Assessing the generalizability of a performance predictive model

Nikolikj, A.; Cenikj, G.; Ispirova, G.; Vermetten, D.L.; Lang, D.R.; Engelbrecht, P.A.; ... ; Eftimov, T.

Assessing the Generalizability of a Performance Predictive Model

Ana Nikolikj
Jožef Stefan Institute &
Jožef Stefan International
Postgraduate School
Ljubljana, Slovenia

Gjorgjina Cenikj
Jožef Stefan Institute &
Jožef Stefan International
Postgraduate School
Ljubljana, Slovenia

Gordana Ispirova
Jožef Stefan Institute
Ljubljana, Slovenia

Diederick Vermetten
LIACS, Leiden University
The Netherlands

Ryan Dieter Lang
Stellenbosch University
South Africa

Andries Petrus Engelbrecht*
Stellenbosch University
South Africa

Carola Doerr
Sorbonne Université, CNRS, LIP
Paris, France

Peter Korošec
Jožef Stefan Institute
Slovenia

Tome Eftimov
Jožef Stefan Institute
Slovenia

ABSTRACT

A key component of automated algorithm selection and configuration, which in most cases are performed using supervised machine learning (ML) methods is a good-performing predictive model. The predictive model uses the feature representation of a set of problem instances as input data and predicts the algorithm performance achieved on them. Common machine learning models struggle to make predictions for instances with feature representations not covered by the training data, resulting in poor generalization to unseen problems. In this study, we propose a workflow to estimate the generalizability of a predictive model for algorithm performance, trained on one benchmark suite to another. The workflow has been tested by training predictive models across benchmark suites and the results show that generalizability patterns in the landscape feature space are reflected in the performance space.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning**; **Learning latent representations**; **Supervised learning**; • **Theory of computation** → **Design and analysis of algorithms**.

KEYWORDS

meta-learning, single-objective optimization, generalization

ACM Reference Format:
Ana Nikolikj, Gjorgjina Cenikj, Gordana Ispirova, Diederick Vermetten, Ryan Dieter Lang, Andries Petrus Engelbrecht, Carola Doerr, Peter Korošec, and Tome Eftimov. 2023. Assessing the Generalizability of a Performance Predictive Model. In *Genetic and Evolutionary Computation Conference Companion (GECCO '23 Companion), July 15–19, 2023, Lisbon, Portugal*. ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3583133.3590617

*Also with Gulf University for Science and Technology.

GECCO '23 Companion, July 15–19, 2023, Lisbon, Portugal

ACM ISBN 979-8-4007-0120-7/23/07.
https://doi.org/10.1145/3583133.3590617

1 INTRODUCTION

Automated algorithm configuration [1, 12] and selection [7, 8] are among the most researched topics in evolutionary computation. These systems often use predictive machine learning (ML) models which take the feature representation of problem instances as input and predict the performance of an algorithm instance on a problem instance. However, one of the main drawbacks presented in these learning tasks is the low generalizability of the predictive models. The models fail to provide accurate predictions for problem instances whose feature representation is underrepresented or not presented in the training data.

Recent studies [9, 14] show that poor predictive results have been obtained when an ML model for performance prediction is trained on the problem instances from one benchmark suite and then evaluated on problem instances from another benchmark suite. Škvorc et al. [14] present results when a random forest (RF) model trained on the BBOB (i.e., COCO) [5] benchmark suite provides poor results when tested on artificially generated problem instances [15] and vice-versa. Kostovska et al. [9] show that an automated algorithm selector which is based on performance prediction models trained on the BBOB benchmark suite, cannot generalize to problem instances from the Nevergrad's YABBOB [2] benchmark suite.

Our contribution: We propose a workflow to estimate the generalizability of a predictive model from one benchmark suite to another. Problem instances are grouped into clusters based on their features and further use the distribution of each benchmark suite represented as the number of problem instances across the clusters as a meta-representation for each benchmark suite. Similarity between benchmark suites is calculated using this meta-representation. This similarity can indicate if a predictive model can be generalized across benchmark suites. We evaluated the workflow by training predictive models and found that generalizability patterns in the feature space were also present in the performance space.

Data and code availability: The data and the code involved in this study are available at [11].

2 ASSESSING GENERALIZABILITY WORKFLOW

Let us assume that we have m benchmark suites. Each benchmark suite can consist of a different number of problem instances. One out of m benchmark suites is selected to train the supervised ML predictive model ($\mathcal{M}$) and the remaining $m-1$ benchmark suites are used to test the model. To assess the generalizability of the model $\mathcal{M}$ to the different benchmark suites used for testing, we propose the following workflow:

Defining a unified meta-representation on a problem instance level – represent the problem instances from all benchmark suites using the same n meta-features that describe the landscape properties of the problem instances. With this, all problem instances (i.e., the ones selected for training and the remaining ones used for testing) are projected into the same n-dimensional vector space.

Defining a coverage matrix – based on their meta-representation cluster the problem instances from all benchmark suites into k clusters. Next, for each benchmark suite calculate the percentage of problem instances that belong to each cluster. With this, we define k-vector meta-representation on a benchmark suite level that represents the distribution of the benchmark suite across different clusters (i.e different regions in the problem space).

Define the similarity between two benchmark suites – the similarity between a pair of benchmark suites is calculated using their coverage matrix-based meta-representation. The approach uses cosine similarity as a similarity measure [13].

High benchmark suite similarity suggests accurate predictions by a model trained on one suite for the other suite. The low similarity suggests poor generalization and coverage of different problem landscape space regions.

3 EXPERIMENTAL DESIGN

The evaluation of the workflow has been performed in two experiments. More details about them are provided below.

Benchmark suites: In the *first experiment*, we involve the benchmark suite data available from a previous study [10], where the BBOB, CEC 2013, CEC 2014, CEC 2015, and CEC 2017 benchmark suites are used. The CEC benchmark suites change annually, with some overlap in problem instances across different suites, but the definition of the same problem instance differs each year, which may result in varying properties of the benchmarks even with the same problem instance definition. In the experiments, the problem dimension is set to $D = 10$. In the *second experiment*, we select benchmark problem instances that are affine recombinations of pairs of BBOB problem instances, where 9,936 new problem instances are generated for several dimensions [4]. Next, we use the SELECTOR approach [3] to select diverse benchmark problem instances in $D = 5$ based on their 14 landscape features. The benchmark problem instances have been transformed into a graph based on the cosine similarity using their landscape features. Next, the Maximal Independent Set method has been run five times independently to select five benchmark suites (BS1, BS2, BS3, BS4, BS5) that contain around 110 problem instances with minimal overlap. SELECTOR guarantees that the distribution of the problem instances in the five independent selections is similar.

Performance data: In the *first experiment*, performance data for the Covariance Matrix Adaption Evolutionary Strategy (CMA-ES) [6] has been used. The algorithm stops after either reaching 100,000 function evaluations or finding a solution within 10^{-8} of the global optimum. As a target in the regression models, we used the obtained solutions' precision (i.e., the error to the global optimum). In the *second experiment*, we use the performance data of the Diagonal CMA-ES [6]. Here, we also retrieve the precision after a budget of 10,000 function evaluations as a target variable for the regression models. For both experiments, we use their default hyperparameters implementation from the Nevergrad library [2].

Exploratory landscape analysis: For the first experiment, to describe the landscape properties of each problem instance, 64 publicly available ELA features from a previous study [10] are used. In the second experiment, 14 ELA features are used, also available from a previous study [4].

Clustering: The K-Means algorithm clusters problem instances from benchmark suites in both experiments. The Silhouette score is used to estimate cluster number in the *first experiment* and the elbow method with the distortion metric is used in the *second experiment*. We tested different measures to estimate the number of clusters, just to check the sensitivity of the approach using different measures. ELA features are normalized before clustering, and the *scikit-learn* package in Python is used for its implementation.

Predictive models: Random Forest (RF) models (from the *scikit-learn package in Python* with default hyper-parameters) are trained on each benchmark suite, evaluated on remaining suites, and reported using median absolute error (MDAE). Results are analyzed to determine if a pattern from the coverage matrix is also present in automated algorithm performance prediction model performance.

4 RESULTS AND DISCUSSION

Here, the results for both experiments are presented in more detail.

First experiment. The optimal number of clusters has been estimated to 13. Table 1 presents the coverage matrix, where each cell in the table shows the percentage of the total number of instances from the benchmark suite presented in the row, that belongs to each cluster presented in the column. Each row then is used as a meta-representation for each benchmark suite. The results show that the BBOB benchmark suite is the most widely spread in the feature space (i.e., landscape space) as its instances are distributed across nine clusters out of 13, compared to the CEC benchmark suites which condensed to a smaller number of clusters. There are four clusters that consist only of BBOB problem instances. Also, it is visible that the distribution of the CEC 2014 and CEC 2017 problem instances across the clusters is very similar.

Table 1: Coverage matrix calculated with 13 clusters.

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13
BBOB	0.04	0.47	0.08	0.00	0.03	0.08	0.06	0.00	0.00	0.00	0.18	0.04	0.02
CEC2013	0.00	0.00	0.00	0.40	0.04	0.28	0.04	0.08	0.08	0.00	0.00	0.08	0.00
CEC2014	0.00	0.00	0.03	0.30	0.10	0.17	0.10	0.27	0.00	0.03	0.00	0.00	0.00
CEC2015	0.00	0.00	0.00	0.13	0.00	0.13	0.27	0.33	0.00	0.07	0.00	0.07	0.00
CEC2017	0.00	0.00	0.00	0.31	0.07	0.07	0.21	0.24	0.00	0.03	0.00	0.07	0.00

Figure 1 presents a heatmap of the cosine similarity between the benchmark suite meta-representations, and hierarchical clustering dendrogram of the benchmark suites' meta-representations. Based on the cosine similarity between the benchmark suites, the similarity matrix is reorganized such that the more similar benchmark suites are placed together in the dendrogram. The colors in

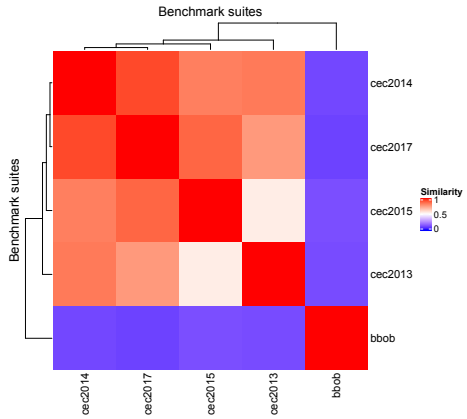

Figure 1: Heatmap of the cosine similarity between benchmark suites representations generated based on 13 clusters.

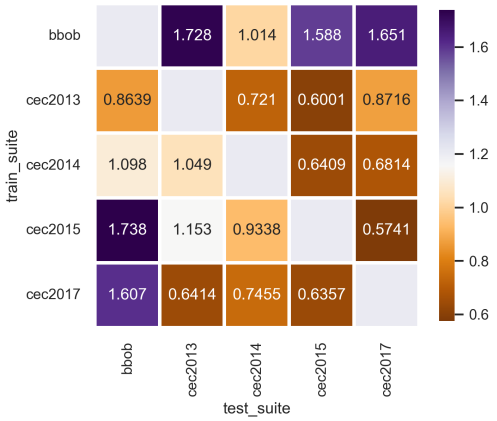

Figure 2: Heatmap showing the MDAEs of an RF model for the performance of CMA. Rows indicate the train benchmark suite and columns indicate the benchmark suite the model was evaluated on.

the plot represent the cosine similarity. The figure shows that all CEC benchmark suites have high similarity. The pairwise cosine similarities between all pairs of CEC benchmark suites are greater than 0.5. This is as expected since the CEC suites have had little modifications through the years. However, comparing them to the BBOB benchmark suite it seems that there is a big difference in how their problem instances are distributed in the feature space, also visible earlier from the coverage matrix. Looking into the clustering result, it follows that CEC 2014 and CEC 2017 are the most similar ones, further both of them are close to CEC2015, and then to CEC 2013, while all of them are placed on another side of the dendrogram compared to the BBOB. This result further points out that we can expect a predictive model trained on CEC 2014 to have the best results when it will be evaluated on CEC 2017 and vice-versa. Further, a model trained on CEC 2014 or CEC 2017 is expected to have good prediction results when it will be evaluated on CEC 2013 and CEC 2015. Also, models trained on CEC 2013 or CEC 2015 will have to generalize the prediction results across CEC benchmark suites. The dissimilarity of CEC benchmark suites with the BBOB benchmark suite indicates that we do not have a guarantee that the model will generalize across them.

To investigate if the similarity patterns in the landscape feature space will also be present in the performance space as model generalizability patterns, we present the evaluation results of an algorithm performance predictive model, when the model is trained on one benchmark suite and evaluated on the remaining ones. The heatmap in Figure 2 presents the RF model errors for the performance prediction of the CMA-ES. Each cell presents the median absolute error of the RF model, trained on the benchmark suite presented in the row, and evaluated on the benchmark suite presented in the column. The results show that a predictive model trained on BBOB produces larger errors across all CEC benchmark suites. A model trained on CEC 2017 provides smaller errors when it is evaluated on CEC 2013, 2014, and 2015, and ends up with a larger error for BBOB. When CEC 2013 is used to train the model, similar errors are obtained across all benchmark suites. A similar effect occurs when CEC 2015 is used for training, good errors are achieved on CEC 2014 and CEC 2017, and the error increases for CEC 2013, ending up with a larger error for BBOB.

The results indicate that a similar distribution of the benchmark suites over the landscape feature space leads to similar model errors on the suites. This study does not guarantee that the training and testing error will be good but it guarantees that they will be in similar ranges. We are not dealing with the quality of the benchmark suites but only comparing the landscape feature distribution of the benchmark suites. However, it is not possible to establish a complete generalizability mapping function between the landscape feature space and the performance space, since these algorithms are stochastic in nature and they all have different behavior.

Second experiment: Using the elbow method of the distortion metric curve, the optimal number of clusters is determined to be four. Table 2 presents the distribution of the five artificial benchmark suites across the four clusters. From the table, it follows that all benchmark suites have similar distribution across the clusters, thus following the same distribution across the feature space. In addition, for each pair of benchmark suites we analyzed the cosine similarity between their meta-representations based on the coverage matrix. The obtained cosine similarities of the meta-representations exceeded 0.98 for all pairs of benchmark suites. This result indicates that a model trained on any one of these benchmark suites should be easy to generalize to the remaining benchmark suites.

Table 2: Coverage matrix calculated with four clusters.

	C1	C2	C3	C4
BS1	0.14	0.21	0.32	0.32
BS2	0.16	0.26	0.33	0.25
BS3	0.16	0.27	0.32	0.25
BS4	0.20	0.22	0.31	0.27
BS5	0.15	0.26	0.34	0.25

The evaluation results (MDAE) of the predictive models trained on each of the five artificial benchmark suites and evaluated on the remaining four for the diagonal CMA-ES are presented in Figure 3. The rows indicate the benchmark suite on which the model has been trained and the columns indicate the benchmark suite on which the model has been evaluated.

The results show that the patterns that are visible in the landscape feature space are also reflected in the model performance space. Models trained on all five benchmark suites separately are

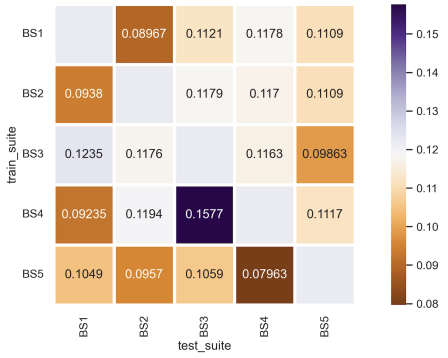

Figure 3: Heatmap showing the MDAEs of an RF model for the performance of diagonal CMA-ES. Rows indicate the train benchmark suite and columns indicate the benchmark suite the model was evaluated on.

generalizable or have similar errors on all the remaining benchmarking suites. We also need to point out here that the training MDAE is smaller than all the MDAE obtained on the test benchmark suites, which is expected in a machine learning setup. However, the difference between the training and testing errors is not practically significant (e.g., training MDAE is 0.06 for BS1, 0.04 for BS2, 0.08 for BS3, 0.03 for BS4, and 0.04 for BS5 for diagonal CMA-ES).

To show that different distributions in the feature landscape space lead to worse model performance,we assessed all instances in the fourth cluster. This cluster, referred to as BS6, includes samples from all five artificially generated benchmark suites and was randomly selected to ensure a different feature landscape distribution than the other five suites. The BS6 instances cover only one region of the feature landscape and do not include samples from the three other regions present in the clustering result that are part of the other five suites. Table 3 presents RF errors when the model is trained on BS6 and evaluated for automated algorithm performance prediction on the other five benchmark suites. It is obvious that the error models are worse (compared with the errors presented in Figure 3. The results prove that different feature landscape distribution decreases the generalization of a predictive model.

Table 3: RF errors when the model is trained on BS6 and evaluated on the other five benchmark suites.

	BS1	BS2	BS3	BS4	BS5
DE	0.185922	0.182945	0.183876	0.185423	0.202225
RSPSO	0.479039	0.492742	0.474655	0.506378	0.517526
diag CMA-ES	0.240901	0.240292	0.237267	0.25774	0.240292

5 CONCLUSION

Our study proposes a workflow for estimating the generalizability of performance predictive models in automated algorithm selection and configuration. The workflow involves converting problem instances into a common meta-representation and clustering them to create a benchmark suite meta-representation. The similarity between benchmark suites is then calculated to indicate generalizability between models. Two experiments were conducted, one with commonly used benchmark suites and the other with artificially generated suites. Results show that generalizability patterns in the feature landscape space also exist in the performance space, assisting in predicting model performance on new instances. However, the workflow is dependent on the quality of the feature representation and may be affected by different performances of the algorithm with similar feature representations. Future work will test the workflow with different feature representations and families of supervised machine learning methods.

ACKNOWLEDGMENTS

The authors acknowledge the support of the Slovenian Research Agency through program grant P2-0098, project grants N2-0239 to TE and J2-4460 to PK, young researcher grant No. PR-12393 to GC, and a bilateral project between Slovenia and France grant No. BI-FR/23-24-PROTEUS-001 (PR-12040). Our work is also supported by ANR-22-ERCS-0003-01 project VARIATION. The authors also acknowledge the Centre for High Performance Computing (CHPC), South Africa, for providing computational resources to this research project.

REFERENCES

[1] Nacim Belkhir, Johann Dréo, Pierre Savéant, and Marc Schoenauer. 2017. Per instance algorithm configuration of CMA-ES with limited budget. In *Proc. of Genetic and Evolutionary Computation (GECCO'17)*. ACM, 681–688. https://doi.org/10.1145/3071178.3071343

[2] Pauline Bennet, Carola Doerr, Antoine Moreau, Jeremy Rapin, Fabien Teytaud, and Olivier Teytaud. 2021. Nevergrad: black-box optimization platform. *ACM SIGEVOlution* 14, 1 (2021), 8–15.

[3] Gjorgjina Cenikj, Ryan Dieter Lang, Andries Petrus Engelbrecht, Carola Doerr, Peter Korošec, and Tome Eftimov. 2022. SELECTOR: Selecting a Representative Benchmark Suite for Reproducible Statistical Comparison. In *Proceedings of the Genetic and Evolutionary Computation Conference* (Boston, Massachusetts) *(GECCO '22)*. Association for Computing Machinery, New York, NY, USA, 620–629. https://doi.org/10.1145/3512290.3528809

[4] Konstantin Dietrich and Olaf Mersmann. 2022. Increasing the Diversity of Benchmark Function Sets Through Affine Recombination. In *Parallel Problem Solving from Nature–PPSN XVII: 17th International Conference, PPSN 2022, Dortmund, Germany, September 10–14, 2022, Proceedings, Part I*. Springer, 590–602.

[5] Nikolaus Hansen, Anne Auger, Raymond Ros, Olaf Mersmann, Tea Tušar, and Dimo Brockhoff. 2021. COCO: A platform for comparing continuous optimizers in a black-box setting. *Optimization Methods and Software* 36, 1 (2021), 114–144.

[6] Nikolaus Hansen and Andreas Ostermeier. 2001. Completely Derandomized Self-Adaptation in Evolution Strategies. *Evolutionary Computation* 9, 2 (2001), 159–195. https://doi.org/10.1162/106365601750190398

[7] Anja Jankovic and Carola Doerr. 2020. Landscape-aware fixed-budget performance regression and algorithm selection for modular CMA-ES variants. In *GECCO*. ACM, 841–849.

[8] Pascal Kerschke and Heike Trautmann. 2019. Automated algorithm selection on continuous black-box problems by combining exploratory landscape analysis and machine learning. *Evolutionary computation* 27, 1 (2019), 99–127.

[9] Ana Kostovska, Anja Jankovic, Diederick Vermetten, Jacob de Nobel, Hao Wang, Tome Eftimov, and Carola Doerr. 2022. Per-run algorithm selection with warm-starting using trajectory-based features. In *Parallel Problem Solving from Nature–PPSN XVII: 17th International Conference, PPSN 2022, Dortmund, Germany, September 10–14, 2022, Proceedings, Part I*. Springer, 46–60.

[10] Ryan Dieter Lang and Andries Petrus Engelbrecht. 2021. An Exploratory Landscape Analysis-Based Benchmark Suite. *Algorithms* 14, 3 (2021), 78.

[11] Ana Nikolikj. 2023. *Prediction Model Generalizability.* https://github.com/anikolik/assessing-generalizability-of-prediction-models

[12] Raphael Patrick Prager, Heike Trautmann, Hao Wang, Thomas HW Bäck, and Pascal Kerschke. 2020. Per-instance configuration of the modularized CMA-ES by means of classifier chains and exploratory landscape analysis. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 996–1003.

[13] Amit Singhal et al. 2001. Modern information retrieval: A brief overview. *IEEE Data Eng. Bull.* 24, 4 (2001), 35–43.

[14] Urban Škvorc, Tome Eftimov, and Peter Korošec. 2022. Transfer Learning Analysis of Multi-Class Classification for Landscape-Aware Algorithm Selection. *Mathematics* 10, 3 (2022), 432.

[15] Ye Tian, Shichen Peng, Xingyi Zhang, Tobias Rodemann, Kay Chen Tan, and Yaochu Jin. 2020. A recommender system for metaheuristic algorithms for continuous optimization based on deep recurrent neural networks. *IEEE transactions on artificial intelligence* 1, 1 (2020), 5–18.