4.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright;

3. am aware of the fact that the author retains the rights specified in points 1 and 2;

4. confirm that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Gregor Rehand
15/05/2025