We Won't Get Fooled Again: When Performance Metric Malfunction Affects the Landscape of Hyperparameter Optimization Problems

Kalifou René Traoré[1,2(✉)], Andrés Camero[2,3], and Xiao Xiang Zhu[1]

[1] Data Science in Earth Observation, Technical University of Munich, Arcisstrasse 21, Munich, Germany
xiaoxiang.zhu@tum.de
[2] Remote Sensing Institute, German Aerospace Center (DLR), Münchener Strasse 20, Weßling, Germany
{kalifou.traore,andres.camerounzueta}@dlr.de
[3] Helmholtz AI, Neuherberg, Germany

Abstract. Hyperparameter optimization (HPO) is a well-studied research field. However, the effects and interactions of the components in an HPO pipeline are not yet well investigated. Then, we ask ourselves: *Can the landscape of HPO be biased by the pipeline used to evaluate individual configurations?* To address this question, we proposed to analyze the effect of the HPO pipeline on HPO problems using fitness landscape analysis. Particularly, we studied over 119 generic classification instances from either the DS-2019 (CNN) and YAHPO (XGBoost) HPO benchmark data sets, looking for patterns that could indicate evaluation pipeline malfunction, and relate them to HPO performance. Our main findings are: (i) In most instances, large groups of diverse hyperparameters (i.e., multiple configurations) yield the same *ill* performance, most likely associated with majority class prediction models (predictive accuracy) or models unable to attribute an appropriate class to observations (log loss); (ii) in these cases, a worsened correlation between the observed fitness and average fitness in the neighborhood is observed, potentially making harder the deployment of local-search-based HPO strategies. (iii) these effects are observed across different HPO scenarios (tuning CNN or XGBoost algorithms). Finally, we concluded that the HPO pipeline definition might negatively affect the HPO landscape.

Keywords: Hyperparameter Optimization · Fitness Landscape Analysis · Benchmarking

1 Introduction and Related Work

Modern data-driven approaches dealing with large-scale data require domain, data science, and technical expertise. The variety of application tasks (e.g., classification and object detection) often require designing models that are not necessarily reusable in other tasks, and this process is both resource-demanding

B. Dorronsoro et al. (Eds.): OLA 2023, CCIS 1824, pp. 148–160, 2023.
https://doi.org/10.1007/978-3-031-34020-8_11

and error-prone [3,8,12]. Thus, automating the design of ML pipelines, a.k.a. *AutoML* [6], is much more desirable.

AutoML is usually split into four main activities: data preparation, feature engineering, model generation, and model estimation [5]. *Hyperparameter optimization* (HPO [1]) is an important task in model generation. HPO aims at automatically tuning the hyperparameters of learning algorithms, and as with all optimization problems, it is facing the process of minimizing/maximizing a target function (e.g., the performance metric of the model) subject to a set of constraints. HPO is a well-studied field [1], but the effects and interaction between the components of its pipeline are not yet well investigated. Recently, authors [10] have proposed to characterize the search space of AutoML pipelines using *fitness landscape analysis* (FLA [11]). In the same line, [15] proposed a FLA-base framework to characterize NAS problems, and applied it to a multi-sensor data fusion problem [14]. Despite the great results and insights provided by these studies, the relation between HPO and the rest of the HPO pipeline remains barely explored.

Therefore, in this study, we pose the following research question: ***Can the landscape of HPO be biased by the pipeline used to evaluate individual configurations?*** To address this question, we propose to study HPO in the context of AutoML using FLA. Particularly, using *fitness distance correlation* (FDC [7]), *locality* and *neutrality* [2], we aim at patterns that arise from evaluation pipeline issues and assess how they could alter the landscapes of HPO problems. The results on over 119 instances from either the DS-2019 [13] or the YAHPO [9] HPO benchmarks show the existence of large groups of diverse HP configurations that yield the same *ill* fitness value. This *illness* could be explained by the fitness metric selection (e.g., predictive accuracy and log loss), that *induce* various *suboptimal* model behavior, in scenarios of *different natures* (tuning CNN and XGBoost algorithms). More precisely, for the predictive accuracy, we suspect the generation of majority class predictors. In the case of the log loss criterion, the generation of models unable to classify. A complementary analysis of locality shows that the resulting landscapes are more rugged, with lesser correlation between the observed fitness and the fitness in the neighborhood. In other words, these problems are hard to tackle using a local-search strategy.

The rest of the paper is as follows: The next section introduces the methodology used in the study, Sect. 3 presents results of landscape analysis on HPO problems, and Sect. 4 provides conclusions.

2 Methodology

Given a HPO problem, let S be the HP configuration space, f the fitness function that assigns a value $f(x) \in \mathbb{R}$ to all configurations $x \in S$, and $N(x)$ a neighborhood operator that provides a structure to S. Then, the fitness landscape is defined as $\mathcal{L} = (S, f, N)$.

We are interested in exploiting the landscape definition to study the relation between the HPO landscape and the HPO pipeline, and check whether the

pipeline may bias the HPO landscape. Particularly, we propose to use the FDC and *locality* to characterize this relation. The motivation is that issues related to the evaluation pipeline should affect the fitness of configurations irrespectively of their configuration, and thus their distance to the optimum. In other words, repetitive or grouping patterns (such as lines) might appear when visualizing distributions of distances to the optimum. Moreover, the locality of the configuration space should be arbitrarily affected, i.e., some configurations should present an unexpected or *random* behavior (in relation to the neighborhood).

Without loss of generality, we consider the problem of tuning the HPs of a fixed model, e.g. neural network architecture or XGBoost ensemble, to perform a task (e.g., classification). Typically, the HP configuration space consists of mixed type features (continuous, discrete or categorical). Thus, we propose to evaluate the distance between individuals using a dedicated similarity function, $\delta(x, y)$, introduced by [4]. Then, we define a neighborhood function $N(x) = \{y \in S \mid \delta(x, y) < \Delta\}$.

The FDC is often interpreted as a measure of the existence of search trajectories from randomly picked configurations to the known global optimum. In practice, the FDC is not collected as a correlation score but visualized as the distribution of fitness versus distance to the global optimum. It writes as: $\text{FDC}(f, x^*, S) = \{(\delta(x^*, y), f(y)) \mid \forall y \in S\}$, where $x^* \in S$ is the global optimum. On the other hand, *locality* corresponds to the relationship between the observed fitness and the distribution of average fitness in the neighborhood [2].

Moreover, the *neutrality degree* [2] provides an additional picture of the interaction between solutions in the landscape. It is defined as $N_d(x) = |\{x^{'} \in N(x) \mid\mid f(x^{'}) - f(x) \mid < \epsilon\}|$, and is interpreted as the number of neighbors of x that have a *similar* fitness. In this case, we set $\epsilon = \texttt{max}_{\text{fitness}}/C$. Besides, Table 1 and 2 give a description of abbreviations and symbols used in the paper.

Table 1. Table of abbreviations used in the paper.

Abbreviation	Description
HP	Hyperparameter
HPO	Hyperparameter Optimization
CV	Computer Vision
FLA	Fitness Landscape Analysis
FDC	Fitness Distance Correlation
CNN	Convolutional Neural Network
MMCE	Mean Misclassification Error

3 Results

To evaluate the proposed methodology, we propose to analyze the **DS-2019** and **YAHPO** HPO benchmark data sets. DS-2019 consists of a tabular benchmark for the scenario of tuning the HPs of a (fixed) convolutional neural network

(CNN), a ResNet-18, on ten instances of CV classification. For each instance, 15 hyperparameters should be optimized, including the batch size, number of epochs, and momentum, among others.

Table 2. Table of symbols used in the paper.

Symbol	Description
S	Hyperparameter Configuration Space
N	Neighborhood operator
f	Fitness evaluation function
$\mathcal{L}$	Fitness landscape derived from the combination of S, N, f
$\texttt{max}_{\text{fitness}}$	Maximum fitness value observed in S
$\texttt{max}_{\text{dist}}$	Maximum pairwise distance (to the optimum) measured in S
$N_d(x)$	Neutrality degree of a HP solution x
$\delta(x, y)$	Gower distance between HP solutions x and y
Δ	Threshold (in Gower distance) used by N to assign neighbors
ϵ	Threshold (in fitness) used by N_d to assign neutral neighbors
C	Constant used to define the fitness neutrality threshold ϵ

YAHPO consists of a tabular benchmark for the scenarios of tuning various learning algorithms (e.g. XGBoost, Neural Networks) for 119 instances of classification, in various domains of applications. Many of the instances were obtained from the collaborative open-source OpenML platform, gathering an ever-growing number of machine learning instances. In this study, we focus on the specific scenario tuning a set of 15 HPs for the XGBoost learning algorithm.

The code used for the experiments is available following the anonymized link: https://github.com/anonymous-for-open-review/late-breaking-automlConf-2022.

3.1 Classification Accuracy

The following paragraphs introduce results when evaluating solutions using the metric of *predictive accuracy* (DS-2019). In the case of YAHPO, the available metric is the *mean misclassification error* (*MMCE* = *1* − *predictive accuracy*), and we focus on the instances *WDBC, YEAST, MINIBOONE, and ISOLET*. Similar observations are made for the rest of the 119 instances.

Fitness Distance Correlation (FDC)
First, for each instance, we randomly sampled 1000 HP configurations and computed the FDC. Results are shown in Figs. 1 and 2, respectively, for DS-2019 and YAHPO.

Overall, the distances to the global optimum cover a wide range of values: the distribution of distances is wide and uniform in most instances. This is the case for both benchmarks. This suggests a large diversity in the HP configurations (with respect to the optimum), for the sample and potentially the whole configuration space. This is true for both benchmarks, with slightly more narrow distributions of distances for YAHPO. This suggests slightly less diversity within the XGBoost HP configuration space (YAHPO), than one of the CNN classifiers (DS-2019).

In most instances of DS-2019, the fitness also covers a wide range of values, as opposed to relatively more narrow distributions of fitness on YAHPO. This suggests a larger influence on HP configuration on the fitness of CNN classifiers (DS-2019) than on more classical and notoriously robust ensembles of models, i.e XGBoost (YAHPO). Overall, the distributions of fitness all seem to be multimodal, with a principal mode for large fitness values (i.e., good configurations), and another mode for *odd* values, for both benchmarks.

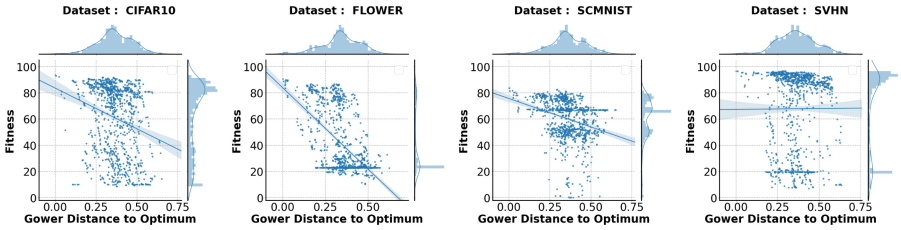

Fig. 1. FDC plot for instances from the DS2019 benchmark, and the corresponding regression line in blue. The fitness function is the *predictive accuracy*. (Color figure online)

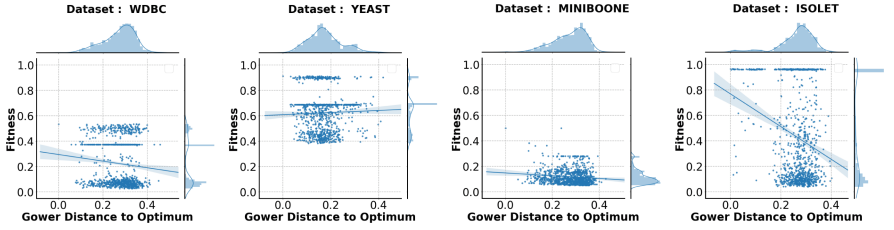

Fig. 2. FDC plot for a few instances from the YAHPO HPO benchmark, and the corresponding regression line in blue. The fitness function is the MMCE. (Color figure online)

Besides, we checked the data distribution for each instance, and we notice that the *odd* modes could be correlated to the majority class. Note that the fitness metric used is predictive accuracy for DS-2019, and its opposite the MMCE for YAHPO. For example in DS-2019, on FLOWER it is around 25%, SCMNIST around 65% and SVHN around 20%. In YAHPO, on WDBC it is around 38%, on YEAST around 70%, and ISOLET around 97%, among others. In particular, configurations are affected regardless of the distance to the optimum. In other words, very diverse configurations yield the same fitness value. This phenomenon could be attached to issues with the learning process, failing to properly fit the data and being stuck in poor local optima (i.e., majority class prediction), preventing them to reach the fitness that their HP configuration would normally yield. Besides, there is no clear global correlation between the observed fitness and distance to the global optimum. This could be caused by the multi-modal nature of the distributions of fitness.

Neighborhood

Next, we seek to identify how the observed artifacts, i.e., the *majority* class predictors, affect the locality of landscapes. Figures 3 and 4 show the distribution of average neighbor fitness as a function of the observed fitness, respectively, for instances from DS-2019 and YAHPO. The black dash-dotted line represents the bisector, i.e., the line connecting all points of equal value on both axis. To generate the plots, we used the previously sampled configurations, and identified the maximal pairwise distance (of any individual) to the optimum $\texttt{max}_{\text{dist}}$, and maximum observed fitness $\texttt{max}_{\text{fitness}}$. Given a constant $C = 40$, we discretize the range of fitness values into intervals, where a step is equal to the maximum observed fitness $\texttt{max}_{\text{fitness}}$ divided by C. In order to decide if a configuration is a neighbor, we set $\Delta = \texttt{max}_{\text{dist}}/C$.

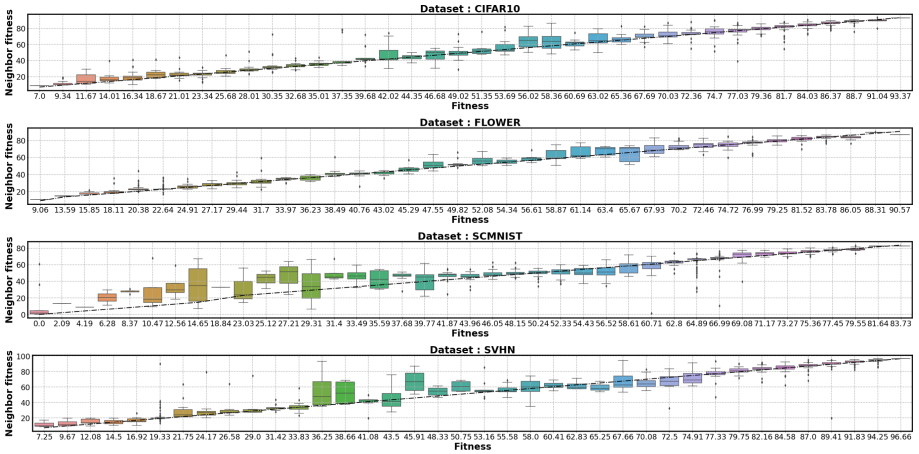

Fig. 3. Distribution of the average fitness of neighbors as a function of the observed fitness (*predictive accuracy*), for a few instances of the DS2019 benchmark.

Overall, we observe in many instances of DS-2019 a strong correlation between the observed fitness and the average fitness in the neighborhood. Indeed, the box-plots are aligned with the bisector. From the perspective of local search, it is easy to navigate the configuration space by consistently improving the fitness, from randomly distant and bad configurations to configurations of high fitness, for instances from DS-2019. This is less the case in YAHPO, as show in Fig. 4. Indeed, we observe a weaker correlation between the two variables. In the presence of unexpected mode in the distributions of fitness (see Fig. 2), e.g. YEAST and ISOLET, we find that a majority of neighbors tend to have the fitness of the observed mode, respectively around 70% and 97%. This suggests that the respective landscapes might have many local optima surrounded by plane areas at the odd fitness value. Thus, the chances for local search of being stuck are higher in such instances. Besides, the instances with more uniform and wider distribution of fitness (Fig. 1) tend to have a near perfect correlation. On the other hand, the more the distributions are multi-modal and with peaky modes, the worse the correlation between the variables of interest. This is the case for both benchmarks.

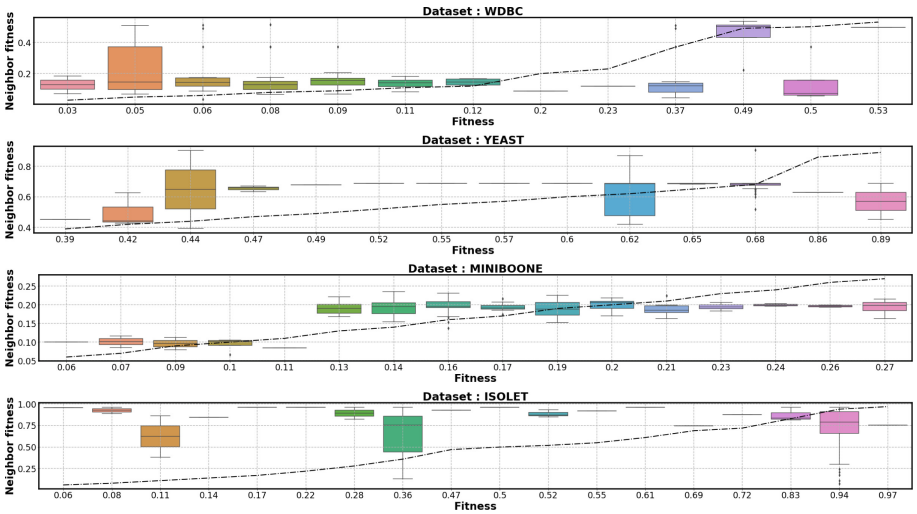

Fig. 4. Distribution of the average fitness of neighbors as a function of the observed fitness (MMCE), for a few instances from the YAHPO benchmark.

To summarize, results indicate that the evaluation protocol could have an impact on the easiness and practicability of HPO landscapes, assessed by the correlation.

Neutrality

Next, we look into the neutrality of the landscape. Figures 5 and 6 show the distribution of neutral neighbor counts as a function of the current fitness, respectively, for instances from DS-2019 and YAHPO.

For instances from DS-2019, the neutrality degree is equal to or greater than one, for most ranges of fitness values. In order words, most configurations have at least one neutral neighbor. Also, note that the FDC and *locality* results for CIFAR-10 are *good*, while for SCMNIST adn SVHN, with a multi-modal distribution of fitness (Fig. 1), coupled with lower *local* correlation (i.e., between the fitness and the fitness in the neighborhood, Fig. 3), the results are *bad*. Regarding CIFAR-10, the neutrality degree is on average consistently greater than two. In other words, most configurations have two or (many) more *neutral* neighbors. On the other hand, for SCMNIST and SVHN, the neutrality degree is inconsistent and with lower values on average. In particular, N_d is lower for fitness values ranging from 6.28 to 43.98% for SCMNIST, i.e., generally bad configurations have fewer neutral neighbors than *mid* and *good* configurations. Also, as expected, there is a huge number of neutral neighbors around the majority class prediction fitness: around 65% for SCMNIST and 20% for SVHN.

In YAHPO, we find that the range of fitness for which one can find solutions with neutral neighbors is limited. In practice, it represents a fraction of the range covered by all evaluated solutions. Besides, the neutrality degree is generally above 2, with larger counts associated to the modes in the distributions of fitness (See Fig. 2). These two facts suggest that the landscapes might be highly rugged with many local optima (no neutral neighbors), with areas centered towards values of the observed modes.

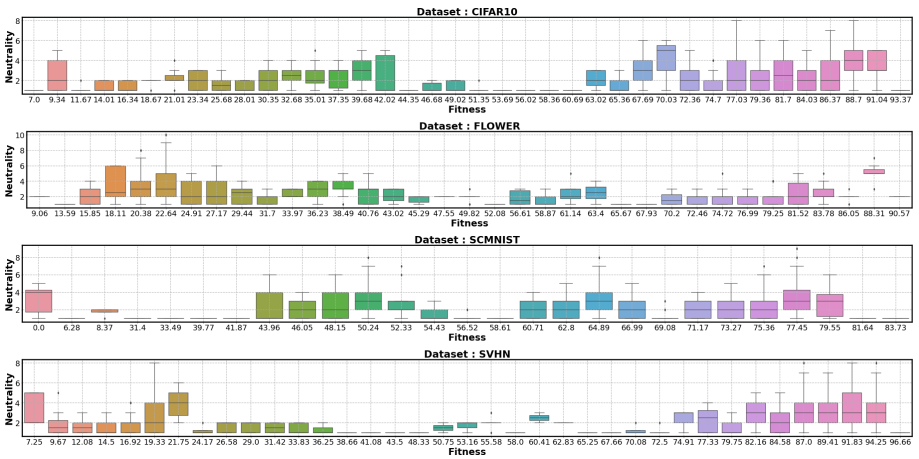

Fig. 5. Neutrality degree as a function of the observed fitness (*predictive accuracy*), for a few instances from the DS2019 benchmark.

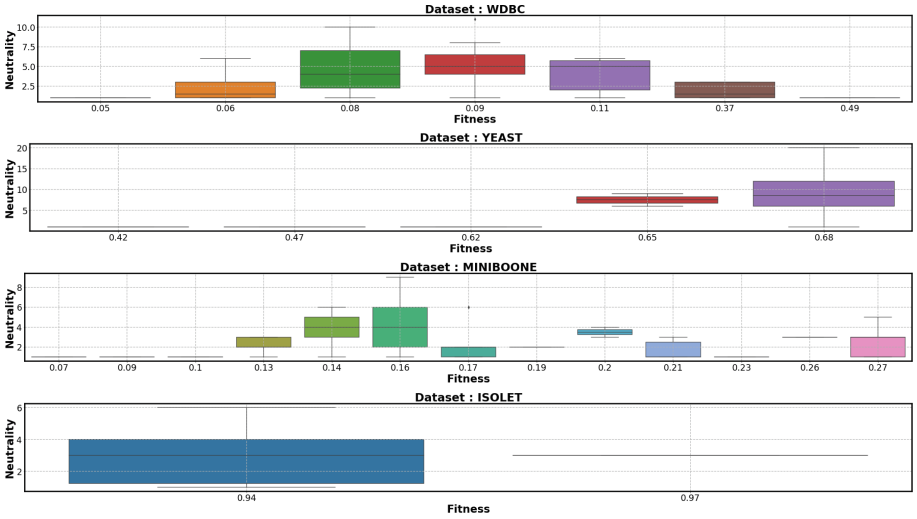

Fig. 6. Neutrality degree as a function of the observed fitness (MMCE), for a few instances from the YAHPO benchmark.

As a summary, the evaluation pipeline *malfunction* is responsible for an *imbalanced* landscape, i.e., the AutoML pipeline generates arbitrary *peaks* of fitness (low N_d) in areas of expected continuous fitness.

3.2 Log Loss

The following paragraphs present the results of the analysis when evaluating solutions using the Log Loss for classification. This is done on a sub-sample of the YAHPO benchmark, namely *SEMEION, VEHICLE, SEGMENT, KC1.* Similar observations are made for the remaining 115 instances.

Fitness Distance Correlation (FDC)
First, we look at the FDC for the four instances *SEMEION, VEHICLE, SEGMENT, KC1*, as shown in Fig. 7.

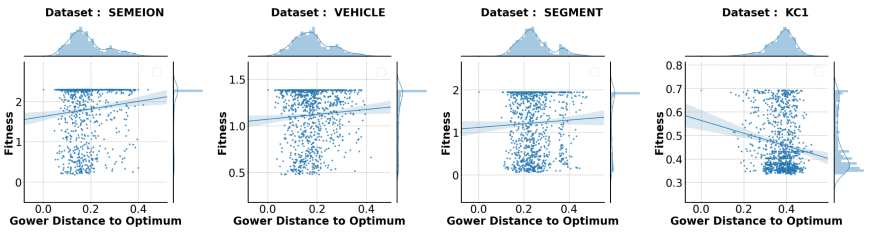

Fig. 7. FDC plot for a few instances from the YAHPO HPO benchmark, and the corresponding regression line in blue. The fitness function is the Log Loss. (Color figure online)

Similarly to results gathered in Sect. 3.1 (*predictive classification accuracy*), we find that the distributions of fitness are also covering a wide range of values, and are multi-modal. Several distributions also have the artifact identified previously: an unexpected mode around large (*poor*) fitness values, associated with HP configurations of highly variable *Gower* dissimilarity to the optimum. For instance, it is around the Log Loss value of 2.25 for SEMEION, 1.55 for VEHICLE, and 1.98 for SEGMENT. It affects HP configurations at Gower distances 0.15 to 0.45, i.e. covering the whole range of dissimilarity to the optimal HP. This phenomenon is also observed for a majority of the 119 analyzed instances of YAHPO. Besides, another mode can exist (around 0.35 for KC1) at low fitness values, i.e *good* HP configurations.

When looking at the nature of the instances (number of classes), we find that the unexpected mode correlates with the fitness value yielded by the Log Loss metric when attributing an *equal* probability of occurring to all classes, for all observations. In other words, solutions associated with the mode are likely solutions with *no classification ability*. This phenomenon could occur since the Log Loss metric does not penalize such behavior, a local optimum to which many solutions could naturally converge to.

Neighborhood

Next, we look at the neighborhood in the landscapes generated by the Log Loss metric. This is shown in Fig. 8. Overall, we find that the correlation between the average fitness of neighbors and the observed fitness, is weak in the case of distributions of fitness with the identified artifact (*peaks*). Most neighbors have a fitness value of the unexpected mode, e.g. a Log Loss value of 2.25 for SEMEION

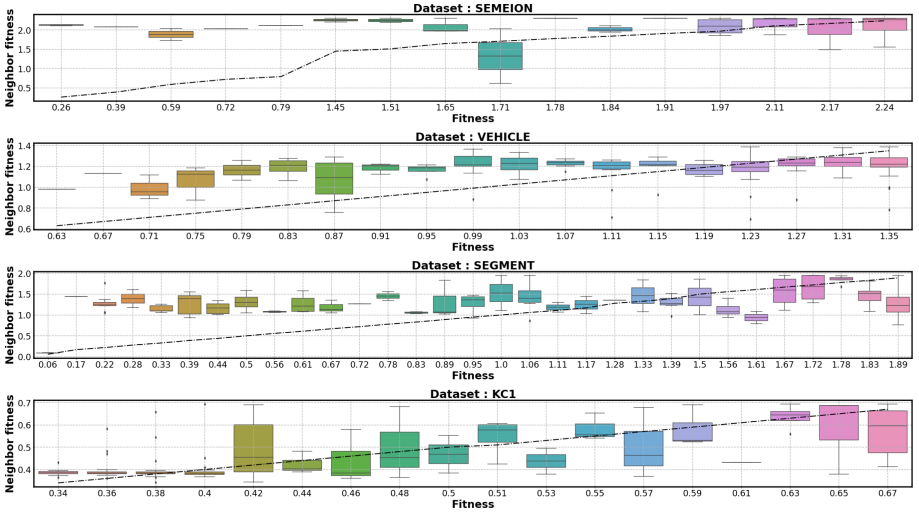

Fig. 8. Distribution of the average fitness of neighbors as a function of the observed fitness (Log Loss), for a few instances from the YAHPO benchmark.

and 1.35 for VEHICLE. This observation is in line with those made when using the metric of MMCE (see Figs. 1 and 2).

To summarize, using the Log Loss as an evaluation metric might also negatively impact the easiness of the associated landscapes, by increasing their ruggedness and decreasing their fitness potential.

Neutrality

Next, we look into the neutrality of the landscapes, as shown in Fig. 9. Similar to the analysis provided when evaluating with the MMCE, we find that few solutions have a neutral neighbor, and these are found within a restricted range of fitness. This suggests highly rugged landscapes. Besides, the highest counts of neutral neighbors are for solutions associated with the modes in the distributions of fitness (see Fig. 7). For instance, at a Log Loss value of 2.2 for SEMEION, and between 1.23 and 1.35 for VEHICLE. In other words, the landscapes are highly rugged (numerous local minima), and surrounded by a plateau of HP configurations associated with the high Log Loss values of the observed mode.

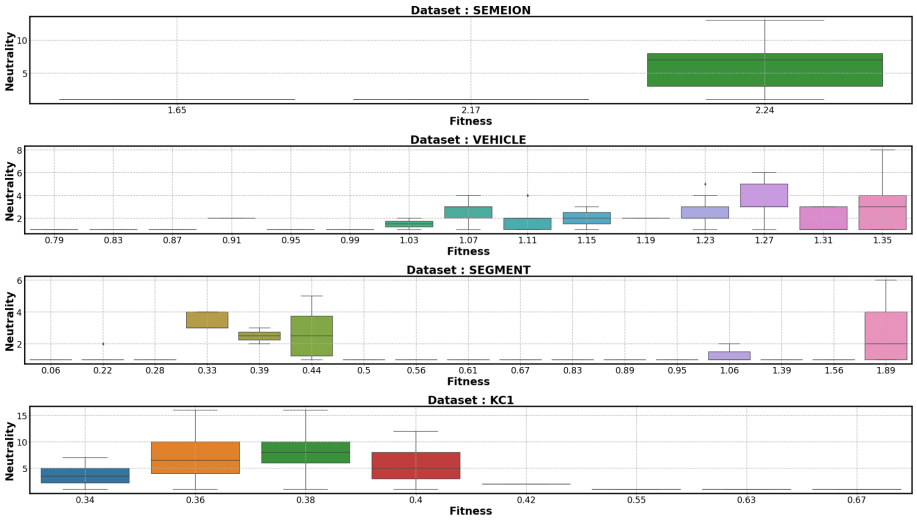

Fig. 9. Neutrality degree as a function of the observed fitness (Log Loss), for a few instances from the YAHPO benchmark.

4 Conclusions and Future Work

In this paper, we investigate if AutoML pipelines can negatively affect the landscape of HPO problems. More precisely, we address the following question: *Can the landscape of HPO be biased by the pipeline used to evaluate individual configurations?* To tackle this question, we have studied the fitness landscape of over 119 HPO instances obtained from either the DS-2019 (CNN) or YAHPO

(XGBoost) HPO benchmark data sets, using the concepts of *fitness distance correlation*, *locality*, and *neutrality*.

The FDC analysis shows unhealthy patterns in many HPO instances, with large groups of very diverse HP configurations with the same *ill* fitness value. These resulting peaks in fitness appear to be outliers in the respective distributions. Looking at the locality (fitness versus fitness in the neighborhood), we observe two things: First, there is a correlation between both variables of interest in healthy landscapes, suggesting that an *easy* path from randomly picked HP configurations could lead to the best performers, i.e., local-search may *do the job*. Second, for HPO problems negatively affected by the mentioned *illnesses* (i.e., the majority class predictors, or the inability to classify), the correlation between the current fitness and fitness in the neighborhood is worsened, indicating more rugged local landscapes.

Even though the *majority class prediction* problem for models trained and evaluated using some metrics (e.g., accuracy) is well known, the results show that the problem may not be taken seriously into account. This is also the case with the *inability to classify* arising when using the Log Loss as a training and evaluation criterion.

Thus, a great amount of resources is wasted when addressing HPO (i.e., many *simple* majority class or *inable* models are evaluated). Furthermore, the evidence shows that *the landscape of HPO problems could be negatively affected by the evaluation pipeline being used*.

Future work will further investigate how such artifacts affect HPO algorithms in practice.

Acknowledgements. Authors acknowledge support by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement No. [ERC-2016-StG-714087], Acronym: *So2Sat*), by the Helmholtz Association through the Framework of Helmholtz AI [grant number: ZT-I-PF-5-01] - Local Unit "Munich Unit @Aeronautics, Space and Transport (MASTr)" and Helmholtz Excellent Professorship "Data Science in Earth Observation - Big Data Fusion for Urban Research" (W2-W3-100), by the German Federal Ministry of Education and Research (BMBF) in the framework of the international future AI lab "AI4EO – Artificial Intelligence for Earth Observation: Reasoning, Uncertainties, Ethics and Beyond" (Grant number: 01DD20001) and the grant DeToL. The authors also acknowledge support by DAAD for a Doctoral Research Fellowship.

References

1. Bischl, B., et al.: Hyperparameter optimization: foundations, algorithms, best practices and open challenges (2021). https://doi.org/10.48550/ARXIV.2107.05847, https://arxiv.org/abs/2107.05847
2. Clergue, M., Verel, S., Formenti, E.: An iterated local search to find many solutions of the 6-states firing squad synchronization problem. Appl. Soft Comput. **66**, 449–461 (2018). https://doi.org/10.1016/j.asoc.2018.01.026, https://www.sciencedirect.com/science/article/pii/S1568494618300322

3. Elsken, T., Metzen, J.H., Hutter, F.: Neural architecture search: a survey. J. Mach. Learn. Res. **20**(1), 1997–2017 (2019)
4. Gower, J.C.: A general coefficient of similarity and some of its properties. Biometrics **27**(4), 857–871 (1971). http://www.jstor.org/stable/2528823
5. He, X., Zhao, K., Chu, X.: AutoML: a survey of the state-of-the-art. Knowl.-Based Syst. **212**, 106622 (2021)
6. Hutter, F., Kotthoff, L., Vanschoren, J.: Automated Machine Learning - Methods, Systems, Challenges. Springer, Berlin (2019). https://doi.org/10.1007/978-3-030-05318-5
7. Jones, T., Forrest, S.: Fitness distance correlation as a measure of problem difficulty for genetic algorithms. In: Proceedings of the 6th International Conference on Genetic Algorithms, pp. 184–192. Morgan Kaufmann Publishers Inc., San Francisco (1995)
8. Ojha, V.K., Abraham, A., Snášel, V.: Metaheuristic design of feedforward neural networks: a review of two decades of research. Eng. Appl. Arti. Intell. **60**, 97–116 (2017). https://doi.org/10.1016/j.engappai.2017.01.013, https://www.sciencedirect.com/science/article/pii/S0952197617300234
9. Pfisterer, F., Schneider, L., Moosbauer, J., Binder, M., Bischl, B.: YAHPO gym - an efficient multi-objective multi-fidelity benchmark for hyperparameter optimization (2021)
10. Pimenta, C.G., de Sá, A.G.C., Ochoa, G., Pappa, G.L.: Fitness landscape analysis of automated machine learning search spaces. In: Paquete, L., Zarges, C. (eds.) EvoCOP 2020. LNCS, vol. 12102, pp. 114–130. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-43680-3_8
11. Pitzer, E., Affenzeller, M.: A comprehensive survey on fitness landscape analysis. In: Fodor, J., Klempous, R., Suárez Araujo, C.P. (eds.) Recent Advances in Intelligent Engineering Systems. Studies in Computational Intelligence, vol. 378, pp. 161–191. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-23229-9_8
12. Ren, P., Xiao, Y., Chang, X., Huang, P.y., Li, Z., Chen, X., Wang, X.: A comprehensive survey of neural architecture search: challenges and solutions. ACM Comput. Surv. **54**(4) (2021). https://doi.org/10.1145/3447582
13. Sharma, A., van Rijn, J.N., Hutter, F., Müller, A.: Hyperparameter importance for image classification by residual neural networks. In: Kralj Novak, P., Šmuc, T., Džeroski, S. (eds.) DS 2019. LNCS (LNAI), vol. 11828, pp. 112–126. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-33778-0_10
14. Traoré, K.R., Camero, A., Zhu, X.X.: Landscape of neural architecture search across sensors: how much do they differ ? ISPRS Ann. Photogr. Remote Sens. Spat. Inf. Sci. **V-3-2022**, 217–224 (2022). https://doi.org/10.5194/isprs-annals-V-3-2022-217-2022, https://www.isprs-ann-photogramm-remote-sens-spatial-inf-sci.net/V-3-2022/217/2022/
15. Traoré, K.R., Camero, A., Zhu, X.X.: Fitness landscape footprint: a framework to compare neural architecture search problems (2021)