Overfitting in Combined Algorithm Selection and Hyperparameter Optimization

Sietse Schröder[0009−0002−9130−4546], Mitra Baratchi[0000−0002−1279−9310], and Jan N. van Rijn[0000−0003−2898−2168]

Leiden Institute of Advanced Computer Science (LIACS), Leiden University
sietseschroder@gmail.com

Abstract. Hyperparameter optimization (HPO) aims to design machine learning algorithms that generalize well to unseen data by repeatedly evaluating hyperparameter configurations using a validation procedure. When the validation performance of these configurations is overly optimistic compared to the performance on an unseen test set, this is referred to as meta-overfitting. We decompose meta-overfitting into two types: (i) selection-based and (ii) adaptive overfitting. Selection-based overfitting occurs when testing many configurations, which increases the chance of finding a configuration that performs well on the validation set by chance but performs suboptimal on the test set. Adaptive overfitting arises from advanced HPO methods, such as Bayesian optimization, which iteratively utilize validation results of earlier configurations to propose new configurations increasingly tailored to the specific validation set. We provide one of the largest empirical studies of meta-overfitting in the context of HPO for the Combined Algorithm Selection and Hyperparameter Optimization (CASH) problem, analyzing random search and Bayesian optimization for 48 classification and 16 regression datasets using holdout validation. We show evidence of adaptive overfitting in Bayesian optimization for 41 classification datasets, and consistent with prior work, we show that multiclass datasets are less affected by this phenomenon. Additionally, we find that optimization procedures for regression datasets are surprisingly resilient to adaptive overfitting. Furthermore, we explore the effect of various design choices in the validation procedure (i.e., 10-fold cross-validation and varying holdout set sizes) on meta-overfitting.

Keywords: Overfitting · AutoML · Supervised Learning

1 Introduction

Designing a machine learning pipeline involves selecting various components and tuning their hyperparameters. Hyperparameter optimization (HPO) supports the human in the loop by automatically searching for an optimal pipeline that best generalizes to unseen data, reducing manual effort and empowering non-experts to create complex machine learning workflows [3]. Simultaneously optimizing both the pipeline components as well as their hyperparameters is defined as the combined algorithm selection and hyperparameter optimization (CASH) problem [23].

The generalization performance of hyperparameter configurations is commonly estimated using nested resampling techniques such as k-fold cross-validation or a holdout set. To evaluate an HPO method on unseen data, the dataset is first split into a train and test set. Since the test set can only be used once, the train set is further divided into a train and validation set to evaluate the various configurations suggested throughout the HPO process. Typically, the best-performing configuration on the validation set is selected and evaluated on the test set. Instead of a single holdout set, cross-validation can be applied at either or both levels to improve robustness. These resampling splits are generally predetermined and remain fixed throughout the HPO process. In practical learning scenarios, limited data often restricts the size of resampling splits used for evaluation, leading to less statistically robust estimates of generalization performance [18]. When the performance estimates obtained during validation are overly optimistic compared to the actual performance the pipeline achieves on unseen test data, this is referred to as meta-overfitting, described by the meta-overfitting error (MOE) [1], which measures the difference between the validation estimation and the actual generalization performance. In the context of HPO, meta-overfitting arises because many hyperparameter configurations are evaluated, some naturally performing well on validation data due to chance alone [18]. Since HPO optimizes for the best validation performance, it tends to favor these overfitted configurations over others that might generalize better to unseen data. In this work, we refer to this type of meta-overfitting as *selection-based overfitting.*

Moreover, we hypothesize that meta-overfitting occurs due to the sequential nature of many hyperparameter optimization strategies, which leverage information from previous evaluations to propose configurations increasingly tailored to the specific validation procedure. While this approach may improve validation performance, it does not necessarily enhance the ability to generalize to unseen data. This phenomenon is in a more general context called adaptive data analysis [6], highlighting the risks of adaptively refining solutions based on limited validation data. When this adaptivity leads to overly optimistic performance estimates, it is termed *adaptive overfitting.* Given that state-of-the-art HPO techniques, such as Bayesian optimization, inherently rely on adaptive mechanisms to iteratively refine hyperparameter configurations, it is critical to assess whether the solutions they propose are overfitted due to either adaptive or selection-based effects.

In this study, we conduct a comprehensive empirical investigation into meta-overfitting arising from random search and Bayesian optimization, offering one of the first large-scale analyses of this phenomenon in HPO for the CASH problem. Our key contributions are as follows:

1. We advance the understanding of meta-overfitting in HPO by examining its effects across 64 diverse classification and regression datasets.

2. On four binary classification datasets, we explore the mitigating effect of 10-fold cross-validation (10CV).
3. We identify adaptive overfitting as a significant contributor to meta-overfitting in HPO and provide, to the best of our knowledge, the first evidence of its presence in Bayesian optimization across most of the classification datasets analyzed.
4. We examine the impact of varying validation set sizes on selection-based and adaptive overfitting, extrapolating the practical implications of meta-overfitting across scenarios with differing levels of data availability.

2 Related Work

In this section, we review the background of selection-based and adaptive overfitting and argue their inherent connection to HPO techniques. Furthermore, we review related work on meta-overfitting in HPO, specifically when using state-of-the-art Bayesian optimization.

Selection-based overfitting: Selecting the hypothesis with the highest cross-validation performance from multiple evaluations can lead to biased results and poor generalization [16, 18], the phenomenon we refer to as selection-based overfitting. Subsequent research demonstrated that algorithms for variable selection, also referred to as feature selection, frequently overfit the cross-validation process [21]. Empirical studies reveal that manual model selection and hyperparameter tuning frequently result in models that perform better on validation data than they do on unseen test sets, regardless of whether this performance has been determined by a single nested holdout set or nested cross-validation procedures [5]. Circling back, since HPO algorithms repeatedly evaluate hyperparameter configurations [3], they are susceptible to selection-based overfitting.

Adaptive data analysis and overfitting: Beyond the context of HPO, a famous example of adapting analyses based on validation results is Freedman's paradox [10], which demonstrates that selecting randomly sampled features based on their correlation with a target can yield statistically significant regression results on the validation set, ultimately leading to poor generalization. Similarly, adaptive data analysis, introduced by Dwork et al. [6], explores how adapting models or hypotheses based on intermediate evaluation undermines the statistical validity of the validation results. They show the risk of overfitting to the validation process, often referred to as adaptive overfitting, grows with the number of adaptively chosen evaluations but decreases as the size of the validation dataset increases. Theoretically, multiclass classification was shown to be more resilient to adaptive overfitting compared to binary classification [8]. This effect was empirically observed in HPO using Bayesian optimization in this work.

Despite theoretical concerns, a meta-analysis of Kaggle competitions reported minimal signs of adaptive overfitting [22]. Similarly, studies on widely-used benchmarks like MNIST, CIFAR-10, and ImageNet have found limited evidence of adaptive overfitting [19, 20, 24], suggesting that the practical impact of adaptive overfitting may be context-dependent. Investigating adaptive overfitting in the context of HPO is particularly interesting since Bayesian optimization, which is one of the most commonly used techniques for HPO, inherently selects new hyperparameter configurations based on the performance of previous configurations evaluated on the validation set [2].

Overfitting in HPO: Meta-overfitting in HPO, also known as overtuning or oversearching [16, 17], remains a relatively underexplored topic in the literature [3, 15]. Recent studies have begun addressing this gap. Lévesque [13] conducted a comprehensive evaluation, revealing significant meta-overfitting in Bayesian optimization when tuning SVM hyperparameters across 118 datasets. This study also demonstrated that reshuffling the validation split after each evaluation improved both stability and overall performance. Subsequent research confirmed these findings, showing that generalization performance with reshuffled holdout splits per iteration often matches that of 5-fold cross-validation [15]. Furthermore, early stopping was identified as a viable approach to mitigate meta-overfitting when optimizing hyperparameters for XGBoost and random forests [14]. Fabris and Freitas [7] identified evidence of meta-overfitting in the AutoML tool Auto-sklearn [9], offering a notable exploration of this phenomenon in the context of the CASH problem. However, the generalizability of their results is constrained by two factors: the reliance on a single run per dataset and the incorporation of post-hoc ensembles (i.e., instead of utilizing the best configuration based on the HPO procedure, several of the best configurations are combined into an ensemble). The use of ensembles is reported to improve generalization over individual pipelines, thus reducing meta-overfitting [9].

Our work extends prior research by analyzing 64 datasets (with 100 repetitions each), spanning classification and regression, and uniquely isolating adaptive overfitting as a key contributor to meta-overfitting in HPO for the CASH problem.

3 Problem Definition

We formalize the CASH problem and describe our method for quantifying meta-overfitting in HPO. The CASH problem involves finding the optimal combination of a learning algorithm $A \in \mathcal{A}$ and its hyperparameters $\lambda \in \Lambda$ to minimize a loss function $\mathcal{L}$ (e.g., error rate, RMSE) on validation data $\mathcal{D}_{val}$ after training on $\mathcal{D}_{train}$. As introduced by Thornton et al. [23], the CASH problem is defined as:

$$A^*, \lambda^* \in \underset{A \in \mathcal{A}, \lambda \in \Lambda}{\arg\min}\, \mathcal{L}(A_\lambda, \mathcal{D}_{train}, \mathcal{D}_{val}) \tag{1}$$

During the HPO process, many hyperparameter configurations are evaluated on $\mathcal{D}_{val}$, possibly with an overly optimistic estimation compared to evaluation on a completely unseen, ideally large, test set $\mathcal{D}_{test}$, which remains inaccessible to the HPO algorithm during optimization. We measure this effect with the *meta-overfitting error (MOE)*, which we formally define as:

$$MOE(A_\lambda, \mathcal{D}_{train}, \mathcal{D}_{val}, \mathcal{D}_{test}) := \mathcal{L}(A_\lambda, \mathcal{D}_{train}, \mathcal{D}_{test}) - \mathcal{L}(A_\lambda, \mathcal{D}_{train}, \mathcal{D}_{val}) \quad (2)$$

In this work, an *iteration* is defined as a single step in the HPO process where one configuration is proposed and evaluated. A *repetition* refers to a complete HPO procedure, consisting of multiple iterations. We assess meta-overfitting in HPO using two metrics: the average meta-overfitting error (MOE_{avg}) and the resulting meta-overfitting error (MOE_{res}). Let $[A_\lambda]_{i,j}$ denote the hyperparameter configuration from iteration i during repetition j. Additionally, let $[A_\lambda]^*_{i,j}$ be the best-found hyperparameter configuration in an HPO process for a given repetition j when completed until iteration i. The resulting meta-overfitting error is defined as:

$$MOE_{res}(\mathcal{D}_{train}, \mathcal{D}_{val}, \mathcal{D}_{test}, i) := \frac{1}{n}\sum_{j=1}^{n} MOE([A_\lambda]^*_{i,j}, \mathcal{D}_{train}, \mathcal{D}_{val}, \mathcal{D}_{test}) \quad (3)$$

In the context of HPO applications, this corresponds to the meta-overfitting error of the configuration that the algorithm would select if stopped at that iteration, averaged over various repetitions.

Furthermore, we define the average meta-overfitting error for iteration i as the mean meta-overfitting error of all configurations suggested in that iteration across n repetitions:

$$MOE_{avg}(\mathcal{D}_{train}, \mathcal{D}_{val}, \mathcal{D}_{test}, i) := \frac{1}{n}\sum_{j=1}^{n} MOE([A_\lambda]_{i,j}, \mathcal{D}_{train}, \mathcal{D}_{val}, \mathcal{D}_{test}) \quad (4)$$

This measure focuses on the meta-overfitting of configurations selected specifically at iteration i, rather than tracking the best configuration found up to that point (as in Equation 3). If the MOE_{avg} increases across iterations of i, we found empirical evidence for adaptive overfitting, since the effect of selection-based meta-overfitting has been accounted for.

4 Experimental setup

We aim to experimentally assess the impact of both selection-based overfitting and adaptive overfitting over a wide range of datasets. Additionally, we aim to assess which factors can mitigate this effect.

Datasets: We investigate meta-overfitting using holdout evaluation on 64 datasets: 48 classification datasets from the OpenML-CC18 benchmark [4] and 16 regression datasets from the AutoML benchmark [11]. Datasets with more than 50 features and two regression datasets (due to compatibility issues) were excluded.

For experiments on 10-fold cross-validation and varying validation set sizes, we used the same four binary classification datasets, chosen for their large size (necessary for validation size experiments) and to manage computational resource constraints.

Search space: The search space consists of all algorithms and hyperparameters that can be searched over by the hyperparameter optimization algorithm. The pipeline includes the following components: two categorical value encoders, three missing value imputers, four feature selectors, eight scalers, three dimensionality reduction techniques, 17 classifiers, and 15 regressors. Additionally, most of these components are configurable with hyperparameters.

Implementation: Bayesian optimization was implemented using SMAC3 [12]. For comprehensive details on the search space, datasets, and experimental implementation, we refer to our GitHub repository.[1]

Experiments: The following three experiments were conducted:

1. **Large-scale holdout validation**: We employ a holdout set for both validation, used by the HPO algorithm to evaluate and select configurations, and test, which remains fully inaccessible to the HPO algorithm. For datasets over 3 000 instances, splits were fixed at 1 000 (train), 500 (validation), and the remainder as test data. Smaller datasets used a 40/20/40% split. This division between training, validation and test data is broadly adopted in machine learning [3].
2. **10-fold cross-validation (10CV)**: Due to computational constraints, experiments were restricted to four datasets. We performed cross-validation, generating 10 different validation scores, which were averaged to estimate validation performance. For testing, we used a single large holdout set. Rather than retraining models on the full training set to obtain test performance, we evaluated the predictions from all 10 models on the test set and averaged their individual performances.
3. **Holdout with varying validation sizes**: Validation sizes ranged from 100 to 10 000 instances, with 1 000 train instances fixed for comparability. The experiment was conducted on the same four datasets used in the 10-fold cross-validation experiment.

Each dataset was evaluated using 100 repetitions, with 250 iterations of both random search and Bayesian optimization per repetition, to assess the various forms of meta-overfitting across two HPO methods with different search characteristics. This resulted in 6.6 million pipelines trained and tested on validation and test sets. The experiments spanned roughly 13 days, utilizing 20 Intel Xeon 2.20GHz machines with two cores each.

[1] `https://github.com/ADA-research/OverfittingCASH`

5 Results and Discussion

We present our results visually using the *adult* dataset, which consists of 15 features, 48,842 instances, and a binary target variable. We extend these findings to all 64 datasets analyzed. First, we examine the impact of meta-overfitting (Section 5.1). Next, we investigate adaptive overfitting and provide empirical evidence of its occurrence (Section 5.2). Finally, we analyze how the validation set size influences meta-overfitting (Section 5.3).

5.1 Analysis of meta-overfitting in HPO

As discussed previously, meta-overfitting can be categorized into two types: (i) selection-based overfitting and (ii) adaptive overfitting. In Figure 1, we demonstrate this effect by comparing performance evaluations on both the validation and the test set, where each point represents a hyperparameter configuration suggested by an HPO algorithm. Note that generally configurations are evaluated only on the validation set, and the best-performing configuration is then assessed on the test set. However, for experimental purposes, we evaluate all configurations on both datasets. The validation set consists of 500 instances, while the independent test set contains approximately 47 000 instances. Configurations located to the right of the diagonal in the figure perform better on the validation set than on the test set and thus exhibit meta-overfitting. Because an HPO procedure typically selects the right-most configuration, which is generally (but not necessarily) on the right side of the diagonal, it is likely to produce a meta-overfitted configuration – the effect we refer to as selection-based meta-overfitting. This effect becomes apparent in the case of random search, illustrated in Figure 1 (left). Additionally, Bayesian optimization tends to concentrate its search around more overfitted regions in later iterations, as it gains more information from previous evaluations on the validation set, as illustrated in Figure 1 (right). The additional meta-overfitting introduced by this effect is referred to as adaptive overfitting. Both types of meta-overfitting are visually present across the datasets analyzed.

To better understand and quantify meta-overfitting in HPO, we analyze the hyperparameter configurations ultimately chosen during the optimization process. Specifically, at each iteration, we select the configuration that achieves the highest performance according to the validation procedure up to that point. The extent of meta-overfitting exhibited by these chosen configurations is quantified using the resulting meta-overfitting error, as defined in Equation 3.

Figure 2 (left) illustrates the average progression of accuracy scores throughout the HPO optimization process. Two key insights can be drawn from this figure. First, the difference between the best and selected configurations shows that there are configurations that outperform the selected ones but are not chosen, highlighting the potential for performance improvement if more reliable validation estimates were employed. Second, after a certain number of iterations, no significantly better configurations are identified for the test set, while the validation performance continues to improve, particularly in the case of Bayesian

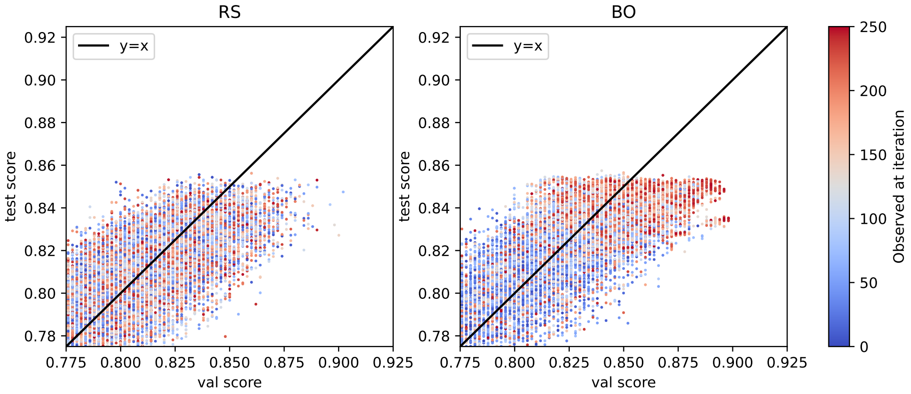

Fig. 1: Test and validation accuracy scores of highest performing configurations produced by random search (left) and Bayesian optimization (right) on the *adult* dataset. The color indicates in what iteration they were suggested.

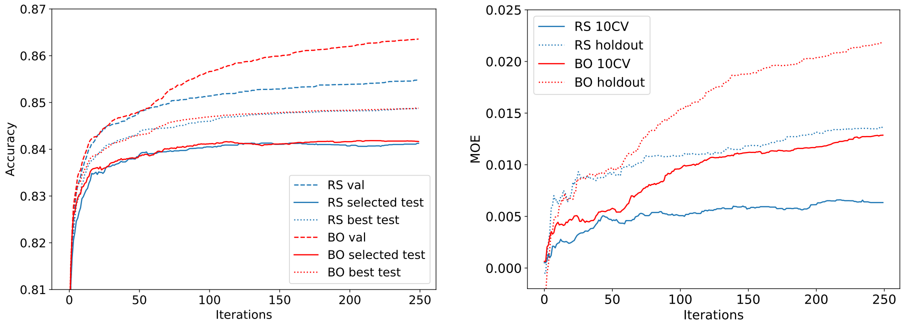

Fig. 2: **Left:** Average accuracy of the selected configuration from Bayesian optimization and random search on the holdout validation and test sets, along with the test set accuracy of the best configuration on the test set, using the *adult* dataset. **Right:** Resulting MOE for holdout validation and 10CV from Bayesian optimization and random search on the *adult* dataset.

optimization. This suggests that the selected configurations increasingly overfit the validation set, with no corresponding improvement in generalization performance beyond a certain point in the optimization process. Figure 2 (right) illustrates the meta-overfitting error of the configurations selected during each iteration, averaged across 100 repetitions. These findings highlight the practical implications of meta-overfitting. Specifically, after 250 iterations of Bayesian optimization, the selected configuration is expected to exhibit an average overfitting error of 1.3% when using 10-fold cross-validation as the internal validation procedure. This error increases to 2.2% when holdout validation is used.

Across datasets: These results are generalizable to other datasets. On the four binary classification datasets we used to investigate 10-fold cross-validation, the final configurations identified by Bayesian optimization exhibited meta-overfitting errors ranging from 1.0% to 3.7% in accuracy, with an average of 1.8%. These findings raise important questions about the use of 10-fold cross-validation as a benchmarking practice, especially when the same fixed cross-validation strategy is applied for both internal validation and final assessment. In such cases, a portion of the observed performance improvement may result from overfitting the validation procedure rather than genuinely enhancing the generalizability of the hyperparameter optimization process.

Furthermore, on 29 binary classification datasets using holdout validation, we observe an average meta-overfitting error of 3.3% in accuracy for the final selected configuration identified by Bayesian optimization, with a median of 2.3%. Similarly, for 19 multiclass classification datasets, the average meta-overfitting error is 2.9% accuracy, with a median of 2.3%. These findings are particularly relevant to common practices in widely used HPO frameworks, such as Auto-sklearn [9], which employs internal holdout validation by default. They highlight the importance of addressing meta-overfitting when designing robust and reliable machine learning pipelines with HPO.

5.2 Adaptive overfitting across datasets

In this section, we present evidence of adaptive overfitting in Bayesian optimization applied to the CASH problem. Adaptive overfitting occurs when hyperparameter configurations become increasingly tailored to the validation procedure. This arises because Bayesian optimization incorporates growing amounts of information from prior evaluations into its surrogate model, leading to configurations that are progressively fine-tuned to the validation set. Therefore, adaptive overfitting can be observed as a linear relationship between the degree of meta-overfitting in a hyperparameter configuration and the iteration in which that configuration is suggested by Bayesian optimization. This means that hyperparameter configurations proposed in later iterations tend to exhibit greater levels of meta-overfitting, on average. In contrast, for random search, the iterations are independent of one another. As a result, there is no systematic pattern of meta-overfitting across iterations, and we would expect the average meta-overfitting for any given iteration to be approximately zero.

Figure 3 (left) illustrates the linear relationship between the average meta-overfitting error (defined in Equation 4) of a hyperparameter configuration and the iteration in which it was suggested on the *adult* dataset. This relationship is evident regardless of whether holdout evaluation or 10-fold cross-validation is employed, providing strong evidence of adaptive overfitting in both validation procedures. This implies that configurations suggested in the early iterations of Bayesian optimization are generally not overfitted, but configurations proposed in the final iterations tend to exhibit an average overfitting of approximately 0.5% accuracy for 10-fold cross-validation and 0.8% accuracy for holdout validation.

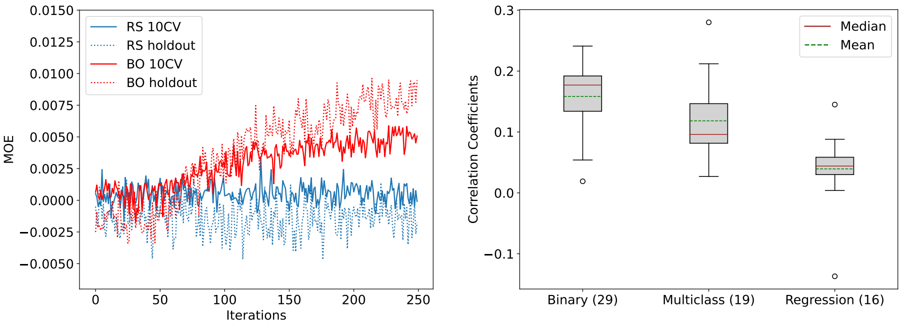

Fig. 3: **Left:** Average MOE of different HPO algorithms using holdout validation and 10CV on the *adult* dataset, showing a linear relationship for Bayesian optimization. **Right:** Box-plots of Spearman's coefficients of all 64 datasets considered using holdout validation, calculated using 100 repetitions per dataset.

Extrapolating these observations to the full range of 64 datasets, we quantify adaptive overfitting using Spearman's rank-order correlation coefficient, which is suited for assessing relationships between variables that are not all normally distributed, such as the iterations uniformly spanning from 1 to 250. A higher coefficient indicates a stronger linear relationship between iterations and the average meta-overfitting error, signifying a more pronounced effect of adaptive overfitting. To account for testing multiple hypotheses, we apply the highly conservative Bonferroni correction within dataset types (binary, multiclass, and regression). This adjustment raises the threshold for statistical significance, ensuring that observed linear relationships are unlikely to be false positives. Using this approach, we assess the linear relationships between the average meta-overfitting error of a hyperparameter configuration and the iteration in which it was suggested. When using the holdout method, we find statistically significant results for 25 out of 29 binary classification datasets, 16 out of 19 multiclass classification datasets, and 3 out of 16 regression datasets. For 10-fold cross-validation, all 4 binary classification datasets show significant adaptive overfitting. Figure 3 (right) shows the distribution of Spearman's correlation coefficients of all datasets in each problem type for the holdout method. We statistically compare the coefficients of binary and multiclass datasets. Shapiro-Wilk tests confirm normality for both samples, and Levene's test verifies homogeneity of variances, allowing the use of an independent samples t-test. Supported by this test, we conclude that multiclass classification datasets exhibit significantly less adaptive overfitting compared to binary classification datasets ($t(46) = 2.3,\ p = .03$), confirming prior work [8]. However, further research is needed to substantiate this finding and explore the influence of the number of classes on adaptive overfitting. Additionally, our results indicate that regression datasets are highly resilient to adaptive overfitting.

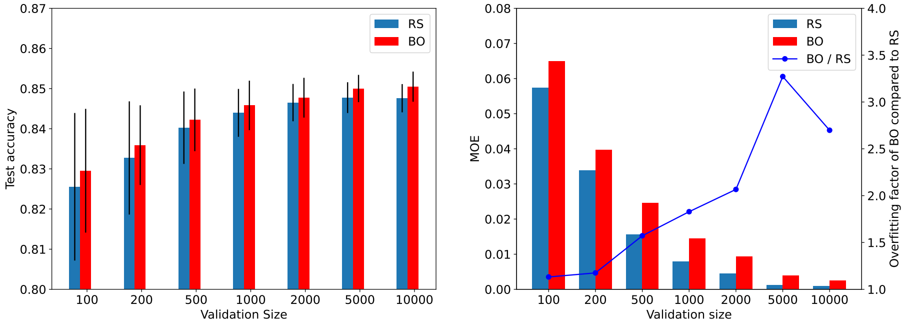

Fig. 4: **Left:** Effect of the validation set size on the stability and generalization of HPO for the *adult* dataset. **Right:** Meta-overfitting error for random search and Bayesian optimization across validation set sizes for the *adult* dataset. The BO/RS line illustrates the factor by which the selected configurations of Bayesian optimization exhibit greater resulting MOE compared to random search.

5.3 Effects of different validation set sizes

Finally, we study the effect of validation set size on meta-overfitting and the generalizability of HPO algorithms using four binary classification datasets, as motivated in Section 4. Figure 4 (left) illustrates how larger validation sets improve stability and generalization performance. Figure 4 (right) highlights differences in meta-overfitting for random search and Bayesian optimization across validation sizes. For small validation sets, the selection-based overfitting in random search is comparable to the meta-overfitting in Bayesian optimization, suggesting a minimal role for adaptive overfitting. As validation sizes increase, the meta-overfitting in random search diminishes, as expected. However, for Bayesian optimization, meta-overfitting maintains a factor for increasing validation sizes. For instance, configurations selected by Bayesian optimization are 1.1 times more overfitted than those of random search at 100-instance validation sets, rising to 3.3 times for 5 000 instances. Similar results are observed across all four datasets. These findings suggest adaptive overfitting is still persistent with large validation sets, warranting caution even when validation sets contain thousands of instances. However, despite the effects of meta-overfitting, it remains clear that Bayesian optimization performs better than random search.

6 Conclusions

This study explores meta-overfitting in HPO for the CASH problem through an empirical analysis of 64 datasets covering classification and regression. We formalized meta-overfitting and decomposed it into two primary sources: (i) selection-based and (ii) adaptive overfitting. We show that both random search and Bayesian optimization exhibit meta-overfitting. Our results empirically confirm the logical fact that in random search, meta-overfitting arises solely from selection-based overfitting, whereas in Bayesian optimization, there is also evidence of adaptive overfitting. Using a single holdout validation set, Bayesian optimization produced configurations with an average meta-overfitting error of 3.3% accuracy on binary classification datasets and 2.9% accuracy on multiclass datasets. While 10-fold cross-validation reduced this error, configurations still showed an average meta-overfitting error of 1.8% accuracy on four binary classification datasets. Furthermore, we identified statistically significant adaptive overfitting in Bayesian optimization for HPO in the CASH problem, an effect not previously measured in this context to the best of our knowledge. Significant adaptive overfitting was observed in 25 of the 29 binary classification datasets and 16 of the 19 multiclass classification datasets, with multiclass datasets showing significantly less adaptive overfitting, consistent with prior theory [8]. Regression datasets, however, appeared notably resistant to this effect. Finally, we investigated how validation set size affects meta-overfitting. In random search, the meta-overfitting error of selected hyperparameter configurations diminished with larger validation sets. However, in Bayesian optimization, the meta-overfitting error remained more robust, suggesting that adaptive overfitting can be a concern even with validation sets containing thousands of instances.

Recommendations: We advise machine learning practitioners to be cautious about potentially overfitted results being produced by HPO algorithms, particularly when validation results are directly used to guide the suggestion of new hyperparameter configurations. Similarly, we recommend avoiding fixed 10-fold cross-validation when benchmarking HPO algorithms and instead resampling the training and validation splits, either at each iteration, following [15], or at least during the evaluation of the selected hyperparameter configuration.

Future work: Future research could further explore meta-overfitting, especially in relation to HPO techniques like ensembling and meta-learning, commonly used in state-of-the-art HPO frameworks. Investigating adaptive overfitting in HPO and applying strategies from adaptive data analysis to mitigate its effects also presents a compelling avenue for future studies.

Disclosure-of-interests: The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Arora, S., Zhang, Y.: Rip van Winkle's razor: A simple estimate of overfit to test data. Computing Research Repository **abs/2102.13189** (2021)
2. Baratchi, M., Wang, C., Limmer, S., van Rijn, J.N., Hoos, H.H., Bäck, T., Olhofer, M.: Automated machine learning: Past, present and future. Artificial Intelligence Review **57**(5), 122 (2024)

3. Bischl, B., Binder, M., Lang, M., Pielok, T., Richter, J., Coors, S., Thomas, J., Ullmann, T., Becker, M., Boulesteix, A., Deng, D., Lindauer, M.: Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery **13**(2) (2023)
4. Bischl, B., Casalicchio, G., Feurer, M., Gijsbers, P., Hutter, F., Lang, M., Mantovani, R.G., van Rijn, J.N., Vanschoren, J.: OpenML benchmarking suites. In: Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks (2021)
5. Cawley, G.C., Talbot, N.L.C.: On over-fitting in model selection and subsequent selection bias in performance evaluation. Journal of Machine Learning Research **11**, 2079–2107 (2010)
6. Dwork, C., Feldman, V., Hardt, M., Pitassi, T., Reingold, O., Roth, A.L.: Preserving statistical validity in adaptive data analysis. In: Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015. pp. 117–126. ACM (2015)
7. Fabris, F., Freitas, A.A.: Analysing the overfit of the auto-sklearn automated machine learning tool. In: Machine Learning, Optimization, and Data Science - 5th International Conference, LOD 2019. Lecture Notes in Computer Science, vol. 11943, pp. 508–520. Springer (2019)
8. Feldman, V., Frostig, R., Hardt, M.: The advantages of multiple classes for reducing overfitting from test set reuse. In: Proceedings of the 36th International Conference on Machine Learning, ICML 2019. Proceedings of Machine Learning Research, vol. 97, pp. 1892–1900. PMLR (2019)
9. Feurer, M., Klein, A., Eggensperger, K., Springenberg, J.T., Blum, M., Hutter, F.: Efficient and robust automated machine learning. In: Advances in Neural Information Processing Systems 28, NeurIPS 2015. pp. 2962–2970 (2015)
10. Freedman, D.A.: A note on screening regression equations. The American Statistician **37**(2), 152–155 (1983)
11. Gijsbers, P., Bueno, M.L.P., Coors, S., LeDell, E., Poirier, S., Thomas, J., Bischl, B., Vanschoren, J.: AMLB: an AutoML benchmark. Journal of Machine Learning Research **25**, 101:1–101:65 (2024)
12. Lindauer, M., Eggensperger, K., Feurer, M., Biedenkapp, A., Deng, D., Benjamins, C., Ruhkopf, T., Sass, R., Hutter, F.: SMAC3: A versatile Bayesian optimization package for hyperparameter optimization. Journal of Machine Learning Research **23**, 54:1–54:9 (2022)
13. Lévesque, J.C.: Bayesian hyperparameter optimization: Overfitting, ensembles and conditional spaces. Ph.D. thesis, Université Laval (2018), `http://hdl.handle.net/20.500.11794/28364`
14. Makarova, A., Shen, H., Perrone, V., Klein, A., Faddoul, J.B., Krause, A., Seeger, M.W., Archambeau, C.: Overfitting in Bayesian optimization: An empirical study and early-stopping solution. Computing Research Repository **abs/2104.08166** (2021)
15. Nagler, T., Schneider, L., Bischl, B., Feurer, M.: Reshuffling resampling splits can improve generalization of hyperparameter optimization. In: Advances in Neural Information Processing Systems 38, NeurIPS 2024 (2024)
16. Ng, A.Y.: Preventing "overfitting" of cross-validation data. In: Proceedings of the Fourteenth International Conference on Machine Learning, ICML 1997. pp. 245–253. Morgan Kaufmann (1997)
17. Quinlan, J.R., Cameron-Jones, R.M.: Oversearching and layered search in empirical learning. In: Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, IJCAI 95. pp. 1019–1024. Morgan Kaufmann (1995)

18. Rao, R.B., Fung, G.: On the dangers of cross-validation. An experimental evaluation. In: Proceedings of the SIAM International Conference on Data Mining, SDM 2008. pp. 588–596. SIAM (2008)
19. Recht, B., Roelofs, R., Schmidt, L., Shankar, V.: Do CIFAR-10 classifiers generalize to CIFAR-10? Computing Research Repository **abs/1806.00451** (2018)
20. Recht, B., Roelofs, R., Schmidt, L., Shankar, V.: Do ImageNet classifiers generalize to ImageNet? In: Proceedings of the 36th International Conference on Machine Learning, ICML 2019. Proceedings of Machine Learning Research, vol. 97, pp. 5389–5400. PMLR (2019)
21. Reunanen, J.: Overfitting in making comparisons between variable selection methods. Journal of Machine Learning Research **3**, 1371–1382 (2003)
22. Roelofs, R., Shankar, V., Recht, B., Fridovich-Keil, S., Hardt, M., Miller, J., Schmidt, L.: A meta-analysis of overfitting in machine learning. In: Advances in Neural Information Processing Systems 32, NeurIPS 2019. pp. 9175–9185 (2019)
23. Thornton, C., Hutter, F., Hoos, H.H., Leyton-Brown, K.: Auto-WEKA: Combined selection and hyperparameter optimization of classification algorithms. In: The 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2013. pp. 847–855. ACM (2013)
24. Yadav, C., Bottou, L.: Cold case: The lost MNIST digits. In: Advances in Neural Information Processing Systems 32, NeurIPS 2019. pp. 13443–13452 (2019)