Machine Learning Research
2022; 7(1): 1-7
http://www.sciencepublishinggroup.com/j/mlr
doi: 10.11648/j.mlr.20220701.11
ISSN: 2637-5672 (Print); ISSN: 2637-5680 (Online)

Automated Machine Learning Models and State-Of-The-Art Effort in Mitigating Combined Algorithm Selection and Hyperparameter Optimization Problems: A Review

Nwokonkwo Obi Chukwuemeka[1], John-Otumu Adetokunbo MacGregor[2], Nnadi Leonard Chukwualuka[1], Ogene Ferguson[3]

[1]Department of Information Technology, Federal University of Technology, Owerri, Nigeria

[2]Department of Computer Science, Morgan State University, Baltimore, USA

[3]Department of Computer Science & Info Tech, Petroleum Training Institute, Effurun, Nigeria

Email address:
obi.nwokonkwo@futo.edu.ng (Nwokonkwo O. C.), adetokunbo.otumu@morgan.edu (John-Otumu A. M.), leonardnnadi@gmail.com (Nnadi L. C.), ogene_f@pti.edu.ng (Ogene F.)

To cite this article:
Nwokonkwo Obi Chukwuemeka, John-Otumu Adetokunbo MacGregor, Nnadi Leonard Chukwualuka, Ogene Ferguson. Automated Machine Learning Models and State-Of-The-Art Effort in Mitigating Combined Algorithm Selection and Hyperparameter Optimization Problems: A Review. *Machine Learning Research.* Vol. 7, No. 1, 2022, pp. 1-7. doi: 10.11648/j.mlr.20220701.11

Received: December 20, 2021; **Accepted**: January 8, 2022; **Published**: January 14, 2022

Abstract: Automated machine learning (AutoML) models is one of several machine learning algorithms that can be used to automate the solution of real-world problems. It automates the selection, composition, and parameterization processes of the machine learning models in particular. Machine learning could be more user-friendly when it is automated, and it often produces faster, and more accurate results than hand-coded machine learning methods. For more than ten years, AutoML for supervised learning has been the main focus of research under the discipline of artificial intelligence, and significant progress has been made theeafter; consider the usefulness of AutoML methods in the most popular machine learning toolkits, as well as the AutoML mechanisms in large scale platforms such as Microsoft Azure. This paper provides a methodical analysis of the AutoML workflow as well as the state-of-the-art effort in dealing with the challenges involving Combined Algorithm Selection and Hyperparameter Optimization by gathering information about AutoML from several published articles from different online repositories in order to delve more into the methods used in different domains and the level of accuracy obtained. Findings revealed that the next generation of machine learning and artificial intelligence research is focused on automating the other phases of the whole end-to-end machine learning pipeline, from data comprehension to model deployment. With significantly better deep learning algorithms and big datasets, AutoML is predicted to be able to handle most of the data cleaning process in the future. AutoML will evolve into a highly human-competitive system that will change the way we think about data research.

Keywords: Transfer Learning, Machine Learning, Hyperparameter, Automation, Artificial Intelligence

1. Introduction

Machine learning (ML) is becoming increasingly popular in a variety of fields for the creation of machine learning models, the requirement for fast data extraction, model training, model assessment, and model deployment (pipeline) has never been more critical [1]. Many corporate and governmental organizations have realized that data analysis is a valuable tool for acquiring understandings on exactly how to enhance their corporate model, making decision-, and even goods [2]. The technique of applying ML models to real-world issues via automation is known as automated machine learning (AutoML) [3]. AutoML automates the selection, construction, and parameterization of ML models. ML algorithms that are automated are more user-friendly and typically provide faster, more accurate results than hand-coded algorithms [3].

ML has also advanced significantly. For example, in the game of GO, AlphaGO [4] defeated the human champion. According to He, K. et al. [5] ResNet outperformed humans in image recognition, and Microsoft's voice system came close to matching human speech transcription [6].

AutoML using classification techniques under supervised learning has remained the main crux of research for over 10 years and significant progress has been made in terms of its utility [7]. The AutoML processes in huge scaled systems or platforms such as Microsoft Azure or H2O.ai and the AutoML systems in the extremely widespread machine learning toolkits [8-10]. Indeed, AutoML is a serious issue in ML today, with a lot of interest from industry, academia, and even the general public.

In this study, we give a methodical assessment of the AutoML workflow as well as a state of the art (SOTA) effort in dealing with the problems of Combined Algorithm Selection and Hyperparameter Optimization (CASH). The study highlights the remaining steps of the whole end-to-end ML pipeline from data pre-modeling all the way through to post modeling.

1.1. AutoML Workflow

AutoML is a platform or open-source library that automates each stage in the ML process, from handling a raw dataset to deploying a realistic ML model. In classical ML, models are built by hand, and each stage of the process must be handled separately [3].

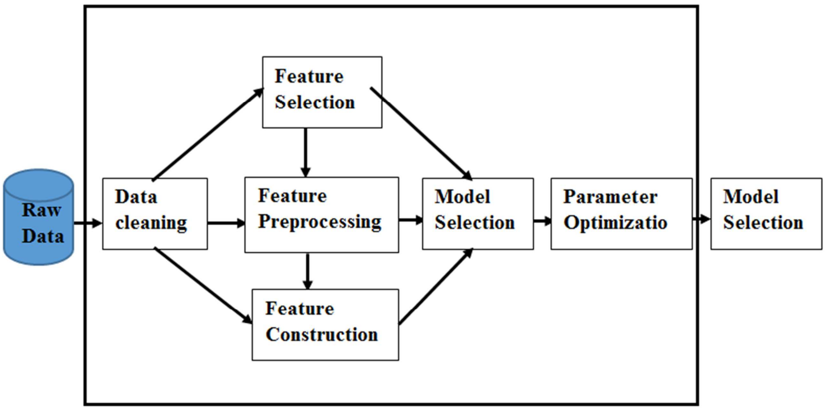

Figure 1. *AutoML Workflow [11].*

AutoML identifies and applies the best ML algorithm for a given task. It accomplishes this by employing two concepts:

a. Neural architecture search, which aids in the creation of neural networks by automating the process. This makes it easier for AutoML models to find new architectures for situations that require them.

b. Transfer learning, in which previously trained models apply their knowledge to fresh data sets. AutoML can use transfer learning to apply existing structures to new problems.

Users with only a rudimentary understanding of ML and Deep Learning (DL), Artificial Intelligence (AI) experts can interact with the models using a programming language such as Python.

Neural Architecture Search (NAS)

In recent years, DL has made tremendous progress in a variety of applications, including speech recognition, machine translation and image recognition. Deep convolutional neural networks (DCNN) are a critical component of this progress [12]. The NAS is a tool for computerizing architectural planning, is the next step of in the automation of machine learning. NAS approaches have already surpassed manually created systems on various tasks, including image classification, object recognition, and semantic segmentation [13].

In Feurer & Hutter [21], NAS is a subset of AutoML that has a lot in common with hyperparameter optimization and meta-learning. The three aspects by which we characterize NAS approaches are the search space, search tactic, and routine assessment approach (Elsken et al., 2019).

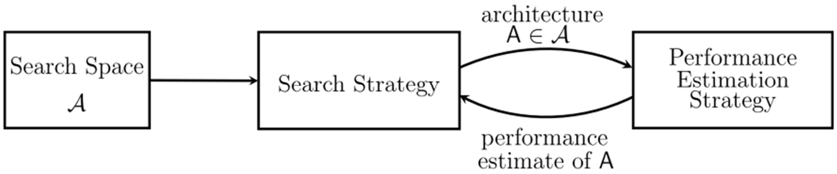

Figure 2. *NAS Techniques [12].*

A search technique chooses a design “A” from a prearranged search space “A” in Figure 2. The architecture is given to a performance evaluation approach, which returns the projected performance of “A” to the search approach.

1.1.1. Searching for Space

The search space defines which architectural representations are theoretically viable. Understanding the characteristics of well-suited designs can help to restrict the search area and streamline the process. However, this presents a human bias, which may obstruct the discovery of novel architectural building blocks that go away from human comprehension.

1.1.2. Search Methodology

The search strategy outlines exactly how to look for the search space for information (which is regularly exponentially huge or even boundless).

It addresses the classic exploration-exploitation trade-off, in which it is desirable to quickly identify high-performing designs while avoiding early convergence to a region of low-performing structures.

1.1.3. Performance Estimation Strategy

The purpose of NAS is normally to discover designs that work effectively with data that has never been seen before. The most straightforward way is to train and validate the architecture on data, but this is computationally expensive and restricts the number of architectures that may be investigated.

The easiest choice is to achieve a basic training and confirmation of the architecture on data, but this is computationally costly and reduces the number of designs that can be investigated. Performance Evaluation belongs to the process of valuing this operation: the simplest option is to perform a standard training and validation of the architecture on data, but this is computationally expensive and limits the number of designs that can be investigated. As a result, a lot of recent research has been devoted to figuring out how to reduce the cost of these performance estimations.

1.2. Transfer Learning

In ML, transfer learning (TL) refers to the reuse of a formerly trained model on a new-found problem. In transfer learning, a computer leverages information from a previous assignment to improve generalization about a new task. For example, while training a classifier to predict whether a picture contains food, you may use the knowledge gathered during training to recognize beverages [14]. It is therefore possible to use what was learnt in one activity to help us generalize on what we have learnt in another. The weights learned by a network at "task A" are transferred to a new "task B.".

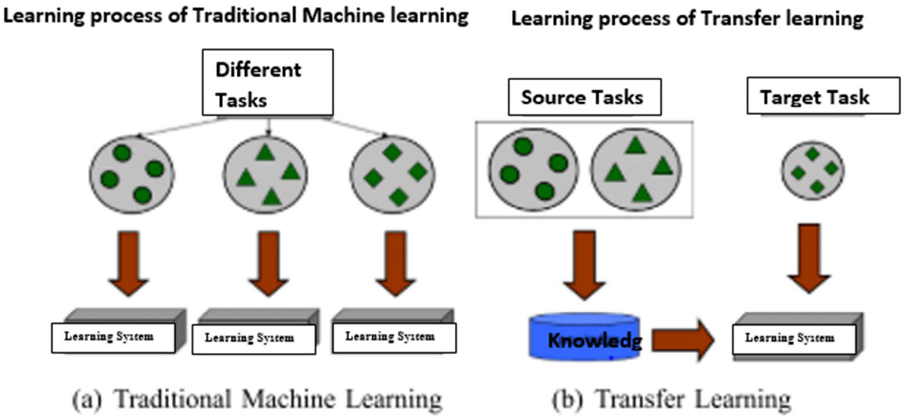

Figure 3. *Different Learning Processes between Traditional Machine Learning and Transfer Learning [15].*

Table 1. *Traditional Machine Learning Vs Various Transfer Learning Settings [15].*

Learning Settings		Source and Target Domain	Source and Target Tasks
Traditional Machine Learning		The same	The same
Transfer Learning	Inductive Transfer Learning	The same	Different but related
		Different but related	Different but related
	Transductive Transfer Learning	Different but related	The same

Table 2. *Different Settings of Transfer Learning [15].*

Transfer Learning Settings	Related Areas	Source Domain Labels	Target Domain Labels	Tasks
Inductive Transfer Learning	Multi-task Learning	Available	Available	Regression Classification
	Self-taught Learning	Unavailable	Available	Regression Classification
Transductive Transfer Learning	Domain Adaptation, Sample Selection Bias, Co-variate Shift	Available	Unavailable	Regression Classification
Unsupervised Transfer Learning		Unavailable	Unavailable	Clustering, Dimensionality Reduction

2. Related Works

This section reviewed some related works done by different research scholars under Automated Machine Learning, domain, methodology, packages, and programming languages implemented.

Table 3. *Methodical review on related works.*

S/N	Domain	Methods	Findings
1	Automated Feature Generation and Selection [16]	AutoLearn Regression based feature learning Python	Their experimental evaluation of the features learned through our model on 18 UC Irvine and 7 Gene expression datasets across different domains shows that the features learned through our model can improve overall prediction accuracy by 13.28 percent compared to original feature space and 5.87 percent compared to other top performing models, across 8 different classifiers without using machine learning of any domain knowledge.
2	Automated ML for predictive quality in production [17]	Python, AutoML Preprocessing + Auto-sklearn	AutoML still necessitates programming skills and is surpassed by manual implementation, which has an F1 value of 73 percent compared to AutoML's F1 value of 48 percent.
3	Pipeline Selection [18]	TensorOboe Python	When compared to Decision tree, Extra tree, Gradient boosting, Gaussian naive Bayes, KNN, Logistic regression, Multilayer perceptron, Perceptron, Random Forest Linear, and SVM for runtime prediction accuracy on OpenML datasets within a factor of 2 and within a factor of 4, Adaboost has a better accuracy of 73.6 percent and 86.9%, respectively.
4	A distributed, collaborative, and scalable system for automated ML [19]	ATM: Auto Tuned Model Python	They demonstrated the usefulness of ATM on 420 datasets from OpenML and train over 3 million classifiers. Their initial results show ATM can beat human-generated solutions for 30% of the datasets, and can do so in $1/100^{th}$ of the time.
5	Neural AutoML for DL [20]	LEAF	LEAF is capable of outperforming current state-of-the-art AutoML systems as well as the best hand-crafted solutions.
6	Combined Selection and Hyperparameter Optimization of Classification Algorithms [9]	AUTO-WEKA Waikato Environment for Knowledge Analysis (Weka)	The performance of Auto-WEKA classification is frequently superior to that of traditional selection and hyperparameter optimization approaches.
7	Hyperparameter Optimization [21]	Model-free methods Multi-fidelity Optimization Bayesian optimization	
8	DL pipeline [24]	Neural Architecture Search (NAS) Hyperparameter optimization (HPO) Python	Based on the PTB dataset and state-of-the-art models. The better the performance, the smaller the perplexity. It reveals that the auto NAS model has a difficulty of 64, but the human experts' model has a perplexity of 54.55.
9	Tree-based Pipeline Optimization Tool for Automating Data Science [11]	Tree-based Pipeline Optimization Tool (TPOT) Python	With larger data sets, their trials attain a categorization accuracy of 80%.
10	Automated ML Techniques for Data Streams [22]	meta-learning technique for online algorithm selection based on meta- feature extraction	The meta-learning strategy used by a heterogeneous ensemble of online learners produced better results than some of the online learners working alone, but it was worse than the online ensembles using ADMIN.
11	Automated ML via hierarchical planning [23]	ML-plan	According to their research, ML-Plan implementation is very competitive and frequently beats state-of-the-art techniques such as Auto-WEKA, auto-sklearn, and TPOT.
12	TL in Collaborative Filtering for Sparsity Reduction [25]	matrix-based transfer learning framework, coordinate system transfer or CST	The effectiveness of coordinate system transfer, or CST, in alleviating the data sparsity problem in collaborative filtering. Several state-of-the-art methods for this problem are greatly outperformed by their strategy.
13	Auto-sklearn: Efficient and Robust AutoML [8]	Python Auto-sklearn	Their system won six of the ten phases of the first ChaLearn AutoML challenge, and a thorough study of over 100 different datasets reveals that it significantly beats the prior state of the art in AutoML.
14	Automating feature Engineering in relational databases [26]	One Button Machine, or OneBM for short	OneBM was validated in Kaggle contests, where it performed as well as the top 16 percent to 24 percent of data scientists in three competitions. In a Kaggle Competition, OneBM outperformed the state-of-the-art system in terms of prediction accuracy and placing on the Kaggle scoreboard.
15	Automatic Feature Generation and Selection [27]	ExploreKit Java	They show that ExploreKit can achieve classification-error reduction of 20% overall.
16	Learning Feature Engineering for Classification [28]	Learning Feature Engineering (LFE)	Their empirical findings reveal that LFE beats other feature engineering approaches for the vast majority of datasets (89 percent) while incurring a significantly lower computing cost.
17	Predictive Entropy Search for Multi-objective Bayesian Optimization with Constraints [29]	Predictive Entropy Search for Multi-objective Bayesian Optimization with Constraints	PESMOC is able to deliver better recommendations with a lesser number of assessments than a random search method, according to the results.
18	A Novel Bandit-Based Approach to Hyperparameter Optimization [30]	Hyperband	On a set of hyperparameter optimization problems, Hyperband used popular Bayesian optimization methods. On a range of deep-learning and kernel-based learning issues, they find that Hyperband can deliver an order-of-magnitude speedup over our competitor set.
19	NAS [31]	Simple regression	They demonstrate that their performance prediction models and early stopping method are cutting-edge in terms of prediction accuracy and

S/N	Domain	Methods	Findings
			speedup while still identifying the best model configurations.
20	Towards Automatically Tuned Neural Networks [32]	Auto-Net Auto-sklearn Python	The combination of Auto-Net and Auto-sklearn generally outperforms either alone, reporting the first results on winning competition datasets versus human specialists using automatically tuned neural networks.

3. Combined Model Selection and Hyperparameter Optimization Problems

Using a combination of the building blocks, such as feature engineering, Meta-learning, and Architectural search, the difficulty of coupled model selection and hyperparameter optimization can be overcome. Finally, a thorough solution identifies the ideal ML pipeline for raw (unprocessed) feature vectors in the briefest time for a given number of computer resources. It is noted that since 2015 a series of AutoML competitions has been organized every year [33]. A comprehensive pipeline consists of data cleaning, feature engineering (selection and creation), model selection, hyperparameter tuning, and finally creating an ensemble of the best trained models to achieve excellent performance on unseen test data. Optimizing the entire ML pipeline which is not certainly differentiable from start to finish is a tough task, and several methodologies have been employed to investigate various options [2].

Hyperparameter Optimization

Bayesian optimization, which is a vital stage in tackling the entire CASH issue, is the most notable example of comparable methodologies. The goal is to develop a model that can forecast loss and variation for each input. After each optimization phase, the model (or present belief) is updated using a posteriori knowledge, hence the name Bayesian [2].

To pick where to sample the next real loss, a sampling function is built that trades off areas of low expected loss (exploitation) with regions of high variation (exploration). Gaussian Processes are commonly used in Bayesian optimization; however, Random Forests have been used to model the loss surface of the hyperparameters as a Gaussian distribution in Sequential Model-based optimization for general Algorithm Configuration (SMAC) or the Tree-structured [34].

Successive Halving [35], and its built-on Hyperband [30] are model-free approaches that utilize real-time optimization progress to narrow down a set of competing hyperparameter configurations throughout the course of a whole optimization cycle, perhaps with multiple restarts. Evolutionary Strategies are a small variant on this, since they allow for perturbations of particular configurations during training. Multiple iterations of the optimizer can be unrolled in the particular situation when both the optimize and the optimizer are differentiable, and an update for the hyperparameters can be calculated using gradient descent and backpropagation [36].

Pipeline Optimization

Various perspectives on the topic are used to develop entire ML pipelines, as well as feature prepossessing, hyperparameter tweaking, model selection, and ensemble formation. Various pipeline optimization strategies were inspired by Bayesian optimization, genetic programming, and binary optimization.

Auto-sklearn uses meta-learning, Bayesian optimization, and ensemble building to solve the CASH issue. It begins by extracting meta-features from a new dataset, such as job type (classification or regression), number of classes, feature vector dimensions, sample count, and so on. These meta-features are used by auto-meta-learner sklearn's to start the optimization process based on previous experience with similar data sets (similar according to the Meta features). Following that, Bayesian optimization is used to optimize the preprocessing and model hyperparameters on a regular basis. Finally, an ensemble of models trained through iterative optimization is used to produce a robust classifier or regression model [2].

Tree-based Pipeline Optimization Tool (TPOT) is a genetic programming-based ML pipeline optimization technique. It seeks out the best pipeline for a given classification or regression task, considering feature processing, model, and hyperparameters. The feature processing module in TPOT works in conjunction with the feature selection and generation modules. The feature construction block does the kernel trick [2] or dimensionality reduction procedures. The pipelines are optimized via genetic programming: the method generates many tree-based pipelines at random at first. The top 20% of the population is then picked based on cross-validation accuracy, and 5 descendants are generated from each by changing a point in the pipeline at random. The algorithm continues until a halting requirement is met. Finally, the ATM framework according to Swearingen, T. et al. [19] combines multi-armed bandit optimization and hybrid Bayesian optimization to discover the finest models.

4. Conclusion

Automated machine learning's purpose is to make it easier for users to create machine learning systems. AutoML is a well-recognized field with extensive applicability in the data science era, ranging from the optimization of fixed model hyperparameters through model type selection and whole model/pipeline development, as well as the autonomous construction of deep learning architectures. Significant progress was made in the early years, with very successful techniques readily available to those with only rudimentary ML skills. Similarly, strategies can make even machine learning experts' design jobs easier. In this research, we present a systematic investigation, AutoML approach, and state-of-the-art (SOTA) effort in dealing with the CASH Optimization problems. We can now employ AutoML techniques to solve problems that previously required a lot of effort. AutoML has a lot to look forward to in the years ahead, given the progress made in the field of automated machine learning thus far. Approaches to addressing explainable AutoML models, AutoML in feature engineering, AutoML for non-tabular data, Large-scale AutoML, and AutoML Transfer Learning will all be exciting to learn about with respect to deep learning.

5. Recommendations

AutoML is expected to be able to handle most of the data cleaning processes in the future by employing considerably improved deep learning algorithms with massive datasets. AutoML will become a highly human-competitive system that will revolutionize data science as we know it.

References

[1] Chauhan, K., Jani, S., Thakkar, D., Dave, R., Bhatia, J., Tanwar, S., & Obaidat, M. S. (2020). *Automated Machine Learning: The New Wave of Machine Learning*. (Icimia), 205–212.

[2] Tuggener, L., Amirian, M., Rombach, K., Stefan, L., Varlet, A., Westermann, C., … Ag, P. (2019). *Automated Machine Learning in Practice: State of the Art and Recent Results*. 31–36. https://doi.org/10.1109/SDS.2019.00-11.

[3] Lutkevich, B. (2020). What is automated machine learning (AutoML). Retrieved from https://searchenterpriseai.techtarget.com/definition/automated-machine-learning-AutoML.

[4] Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., and Hassabis, D. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature*, *529* (7587), 484–489. https://doi.org/10.1038/nature16961.

[5] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2016-Decem*, 770–778. https://doi.org/10.1109/CVPR.2016.90.

[6] Yao, Q., Wang, M., Chen, Y., Dai, W., Li, Y., Tu, W., … Yu, Y. (2018). *Taking Human out of Learning Applications: A Survey on Automated Machine Learning*. 1–20. Retrieved from http://arxiv.org/abs/1810.13306.

[7] Mirtaheri, S. M., & Dinc, M. E. (2014). *A Brief History of Web Crawlers*. (May).

[8] Feurer, M., Klein, A., Eggensperger, K., Springenberg, J. T., Blum, M., & Hutter, F. (2019). *Auto-sklearn: Efficient and Robust Automated Machine Learning*.

[9] Thornton, C., & Leyton-brown, K. (2013). *Auto-WEKA: Combined Selection and Hyperparameter Optimization of Classification Algorithms*. 847–855.

[10] Escalante, H. J. (2020). *Automated Machine Learning -- a brief review at the end of the early years*. 1–17. Retrieved from http://arxiv.org/abs/2008.08516.

[11] Olson, R. S., Bartley, N., Urbanowicz, R. J., & Moore, J. H. (2016). *Evaluation of a Tree-based Pipeline Optimization Tool for Automating Data Science*.

[12] Elsken, T., Metzen, J. H., & Hutter, F. (2019). *Neural Architecture Search: A Survey*. *20*, 1–21.

[13] Cai, H., Chen, T., Zhang, W., Yu, Y., & Wang, J. (2017). *Efficient Architecture Search by Network Transformation*. 2787–2794.

[14] Donges, N. (2021). What Is Transfer Learning A Simple Guide Built In. Retrieved from https://builtin.com/data-science/transfer-learning.

[15] Pan, S. J., & Fellow, Q. Y. (2009). *A Survey on Transfer Learning*. 1–15.

[16] Kaul, A., Maheshwary, S., & Pudi, V. (2017). *AutoLearn - Automated Feature Generation and Selection*. https://doi.org/10.1109/ICDM.2017.31.

[17] Krauß, J., Pacheco, B. M., Zang, H. M., Heinrich, R., Design, C., Stief, P., … Siadat, A. (2020). ScienceDirect ScienceDirect methodology to analyze the functional and physical architecture Automated machine learning for predictive quality in production existing products for an assembly oriented product family identification. *Procedia CIRP*, *93*, 443–448. https://doi.org/10.1016/j.procir.2020.04.039.

[18] Yang, C., Fan, J., Wu, Z., & Udell, M. (2020). *AutoML Pipeline Selection: Efficiently Navigating the Combinatorial Space*.

[19] Swearingen, T., Drevo, W., Cyphers, B., Cuesta-Infante, A., Ross, A., & Veeramachaneni, K. (2017). ATM: A distributed, collaborative, scalable system for automated machine learning. *Proceedings - 2017 IEEE International Conference on Big Data, Big Data 2017*, *2018-January* (December), 151–162. https://doi.org/10.1109/BigData.2017.8257923.

[20] Liang, J., Meyerson, E., Hodjat, B., Fink, D., Mutch, K., & Miikkulainen, R. (2019). Evolutionary neural automl for deep learning. *GECCO 2019 - Proceedings of the 2019 Genetic and Evolutionary Computation Conference*, 401–409. https://doi.org/10.1145/3321707.3321721.

[21] Feurer, M., & Hutter, F. (2019). *Hyperparameter Optimization*. 3–33.

[22] Imbrea, A. (2021). *Automated Machine Learning Techniques for Data Streams*. Retrieved from http://arxiv.org/abs/2106.07317.

[23] Mohr, F., Wever, M., & Hüllermeier, E. (2018). ML-Plan: Automated machine learning via hierarchical planning. *Machine Learning*. https://doi.org/10.1007/s10994-018-5735-z.

[24] He, X., Zhao, K., & Chu, X. (2021). AutoML: A survey of the state-of-the-art. *Knowledge-Based Systems*, *212* (Dl). https://doi.org/10.1016/j.knosys.2020.106622.

[25] Pan, W., Xiang, E. W., Liu, N. N., & Yang, Q. (2010). *Transfer Learning in Collaborative Filtering for Sparsity Reduction*. 230–235.

[26] Lam, H. T., Thiebaut, J., & Sinn, M. (2016). *One button machine for automating feature engineering in relational databases*.

[27] Katz, G., & Song, D. (2016). *ExploreKit: Automatic Feature Generation and Selection*. https://doi.org/10.1109/ICDM.2016.176.

[28] Nargesian, F., Samulowitz, H., Khurana, U., Khalil, E. B., & Turaga, D. (2017). *Learning Feature Engineering for Classification*. 2529–2535.

[29] Garrido-merch, E. C., Tom, F., Hern, D., & Tom, F. (2016). *Predictive Entropy Search for Multi-objective Bayesian Optimization with Constraints*.

[30] Li, L., Jamieson, K., & Desalvo, G. (2018). *Hyperband: A Novel Bandit-Based Approach to*. *18*, 1–52.

[31] Baker, B., Gupta, O., and Raskar, N. N. (2018). Accelerating neural architecture search using performance prediction. *6th International Conference on Learning Representations, ICLR 2018 - Workshop Track Proceedings*, 1–19.

[32] Mendoza, H. (2016). *Towards Automatically-Tuned Neural Networks*. 1–8.

[33] Guyon, I., Sun-hosoya, L., Boullé, M., Escalante, H., Liu, Z., Jajetic, D., … Guyon, I. (2018). *Analysis of the AutoML Challenge series 2015-2018 To cite this version: HAL Id: hal-01906197 Analysis of the AutoML Challenge series 2015-2018*.

[34] Hutter, F., Hoos, H. H., & Leyton-brown, K. (2011). *Sequential Model-Based Optimization for General Algorithm Configuration*.

[35] Jamieson, K., & Talwalkar, A. (2016). Non-stochastic best arm identification and hyperparameter optimization. *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, AISTATS 2016*, 240–248.

[36] Maclaurin, D., Duvenaud, D., and R. A. (2015). Hyper-Parameter Initialization for Squared Exponential Kernel-based Gaussian Process Regression Nalika Ulapane - Academia. Retrieved from https://www.academia.edu/45378935/Hyper_Parameter_Initialization_for_Squared_Exponential_Kernel_based_Gaussian_Process_Regression.