Foundations of Trustworthy AI – Integrating Reasoning, Learning and Optimization
TAILOR
Grant Agreement Number 952215

AutoAI Benchmarks v.1 Report

Document type (nature)	Report
Deliverable No	7.5
Work package number(s)	7
Date	Due 31 August 2021
Responsible Beneficiary	ULEI, ID #7
Author(s)	Koen van der Blom
Publicity level	Public
Short description (Please insert the text in the Description of Deliverables in the Appendix 1.)	Version 1 of AutoAI Benchmarks: Curated, regular evaluations of AutoAI techniques and their contribution to trustworthiness, to measure and monitor progress in the field.

History			
Revision	**Date**	**Modification**	**Author**
1.0	31 July 2021	first version	Koen van der Blom

Document Review		
Reviewer	**Partner ID / Acronym**	**Date of report approval**
Michela Milano	#10, UNIBO	August 31, 2021
Fosca Giannotti	#2, CNR	September 6, 2021

For review, the template provided in Folder B on Drive has been used. The review documents are saved in the dedicated EMDESK folder.

This document is a public report. However, the information herein is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

Table of Contents

Summary of the report

This report gives an overview of existing benchmarks in the area of automated AI (AutoAI), and more specifically, in the areas of the five AutoAI topics covered in T7.1-T7.5 of the TAILOR project. Along with pointers to existing resources, we provide initial thoughts on the design of additional benchmarks and, where appropriate, mention existing work that can facilitate these efforts.

Organisation

The following experts from the TAILOR consortium have been involved in the writing of this report, based on materials collected across a broad range of project partners and following the Second TAILOR WP7 Workshop on 3 June 2021:

Partner ID / Acronym	Name	Role
#7, ULEI	Holger Hoos Koen van der Blom Mitra Baratchi Jan N. van Rijn	WP7 Leader
#17, ALU-FR	Frank Hutter	Task Leader
#3, INRIA	Marc Schoenauer	Task Leader
#12, TUE	Joaquin Vanschoren	Task Leader

1. Introduction

The development of AutoAI techniques, and assessment of their quality in terms of performance and trustworthiness hinges on high quality benchmarks. Many AI systems are manually constructed or configured to perform a specific task, AutoAI aims to automate (parts of) this construction process. Take for instance the widely studied AutoML task of automated selection of classification algorithms and the optimisation of their hyperparameters (e.g. AutoWEKA), or the automated optimisation of machine learning pipelines which considers the same type of approaches (e.g. Auto-sklearn).

When it comes to AI benchmarks, they just have to enable the comparison of AI systems, whereas AutoAI benchmarks need to enable the operation of the AutoAI techniques in addition to enabling the comparison of the AI systems resulting from using AutoAI. For the example above, this is the difference between needing just training data for the ML algorithms and also needing to specify the AutoML scenario. This scenario includes components such as which ML algorithms are considered and, for instance, the time budget available to tune the hyperparameters. So, although AutoAI benchmarks may share a common part with regular AI benchmarks, they also require additional components. See Fig.1 for a simple example.

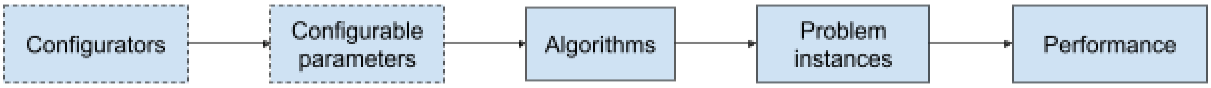

Fig. 1: Simplified example of benchmarking for algorithm configuration (AC). Blocks with solid borders are present in regular algorithm benchmarking scenarios. Blocks with dashed borders come into play for AC (e.g., AutoML) benchmarking.

As part of WP7 one of the goals is to create awareness about the available AutoAI benchmarks, and another goal is to identify gaps for the development of new benchmarks to complement existing work. To this end, we aim to capture the most important benchmarks here, but not necessarily an exhaustive list. This first report aims to establish the situation at the start of the TAILOR project, while the future bi-annual reports will describe benchmarks that were newly generated, collected, or were previously missed.

In the following, existing AutoAI benchmarks are given per research task in WP7; this was done to ensure broad coverage of our initial survey. In addition, non-AutoAI benchmarks with potential to be extended to AutoAI are listed, with particular attention to areas where no AutoAI benchmarks are available yet. Work from TAILOR network members is highlighted in boldface, both for the AutoAI and non-AutoAI benchmarks.

Throughout the remainder of the duration of TAILOR, four updates on AutoAI benchmarks will be provided. These will introduce benchmarks newly developed as part of the work on TAILOR WP7 as well as ones provided by the community at large.

AutoAI is still a young field, and therefore, quite naturally, the corresponding benchmarks have also been developed recently. As such, no clear general framework for AutoAI benchmarking currently exists. In fact, with its ongoing extension beyond machine learning, it is not clear this can already be defined. One important challenge is to develop this framework. Over the course of the project, this will be an area to work on, and advancements will be included in the corresponding updates to this report. In particular, general concepts are needed for what is important and useful to consider for an AutoAI benchmark.

Another extension we will aim to include in the updates to this report are summaries of the main characteristics of each included benchmark, such as problem and data type, size and other characteristics of benchmark sets, as well as type and characteristics of (target) algorithms.

2. AutoML in the wild [T7.2, ALU-FR]

“AutoML in the wild” aims to facilitate the usability of machine learning by non-machine-learning-experts. Our efforts in TAILOR concentrate on two research lines:
1. Making Neural Architecture Search (NAS) usable in the wild
2. Design methods to automatically handle messy real-world data in AutoML.

In particular in NAS, benchmarking is heavily used. Task leader Frank Hutter, together with collaborators from Google, introduced the first tabular NAS benchmark, NAS-Bench-101. The research community is using this benchmark heavily and also created almost a dozen new tabular NAS benchmarks since. In the following, we typeset NAS benchmarks by TAILOR participants in boldface.

- **Chris Ying, Aaron Klein, Esteban Real, Eric Christiansen, Kevin Murphy, Frank Hutter, "Nas-bench-101: Towards reproducible neural architecture search," Proceedings of the International Conference on Machine Learning, 2019.**
- Xuanyi Dong and Yi Yang, “NAS-Bench-201: Extending the Scope of Reproducible Neural Architecture Search,” Proceedings of the International Conference on Learning Representations, 2019.
- Chaojian Li, Zhongzhi Yu, Yonggan Fu, Yongan Zhang, Yang Zhao, Haoran You, Qixuan Yu, Yue Wang, Yingyan Lin. HW-NAS-Bench: Hardware-Aware Neural Architecture Search Benchmark. Proceedings of the International Conference on Learning Representations, 2021.
- Nikita Klyuchnikov, Ilya Trofimov, Ekaterina Artemova, Mikhail Salnikov, Maxim Fedorov, Evgeny Burnaev. “NAS-Bench-NLP: Neural Architecture Search Benchmark for Natural Language Processing”, arXiv 2020
- Abhinav Mehrotra, Alberto Gil C. P. Ramos, Sourav Bhattacharya, Łukasz Dudziak, Ravichander Vipperla, Thomas Chau, Mohamed S Abdelfattah, Samin Ishtiaq, Nicholas Donald Lane. “NAS-Bench-ASR: Reproducible Neural Architecture Search for Speech Recognition”, Proceedings of the International Conference on Learning Representations, 2021.
- **Arber Zela, Julien Siems, Frank Hutter. “NAS-Bench-1Shot1: Benchmarking and Dissecting One-shot Neural Architecture Search”. Proceedings of the International Conference on Learning Representations, 2020.**
- **Julien Siems, Lucas Zimmer, Arber Zela, Jovita Lukasik, Margret Keuper, Frank Hutter. “NAS-Bench-301 and the Case for Surrogate Benchmarks for Neural Architecture Search”, arXiv 2020**

- **Shen Yan, Colin White, Yash Savani, Frank Hutter. "NAS-Bench-x11 and the Power of Learning Curves", CVPR workshop on NAS 2021.**

Work is underway at the University of Freiburg to provide all of these NAS benchmarks through a unified interface.

Related to work on NAS is hyperparameter optimization. There are several benchmarks concerning this:

- **HPOlib: https://github.com/automl/HPOlib**
- **HPOBench: https://github.com/automl/HPOBench**
- Bayesmark: https://github.com/uber/bayesmark

Relatedly, there is also a benchmark on dynamic algorithm configuration (DAC), which covers the dynamic optimization of hyperparameters:
https://github.com/automl/DACBench

- **Theresa Eimer, Andre Biedenkapp, Maximilian Reimer, Steven Adriaensen, Frank Hutter, Marius Lindauer, DACBench: A Benchmark Library for Dynamic Algorithm Configuration, IJCAI 2021**

Regarding the handling of messy data, there are also a few works:

- The Data wrangling dataset repository: http://dmip.webs.upv.es/datawrangling/catalog.html
- Spreadsheets: Benchmarks and quality assurance techniques for spreadsheets: https://spreadsheets.ist.tugraz.at/

Further, TU/e (Joaquin Vanschoren) will present a paper on handling messy data, especially string categorical features, at the forthcoming ECMLPKDD workshop on Automated Data Science:

- **John W. van Lith and Joaquin Vanschoren. From strings to data science: a practical framework for automated string handling. ECMLPKDD workshop on Automated Data Science.**

3. Beyond standard supervised learning [T7.2, ULEI]

Work on AutoML, an important special case of AutoAI, has so far largely been limited to standard supervised learning scenarios on tabular data. This "Beyond standard supervised learning" task aims to expand the scope of AutoML to more diverse and richer learning settings. To achieve this, two lines of research are being pursued:

1. Bring together AutoML- and domain experts to design flexible AutoML frameworks - including algorithm configuration and meta-learning - for domains such as multi-target regression, unsupervised learning, semi-supervised learning and learning on spatio-temporal data.
2. Collect and make publicly available (whenever possible) benchmark data and scenarios for these new settings, building on existing repositories and libraries.

Work on multi-target regression is limited, but a benchmark framework for multi-label classification was proposed by:

- Wever M, Tornede A, Mohr F, Hullermeier E. AutoML for Multi-Label Classification: Overview and Empirical Evaluation. IEEE Transactions on Pattern Analysis & Machine Intelligence. 2021: https://ieeexplore.ieee.org/abstract/document/9321731

Some initial work on semi-supervised AutoML exists, but it is not very clear which datasets were used or how they were adapted to fit the semi-supervised+AutoML scenario, nor was anything made publicly available for future benchmarking use. For instance:

- Li, Y.-F., Wang, H., Wei, T., & Tu, W.-W. (2019). Towards Automated Semi-Supervised Learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, *33*(01), 4237-4244. https://doi.org/10.1609/aaai.v33i01.33014237

Unsupervised and spatio-temporal learning do not yet seem to have AutoML benchmarks in place, and are thus directions to focus on in further work in WP7 of TAILOR. Fortunately, a large variety of existing (non-AutoML) work exists that can be built on.

For human trajectory prediction, there are currently no benchmarks available that are suited for AutoAI research. However, there have been efforts to identify publicly available datasets that can be used for this purpose. For instance:

- Amirian J, Zhang B, Castro FV, Baldelomar JJ, Hayet JB, Pettré J. Opentraj: Assessing prediction complexity in human trajectories datasets. InProceedings of the Asian Conference on Computer Vision 2020. https://github.com/crowdbotp/OpenTraj

For other spatio-temporal AutoML applications, spatial datasets could be used to develop AutoML benchmarks. Such datasets include:

- AIREO for Earth observation data: https://eo4society.esa.int/projects/aireo/

In addition, as part of the following recently accepted publication, we have identified 7 datasets (and collected them in a single repository) that can be used for AutoML research for remote sensing image classification tasks. This should also make it easier for others to experiment on this wide range of datasets in the future:

- **Palacios Salinas, N. Rosaura and Baratchi, M. and van Rijn, J. N. and Vollrath, A, “Automated Machine Learning for Satellite Data: Integrating Remote Sensing Pre-trained Models into AutoML Systems”, in Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, ECML-PKDD 2021, 2021. https://github.com/palaciosnrps/automl-rs-project**

For sound classification datasets are available that could be used in AutoAI benchmarks:

- **Ma, Jeannette Shijie, Marcello A. Gómez Maureira, and Jan N. van Rijn. "Eating Sound Dataset for 20 Food Types and Sound Classification Using Convolutional Neural Networks." In Companion Publication of the 2020 International Conference on Multimodal Interaction, pp. 348-351. 2020.**

Other domains with existing (non-AutoML) benchmarks that could be extended include:

- High-dimensional particle and astrophysics optimization benchmark: https://arxiv.org/abs/2101.04525
- **OpenML: https://www.openml.org/**
- OpenAI gym: https://gym.openai.com/
- OpenAI baselines: https://github.com/openai/baselines

- DeepMind's RL benchmark problem library b-suite: https://deepmind.com/research/open-source/bsuite

In addition, novel real-world datasets for AutoML benchmarking may be introduced through our work with partners from industry.

For these new domains, it should be investigated how they might be integrated in existing AutoML benchmarking libraries covering automated algorithm configuration:

- **AClib: Algorithm Configuration Library: https://aclib.net/**
- **AClib: Algorithm Configuration Library 2.0: https://bitbucket.org/mlindauer/aclib2/src/master/**
- **Theresa Eimer, Andre Biedenkapp, Maximilian Reimer, Steven Adriaensen, Frank Hutter, Marius Lindauer, DACBench: A Benchmark Library for Dynamic Algorithm Configuration, IJCAI 2021.** https://github.com/automl/DACBench

Automated algorithm selection:

- **Bischl B, Kerschke P, Kotthoff L, Lindauer M, Malitsky Y, Fréchette A, Hoos H, Hutter F, Leyton-Brown K, Tierney K, Vanschoren J. Aslib: A benchmark library for algorithm selection. Artificial Intelligence. 2016. http://www.aslib.net**

Automated hyperparameter optimization:

- **HPOlib: https://github.com/automl/HPOlib**
- **HPOBench: https://github.com/automl/HPOBench**
- Bayesmark: https://github.com/uber/bayesmark

On the AutoAI (or more specifically: AutoML) methods side, closely related to meta-learning, several competitions and benchmarks exist on few-shot learning:

- **El Baz, A., Guyon, I., Liu, Z., Treguer, S., van Rijn, J.N. and Vanschoren, J.: Advances in MetaDL: AAAI 2021 challenge and workshop. AAAI Workshop on Meta-Learning and MetaDL Challenge, 1-16, 2021 (to appear, August).**
- miniImageNet from: Vinyals O, Blundell C, Lillicrap T, Wierstra D. Matching networks for one shot learning. Advances in neural information processing systems. 2016: https://paperswithcode.com/dataset/miniimagenet-1
- Omniglot from: Lake BM, Salakhutdinov R, Tenenbaum JB. Human-level concept learning through probabilistic program induction. Science. 2015: https://paperswithcode.com/sota/few-shot-image-classification-on-omniglot-1-1

Recent work on code smell (= inverse of code quality) detection could provide a further useful basis for interesting, new AutoAI benchmarks:

- Madeyski, L., Lewowski, T.: Mlcq: Industry-relevant code smell data set. In: Proceedings of the Evaluation and Assessment in Software Engineering, pp. 342–347(2020)
- **Soomlek, C., van Rijn, J.N., Bonsangue, M.M.: Automatic human-like detection of code smells, Discovery Science 2021 (to appear, October).**

4. Self-monitoring AI systems [T7.3, ULEI]

Self-monitoring AI systems concern systems that are able to monitor the robustness of their performance. Here robustness is considered in a broad sense, and can include natural drift, changes in the system, and adversarial attacks. For AI systems to be reliable they should be able to detect, report on, and ultimately automatically correct themselves. This task aims to produce:

1. General-purpose methods to self-calibrate AI systems, both for machine learning and other areas of AI
2. Metrics to assess those self-calibration approaches.
3. Benchmarks for these problems

Initial work exists to support the detection of incorrect predictions by machine learning systems. First general methods for uncertainty estimation in AutoML were introduced in:

- **Matthias König, Holger H Hoos and Jan N van Rijn. Towards Algorithm-Agnostic Uncertainty Estimation: Predicting Classification Error in an Automated Machine Learning Setting. In ICML Workshop on Automated Machine Learning. 2020.**

In the aforementioned work, the OpenML-CC18 benchmark set was considered:

- **Bischl, B. and Casalicchio, G. and Feurer, M. and Hutter, F. and Lang, M. and Mantovani, R. G. and van Rijn, J. N. and Vanschoren, J. OpenML benchmarking suites, arXiv preprint arXiv:1708.03731. 2019.**

We see a need to evaluate whether this is sufficient, or if specialised benchmarks have to be developed for the uncertainty estimation domain.

For natural language inference (NLI) in a supervised setting, an adversarial benchmark exists; it works by testing machine learning systems and asking adversarial human annotators to break it:

- Nie Y, Williams A, Dinan E, Bansal M, Weston J, Kiela D. Adversarial NLI: A new benchmark for natural language understanding. arXiv preprint arXiv:1910.14599. 2019. https://arxiv.org/pdf/1910.14599.pdf

Other adversarial settings, such as neural network verification are starting to adopt AutoML techniques, but specific benchmarks are still missing:

- **Matthias König, Holger H Hoos and Jan N van Rijn. Speeding Up Neural Network Verification via Automated Algorithm Configuration. In ICLR Workshop on Security and Safety in Machine Learning Systems. 2021.**

Even so, for neural network verification, there is work that could lead to such benchmarks. For instance from the following competition:

- Verification of Neural Networks Competition: https://sites.google.com/view/vnn20/

In addition, while they do not define a benchmark, the following work provides rather extensive evaluations along with tooling to reproduce these.:

- Vincent Tjeng, Kai Xiao, and Russ Tedrake. Evaluating Robustness of Neural Networks with Mixed Integer Programming. In Proceedings of the 7th International Conference on Learning Representations (ICLR 2019), 2019.

The WILDS benchmarking environment considers distribution shifts for realistic situations between the training and test data. Datasets are included that cover both domain generalisation and subpopulation shifts, as well as the combination of the two.

- Koh PW, Sagawa S, Xie SM, Zhang M, Balsubramani A, Hu W, Yasunaga M, Phillips RL, Gao I, Lee T, David E. Wilds: A benchmark of in-the-wild distribution shifts. In International Conference on Machine Learning 2021. https://wilds.stanford.edu/

Like for moving beyond standard supervised learning, here, too, the extension and integration of benchmarks into the same existing platforms for automated algorithm configuration, automated algorithm selection, and automated hyperparameter optimisation should be considered.

5. Multi-objective AutoAI [T7.4, INRIA]

Most works in AutoAI are searching for the best algorithm, algorithm configuration, or full pipeline design and configuration that optimizes the performance of the resulting algorithm/pipeline, i.e., are driven by a single objective. However, and this is especially true in the context of TAILOR, though performance will always be of interest, there are other objectives pertaining to trustworthiness that might also be important (e.g., explainability, frugality, robustness, etc.). There is unfortunately little doubt that improving these trustworthiness properties will result in decreasing performance (e.g. configuring for a broader set of instances increases robustness, but is less performant than configuring for a narrow set of instances): some trade-off needs to be adopted.

Finding trade-offs between antagonist objectives is the goal of multi-objective optimization (MOO), also known as multi-criteria optimization. There exist several approaches for MOO, from Evolutionary Algorithms (e.g. [DebEtAl02, CoeLec02]) to Iterated Local Search (e.g. [BloEtAl15, **DubEtAl15**, DerEtAl16]) to Bayesian Optimization (e.g. [GalEtAl20, EmmEtAl16]), that aim at finding the Pareto set, i.e. the set of best trade-offs for the competing objectives. This set consists of points for which no objective can be improved without degrading another objective.

A first important remark is that there exist several works aiming at algorithm selection, algorithm configuration, and even algorithm design of multi-objective algorithms (MOAs) [BloEtAl17]. However, such AutoAI is not Multi-objective AutoAI: these works look for the design/selection/configuration of MOAs **from a single objective point of view**, optimizing the performance of the resulting algorithm, usually in terms of hypervolume, or some other multi-objective indicator, and do not consider multiple objectives for AutoAI. This is the case with most, if not all, continuous optimization platforms proposing datasets of problems and their state-of-the-art solutions, from COCO to Nevergrad, and has been present in the corresponding competitions (BBOB, BBComp) for comparing continuous optimizers in a black-box setting.

- Hansen, Nikolaus, Anne Auger, Raymond Ros, Olaf Mersmann, Tea Tušar, and Dimo Brockhoff. "COCO: A platform for comparing continuous optimizers in a black-box setting." Optimization Methods and Software 36, no. 1 (2021): 114-144.: https://coco.gforge.inria.fr/
- BBOB: http://numbbo.github.io/workshops/index.html
- Bennet P, Doerr C, Moreau A, Rapin J, Teytaud F, Teytaud O. Nevergrad: black-box optimization platform. ACM SIGEVOlution. 2021: https://github.com/facebookresearch/nevergrad
- BBcomp - Black-box Optimization Competition: https://www.ini.rub.de/PEOPLE/glasmtbl/projects/bbcomp/

Similar work could be done in other domains where benchmarks exist for AS/AC of single-objective combinatorial functions (ASLib, ACLib, Souza). But we will not consider these any more in this section, for the reasons above.

- **AClib: Algorithm Configuration Library: https://aclib.net/**
- **AClib: Algorithm Configuration Library 2.0: https://bitbucket.org/mlindauer/aclib2/src/master/**

- **Bischl B, Kerschke P, Kotthoff L, Lindauer M, Malitsky Y, Fréchette A, Hoos H, Hutter F, Leyton-Brown K, Tierney K, Vanschoren J. Aslib: A benchmark library for algorithm selection. Artificial Intelligence. 2016.** http://www.aslib.net
- **Marcelo de Souza, Marcus Ritt and Manuel López-Ibáñez (forthcoming). Capping Methods for the Automatic Configuration of Optimization Algorithms. Computers & Operations Research.** https://github.com/souzamarcelo/supp-cor-capopt

On the other hand, it seems straightforward to extend the simplest existing (single-objective) AutoAI algorithms, that do some direct meta-optimization of algorithms hyperparameters, to handle multiple objectives, replacing the optimization algorithm at work in the AutoAI at hand with the corresponding multi-objective optimizer. Indeed, this has been proposed for instance for ParamILS [**HutEtAl09**], with the MO-ParamILS platform, where the MO-ParamILS algorithm was applied to find the Pareto-set for several bi-objective algorithm configuration problems, balancing performance and complexity/CPU cost.

- **Blot A, Hoos HH, Jourdan L, Kessaci-Marmion MÉ, Trautmann H. MO-ParamILS: A multi-objective automatic algorithm configuration framework. InInternational Conference on Learning and Intelligent Optimization 2016 May 29 (pp. 32-47). Springer, Cham.**

Another interesting approach is NSGA-Net, in which NSGA-II is used for Neural Architecture Search (NAS), using a specific representation of DNN architectures, to discover the best trade-off between accuracy (on popular datasets like MNIST and Cifar-10) and size of the network (as a proxy for complexity of the learning phase).

- Lu Z, Whalen I, Boddeti V, Dhebar Y, Deb K, Goodman E, Banzhaf W. Nsga-net: neural architecture search using multi-objective genetic algorithm. InProceedings of the Genetic and Evolutionary Computation Conference 2019 Jul 13 (pp. 419-427).

However, these works have remained isolated, and in particular no recognized benchmarks in the area of Multi-objective AutoAI have been proposed yet, nor do existing works even compare their results to those of other Multi-objective AutoAI approaches. There are several reasons for that.

- Comparing the results of two MOAs (i.e., comparing two Pareto fronts) is not straightforward, several measures have been proposed, and hence several rankings among Multi-objective AutoAI algorithms will need to be established
- There is no clear way to measure most of the trustworthiness objectives: whereas existing works use as second objective some measure of complexity (or CPU cost of learning or optimization), and this could, with some twist, be considered as a proxy for explainability (though a DNN with 300000 weights can be considered more explainable than another one with 3M weights, it is in fact not reasonable to talk about explainability in such context).
- Many (single-objective) AutoAI algorithms are not simply made of one meta-optimization algorithm, and hence cannot easily be turned into multi-objective AutoAI. Approaches exist that are based on recommendation processes [MısSeb17], or on multi-armed bandits [**RakEtAl19**] (even though multi-objective MAB algorithms exist); or algorithms that optimize the whole pipeline, like Auto-sklearn [**FeuEtAl19**], TPOT [OlsEtAl16], or MOSAIC [**RakEtAl19**].

Nevertheless, some low hanging fruits do exist, and future work should address the following avenues:

- Continue with the performance/complexity bi-objective AutoAI: in that area at least, some benchmarks could be set up, building on the existing datasets and platforms.
    - A particular issue in the framework of algorithm selection for continuous optimization is that of the performance measure. Two points of view exist:

measure the time (or number of calls to the objective function) needed to reach a given performance, as done in COCO/BBOB, allowing easy aggregation of performances for functions with different orders of magnitude of values; or measure the best objective value reached in a given time, as done in many papers, as well as in Nevergrad and in all BBComp competitions. Running multi-objective algorithm selection/configuration with best value / number of iterations consumed could reconcile both approaches.

- ○ In the combinatorial optimization domain, where the number of plans that are evaluated is also a good proxy for the CPU cost, there exist many specialized libraries, like tsplib for the Travelling Salesperson Problem, or the past instances of competitions for problems like SAT or Planning, and all of them could easily be turned into benchmarks for multi-objective algorithm selection and configuration:
  - TSPLIB: http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/
  - SAT competition: http://www.satcompetition.org/
  - Planning competitions: https://www.icaps-conference.org/competitions/
- ○ In AutoML (for supervised learning), many works deal with optimizing the whole learning pipeline, and a popular benchmark has emerged from the OpenML dataset, the OpenML100. It has since been superseded by OpenML-CC18, which could easily be transformed into a bi-objective AutoAI problem by also minimizing the learning time, provided all runtimes are scaled fairly (e.g., from the training time of some simple dummy algorithm).
  - **Bischl, B. and Casalicchio, G. and Feurer, M. and Hutter, F. and Lang, M. and Mantovani, R. G. and van Rijn, J. N. and Vanschoren, J. Openml benchmarking suites, arXiv preprint arXiv:1708.03731. 2019.**
- ○ NAS (aka AutoDL) also offers a nice benchmark with the NASBench initiatives, offering either tabular datasets:
  - **Chris Ying, Aaron Klein, Esteban Real, Eric Christiansen, Kevin Murphy, Frank Hutter, "Nas-bench-101: Towards reproducible neural architecture search," Proceedings of the International Conference on Machine Learning, 2019.**

  or surrogate datasets of pre-computed performances of many DNN architectures:
  - **Julien Siems, Lucas Zimmer, Arber Zela, Jovita Lukasik, Margret Keuper, Frank Hutter. “NAS-Bench-301 and the Case for Surrogate Benchmarks for Neural Architecture Search”, arXiv 2020**

- Propose indicators beyond complexity, for other trustworthiness objectives, and extend all of the above to other multi-objective settings:
  - ○ Robustness seems a first easy and agnostic target. In optimization, one can simply average the performances over some sampling around the point of interest - though many options remain open regarding such sampling. In Machine Learning, some noise can be added to the training samples, or to the test sample, or both.

    In the specific domain of DNNs, random noise will not demonstrate anything, and robustness against adversarial examples is another challenge that still requires some research - and it seems we are still far from benchmarks.
  - ○ Explainability on the other hand will probably require defining some proxies: complexity is one, though not satisfying on its own, and one difficulty will be to come up with some agnostic indicator (even complexity is model-dependent).

Interaction with users might be mandatory, as explainability depends on the expertise of the target human.
    - Fairness in the context of supervised learning, requires probably even more work, as some biased datasets need to be designed (but many models of biases can be used, from unbalanced training set to purposely biased outcomes), and fairness of the resulting models measured - all fields still subject to active research.
- Using the above indicators, extend existing AutoAI algorithms that operate on complex search spaces using Evolutionary Computation to multi-objective settings, and discover new learning paradigms following the path opened by AutoML-Zero [ReaEtAl20].

6. Ever-learning AutoAI [T7.5, TU/e]

“Ever-learning AutoAI” aims to ensure that AutoAI gets better over time, producing better models with less data, and avoids the computational overhead of starting from scratch for any new use case, or change in scenario.

The science of learning how to learn better or faster through experience is called meta-learning (or learning to learn). This can be done in several ways:

- Keeping the model architecture and design decisions (hyperparameters) fixed, and then learning good initial model parameters and/or how to update them for a certain set of tasks. This is useful when many similar tasks exist, and the model itself is large and flexible (e.g., a large neural network), so that re-tuning the weights, without changing the architecture, is sufficient to adapt to new tasks.
- Meta-learning the model architecture itself. This is the combination of meta-learning and AutoAI, which is more generally applicable since it can also work when new tasks are quite different from previous tasks. This also works with smaller models as they can completely adapt to new tasks. Ideally it also leads to models which are smaller and more tuned to the given task. We will focus mainly on this approach in this WP.
- Doing any of the above, while also choosing or generating the next task to solve. The idea here is to learn simple variations of a task first, and use that experience to learn increasingly hard variations of the task.

A notable related field is continual learning, where there is a single task, but it itself evolves over time (see, e.g., [DelEtAl21]). New data points may come in that are slightly different from those before (concept drift), which may require retraining or even a change in architecture. For instance, if outliers suddenly appear, the learning pipeline may need to be adapted to deal with these outliers.

While this is an active and fast evolving field, it is also quite young, and there exist almost no commonly used benchmarks.
For one particular subfield, few-shot learning, there exists the meta-dataset

- Triantafillou E, Zhu T, Dumoulin V, Lamblin P, Evci U, Xu K, Goroshin R, Gelada C, Swersky K, Manzagol PA, Larochelle H. Meta-dataset: A dataset of datasets for learning to learn from few examples. arXiv preprint arXiv:1903.03096. 2019. https://github.com/google-research/meta-dataset

This is a curated collection of 10 object recognition (vision) tasks. While it is well designed and has some adoption, few researchers test their algorithms on all 10 tasks: since these datasets are large and each requires pretraining large models, this is often prohibitively expensive. Often, only a few tasks are selected (e.g., omniglot and/or mini-imagenet). This limits its impact as a benchmark. Some examples of its use by TAILOR partners are:

- **Elsken et al. 2017, Meta-NAS. By ALU-FR. Uses metalearning to speed up NAS and evaluates on 2 tasks also appearing in the meta-dataset.**
- **Gonzalez and Vanschoren, 2019, Meta-Reinforcement learning for NAS. By TUE. Trains an RL agent to do NAS and evaluates on 4 tasks from the meta-dataset.**

For the combination of metalearning and AutoAI, OpenML is often used as a source of meta-data to either learn which hyperparameters are most important to tune [**VanHut18**], to warm-start the search for good model configurations based on what worked on similar tasks [**FeuEtAl19**], or to train meta-models to predict which configurations will work well [**BraEtAl21**]. Especially the OpenML100 or OpenML-CC18 mentioned earlier are used often, although some authors also use datasets other than those from OpenML.

To improve the state of benchmarking in this field, our efforts in TAILOR concentrate on three research lines:

- Set up central infrastructure, building on OpenML, to collect large amounts of meta-data. This will represent a ‘global memory’ of AI approaches and their performance that will speed up and imbue new rigor to AI research.
- Create meta-datasets, curated sets of many related tasks, to stimulate research into techniques that can learn effectively across tasks, and benchmark them to measure progress in meta-learning.
- Nurture research in meta-learning and transfer learning to leverage this meta-data to build better models, as well as research that combines these techniques with AutoAI approaches to yield ever-learning AutoAI techniques.

Progress

We created the AutoML Benchmark, an open-source benchmarking framework for AutoML frameworks:

- **Gijsbers P, LeDell E, Thomas J, Poirier S, Bischl B, Vanschoren J. An open source AutoML benchmark. arXiv preprint arXiv:1907.00909. 2019. https://openml.github.io/automlbenchmark/**

This benchmark has known widespread adoption since then:

- Over 15 AutoML frameworks were included by their original authors, many of which are from industry and some are D3M performers. This includes AutoGluon (Amazon), AutoSKLearn (U Freiburg), GAMA (U Eindhoven), H2O-AutoML (H2O), ML.NET AutoML (Microsoft), Auto-XGBoost and MLR3AutoML (U Munich), FLAML (Microsoft), MLJAR-AutoML (MLJAR), OBOE (Cornell), LightAutoML (Sberbank AI), hyperopt-sklean (U Waterloo), and MLPlan (U Paderborn).
- A first run was completed in 2019, with results published at the ICML AutoML Workshop. A second run has been started to evaluate a much wider range of AutoML systems on more than 90 classification and regression datasets. We will submit the results to the JMLR journal.

- To ensure fair evaluation, all systems are run with equal resources as docker images on identical hardware on AWS.

One key characteristic of this benchmark is that it is dynamic: the collection of tasks used changes every year, to avoid that people (or AutoML systems) overfit on a certain set of dataset. It shares this characteristic with Dynabench, a dynamic benchmark of NLP tasks created by Facebook:

- Nie Y, Williams A, Dinan E, Bansal M, Weston J, Kiela D. Adversarial NLI: A new benchmark for natural language understanding. arXiv preprint arXiv:1910.14599. 2019

In future work, we aim to extend this benchmark to explicitly allow meta-learning under certain controlled conditions, to encourage the wider adoption of Meta-Learning in these AutoML frameworks.

We are also upgrading OpenML in several ways to improve benchmarking on a wider range of tasks. OpenML already has extensive support for creating new benchmarks and making them easy to use, but was limited in the range of datasets it could efficiently serve.

- The OpenML API has been updated to allow binary data formats (in particular Parquet). This allows the inclusion of more types of datasets into OpenML, including large image and text datasets. We successfully ran and stored experiments on datasets of 8GB and larger.
- The OpenML Python library [**FeuEtAl21**] has been updated to transparently handle binary formats. End users can submit and receive datasets natively from many Python data structures (e.g. Pandas, Dask, Numpy,...), facilitating experimentation (e.g. evaluations of new pipelines). They can also easily run experiments and stream results back to the platform.
- We have close to 200k yearly users and close to 1k daily visitors to the website, in addition to high-frequency bursts of calls to our APIs. Our servers are regularly experiencing heavy loads. We set up a Kubernetes environment and S3 storage to ensure scalability, high availability, and conformance to modern standards.

In future work, we aim to leverage OpenML to create new benchmarks that can serve as a reference for work leveraging meta-learning in AutoAI. For this, we also aim to offer new services that compute and return rich meta-data that can be directly used by any AutoML system. This will drastically reduce the cost of leveraging meta-learning, lower the threshold to use meta-learning more intensively in AutoML systems, while at the same time also making it more systematic, reproducible, and comparable.

Finally, ULEI and TUE co-organised two meta-learning challenges, one at AAAI 2021 and one at NeurIPS 2021. In these challenges, AutoAI systems for NAS are allowed run-up time for Meta-Learning: instead of starting the search from scratch, they are allowed to spend some time to collect metadata about the challenge tasks and use that to search for the best architectures more efficiently. While these challenges only offer resources for a certain amount of time (until the challenge is over), they can serve as inspiration for future benchmarks in this area.

The most recent work in this area can be found in the NeurIPS workshop on meta-learning (https://meta-learn.github.io/2020/) and the ICML 2021 workshop on AutoML (https://sites.google.com/view/automl2021).

Works published since September 2020 (highlighted if part of TAILOR):

1. **Brazdil, P., van Rijn, J.N., Soares, C., Vanschoren, J. Metalearning. Applications to Automated Machine Learning and Data Mining. Springer 2021**
2. **Celik, Bilge, and Joaquin Vanschoren. "Adaptation strategies for automated machine learning on evolving data." IEEE Transactions on Pattern Analysis and Machine Intelligence (2021).**
3. **Thomas Elsken and Benedikt Staffler and Jan Hendrik Metzen and Frank Hutter. Meta-Learning of Neural Architectures for Few-Shot Learning, arxiv 1911.11090**
4. **Feurer, Matthias, et al. "Openml-python: an extensible python api for openml." Journal of Machine Learning Research 22.100 (2021): 1-5.**
5. **Gijsbers, Pieter, et al. "Meta-Learning for Symbolic Hyperparameter Defaults." GECCO 2021 (2021).**
6. **Gijsbers, Pieter, and Joaquin Vanschoren. "GAMA: a General Automated Machine learning Assistant." ECMLPKDD 2020 (2020).**
7. **Weerts, Hilde JP, Andreas C. Mueller, and Joaquin Vanschoren. "Importance of tuning hyperparameters of machine learning algorithms." arXiv preprint arXiv:2007.07588 (2020).**

References

[BloEtAl15] Blot A, Aguirre H, Dhaenens C, Jourdan L, Marmion ME, Tanaka K. Neutral but a winner! How neutrality helps multiobjective local search algorithms. InInternational Conference on Evolutionary Multi-Criterion Optimization 2015 Mar 29 (pp. 34-47). Springer, Cham.

[BloEtAl17] Blot A, Jourdan L, Kessaci MÉ. Automatic design of multi-objective local search algorithms: case study on a bi-objective permutation flowshop scheduling problem. InProceedings of the Genetic and Evolutionary Computation Conference 2017 Jul 1 (pp. 227-234).

[BraEtAl21] Brazdil, P., van Rijn, J.N., Soares, C., Vanschoren, J. Metalearning. Applications to Automated Machine Learning and Data Mining. Springer 2021

[CoeLec02] Coello CC, Lechuga MS. MOPSO: A proposal for multiple objective particle swarm optimization. InProceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600) 2002 May 12 (Vol. 2, pp. 1051-1056). IEEE.

[DebEtAl02] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," in IEEE Transactions on Evolutionary Computation, vol. 6, no. 2, pp. 182-197, April 2002, doi: 10.1109/4235.996017.

[DelEtAl21] Delange, Matthias, et al. "A continual learning survey: Defying forgetting in classification tasks." IEEE Transactions on Pattern Analysis and Machine Intelligence (2021).

[DerEtAl16] Bilel Derbel, Arnaud Liefooghe, Qingfu Zhang, Hernan Aguirre, Kiyoshi Tanaka. Multi-objective Local Search Based on Decomposition. International Conference on Parallel Problem Solving from Nature (PPSN 2016), 2016, Edinburgh, United Kingdom. pp.431 - 441

[DubEtAl15] J. Dubois-Lacoste, M. López-Ibáñez and T. Stützle, "Anytime Pareto local search", Eur. J. Oper. Res., vol. 243, no. 2, pp. 369-385, 2015.

[EmmEtAl16] Emmerich M, Yang K, Deutz A, Wang H, Fonseca CM. A multicriteria generalization of bayesian global optimization. InAdvances in Stochastic and Deterministic Global Optimization 2016 (pp. 229-242). Springer, Cham.

[FeuEtAl19] Feurer M, Klein A, Eggensperger K, Springenberg JT, Blum M, Hutter F. Auto-sklearn: efficient and robust automated machine learning. In Automated Machine Learning 2019 (pp. 113-134). Springer, Cham.

[FeuEtAl21] Feurer, Matthias, et al. "Openml-python: an extensible python api for openml." Journal of Machine Learning Research 22.100 (2021): 1-5.

[GalEtAl20] Galuzio PP, de Vasconcelos Segundo EH, dos Santos Coelho L, Mariani VC. MOBOpt—multi-objective Bayesian optimization. SoftwareX. 2020 Jul 1;12:100520.

[HutEtAl09] Hutter F, Hoos HH, Leyton-Brown K, Stützle T. ParamILS: an automatic algorithm configuration framework. Journal of Artificial Intelligence Research. 2009 Oct 30;36:267-306.

[MısSeb17] Mısır M, Sebag M. Alors: An algorithm recommender system. Artificial Intelligence. 2017 Mar 1;244:291-314.

[OlsEtAl16] R. S. Olson, N. Bartley, R. J. Urbanowicz, and J. H. Moore. Evaluation of a Tree-based Pipeline Optimization Tool for Automating Data Science. In Proc. ACM-GECCO, pages 485–492. ACM Press, 2016

[RakEtAl19] Herilalaina Rakotoarison, Marc Schoenauer, Michèle Sebag. Automated Machine Learning with Monte-Carlo Tree Search. IJCAI-19 - 28th International Joint Conference on Artificial Intelligence, Aug 2019, Macau, China. pp.3296-3303

[ReaEtAl20] Real E, Liang C, So D, Le Q. Automl-zero: Evolving machine learning algorithms from scratch. InInternational Conference on Machine Learning 2020 Nov 21 (pp. 8007-8019).

[VanHut18] Van Rijn, Jan N., and Frank Hutter. "Hyperparameter importance across datasets." Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2018.