



HAL
open science

AMLBID: An auto-explained Automated Machine Learning tool for Big Industrial Data

Moncef Garouani, Adeel Ahmad, Mourad Bouneffa, Mohamed Hamlich

► To cite this version:

Moncef Garouani, Adeel Ahmad, Mourad Bouneffa, Mohamed Hamlich. AMLBID: An auto-explained Automated Machine Learning tool for Big Industrial Data. *SoftwareX*, 2022, 17, pp.100919. 10.1016/j.softx.2021.100919 . hal-03484800

HAL Id: hal-03484800

<https://hal.science/hal-03484800v1>

Submitted on 19 Sep 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NoDerivatives 4.0 International License



Original software publication

AMLBID: An auto-explained Automated Machine Learning tool for Big Industrial Data

Moncef Garouani ^{a,b,c,*}, Adeel Ahmad ^a, Mourad Bouneffa ^a, Mohamed Hamlich ^b

^a Univ. Littoral Cote d'Opale, UR 4491, LISIC, Laboratoire d'Informatique Signal et Image de la Cote d'Opale, F-62100 Calais, France

^b CCPS Laboratory, ENSAM, University of Hassan II, Casablanca, Morocco

^c Study and Research Center for Engineering and Management(CERIM), HESTIM, Casablanca, Morocco



ARTICLE INFO

Article history:

Received 15 June 2021

Received in revised form 22 November 2021

Accepted 22 November 2021

Keywords:

Machine learning

AutoML

Meta-learning

Decision-support systems

Explainable AI

Big industrial data

ABSTRACT

The Machine Learning (ML) based solutions in manufacturing industrial contexts often require skilled resources. More practical non-expert software solutions are then desired to enhance the usability of ML algorithms. The algorithm selection and configuration is one of the most difficult tasks for users like manufacturing specialists. The identification of the most appropriate algorithm in an automatic manner is among the major research challenges to achieve optimal performance of ML tools. In this paper, we present an auto-explained Automated Machine Learning tool for Big Industrial Data (AMLBID) to better cope with the prominent challenges posed by the evolution of Big Industrial Data. It is a meta-learning based decision support system for the automated selection and tuning of implied hyperparameters for ML algorithms. Moreover, the framework is equipped with an explainer module that makes the outcomes transparent and interpretable for well-performing ML systems.

© 2021 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

Code metadata

Current code version	v0.1
Permanent link to code/repository used for this code version	https://github.com/ElsevierSoftwareX/SOFTX-D-21-00111
Code Ocean compute capsule	https://codeocean.com/capsule/9828764/tree
Legal Code License	MIT License
Code versioning system used	git
Software code languages, tools, and services used	Python
Compilation requirements, operating environments & dependencies	Python 3.x; Jupyter; dash; sklearn; See requirements file
If available Link to developer documentation/manual	https://github.com/LeMGarouani/AMLBID/blob/main/README.md
Support email for questions	mgarouani@gmail.com

1. Motivation and significance

Industrial Big Data refers to the large amount of diversified data that is generated continuously, in real time by the network of industrial equipment [1]. The continuous digital transformation of the manufacturing industry has led to the widespread adoption of ML solutions [2]. Although, in many industrial areas,

the ML sufficiently assists the large data analysis for decision-making purposes, the human interventions are often required. The domain experts master better the application area. For example, the domain experts are able to provide characteristics of the application, which can help to improve the performance of the algorithms. However, they are not necessarily ML experts. Consequently, the large number of available ML algorithms and hyperparameters configurations could lead to infeasible exhaustive search executions. Therefore, in this context, the expertise of data-scientists is highly desired for the identification of the most appropriate algorithm configurations [3,4].

The selection of an algorithm or a family of algorithms that are more likely to perform better on a given combination of

* Corresponding author at: Univ. Littoral Cote d'Opale, UR 4491, LISIC, Laboratoire d'Informatique Signal et Image de la Cote d'Opale, F-62100 Calais, France.

E-mail address: moncef.garouani@etu.univ-littoral.fr (Moncef Garouani).

datasets and their evaluation measures is a critical task [3]. The ML algorithms generally have two kinds of parameters :

- the ordinary parameters that the model learns and optimizes automatically based on its normal behavior during the learning phase,
- the hyperparameters (categorical and continuous), which are usually manually set before the beginning of the model training.

In the context of the manufacturing industry, a major challenge is the selection of the feasible ML algorithm and the tuning of related hyperparameters. The algorithm selection and its configuration (tuning of hyperparameters) is a complex process because the ML algorithms are used as a “black-box”. The performance is affected by the characteristics of the datasets and the configuration of algorithms hyperparameters [5]. The selection and configuration of appropriate algorithm(s) is an error prone and time-consuming process due to the prevailing flaws while establishing the multiple configurations. It hence emphasizes the need to automate this process.

The Automated Machine Learning (AutoML) [6] is a decision support system that partially or totally automates the ML pipeline. The major goal of this research field is to enable non-expert ML developers to effectively utilize “off-the-shelf” solutions, which would save time and effort for practitioners [7,8]. At its core, the AutoML strives to achieve the performance criteria (e.g. accuracy, recall, F1 score) in order to solve the respective ML tasks such as classification, regression, or clustering. In the intervening period, the AutoML optimizes a given performance criterion [9] to solve the particular task with respect to the dataset.

Multiple approaches have been proposed to tackle the above problem [9–13] owing to the immense potential of AutoML. In this regard, several tools are available in the research community such as Auto-sklearn [5], TPOT [12], and AutoWEKA [14]. There are also several commercial tools such as RapidMiner [15], H2O Driverless AI [16], Data Robot [17], and MATLAB ML toolbox [18].

We observe that many industrial actors are competing around the goal of automating the machine learning [19]. They are mostly focused on various budget-limited tasks dealing with the supervised learning. However, they typically come up with the black-box solutions and lack the effective explanations of the predicted performance factors. It is also worth noting that the cost of these solutions tends to higher due to the involved computational complexity and the time required to generate recommendations [3].

Generally, in the most of the existing AutoML systems, the visibility is limited on the prominent exhibition of input and output parameters. They rather conceal the visibility of inherent associations among them. Instead of that, the confidence of users can be increased with the transparency of the automatic results in AutoML systems. The user confidence in AutoML systems is important because conventionally AutoML systems are used as the Decision Support Systems (DSS). Therefore, the acceptability and the trust-in factors of an AutoML support system are highly dependent on the transparency of the recommendation generation process [20].

In this paper, we present AMLBID, a transparent, interpretable and auto-explainable meta-learning based tool [21] that identifies the optimal or near-optimal ML configuration for a given problem. It also explains the rationale traceability behind a recommendation. The tool, as a decision support system, is able to simulate the role of the ML expert because it is based on meta-learning approach.

2. Software description

It is often difficult to build an accurate predictive model based on ML for an industrial problem that is easy to be interpreted by non-expert ML developers [6,22]. The key idea behind the transparent and auto-explainable AutoML vision is to separate the recommendations from the explanations by using two modules simultaneously, as shown in Fig. 1. The *Recommender module* (AMLBID) for the recommendations and the *Explanatory module* (AMLExplainer) for explanations. The first module is used to provide the most appropriate ML configuration(s) for a given problem. It is aimed at maximizing the requested predictive metric (e.g. Accuracy, Recall, Precision). The second module is used to provide the rationale behind the recommended ML configuration(s) as well as auto-generated explanations to better understand the inner workings of the model in an interpretable manner through an interactive multi-view tool.

2.1. Software architecture

The workflow of the proposed self-explanatory AutoML system consists of two major components :

- the AutoML component, which presents the AutoML process at the different levels of abstraction from ML configuration's recommendation to the refinement,
- and the explanatory one, which allows users to inspect both the process of decision generation and the inner working of the recommended ML model.

In the following sections, we discuss these modules in brief detail.

2.1.1. Recommendation module

The AutoML tool for Big Industrial Data (AMLBID) is a meta-learning based system in order to automate the problem of algorithm selection and its configuration. It uses a recommendation system that is bootstrapped with a knowledge base. The current knowledge base is derived from a large set of experiments conducted on 400 real-world manufacturing classification datasets which are collected from the popular repositories, such as OpenML,¹ UCI² and Kaggle³. It accumulates the generation of more than 4 millions evaluated ML configurations (pipelines). Each pipeline consists of a choice of a ML model and the configuration of its hyperparameters. The system is able to identify effective pipelines without performing expensive computational analysis. For this purpose, the system explores the interactions between meta-features (characteristics) of the datasets and the pipelines topologies.

The recommendation phase is initiated with the occurrence of a dataset as a new input of the AutoML process. At this stage, the user selects a predictive analytical metric (e.g. Precision, Accuracy, Recall) to be used for the analysis. AMLBID then automatically provides a set of ML algorithms and recommended configuration of their related hyperparameters, so that the predictive performance becomes the first-rate performance.

2.1.2. Explainer module

AMLExplainer and AMLBID are implemented following a client-server architecture. The server coordinates the interactions between AMLExplainer and the AutoML recommendation tool. The client-side scripts manage the visual user interfaces including the visualization of data summaries on multiple levels of the recommended models. Meanwhile, AMLExplainer guides the

¹ <https://www.openml.org/>.

² <https://archive.ics.uci.edu/>.

³ <https://www.kaggle.com/>.

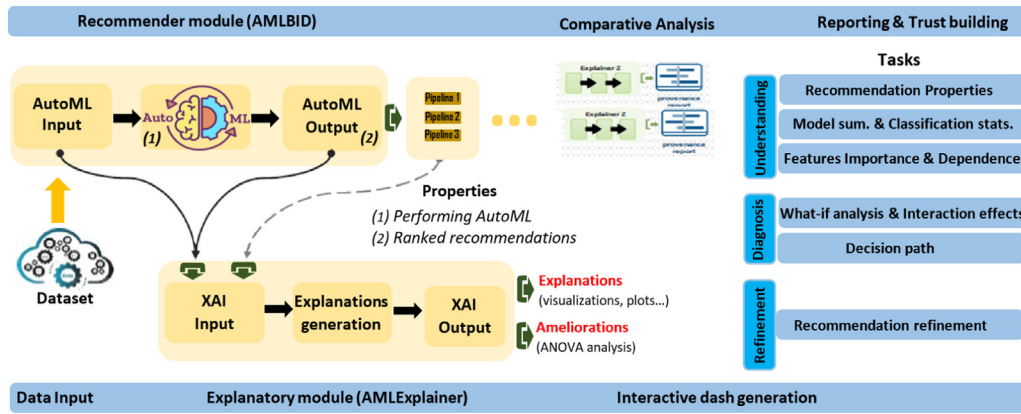


Fig. 1. Workflow of the white-box internal structures of AutoML.

Table 1

The configuration of ML algorithms and hyperparameters as tuned in the current experiments.

ML algorithm	Tuned hyperparameters
Logistic Regression (LR)	C, Penalty, Fit_intercept, Dual
Stochastic Gradient Descent (SGD)	Loss, Penalty, Alpha, Learning_rate, Fit_intercept, L1_ratio, Eta0, Power_t
Support Vector Classifier (SVM)	Kernel, C, Gamma, Degree, Coef0
Decision Tree (DT)	Min_simple_leaf, Min_simple_split, Criterion, Max_features
Random Forest (RF) & Extra Trees (ET)	N_estimators, Min_simple_split, Max_features, Min_simple_leaf, Min_weight_fraction_leaf
AdaBoost (AB)	N_estimators, Learning_rate, Algorithm, Base_estimator_max_depth
Gradient Boosting (GB)	N_estimators, Learning_rate, Criterion, Loss, Max_depth, Min_simple_leaf, Min_simple_split

end-users to improve the predictive performances, in case of the unsatisfying results. Hence, it can increase the transparency, controllability, and reliability of AutoML DSS.

2.2. Software functionalities

The current version of AMLBIID is available on the PyPI package index in form of a Python-package⁴ to facilitate its distribution and use. It presents a meta-learning based framework with major objective to automate the process of algorithm selection and the tuning of hyperparameters in supervised ML along with rational traceability. The available literature witness that the majority of state-of-the-art tools evaluate a set of pipelines by actually executing them on a given dataset prior to the recommendation. It is observed that such executions may require considerable computing time while consuming precious resources as per their availability [22]. The proposed system (AMLBIID) immediately produces a list of potential top-ranked pipelines using its knowledge base at an imperceptible computational time, hence it notably economizes the cost of resources and their provisional availability.

The available version of AMLBIID in its present form supports 08 different classification algorithms from the popular Python-based ML library Scikit-learn. The Table 1 gives the detailed description of the supported algorithms and the tuned hyperparameters.

Broadly, AMLBIID is an interactive tool to guide the end-users for improving the utility and usability of the AutoML process with the following salient features:

- It automatically (and accurately) selects the most appropriate ML algorithm(s) with related hyperparameters configuration through the use of a collaborative knowledge-base.

The knowledge-base is continuously improved by running more tasks. It makes AMLBIID smarter by achieving more experience, based on the growing knowledge-base.

- It provides assistance when AutoML returns unsatisfying results, in order to improve the predictive performances. That is achieved by assessing the importance and the correlation among the algorithm hyperparameters.
- The framework is equipped with an explanation module, which allows the end-user to understand the diagnostic design of the returned ML models using various explanation techniques. In particular, the explanation artifact allows the end-user to:
 - investigate the reasoning behind the AutoML recommendation generation process,
 - and explore the predictions of a recommendation in a trustful manner, through linked visual summaries in form of graphical, tabular, or textual information for a higher trust.

Therefore, AMLBIID enables the end users to ask a series of *what-if* scenarios while probing the opportunities to use predictive models. It can improve outcomes and reduce costs for various tasks such as the dependencies of classical collaborations of domain experts and data-scientists.

3. Illustrative examples

AMLBIID broadly has two major modules, the AMLBIID_Recommendation module and the AMLBIID_Explainer module. The AMLBIID_Recommendation module recommends and builds highly-tuned ML pipelines, whereas, the AMLBIID_Explainer module is used to intercept the inner working of the generated pipeline(s). In the following sections we discuss the functionality of these modules in further detail.

⁴ <https://pypi.org/project/AMLBIID/>.

3.1. The recommendation module

Listing 1 summarizes the interactions required to use AML-BID in order to recommend a pipeline. Subsequently, AMLBID attributes a score to the chosen pipelines and export the best pipeline to a dynamically stored .py file.

As shown on line 5, we define the root directory of the dataset to be loaded. The recommend function (as shown on line 8) initializes the meta-learning process to find the highest-scoring pipeline according to the desired performance criteria. Then, the recommended pipeline is trained on the train-set of the provided samples (as shown on line 10). After the execution of this code, Recommended_pipeline.py (for instance, as given in listing 2) is generated dynamically using the export function (as shown on line 15). It contains the corresponding Python code for the optimized pipeline.

```

1 from AMLBID.recommender import AMLBID_Recommender
2 from AMLBID.loader import *
3
4 #Load dataset
5 Data, X_train, Y_train, X_test, Y_test=load_data("Dataset.csv")
6
7 #Generate the optimal configuration
8 model=AMLBID_Recommender.recommend(Data, metric="Accuracy",
9                                     mode="Recommender")
10 model.fit(X_train, Y_train)
11
12 print(model.score(X_test, Y_test))
13
14 #Export configuration's corresponding Python code
15 model.export('Recommended_pipeline.py')
```

Listing 1: Illustrative code example of recommendation module.

```

1 import numpy as np
2 import pandas as pd
3 from sklearn.tree import DecisionTreeClassifier
4 from sklearn.metrics import classification_report
5 from sklearn.model_selection import train_test_split
6
7 data = pd.read_csv("Dataset.csv")
8
9 X = data.drop('class', axis=1)
10 Y = data['class']
11
12 X_train, X_test, Y_train, Y_test = train_test_split(X, Y,
13                                                    test_size=0.3, random_state=42)
14
15 model= DecisionTreeClassifier(criterion='entropy',
16                              max_features=0.5672564,
17                              min_samples_leaf=5,
18                              min_samples_split=20)
19 model.fit(X_train, Y_train)
20
21 Y_pred = model.predict(X_test)
22 score = model.score(X_test, Y_test)
23
24 print(classification_report(Y_test, Y_pred))
25 print(' Pipeline test accuracy: %.3f' % score)
```

Listing 2: Generated python file.

3.2. The explainer module

The AMLBID_Explainer module allows users to inspect the insights of the recommendation module and the decision generation process. Its use is illustrated in listing 3. It provides explanations on several levels of abstraction like the importance of features and the contribution of features to the individual predictions (with the help of SHAP tool [23] that finds the shapely values of a contribution for some prediction), "what-if" analysis, visualization of individual decision path, the weight of hyperparameters, and correlations. A partial vision of the AMLBID_Explainer component is shown in Fig. 2.

```

1 from AMLBID.recommender import AMLBID_Recommender
2 from AMLBID.explainer import AMLBID_Explainer
3 from AMLBID.loader import *
4
5 #Load dataset
6 Data, X_train, Y_train, X_test, Y_test=load_data("Dataset.csv")
7
8 #Generate the optimal configurations
9 model, config=AMLBID_Recommender.recommend(Data,
10                                             metric="Accuracy",
11                                             mode="Recommender_Explainer")
12 model.fit(X_train, Y_train)
13
14 #Generate the interactive explanatory dash
15 Explainer = AMLBID_Explainer.explain(model, config,
16                                     X_test, Y_test)
17 Explainer.dash()
```

Listing 3: Illustrative code example of recommendation_explainer module.

4. Impact

The ML modeling process generally operates as a highly iterative exploratory process. In reality, there is no one-size-fits-all model solution, i.e, there does not exist a single model or algorithm which can be used to achieve the highest accuracy for all datasets in a certain application domain. On that account, undertaking a large number of ML algorithms with different hyperparameters configurations would not yield a practical solution, rather it would be an inefficient, tedious, and time consuming process. In this context, the application of AMLBID is twofold: primarily it makes possible for non-data science specialists (practitioners and researchers) to build robust ML pipelines without the need for specialist assistance or intervention and even having to write a single line of code. Subsequently, the white-box specificity of the proposed AutoML tool makes it possible to interactively inspect the inner-workings of the ML predictive models without having to depend on a data scientist to generate and interpret all the extreme plots and tables.

The main objective of the AMLBID has been focused towards the design of a decision support system in order to support the non-expert practitioners and researchers. Prospectively, we intend its use in the domain of industry 4.0 [24] to take maximum benefit of ML techniques to optimize the automated manufacturing processes. In our previous works [25,26], we studied the effectiveness of the recommendation module for the selection and parameterization of ML for problems concerning the manufacturing industry. The evaluation results respond the basic research question that is *how some ML oriented manufacturing works could be further improved, simply through the use of a better ML algorithm configuration using the AMLBID*. Since AMLBID is built upon the meta-learning concept, in the broader sense, it is not only beneficial for the manufacturing actors and researchers but also for many other areas. Nevertheless, the AMLBID is useful for academic purposes, helping academia to build and understand ML predictive models behavior.

5. Conclusion

The machine learning based applications are increasingly desired due to their robustness for the large data analysis. Also, they can rapidly integrate "off-the-shelf" solutions in multiple areas. However, the non-expert data analysts are more inclined to adapt the ML based solutions that are more easily persuadable, among diverse algorithms, with the help of their rational traceability. We argue that the adaptability of the powerful decision support systems based on the ML based solutions can be further enhanced with the help of comprehensive instructions regarding the recommended pipelines and their insights. Thus,

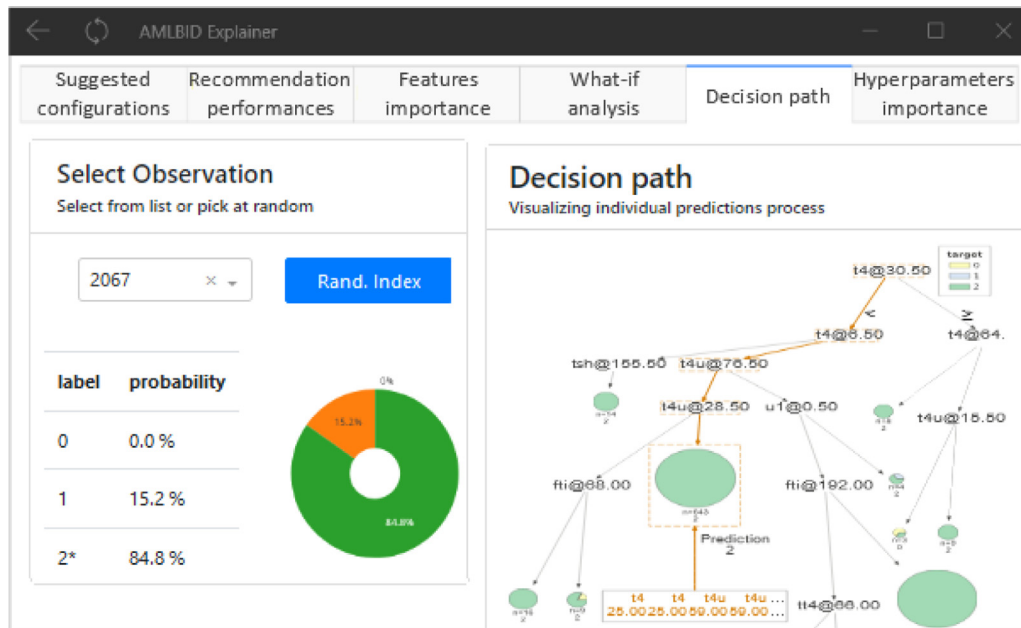


Fig. 2. Functional dashboard of AMLBID_Explainer showing the decision path of a predicted instance.

making them more trustworthy instead of black-box solutions. In this work, we present AMLBID, a novel transparent and auto-explained AutoML support system. It also includes an interactive visualization module (AMLBID_Explainer) that allows ML experts and neophytes to easily inspect and analyze the automatic results of an AutoML decision support system.

We propose to use the general explanation methods of AutoML systems; the advantage of AMLBID on account of performance comes from the remarkable time gain and prevention of resource bottlenecks. In this context, the contemporary ML based solutions are required to train multiple ML models from scratch for the fresh datasets prior to generate the list of recommended pipelines. It competes favorably to the state-of-the-art AutoML systems in the fields that are less tolerant to the delays such as the manufacturing industry. Whilst, the AMLBID particularly addresses this challenge to promptly produce a list of potential top-ranked ML configurations using its collaborative knowledge base. In practice, the confidentiality of the analyzed datasets is respected by the fact that the knowledge-base of AMLBID consists of the meta-features of datasets and not the actual data. At present, we are planning to expand AMLBID to support algorithms of regression, deep learning and distributed ML libraries (e.g., Spark ML [27]) since we are dealing with Industrial Big Data.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors thank the *Université du Littoral Côte d'Opale (ULCO)*, France, *School of engineering's and business' sciences and technics (HESTIM)*, Morocco and *CNRST Morocco* for the partial financial support, and the *CALCULCO* computing platform, supported by *SCoSI/ULCO (Service COmmun du Système d'Information de l'Université du Littoral Côte d'Opale)* for the computational facilities.

References

- [1] Millman N. Big data to unlock value from the industrial internet of things. *ComputerWeekly.Com* 2015. <https://www.computerweekly.com/opinion/Big-data-to-unlock-value-from-the-Industrial-Internet-of-Things>.
- [2] Tao F, Qi Q, Liu A, Kusiak A. Data-driven smart manufacturing. *J Manuf Syst* 2018;48:157–69. <http://dx.doi.org/10.1016/j.jmsy.2018.01.006>, Special Issue on Smart Manufacturing.
- [3] Cohen-Shapira N, Rokach L, Shapira B, Katz G, Vainshtein R. AutoGRD: model recommendation through graphical dataset representation. In: *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. New York, NY, USA; 2019, p. 821–30. <http://dx.doi.org/10.1145/3357384.3357896>.
- [4] Muñoz MA, Sun Y, Kirley M, Halgamuge SK. Algorithm selection for black-box continuous optimization problems: A survey on methods and challenges. *Inform Sci* 2015;317:224–45. <http://dx.doi.org/10.1016/j.ins.2015.05.010>.
- [5] Feuerer M, Klein A, Eggensperger K, Springenberg JT, Blum M, Hutter F. Efficient and robust automated machine learning. In: *Proceedings of the 28th international conference on neural information processing systems*, vol. 2. 2015, p. 2755–63.
- [6] Reif M, Shafait F, Goldstein M, Breuel T, Dengel A. Automatic classifier selection for non-experts. *Pattern Anal Appl* 2014;17(1):83–96. <http://dx.doi.org/10.1007/s10044-012-0280-z>.
- [7] Hutter F, Kotthoff L, Vanschoren J, editors. *Automated machine learning: methods, systems, challenges*. The springer series on challenges in machine learning, 2019. <http://dx.doi.org/10.1007/978-3-030-05318-5>.
- [8] Waring J, Lindvall C, Umeton R. Automated machine learning: Review of the state-of-the-art and opportunities for healthcare. *Artif Intell Med* 2020;104:101822. <http://dx.doi.org/10.1016/j.artmed.2020.101822>.
- [9] Drori I, Krishnamurthy Y, Rampin R, Lourenço R, One J, Cho K, Silva C, Freire J. AlphaD3M: Machine learning pipeline synthesis. In: *AutoML workshop at ICML*. 2018.
- [10] Hutter F, Hoos HH, Leyton-Brown K. Sequential model-based optimization for general algorithm configuration. In: *Coello CAC, editor. Learning and intelligent optimization. Lecture notes in computer science*, Berlin, Heidelberg; 2011, p. 507–23. <http://dx.doi.org/10.1007/978-3-642-25566-3>.
- [11] Laadan D, Vainshtein R, Curiel Y, Katz G, Rokach L. RankML: a meta learning-based approach for pre-ranking machine learning pipelines. 2019. [arXiv:1911.00108](https://arxiv.org/abs/1911.00108).
- [12] Olson RS, Moore JH. TPOT: a tree-based pipeline optimization tool for automating machine learning. In: *Hutter F, Kotthoff L, Vanschoren J, editors. Automated machine learning: methods, systems, challenges*. The springer series on challenges in machine learning, 2019, p. 151–60. <http://dx.doi.org/10.1007/978-3-030-05318-5>.
- [13] Li L, Jamieson KG, DeSalvo G, Rostamizadeh A, Talwalkar A. Efficient hyperparameter optimization and infinitely many armed bandits, vol. 16. 2016, *CoRR*, Abs/1603.06560.

- [14] Kotthoff L, Thornton C, Hoos HH, Hutter F, Leyton-Brown K. Auto-WEKA: automatic model selection and hyperparameter optimization in WEKA. In: Hutter F, Kotthoff L, Vanschoren J, editors. Automated machine learning: methods, systems, challenges. The springer series on challenges in machine learning, Cham; 2019, p. 81–95. <http://dx.doi.org/10.1007/978-3-030-05318-5>.
- [15] RapidMiner, data science & machine learning platform. <https://rapidminer.com>.
- [16] H2O.ai, AI cloud platform. <https://www.h2o.ai/>.
- [17] DataRobot, AI cloud - the next generation of AI. <https://www.datarobot.com/>.
- [18] Machine Learning avec MATLAB. <https://fr.mathworks.com/solutions/machine-learning.html>.
- [19] Guyon I, Sun-Hosoya L, Boullé M, Escalante HJ, Escalera S, Liu Z, Jajetic D, Ray B, Saeed M, Sebag M, Statnikov A, Tu W-W, Viegas E. Analysis of the automl challenge series 2015–2018. In: Automated machine learning: methods, systems, challenges. 2019, p. 177–219. <http://dx.doi.org/10.1007/978-3-030-05318-5>.
- [20] Samek W, Müller K-R. Towards explainable artificial intelligence. In: Explainable AI: Interpreting, Explaining and Visualizing Deep Learning. 2019, p. 5–22. <http://dx.doi.org/10.1007/978-3-030-28954-6>.
- [21] Lemke C, Budka M, Gabrys B. Metalearning: A survey of trends and technologies. *Artif Intell Rev* 2015;44(1):117–30. <http://dx.doi.org/10.1007/s10462-013-9406-y>.
- [22] Garouani M, Ahmad A, Bouneffa M, Hamlich M, Bourguin G, Lewandowski A. Towards big industrial data mining through explainable automated machine learning. 2021, <http://dx.doi.org/10.21203/rs.3.rs-755783/v1>.
- [23] Lundberg SM, Lee S-I. A unified approach to interpreting model predictions. In: Proceedings of the 31st international conference on neural information processing systems. 2017, p. 4768–77, [arXiv:1705.07874](https://arxiv.org/abs/1705.07874).
- [24] Tao F, Qi Q, Liu A, Kusiak A. Data-driven smart manufacturing. In: Special Issue on Smart Manufacturing, *J Manuf Syst In: Special Issue on Smart Manufacturing*, 2018;48:157–69. <http://dx.doi.org/10.1016/j.jmsy.2018.01.006>.
- [25] Garouani M, Ahmad A, Bouneffa M, Lewandowski A, Bourguin G, Hamlich M. Towards the automation of industrial data science: a meta-learning based approach. In: 23rd international conference on enterprise information systems. 2021, p. 709–16. <http://dx.doi.org/10.5220/0010457107090716>.
- [26] Garouani M, Hamlich M, Ahmad A, Bouneffa M, Bourguin G, Lewandowski A. Towards an automatic assistance framework for the selection and configuration of machine-learning-based data analytics solutions in industry 4.0. In: The fifth international conference on big data and internet of things. [in press].
- [27] Zaharia M, Xin RS, Wendell P, Das T, Armbrust M, Dave A, Meng X, Rosen J, Venkataraman S, Franklin MJ, Ghodsi A, Gonzalez J, Shenker S, Stoica I. Apache spark: A unified engine for big data processing. *Commun ACM* 2016;59(11):56–65. <http://dx.doi.org/10.1145/2934664>.