Automating Machine Learning Pipeline Design via Metalearning

Edesio Alcobaça[1], André C. P. L. F. de Carvalho[1]

[1]Institute of Mathematics and Computer Sciences – University of São Paulo (ICMC-USP)
São Carlos – SP – Brazil

e.alcobaca@gmail.com, andre@icmc.usp.br

Abstract. *Although Automated Machine Learning (AutoML) systems allow the use of Machine Learning (ML) to automate the design of ML pipelines, they typically search over fixed, task-agnostic configuration spaces, leading to high computational costs. This paper overviews a Ph.D. thesis that proposes a paradigm shift: using Metalearning (MtL) to dynamically build task-specific search spaces. Unlike prior approaches that either optimize within a fixed search space or directly recommend algorithms without an optimization step, this thesis introduces the Dynamic Pipeline CASH problem, which extends the CASH formulation to incorporate meta-model-driven search space creation for pipelines. The thesis contributes a systematic literature review identifying meta-knowledge as the unifying thread across AutoML subfields, applied studies reinforcing the importance of algorithm selection and tuning, a large-scale benchmark of over one million pipeline configurations, and the pymfe package for reproducible meta-feature extraction. These building blocks converge into a novel MtL framework that dynamically reduces search spaces while maintaining competitive performance.*

1. Introduction

The growing volume of data generated from diverse sources has intensified the demand for computational methods capable of extracting patterns and supporting decision-making. Machine Learning (ML) has emerged as a central discipline in this context, enabling the induction of models that learn from data to perform prediction tasks. However, building effective ML solutions from raw data is a complex process that extends well beyond the application of a single algorithm. Data often requires preprocessing to handle missing values, noise, and feature transformations, and the selection of a suitable learning algorithm, together with its hyperparameter configuration, depends on the characteristics of each problem [Brazdil et al. 2022; Hutter et al. 2019].

The incorporation of these aspects in the design of a ML solution, creating a coherent pipeline, is known as end-to-end ML. The design of these pipelines is typically a labor-intensive task that requires domain expertise and extensive experimentation. Automated Machine Learning (AutoML) was proposed to address this challenge by automating the selection and configuration of pipeline components, thereby reducing the requirement of human intervention [Hutter et al. 2019]. Systems such as Auto-WEKA [Thornton et al. 2013], Auto-Sklearn [Feurer et al. 2015a], and TPOT [Olson et al. 2016] have shown that competitive pipelines can be found automatically. Nevertheless, these systems still face important limitations, including high computational costs arising from large search spaces, the risk of overfitting introduced by aggressive optimization strategies [Fabris and Freitas 2019], and difficulty in navigating complex hyperparameter landscapes [Thornton et al. 2013; Olson et al. 2016; de Sá et al. 2017].

Metalearning (MtL) offers a principled approach to mitigate these limitations by leveraging knowledge accumulated from previous learning experiences [Brazdil et al. 2022]. Rather than searching from scratch for every new task, MtL uses characterizations of datasets, known as meta-features, and historical performance records to recommend algorithms, hyperparameters, or even entire pipeline configurations [Brazdil et al. 2022; Feurer et al. 2015b; Mantovani et al. 2019]. Despite these promising capabilities, the use of MtL to dynamically design task-specific search spaces for pipeline optimization remains largely unexplored.

This paper presents an overview of the main contributions of the first author's Ph.D. thesis, which investigates how MtL can be used to improve the efficiency of AutoML pipeline design. The thesis addresses three interrelated research questions, progressing from a conceptual analysis of the AutoML landscape to an empirical investigation of pipeline configurations and, ultimately, to the proposal of a MtL framework for dynamic search space construction. The remainder of this paper is organized as follows. Section 2 states the research problem, objective, questions, and hypotheses. Section 3 summarizes the theoretical foundation and related work. Section 4 presents the main contributions and results. Section 5 concludes the paper with a discussion of the findings and directions for future work.

2. Problem, objective, research questions, and hypotheses

Current AutoML methods typically search over a fixed configuration space that includes all available preprocessing techniques and classification algorithms, along with their respective hyperparameter ranges [Hutter et al. 2019]. This design leads to three related problems: (i) the computational cost of exploring large search spaces can be prohibitive, especially under constrained time budgets [Thornton et al. 2013; Olson et al. 2016]; (ii) optimization strategies may overfit to validation data when exploring too many configurations [Thornton et al. 2013; Fabris and Freitas 2019]; and (iii) many of the evaluated configurations are unlikely to perform well for the task at hand, wasting computational resources [Olson et al. 2016; de Sá et al. 2017].

These observations motivate the central objective of the thesis: to investigate whether MtL can be used to dynamically generate task-specific search spaces for ML pipeline design, thereby reducing computational costs while preserving predictive performance. The thesis is guided by three research questions, structured to reflect a progression from conceptual grounding to empirical analysis and methodological innovation.

Q1: *What is the historical background of AutoML? How does meta-knowledge bridge different subareas of ML, and what are the main applications and emerging research directions?*

Q2: *How do different combinations of preprocessing techniques and ML classifiers affect classification performance across diverse tasks? Are there consistent patterns of effective or ineffective pipeline configurations?*

Q3: *Can MtL be used to dynamically generate task-specific search spaces for ML* *pipeline design, thereby preserving pipeline performance while reducing the full search space and optimization time?*

These questions are supported by three corresponding hypotheses. ***H1*** posits that a systematic review will reveal that AutoML has a rich background significantly influenced by MtL, transfer learning, and hyperparameter optimization, and that emerging trends focus on interpretability, scalability, and ethical considerations. ***H2*** states that certain combinations of preprocessing techniques and classifiers consistently yield higher or lower performance across tasks, enabling the extraction of reliable meta-knowledge for pipeline recommendations. ***H3*** hypothesizes that MtL can dynamically generate reduced search spaces using historical meta-knowledge, significantly reducing optimization time while achieving performance comparable to full-space search strategies.

3. Theoretical foundation and related work

An ML pipeline consists of a sequence of data transformations followed by a learning algorithm, all of which must be configured appropriately for the task at hand. Formally, the problem of finding an optimal pipeline can be framed as the Combined Algorithm Selection and Hyperparameter optimization (CASH) problem [Thornton et al. 2013]. Given a set of algorithms $\mathcal{A} = \{A_1, \ldots, A_m\}$, each with a corresponding hyperparameter space Λ_j, and a dataset $\mathcal{D}$ split into training and validation folds, the CASH problem seeks the algorithm and hyperparameter configuration that minimizes the expected loss:

$$A^*_{\lambda^*} \in \arg \min_{A_j \in \mathcal{A},\ \lambda \in \Lambda_j} \frac{1}{k} \sum_{i=1}^{k} \mathcal{L}\big(A_j(\mathcal{D}^{(i)}_{\text{train}}, \lambda),\ \mathcal{D}^{(i)}_{\text{val}}\big) \tag{1}$$

where $\mathcal{L}$ denotes a loss function and k is the number of cross-validation folds.

This thesis extends the CASH formulation by proposing the *Dynamic Pipeline CASH* problem, which jointly models pipeline composition and meta-model-driven search space selection. Let $A^p = \{\alpha^p_1, \ldots, \alpha^p_{n_p}\}$ be a set of preprocessors with hyperparameter spaces $\{\Lambda^p_1, \ldots, \Lambda^p_{n_p}\}$, and $A^c = \{\alpha^c_1, \ldots, \alpha^c_{n_c}\}$ a set of classifiers with spaces $\{\Lambda^c_1, \ldots, \Lambda^c_{n_c}\}$. Given a meta-model $f_{\mathcal{M}}$, the meta-features m of a new dataset, and the predicted performance $\hat{y}_{j,l} = f_{\mathcal{M}}(m, \alpha^p_j, \alpha^c_l)$ for each preprocessor-classifier combination, the reduced algorithm set is:

$$A^*(\theta) = \big\{(\alpha^p_j, \alpha^c_l) \in A^p \times A^c : \hat{y}_{j,l} \geq Q_\theta(\hat{Y})\big\}, \tag{2}$$

where Q_θ denotes the θ-quantile of all predicted scores $\hat{Y} = \{\hat{y}_{j,l}\}$. The Dynamic Pipeline CASH problem then optimizes only over the reduced set:

$$\alpha^{p*},\ \lambda^{p*},\ \alpha^{c*},\ \lambda^{c*}\ \in\ \arg \min_{\substack{(\alpha^p_j, \alpha^c_l) \in A^*(\theta) \\ \lambda^p \in \Lambda^p_j,\ \lambda^c \in \Lambda^c_l}} \frac{1}{k} \sum_{i=1}^{k} \mathcal{L}\Big(\big(\alpha^c_l \circ \alpha^p_j\big)(\mathcal{D}^{(i)}_{\text{train}},\ \lambda^p,\ \lambda^c),\ \mathcal{D}^{(i)}_{\text{val}}\Big). \tag{3}$$

This formalization shrinks the feasible set from $|A^p| \times |A^c|$ to $|A^*(\theta)|$ combinations, which is analogous to constraining the hypothesis space $\mathcal{H}_\theta \subset \mathcal{H}_{\text{full}}$, connecting search space reduction to regularization theory.

Several AutoML systems have been developed to address the CASH problem. Auto-WEKA [Thornton et al. 2013] uses Bayesian Optimization (BO) to search over fixed-length pipelines. Auto-Sklearn [Feurer et al. 2015a] extends this approach with MtL-based warm-starting and ensemble post-processing, and its second version [Feurer et al. 2022] further reduces the need for manual configuration. TPOT [Olson et al. 2016] uses genetic programming to evolve variable-length pipeline structures. ML-Plan [Mohr et al. 2018] and RECIPE [de Sá et al. 2017] explore hierarchical planning and grammar-based evolution, respectively.

MtL is concerned with learning from the learning process itself, accumulating meta-knowledge that can be reused to improve future learning tasks [Brazdil et al. 2022]. In the context of algorithm recommendation, MtL operates on meta-datasets composed of meta-examples, where each meta-example corresponds to a dataset characterized by a set of meta-features and associated with the performance of one or more algorithms. Meta-features are numerical descriptors that capture properties of datasets and serve as input to meta-models. They have been categorized into groups such as general, statistical, information-theoretic, model-based, landmarking, and complexity measures [Rivolli et al. 2022]. Several formalizations have been proposed to systematize their extraction [Rivolli et al. 2022]. MtL has been applied to recommend preprocessing algorithms [Parmezan et al. 2017], predict whether hyperparameter tuning is necessary [Mantovani et al. 2019], initialize BO [Feurer et al. 2015b], and select classification algorithms [Brazdil et al. 2022].

Despite the progress in AutoML, relatively little attention has been given to the design of the search space itself. Most systems operate on a fixed, predefined configuration space, regardless of the task. Recent work has begun to explore search space reduction. Xue et al. [2022] proposed automated selection of both the search space and the search strategy. Borboudakis et al. [2023] introduced a sequential hyperparameter space reduction algorithm at the meta-level. Kedziora et al. [2024] explored opportunistic meta-knowledge for configuration space reduction. However, these approaches differ fundamentally from the one proposed in this thesis in three respects: (i) our framework is *optimizer-agnostic*—it constructs a reduced search space S^* that can then be used with any optimizer (Random Search, Bayesian Optimization, evolutionary methods), decoupling search space design from the optimization procedure; (ii) it addresses *end-to-end pipeline design*, jointly selecting preprocessor-classifier combinations, which is a harder problem than algorithm selection alone due to interactions between preprocessing steps and classifier inductive biases; and (iii) it is the first to formally extend the CASH problem to incorporate meta-model-driven search space selection (Dynamic Pipeline CASH), providing a theoretical foundation absent from prior work.

4. Main results

The contributions of this thesis follow a progressive research arc. It begins with a systematic review identifying meta-knowledge as AutoML's unifying thread , followed by applied studies that reinforce the critical need for precise algorithm selection and tuning. These insights motivated a benchmark of over one million pipelines to extract an unbiased meta-knowledge dataset and the development of a reproducible meta-feature extraction tool. Ultimately, these foundational steps culminate in the thesis's central contribution: a novel MtL framework that dynamically generates task-specific search spaces.

4.1. A literature review on AutoML

In Alcobaça and de Carvalho [2026], we conducted a systematic literature review that identified and analyzed 52 survey papers on AutoML, tracing the field's origins to Rice's algorithm selection problem [Rice 1976] and early work on dataset characterization and hyperparameter optimization.

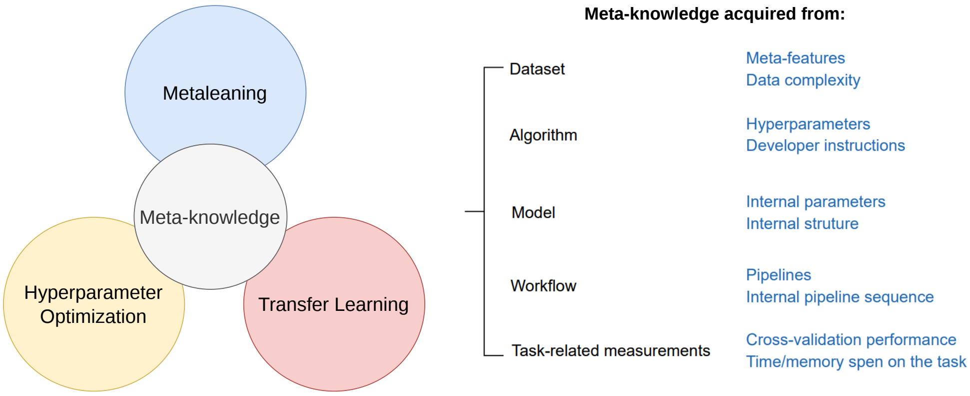

Figure 1. Timeline with distinct names adopted.

A central finding is that MtL, transfer learning, and hyperparameter optimization share the concept of meta-knowledge as a unifying thread. Although traditionally treated as separate subfields, they all rely on accumulating and reusing knowledge from past learning experiences. Figure 1 illustrates this idea. The review proposed a categorization of AutoML approaches along these three dimensions and surveyed their main applications, including algorithm recommendation, automated pipeline design, neural architecture search, and few-shot learning. Emerging research directions were also identified, such as interpretability of AutoML decisions, scalability to large datasets, and integration with federated learning. This contribution addressed **Q1** and provided the conceptual foundation upon which the remaining thesis work was built.

4.2. Applied AutoML studies for preprocessing and modeling

During the early stages of the thesis, several applied ML studies were conducted to verify, in practice, the importance of algorithm selection and hyperparameter tuning. In Alcobaça et al. [2018], we investigated dimensionality reduction in the algorithm recommendation problem, showing that linear techniques such as PCA and LDA could effectively reduce the meta-feature space without sacrificing recommendation performance. In addition, we introduced Meta-Padawan, a preprocessing-based method that combines multiple pre-trained feature descriptors designed to improve generalization in real few-shot image scenarios [El Baz et al. 2022]. This approach tied for 2nd place in the 2021 NeurIPS MetaDL competition.

In Alcobaça et al. [2020a], we applied ML to predict the glass transition temperature (T_g) of oxide glasses from chemical composition, demonstrating that properly tuned models substantially outperformed default configurations and proposing a novel visualization technique to explain the model's behavior. This framework was subsequently adopted in further studies on composition-property relationships of chalcogenide and oxide glasses [Cassar et al. 2021; Mastelini et al. 2022], evidencing its practical value beyond the ML community. These applied studies reinforced the importance of algorithm selection and hyperparameter tuning at both the preprocessing and modeling levels, as stated by **Q2**, thereby motivating the investigation of pipeline configuration interactions and search space recommendation at a larger scale.

4.3. Exploring one million ML pipelines

In Alcobaça and de Carvalho [2025b], we conducted a large-scale benchmarking study to investigate how different combinations of preprocessing techniques and classification algorithms affect pipeline performance. The study evaluated pipelines composed of 13 feature preprocessing techniques combined with 16 classification algorithms across 211 classification datasets from OpenML, resulting in over one million pipeline evaluations. Random Search was deliberately used instead of optimization-based methods to avoid introducing selection bias in the resulting meta-knowledge.

The results revealed that ensemble classifiers (particularly Gradient Boosting, Extra Trees, and AdaBoost) consistently ranked among the top performers, especially when combined with Feature Agglomeration, Polynomial Features, or no preprocessing. Conversely, pipelines involving Kernel PCA or Naive Bayes with kernel-based preprocessing consistently underperformed. This study addressed **Q2** by identifying consistent patterns of effective and ineffective pipeline configurations. Equally important, it produced a publicly available, optimization-free meta-knowledge dataset that served as the foundation for the MtL framework described in Section 4.5.

4.4. MFE: meta-feature extraction package

A critical component of MtL research is the extraction of meta-features from datasets. However, existing tools suffered from limited coverage, lack of standardization, and poor reproducibility [Rivolli et al. 2022]. To address this gap, we developed the pymfe package [Alcobaça et al. 2020b], the most comprehensive open-source tool for meta-feature extraction available to date.

The pymfe [1] package implements a formal framework for meta-feature extraction, supporting 11 meta-feature groups: general, statistical, information-theoretic, model-based, landmarking (simple, relative, and subsampling), clustering-based, concept, itemset, and complexity. Whereas the best existing alternative extracts at most 231 meta-features without a systematic framework, pymfe can extract over 3,800 when combined with multiple summary functions. The package is open-source, with n 51,000+ downloads on PyPI, and has been widely adopted by the MtL and AutoML communities. Finally, it is important to note that this package was inspired by and used in various collaborative MtL research projects during the thesis period [Mantovani et al. 2019, 2020; Garcia et al. 2020].

4.5. Dynamic design of pipeline search spaces via metalearning

The central methodological contribution of the thesis is a MtL framework for dynamically constructing task-specific search spaces for ML pipeline design [Alcobaça and de Carvalho 2025a]. Rather than searching over the entire configuration space, the framework uses a meta-model to predict the expected performance of each preprocessor-classifier

[1] `https://github.com/ealcobaca/pymfe`

combination for a new dataset, retaining only those whose predicted performance exceeds a given quantile threshold θ. This represents a shift in the current AutoML paradigm: instead of optimizing within a fixed search space, the search space itself is designed to fit the task at hand. From a theoretical standpoint, this dynamic reduction acts as a *regularization mechanism* for AutoML: by constraining the set of available algorithms, the hypothesis space shrinks from $\mathcal{H}_{\text{full}}$ to $\mathcal{H}_\theta \subset \mathcal{H}_{\text{full}}$, reducing variance at the cost of a controlled increase in bias—the standard regularization effect. The quantile threshold θ provides an explicit control parameter for this exploration-exploitation trade-off, allowing practitioners to calibrate the aggressiveness of space reduction depending on computational budget and task requirements.

The framework operates in two phases. In the offline phase, meta-features are extracted from training datasets and combined with pipeline performance records, obtained from the benchmarking study (Section 4.3) to build a meta-dataset. A meta-model is then trained to predict the performance for preprocessor-classifier pairs. In the online phase, given a new dataset, the meta-model scores all preprocessor-classifier combinations; those falling below the θ-quantile are discarded, and the remaining ones define a reduced search space for subsequent optimization, as illustrated in Figure 2. Another key methodological contribution is the introduction of *pipeline statistics meta-features*. This novel meta-feature group proved to be the most informative for the meta-model, complementing traditional dataset-level meta-features extracted via pymfe (Section 4.4).

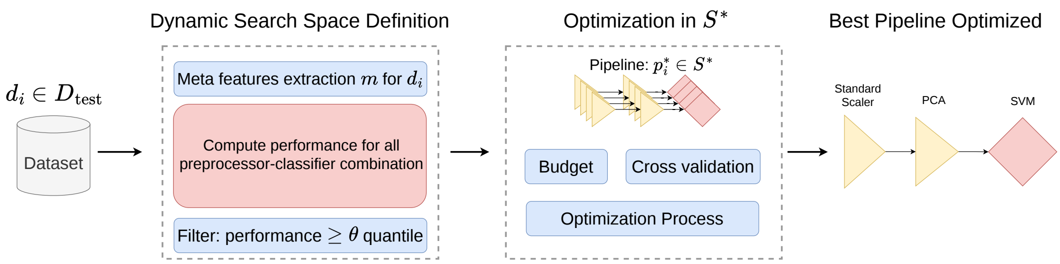

Figure 2. Framework for dynamic ML pipeline design

With a threshold of $\theta = 0.95$, the search space is reduced from 16 preprocessors and 13 classifiers to an average of 4.3 and 4.8, respectively, yielding an algorithm space reduction of approximately 68% and a runtime reduction of approximately 89% while maintaining predictive performance comparable to full-space Random Search. The framework was also integrated with Auto-Sklearn [Feurer et al. 2015a], reducing the average number of components to 7.08 out of 16 preprocessors and 6.50 out of 13 classifiers without statistically significant degradation in predictive performance. This integration demonstrates that the proposed approach is compatible with existing AutoML systems and can be readily adopted as a plug-in module built on top of scikit-learn. These contributions together with pymfe package addressed **Q3**.

4.6. Discussion

Table 1 summarizes the main contributions of the thesis by contrasting the state of the field before and after this research. From the perspective of Computer Science, the thesis made contributions at three levels: conceptual, empirical, and methodological. At the conceptual level, a systematic literature review (Section 4.1) identified meta-knowledge as a unifying concept linking MtL, transfer learning, and hyperparameter optimization, and mapped emerging research directions. At the empirical level, the thesis produced a large-scale benchmarking study (Section 4.3) of over one million pipeline configurations, revealing consistent patterns of effective and ineffective combinations of preprocessing techniques and classification algorithms, and generating a publicly available, optimization-free meta-knowledge dataset. At the methodological level, the thesis proposed a novel MtL framework for dynamically building task-specific search spaces (Section 4.5) and the introduction of pipeline statistics meta-features.

Table 1. Summary of the main contributions of the thesis.

Aspect	Before this thesis	After this thesis
AutoML literature	Fragmented view of MtL, HPO, and transfer learning as separate subfields	Unified perspective via meta-knowledge as a link between areas and systematic review
Applied AutoML Research	Limited ML-driven glass property prediction	AutoML framework adopted for oxide and chalcogenide glass studies
Benchmark + Meta-knowledge Dataset of ML pipelines	No large-scale, optimization-free pipeline performance datasets	1M+ pipeline evaluations across 211 tasks, multiple metrics, publicly available
Meta-feature Tool	Limited tools and number of meta-features, no systematic framework	pymfe: 3,800+ meta-features, 11 groups, systematic framework, 51,000+ downloads
Search Space Design	Fixed, task-agnostic configuration spaces	Dynamic, task-specific search spaces for pipeline design via MtL

Beyond its methodological contributions, the thesis has had tangible practical impact. The pymfe package (Section 4.4) has become the most comprehensive meta-feature extraction tool available, supporting the broader MtL research community. The benchmarking meta-knowledge dataset is a publicly available resource that can be directly used to train new meta-models for ML pipelines. Moreover, applied studies practical outcomes (Section 4.2) complement the thesis's theoretical and methodological advances, showing that its contributions are not only relevant to the ML community but also to applied sciences.

5. Conclusion

In this Ph.D. thesis, we proposed a paradigm shift in the automated design of ML pipelines. By moving away from fixed, task-agnostic configuration spaces, the research demonstrates that MtL can dynamically build task-specific search spaces, significantly mitigating the prohibitive computational costs of AutoML.

Conceptually, the work unifies the AutoML literature by establishing meta-knowledge as the foundational link across its subfields. Empirically, it provides a large-scale benchmark of over one million pipeline evaluations, yielding a publicly available, unbiased meta-dataset. Methodologically, it introduces a novel MtL framework that reduces the algorithm search space while maintaining highly competitive predictive performance. Finally, at the applied level, the thesis delivers tangible impact through the widely adopted pymfe package and applied ML studies.

Several directions for future work remain open. The dynamic search space framework could be extended to regression and multi-target learning tasks. The pymfe package could be expanded to support parallel extraction and regression-specific meta-features. Finally, incorporating user feedback into the search space design process through human-in-the-loop mechanisms would combine the efficiency of automated methods with domain expertise.

References

Alcobaça, E. and de Carvalho, A. C. (2025a). Dynamic design of machine learning pipelines via metalearning. *arXiv preprint arXiv:2508.13436*.

Alcobaça, E. and de Carvalho, A. C. (2025b). Exploring one million machine learning pipelines: A benchmarking study. In *International Conference on Automated Machine Learning*, pages 22–1. PMLR.

Alcobaça, E. and de Carvalho, A. C. (2026). A literature review on automated machine learning. *Artificial Intelligence Review*, 59(1):1–39.

Alcobaça, E., Mantovani, R. G., Rossi, A. L., and De Carvalho, A. C. (2018). Dimensionality reduction for the algorithm recommendation problem. In *2018 7th Brazilian Conference on Intelligent Systems (BRACIS)*, pages 318–323. IEEE.

Alcobaça, E., Mastelini, S. M., Botari, T., Pimentel, B. A., Cassar, D. R., de Leon Ferreira, A. C. P., Zanotto, E. D., et al. (2020a). Explainable machine learning algorithms for predicting glass transition temperatures. *Acta materialia*, 188:92–100.

Alcobaça, E., Siqueira, F., Rivolli, A., Garcia, L. P. F., Oliva, J. T., de Carvalho, A. C., et al. (2020b). Mfe: Towards reproducible meta-feature extraction. *Journal of Machine Learning Reseach*, 21:111–1.

Borboudakis, G., Charonyktakis, P., Paraschakis, K., and Tsamardinos, I. (2023). A meta-level learning algorithm for sequential hyper-parameter space reduction in automl. *arXiv preprint arXiv:2312.06305*.

Brazdil, P., Van Rijn, J. N., Soares, C., and Vanschoren, J. (2022). *Metalearning: Applications to automated machine learning and data mining*. Springer Nature.

Cassar, D. R., Mastelini, S. M., Botari, T., Alcobaca, E., de Carvalho, A. C., and Zanotto, E. D. (2021). Predicting and interpreting oxide glass properties by machine learning using large datasets. *Ceramics international*, 47(17):23958–23972.

de Sá, A. G., Pinto, W. J. G., Oliveira, L. O. V., and Pappa, G. L. (2017). Recipe: a grammar-based framework for automatically evolving classification pipelines. In *European Conference on Genetic Programming*, pages 246–261. Springer.

El Baz, A., Ullah, I., Alcobaça, E., Carvalho, A. C., Chen, H., Ferreira, F., Gouk, H., Guan, C., Guyon, I., Hospedales, T., et al. (2022). Lessons learned from the neurips 2021 metadl challenge: Backbone fine-tuning without episodic meta-learning dominates for few-shot learning image classification. In *NeurIPS 2021 Competitions and Demonstrations Track*, pages 80–96. PMLR.

Fabris, F. and Freitas, A. A. (2019). Analysing the overfit of the auto-sklearn automated machine learning tool. In *International Conference on Machine Learning, Optimization, and Data Science*, pages 508–520. Springer.

Feurer, M., Eggensperger, K., Falkner, S., Lindauer, M., and Hutter, F. (2022). Auto-sklearn 2.0: Hands-free automl via meta-learning. *Journal of Machine Learning Research*, 23(261):1–61.

Feurer, M., Klein, A., Eggensperger, K., Springenberg, J. T., Blum, M., and Hutter, F. (2015a). Efficient and robust automated machine learning. In *Advances in Neural Infor-*

mation Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada, pages 2962–2970.

Feurer, M., Springenberg, J. T., and Hutter, F. (2015b). Initializing bayesian hyperparameter optimization via meta-learning. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA*, pages 1128–1135.

Garcia, L. P., Rivolli, A., Alcobaça, E., Lorena, A. C., and de Carvalho, A. C. (2020). Boosting meta-learning with simulated data complexity measures. *Intelligent Data Analysis*, 24(5):1011–1028.

Hutter, F., Kotthoff, L., and Vanschoren, J., editors (2019). *Automated Machine Learning - Methods, Systems, Challenges*. The Springer Series on Challenges in Machine Learning. Springer.

Kedziora, D. J., Nguyen, T.-D., Musial, K., and Gabrys, B. (2024). On taking advantage of opportunistic meta-knowledge to reduce configuration spaces for automated machine learning. *Expert Systems with Applications*, 239:122359.

Mantovani, R. G., Rossi, A. L., Alcobaça, E., Vanschoren, J., and de Carvalho, A. C. (2019). A meta-learning recommender system for hyperparameter tuning: Predicting when tuning improves svm classifiers. *Information Sciences*, 501:193–221.

Mantovani, R. G., Rossi, A. L. D., Alcobaça, E., Gertrudes, J. C., Junior, S. B., and de Carvalho, A. C. P. d. L. F. (2020). Rethinking default values: A low cost and efficient strategy to define hyperparameters. *arXiv preprint arXiv:2008.00025*.

Mastelini, S. M., Cassar, D. R., Alcobaça, E., Botari, T., de Carvalho, A. C., and Zanotto, E. D. (2022). Machine learning unveils composition-property relationships in chalcogenide glasses. *Acta Materialia*, 240:118302.

Mohr, F., Wever, M., and Hüllermeier, E. (2018). Ml-plan: Automated machine learning via hierarchical planning. *Machine Learning*, 107(8-10):1495–1515.

Olson, R. S., Bartley, N., Urbanowicz, R. J., and Moore, J. H. (2016). Evaluation of a tree-based pipeline optimization tool for automating data science. In Friedrich, T., Neumann, F., and Sutton, A. M., editors, *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference*, pages 485–492. ACM.

Parmezan, A. R. S., Lee, H. D., and Wu, F. C. (2017). Metalearning for choosing feature selection algorithms in data mining: Proposal of a new framework. *Expert Systems with Applications*, 75:1–24.

Rice, J. R. (1976). The algorithm selection problem. *Advances in Computers*, 15:65–118.

Rivolli, A., Garcia, L. P., Soares, C., Vanschoren, J., and de Carvalho, A. C. (2022). Meta-features for meta-learning. *Knowledge-Based Systems*, 240:108101.

Thornton, C., Hutter, F., Hoos, H. H., and Leyton-Brown, K. (2013). Auto-weka: combined selection and hyperparameter optimization of classification algorithms. In *The 19th ACM SIGKDD, 2013*, pages 847–855.

Xue, C., Hu, M., Huang, X., and Li, C.-G. (2022). Automated search space and search strategy selection for automl. *Pattern Recognition*, 124:108474.