OpenML-CTR23 – A curated tabular regression benchmarking suite

Sebastian Fischer[1, 2] **Liana Harutyunyan** **Matthias Feurer**[1, 2] **Bernd Bischl**[1, 2]

[1]LMU Munich
[2]Munich Center for Machine Learning (MCML)

Abstract Benchmark experiments are one of the cornerstones of modern machine learning research. An essential part in the design of such experiments is the selection of datasets. We present the **OpenML** **C**urated **T**abular **R**egression benchmarking suite 2023 (OpenML-CTR23). It is available on OpenML and comprises 35 regression problems that have been selected according to a set of strict criteria. We compare its design with existing regression benchmark suites and also challenge some of the dataset choices of previous efforts. As a first experiment, we compare five machine learning methods of varying complexity on the OpenML-CTR23.

1 Introduction

Machine learning algorithms and their respective implementations should be studied not only through the lens of formal analysis, but also through proper empirical evaluation. Very often, specific details in their construction (which we abstract away in mathematical derivations) influence performance results considerably, and many real-world datasets do not fully satisfy the assumptions we make about data in formal analysis. For this reason, benchmark experiments are an integral part of modern machine learning research. To perform them effectively, researchers need access to a diverse collection of datasets.

In this paper, we present the **OpenML** **C**urated **T**abular **R**egression benchmarking suite 2023 (OpenML-CTR23), a collection of 35 regression problems that meet a large number of quality criteria. We follow many of the design choices of the OpenML-CC18 (Bischl et al., 2021), which is the first benchmarking suite for classification algorithms that was created using rigorous inclusion criteria, and refine them for regression. We also evaluate five (non-deep) machine learning methods of varying complexity on the benchmark suite. These are XGBoost, a Random Forest, a Generalized Additive Model (GAM), a Ridge Regression and a Regression Tree.

First, we discuss related work. Then, we will outline the benchmark suite, its design criteria, and compare it to existing work. In the next section, we describe the experimental results. Finally, we will discuss the broader impact and limitations of our work.

2 Background and Related Work

OpenML (Vanschoren et al., 2014) is a platform for collaborative research in machine learning. As part of this it hosts thousands of easily accessible datasets in a standardized format. It also supports the creation of machine learning tasks, which are concrete problem specifications on datasets (they define the target variable and train-test splits such as k-fold cross-validation). Tasks can be bundled into *benchmarking suites*, which are curated sets of tasks that meet certain quality criteria defined by the creator (Bischl et al., 2021). These make it easier for researchers to quickly find high-quality datasets on which to evaluate their methods. The use of clearly defined inclusion criteria is a substantial improvement over the common practice of selecting datasets without a clear rationale for their selection.

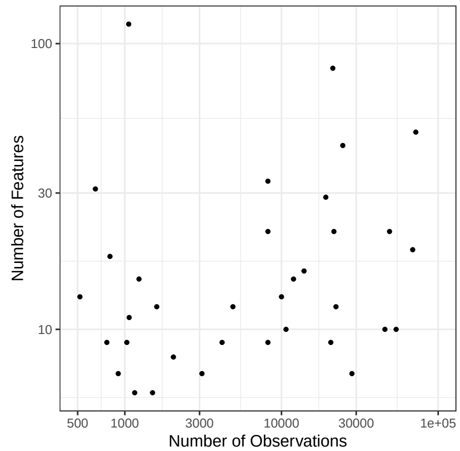

Figure 1: Size of the datasets in the OpenML-CTR23. We show the number of observations on the x-axis and the number of features on the y-axis; both are on log scale.

Most tabular benchmarking suites deal with supervised classification, such as the OpenML-CC18 and the suites mentioned therein (Bischl et al., 2021), but there are also two benchmarking suites for regression, that we discuss in more detail in section 3.2.

Beyond OpenML, there are also other repositories that offer access to large collections of datasets, including Kaggle (Anthony and Howard, 2010), the UCI machine learning repository (Dua and Graff, 2017), and the Penn Machine Learning Benchmark (Olson et al., 2017).

There also exists a large-scale comparison of regression methods using a subset of 42 regression problems from the UCI repository (Fernández-Delgado et al., 2019). However, the focus of this study was to empirically compare methods, not to create an easy-to-use benchmarking suite.

3 Benchmarking Suite

Our main contribution is a curated collection of 35 regression problems available on OpenML.[1] The benchmarking suite is accessible either via the website or the REST API, for which client libraries exist in Python (Feurer et al., 2021), R (Casalicchio et al., 2019), Java (van Rijn, 2016), and Julia.[2] In addition to using existing OpenML datasets, we have also uploaded new datasets. In many cases, we have also re-uploaded datasets to OpenML with more accurate metadata.

We now discuss our design criteria and then compare our proposed suite with existing benchmarking suites. Figure 1 shows the distribution of the number of features and number of observations in a scatter plot and we provide an overview of all datasets in Appendix A.

3.1 Quality Criteria

We follow the design criteria of the OpenML-CC18 (Bischl et al., 2021) as it was the first benchmarking suite to follow clearly defined inclusion criteria. We list these criteria here in order to keep the paper self-contained:

(a) There are between 500 and 100000 observations.

(b) There are less than 5000 features after one-hot encoding all categorical features.

(c) The dataset is not in a sparse format.

[1] `https://www.openml.org/search?type=study&study_type=task&sort=tasks_included&id=353`
[2] `https://www.openml.org/apis`

(d) The observations are i.i.d., which means that we exclude datasets that have time dependencies or require grouped data splits.

(e) The dataset comes with a source or reference that clearly describes it.

(f) We did not consider the dataset to be artificial, but allowed simulated datasets, see Bischl et al. (2021) for more information on the difference.

(g) The data is not a subset of a larger dataset.

In addition, we introduce the following criteria, which are relevant for regression tasks (and ignore the CC18 criteria, which are specific to classification tasks):

(a) There is a numeric target variable with at least 5 different values.

(b) The dataset is not trivially solvable by a linear model, i.e. the training error of a linear model fitted to the whole data has an R^2 of less than 1.

Moreover, we have included the following two criteria to increase the broad usability of our benchmarking suite (see Appendix C for more details):

(a) The dataset does not have ethical concerns.

(b) The use of the dataset for benchmarking is not forbidden.

In addition to the datasets, the OpenML tasks also contain resampling splits, which were determined according to the following rule: If there are less than 1000 observations we use 10 times repeated 10-fold CV. If there are more than 10000 observations we use a 33% holdout split, and for everything between, we use 10-fold CV.

3.2 Comparison with existing OpenML Regression Suites

There are two other regression benchmarking suites available on OpenML, one from the AutoML benchmark (Gijsbers et al., 2022), which we will refer to as *AMLB* from now on, and another from a recent comparison of deep learning methods with tree-based models (Grinsztajn et al., 2022), which we will refer to as *GOVB* (Grinsztajn, Oyallon, and Varoquaux Benchmark). For a more fine-grained discussion on the dataset level (here we only compare design criteria), see Appendix C.
Additional Datasets : the OpenML-CTR23 contains 23 datasets that are not included in any of the existing regression suites.
Quality of Description : we put a strong emphasis on the quality of the dataset description. This excludes datasets from both existing suites for which we were unable to find satisfactory information.
Dataset Size : we focused on medium-sized datasets in the range of 500 to 100000 observations. The GOVB contains datasets from 3000 up to around 5.5 million observations. The AMLB covers datasets from 240 up to 10 million observations. The rationale for this is to make it widely usable by limiting the computational requirements of running experiments on the suite.
Usage Restrictions : we exclude datasets from Kaggle challenges that can only be legally used during the duration of the competition. We find such datasets in the other two suites.
Missing Values : both the AMLB and CTR23 contain datasets with missing values, while they have been removed in the GOVB. Since some learning algorithms handle missing values natively, such global preprocessing steps may put these algorithms at a relative disadvantage.
Removed Features : the GOVB excludes categorical features with more than 20 items and numerical features with less than 10 unique values. Therefore, the resulting tasks might miss important features and do not necessarily respond to real-world problems anymore.

Dataset Difficulty : the CTR23 is the most conservative of the three benchmarking suites in terms of removing datasets based on a difficulty criterion, as we only remove datasets that follow a perfectly linear relationship. Both the AMLB and the GOVB remove datasets when the score difference between selected evaluated machine learning methods are considered to be too small.
Task Splits : Both the AMLB and the CTR23 use OpenML task splits, while the the GOVB does not. In fact, we found 3 datasets in the GOVB that require custom train-splits, which we list in Appendix C.

4 Experiments

We now compare five machine learning models on all CTR23 datasets. The selected methods range from complex black box models (XGBoost, Random Forest) to simple interpretable models (Ridge Regression, Decision Tree). As a middle ground between these two extremes, we also consider a Generalized Additive Model (GAM). With this experiment, we test whether the datasets are sufficiently complex that the simple models are not yet able to adequately capture the functional relationships between the features and the target. We will also compare the experimental results with the results of previous studies to see if they are in agreement.

We run each algorithm once on every task defined by the benchmarking suite and use the root mean-squared error (RMSE) as the evaluation measure. We use the train-test splits provided on OpenML, which we have defined in section 3.2 and conduct a rank-based analysis of the results using the Friedman test (Friedman, 1937) and the Nemenyi post-hoc test (Demšar, 2006). Further details on the experimental setup and specific configurations can be found in the Appendix B. The code and experimental results are available on GitHub.[3]

4.1 Models

We briefly describe the algorithms we compare and refer to Appendix B for additional details and hyperparameter search spaces.
XGBoost (Chen and Guestrin, 2016): we tune 8 hyperparameters for 500 random search iterations (Bergstra and Bengio, 2012) and use a nested resampling procedure.
Random Forest (Breiman, 2001): we use the implementation from the R package *ranger* (Wright and Ziegler, 2017) with the default configuration which is known to work reasonably well (Probst et al., 2019).
Generalized Additive Model: a flexible statistical model that (additively) combines multiple smooth functions of predictor variables (Hastie, 2017). We used the the R package *mgcv* (Wood, 2001) and neither performed hyperparameter tuning nor specified interaction effects.
Ridge Regression (Hoerl and Kennard, 1970): The lambda parameter is tuned using a simple grid search and an inner cross-validation. We use the implementation from the R package *glmnet* (Friedman et al., 2010).
Regression Tree: a single decision tree (Breiman, 1984) as a baseline model without hyperparameter tuning. We use the implementation from the R package *rpart* (Therneau and Atkinson, 2022).

4.2 Results

A global Friedman test showed the results to be significant on the 5% levels. Figure 2 summarizes the results of the post-hoc Nemenyi test. XGBoost is statistically different in all pairwise comparisons and is the clear winner with an average rank of 1.31. The Random Forest comes in second with an average rank of 2.46, but is not significantly different from the GAM, which has an average rank of 2.83. The two worst-performing models are the Ridge Regression with an average rank of 4.17 and the Regression Tree with 4.23.

[3] https://github.com/slds-lmu/paper_2023_regression_suite

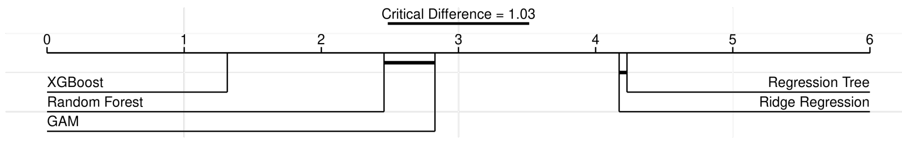

Figure 2: A critical difference plot visualizing the results of the post hoc Nemenyi test for pairwise comparisons. Algorithms that are connected by a thick horizontal line have a rank difference smaller than the critical difference value and are not significantly different on the 5% level.

The top performance of XGBoost is not surprising, as it is the only model other than the much simpler Ridge Regression that we have tuned. This is consistent with the results of Grinsztajn et al. (2022), where XGBoost is also the best performing model for the regression datasets. Fernández-Delgado et al. (2019) also find a gradient boosted tree (although the *gbm* implementation of Greenwell et al. (2022)) to be superior to all other methods considered in our benchmark experiment. They also find the Random Forest (albeit a different implementation) to be superior to the GAM, Ridge Regression and the (rpart) decision tree. As the ranking reflects the complexity of the models considered, we can conclude that the relationships in the CTR23 datasets are challenging enough to be used to benchmark more sophisticated algorithms.

5 Conclusion

Our goal was to provide a high-quality collection of carefully curated regression problems and to make it easily accessible via OpenML. The result of this effort is the OpenML-CTR23, a benchmark suite of 35 regression problems. As design criteria, we adapted those of the CC18 to the regression setting and added two criteria to make it more usable. We then evaluated five machine learning methods of varying complexity, whose performance differed significantly. From this, we concluded that the developed regression suite contains sufficiently challenging datasets to discriminate between simple and complex methods.

While these design criteria are conceptually motivated, they are not experimentally evaluated. An interesting question for further research is how different choices of quality criteria, such as the exclusion of time dependencies, different difficulty criteria, or the inclusion of simulated datasets, affect the results of benchmark experiments.

6 Broader Impact and Limitations

We are not aware of any direct negative impact on society. By providing carefully curated datasets, we hope to help other researchers in two ways. First, they will need to spend less time collecting datasets, as the work has already been done for them. Second, because our primary focus was the creation of a benchmarking suite rather than developing a new method, we probably spent more time selecting datasets than is realistic in a study where a dataset collection is only a by-product. We, therefore, hope that the use of this benchmarking suite will lead to more reliable results in future machine learning research.

Although we were already more conservative about including datasets due to license restrictions, we still included some datasets without explicit licenses, when we felt they were clearly intended for academic use. These are mostly old datasets from a time when dataset licenses were not commonly added. We acknowledge that this is not optimal, but also see it as a step in the right direction and a FAIRer research culture (Stall et al., 2019).

Acknowledgements. The authors of this work take full responsibilities for its content. Sebastian Fischer is supported by the Deutsche Forschungsgemeinschaft (DFG, German Research oundation) – 460135501 (NFDI project MaRDI). This work has been carried out by making use Wyoming's Advanced Research Computing Center, on its Beartooth Compute Environment (https://doi.org/10.15786/M2FY47). The authors gratefully acknowledge the computational and data resources provided by Wyoming's Advanced Research Computing Center (https://www.uwyo.edu/arcc/).

References

Anthony, G. and Howard, J. (2010). Kaggle. `https://www.kaggle.com/`.

Arzamasov, V., Böhm, K., and Jochem, P. (2018). Towards concise models of grid stability. In *2018 IEEE international conference on communications, control, and computing technologies for smart grids (SmartGridComm)*, pages 1–6. IEEE.

Bayle, P., Bayle, A., Janson, L., and Mackey, L. (2020). Cross-validation confidence intervals for test error. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.-F., and Lin, H., editors, *Proceedings of the 33rd International Conference on Advances in Neural Information Processing Systems (NeurIPS'20)*, pages 16339–16350. Curran Associates.

Bergstra, J. and Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13:281–305.

Bertin-Mahieux, T., Ellis, D., Whitman, B., and Lamere, P. (2011). The million song dataset. In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011)*.

Biblarz, T. and Raftery, A. (1993). The effects of family disruption on social mobility. *American sociological review*, pages 97–109.

Bierens, H. and Ginther, D. (2001). Integrated conditional moment testing of quantile regression models. *Empirical Economics*, 26:307–324.

Binder, M., Pfisterer, F., Lang, M., Schneider, L., Kotthoff, L., and Bischl, B. (2021). mlr3pipelines—flexible machine learning pipelines in R. *The Journal of Machine Learning Research*, 22(1):8314–8320.

Bischl, B., Binder, M., Lang, M., Pielok, T., Richter, J., Coors, S., Thomas, J., Ullmann, T., Becker, M., Boulesteix, A., Deng, D., and Lindauer, M. (2023). Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, page e1484.

Bischl, B., Casalicchio, G., Feurer, M., Hutter, F., Lang, M., Mantovani, R., van Rijn, J., and Vanschoren, J. (2021). OpenML benchmarking suites. In Vanschoren, J. and Yeung, S., editors, *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*. Curran Associates.

Bradshaw, G. (1989). Solar flare data set. `http://archive.ics.uci.edu/ml/datasets/solar+flare`.

Breiman, L. (1984). *Classification and regression trees*. Routledge.

Breiman, L. (2001). Random forests. *Machine Learning Journal*, 45:5–32.

Brooks, T., Pope, S., and Marcolini, M. (1989). Airfoil self-noise and prediction. Technical report, NASA.

Casalicchio, G., Bossek, J., Lang, M., Kirchhoff, D., Kerschke, P., Hofner, B., Seibold, H., Vanschoren, J., and Bischl, B. (2019). OpenML: An R package to connect to the machine learning platform OpenML. *Computational Statistics*, 34(3):977–991.

Cassotti, M., Ballabio, D., Todeschini, R., and Consonni, V. (2015). A similarity-based qsar model for predicting acute toxicity towards the fathead minnow (pimephales promelas). *SAR and QSAR in Environmental Research*, 26(3):217–243.

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In Krishnapuram, B., Shah, M., Smola, A., Aggarwal, C., Shen, D., and Rastogi, R., editors, *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'16)*, pages 785–794. ACM Press.

Coraddu, A., Oneto, L., Ghio, A., Savio, S., Anguita, D., and Figari, M. (2016). Machine learning approaches for improving condition-based maintenance of naval propulsion plants. *Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment*, 230(1):136–153.

Cortez, P., Cerdeira, A., Almeida, F., Matos, T., and Reis, J. (2009). Modeling wine preferences by data mining from physicochemical properties. *Decision support systems*, 47(4):547–553.

Cortez, P. and Morais, R. (2007). A data mining approach to predict forest fires using meteorological data. In Neves, J., Santos, M., and Machado, J., editors, *New trends in artificial intelligence : proceedings of the 13th Portuguese Conference on Artificial Intelligence (EPIA 2007)*, pages 512–523. Associação Portuguesa para a Inteligência Artificial (APPIA).

Cortez, P. and Silva, G. (2008). Using data mining to predict secondary school student performance. In Brito, A. and Teixeira, J., editors, *Proceedings of 5th Annual Future Business Technology Conference*, pages 5–12. EUROSIS-ETI.

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30.

Deneke, T., Haile, H., Lafond, S., and Lilius, J. (2014). Video transcoding time prediction for proactive load balancing. In *2014 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE.

Dua, D. and Graff, C. (2017). UCI machine learning repository.

Fairlearn (2016). Revisiting the boston housing dataset. `https://fairlearn.org/main/user_guide/datasets/boston_housing_data.html`.

Feng, J., Lurati, L., Ouyang, H., Robinson, T., Wang, Y., Yuan, S., and Young, S. (2003). Predictive toxicology: benchmarking molecular descriptors and statistical methods. *Journal of chemical information and computer sciences*, 43(5):1463–1470.

Fernandes, K., Vinagre, P., and Cortez, P. (2015). A proactive intelligent decision support system for predicting the popularity of online news. In *Progress in Artificial Intelligence: 17th Portuguese Conference on Artificial Intelligence, EPIA 2015, Coimbra, Portugal, September 8-11, 2015. Proceedings 17*, pages 535–546. Springer.

Fernández-Delgado, M., Sirsat, M., Cernadas, E., Alawadi, S., Barro, S., and Febrero-Bande, M. (2019). An extensive experimental survey of regression methods. *Neural Networks*, 111:11–34.

Feurer, M., van Rijn, J., Kadra, A., Gijsbers, P., Mallik, N., Ravi, S., Müller, A., Vanschoren, J., and Hutter, F. (2021). OpenML-Python: an extensible Python API for OpenML. *Journal of Machine Learning Research*, 22(100):1–5.

Friedman, J., Hastie, T., and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22.

Friedman, M. (1937). The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association*, 32(200):675–701.

Ghahramani, Z. (1996a). The kin datasets. `https://www.cs.toronto.edu/~delve/data/kin/desc.html`.

Ghahramani, Z. (1996b). The pumadyn datasets. *J. Complex*, pages 1–6. `https://www.cs.toronto.edu/~delve/data/pumadyn/desc.html`.

Gijsbers, P., Bueno, M. L. P., Coors, S., LeDell, E., Poirier, S., Thomas, J., Bischl, B., and Vanschoren, J. (2022). Amlb: an automl benchmark. *arXiv:2207.12560 [cs.LG]*.

Greenwell, B., Boehmke, B., Cunningham, J., and Developers, G. (2022). *gbm: Generalized Boosted Regression Models*. R package version 2.1.8.1.

Grinsztajn, L., Oyallon, E., and Varoquaux, G. (2022). Why do tree-based models still outperform deep learning on typical tabular data? In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Hamidieh, K. (2018). A data-driven statistical model for predicting the critical temperature of a superconductor. *Computational Materials Science*, 154:346–354.

Hastie, T. J. (2017). Generalized additive models. In *Statistical models in S*, pages 249–307. Routledge.

Hoerl, A. E. and Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67.

Kaggle (2015). Bike sharing demand dataset. `https://www.kaggle.com/c/bike-sharing-demand/data/`.

Kaggle (2016). House sales in king county, usa. `https://www.kaggle.com/datasets/harlfoxem/housesalesprediction`.

Kaggle (2017). Moneyball. `https://www.kaggle.com/datasets/wduckett/moneyball-mlb-stats-19622012`.

Kaggle (2020). Brazilian houses to rent. `https://www.kaggle.com/datasets/rubenssjr/brasilian-houses-to-rent`.

Kaggle (2021). Fifa 22 complete player dataset. `https://www.kaggle.com/datasets/stefanoleone992/fifa-22-complete-player-dataset`.

Kaggle (2022). Miami housing dataset. `https://www.kaggle.com/datasets/deepcontractor/miami-housing-dataset`.

Kuiper, S. (2008). Introduction to multiple regression: How much is your car worth? *Journal of Statistics Education*, 16(3).

Lang, M., Binder, M., Richter, J., Schratz, P., Pfisterer, F., Coors, S., Au, Q., Casalicchio, G., Kotthoff, L., and Bischl, B. (2019). mlr3: A modern object-oriented machine learning framework in R. *Journal of Open Source Software*, 4(44):1903.

Lang, M., Bischl, B., and Surmann, D. (2017). batchtools: Tools for R to work on batch systems. *Journal of Open Source Software*, 2(10):135.

Michie, D. and Camacho, R. (1994). Building symbolic representations of intuitive real-time skills from performance data. *Machine Intelligence 13*.

Nash, W., Sellers, T., Talbot, S., Cawthorn, A., and Ford, W. (1994). The population biology of abalone (haliotis species) in tasmania. i. blacklip abalone (h. rubra) from the north coast and islands of bass strait. *Sea Fisheries Division, Technical Report*, 48:p411.

Nesha, M. (2019). Wave energy converters data set. `https://archive.ics.uci.edu/ml/datasets/Wave+Energy+Converters`.

Olson, C. A. (1998). A comparison of parametric and semiparametric estimates of the effect of spousal health insurance coverage on weekly hours worked by wives. *Journal of Applied Econometrics*, 13(5):543–565.

Olson, R. S., La Cava, W., Orzechowski, P., Urbanowicz, R. J., and Moore, J. H. (2017). PMLB: a large benchmark suite for machine learning evaluation and comparison. *BioData mining*, 10:1–13.

Ordoni, E., Bach, J., and Fleck, A.-K. (2022). Analyzing and predicting verification of data-aware process models–a case study with spectrum auctions. *IEEE Access*, 10:31699–31713.

Pace, R. K. and Barry, R. (1997a). Quick computation of spatial autoregressive estimators. *Geographical analysis*, 29(3):232–247.

Pace, R. K. and Barry, R. (1997b). Sparse spatial autoregressions. *Statistics & Probability Letters*, 33(3):291–297.

Peeters, S., Melnikov, V., and Hüllermeier, E. (2021). Performance prediction for hardware-software configurations: A case study for video games. In *Advances in Intelligent Data Analysis XIX: 19th International Symposium on Intelligent Data Analysis, IDA 2021, Porto, Portugal, April 26–28, 2021, Proceedings 19*, pages 222–234. Springer.

Probst, P., Wright, M., and Boulesteix, A. (2019). Hyperparameters and tuning strategies for random forest. *WIREs Data Mining and Knowledge Discovery*, 9(3).

R Core Team (2018). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Rana, P. (2013). Physicochemical properties of protein tertiary structure data set.

Rasmussen, C., Neal, R., Hinton, G., van Camp, D., Revow, M., Ghahramani, Z., Kustra, R., and Tibshirani, R. (1996). Computer activity dataset. `http://www.cs.toronto.edu/~delve/data/datasets.html`.

Stall, S., Yarmey, L., Cutcher-Gershenfeld, J., Hanson, B., Lehnert, K., Nosek, B., Parsons, M., Robinson, E., and Wyborn, L. (2019). Make scientific data fair. *Nature*, 570(7759):27–29.

Therneau, T. and Atkinson, B. (2022). *rpart: Recursive Partitioning and Regression Trees*. R package version 4.1.19.

Tsanas, A. and Xifara, A. (2012). Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools. *Energy and buildings*, 49:560–567.

Ushey, K. (2023). *renv: Project Environments*. R package version 0.17.0.

van Rijn, J. N. (2016). *Massively Collaborative Machine Learning*. PhD thesis, Leiden University.

Vanschoren, J., van Rijn, J., Bischl, B., and Torgo, L. (2014). OpenML: Networked science in machine learning. *SIGKDD Explorations*, 15(2):49–60.

Vijayakumar, S. and Schaal, S. (2000). Locally weighted projection regression: An o (n) algorithm for incremental real time learning in high dimensional space. In *Proceedings of the seventeenth international conference on machine learning (ICML 2000)*, volume 1, pages 288–293. Morgan Kaufmann.

Wickham, H. (2011). ggplot2. *Wiley interdisciplinary reviews: computational statistics*, 3(2):180–185.

Wood, S. (2001). mgcv: GAMs and generalized ridge regression for R. *R news*, 1(2):20–25.

Wright, M. and Ziegler, A. (2017). ranger: A fast implementation of random forests for high dimensional data in C++ and R. *Journal of Statistical Software*, 77(1):1–17.

Yeh, I.-C. (1998). Modeling of strength of high-performance concrete using artificial neural networks. *Cement and Concrete research*, 28(12):1797–1808.

Zhou, F., Claire, Q., and King, R. D. (2014). Predicting the geographical origin of music. In *2014 IEEE International Conference on Data Mining*, pages 1115–1120. IEEE.

A Dataset Overview

Table 1 summarizes all datasets contained in the OpenML-CTR23.

B More Details on Benchmark

B.1 Software Environment

The experiments were carried out using the R package *mlr3* (Lang et al., 2019), which is a machine learning framework for the R language (R Core Team, 2018). The data was pre-processed using the mlr3 extension *mlr3pipelines* (Binder et al., 2021). We also used the R package *batchtools* (Lang et al., 2017) to run the experiments on the cluster. Although the experiments were performed in R, the results are included in the GitHub repository as a CSV file.

B.2 Preprocessing

We included the following preprocessing operations:

- We collapse the rarest categorical levels until there are at most 1000 different factor levels.
- If the method cannot handle categorical data, such features are one-hot encoded.
- If the method cannot handle missing data, missing categorical values are imputed using out-of-range imputation.
- If the method cannot handle missing data, missing numerical features are imputed by sampling values from their empirical histogram.

Name	Data ID	Task ID	n	p	Source
abalone	44956	361234	4177	9	Nash et al. (1994)
airfoil_self_noise	44957	361235	1503	6	Brooks et al. (1989)
auction_verification	44958	361236	2043	8	Ordoni et al. (2022)
brazilian_houses	44990	361267	10692	10	Kaggle (2020)
california_housing	44977	361255	20640	9	Pace and Barry (1997b)
cars	44994	361622	804	18	Kuiper (2008)
concrete_compressive_strength	44959	361237	1030	9	Yeh (1998)
cps88wages	44984	361261	28155	7	Bierens and Ginther (2001)
cpu_activity	44978	361256	8192	22	Rasmussen et al. (1996)
diamonds	44979	361257	53940	10	Wickham (2011)
energy_efficiency	44960	361617	768	9	Tsanas and Xifara (2012)
fifa	45012	361272	19178	29	Kaggle (2021)
forest_fires	44962	361618	517	13	Cortez and Morais (2007)
fps_benchmark	44992	361268	24624	44	Peeters et al. (2021)
geographical_origin_of_music	44965	361243	1059	117	Zhou et al. (2014)
grid_stability	44973	361251	10000	13	Arzamasov et al. (2018)
health_insurance	44993	361269	22272	12	Olson (1998)
kin8nm	44980	361258	8192	9	Ghahramani (1996a)
kings_county	44989	361266	21613	22	Kaggle (2016)
miami_housing	44983	361260	13932	16	Kaggle (2022)
Moneyball	41021	361616	1232	15	Kaggle (2017)
naval_propulsion_plant	44969	361247	11934	15	Coraddu et al. (2016)
physiochemical_protein	44963	361241	45730	10	Rana (2013)
pumadyn32nh	44981	361259	8192	33	Ghahramani (1996b)
QSAR_fish_toxicity	44970	361621	908	7	Cassotti et al. (2015)
red_wine	44972	361250	1599	12	Cortez et al. (2009)
sarcos	44976	361254	48933	22	Vijayakumar and Schaal (2000)
socmob	44987	361264	1156	6	Biblarz and Raftery (1993)
solar_flare	44966	361244	1066	11	Bradshaw (1989)
space_ga	45402	361623	3107	7	Pace and Barry (1997a)
student_performance_por	44967	361619	649	31	Cortez and Silva (2008)
superconductivity	44964	361242	21263	82	Hamidieh (2018)
video_transcoding	44974	361252	68784	19	Deneke et al. (2014)
wave_energy	44975	361253	72000	49	Nesha (2019)
white_wine	44971	361249	4898	12	Cortez et al. (2009)

Table 1: Overview of datasets, including the name, OpenML data and task ID, the number of observations (n), the number of features (p), and the source.

B.3 Hyperparameter Settings

XGBoost : the search space is defined in Table 2 and is taken from Bischl et al. (2023). As mentioned in section 4.1, we tune XGBoost using 500 random search iterations and using a nested resampling procedure. In the inner resampling we use 10-fold cross-validation for datasets with less than 1000 observations, 3 folds for datasets between 1000 and 10000 observations and a 33% holdout resampling for everything else.
Random Forest : we use the default configuration from the R package *ranger* (Wright and Ziegler, 2017).

Hyperparameter	Range	Scale
η	$[1 \times 10^{-4}, 1]$	Logscale
n_{rounds}	$[1, 5000]$	
max_depth	$[1, 20]$	
colsample_bytree	$[0.1, 1]$	
colsample_bylevel	$[0.1, 1]$	
λ	$[0.001, 1000]$	Logscale
α	$[0.001, 1000]$	Logscale
subsample	$[0.1, 1]$	

Table 2: Search space for XGBoost

GAM For the generalised additive model we add smooth effects for all numerical features with more than 20 different values. We set the number of knots for each smooth effect to 5 if the ratio of the number of observations to the number of features is less than 10, and to 5 otherwise, thereby avoiding non-identifiable models.
Ridge Regression : we tune the lambda parameter using the default tuning strategy of the *glmnet* package (Friedman et al., 2010). For the inner resampling we use 20-fold cross-validation for datasets with less than 1000 observations and 10-fold cross-validation for all other datasets.
Regression Tree : We tune no hyperparameters and use the default configuration from the *rpart* package (Therneau and Atkinson, 2022).

B.4 Computational Workload and Reproducibility

The total amount of CPU hours for the final experiment was roughly 13000. The code for the experiments[4] is available and open source. Seeds are set for all experiments and an *renv* file describing the computational environment (Ushey, 2023) is included. Instructions on how to reproduce the results are contained in the `README` file of the repository.

B.5 More results

Table 3 contains the RMSE of all five models on all datasets.

[4] `https://github.com/slds-lmu/paper_2023_ci_for_ge`

Task	XGBoost	RF	GAM	Ridge	Tree	
abalone	2.118	2.133	2.120	2.330	2.404	$\times 10^{0}$
airfoil_self_noise	1.170	2.203	4.588	4.930	4.414	$\times 10^{0}$
auction_verification	0.394	2.972	6.140	6.301	3.155	$\times 10^{3}$
brazilian_houses	0.446	0.587	0.321	0.442	1.149	$\times 10^{4}$
california_housing	4.464	5.050	6.193	7.247	7.809	$\times 10^{4}$
cars	2.111	2.486	2.935	3.080	3.422	$\times 10^{3}$
concrete_compressive_strength	0.371	0.529	0.963	1.075	0.900	$\times 10^{1}$
cps88wages	3.800	3.830	3.856	4.120	4.027	$\times 10^{2}$
cpu_activity	2.190	2.461	2.714	9.984	4.767	$\times 10^{0}$
diamonds	0.521	0.540	2.127	1.335	1.311	$\times 10^{3}$
energy_efficiency	0.280	1.082	2.934	3.298	2.575	$\times 10^{0}$
fifa	0.893	0.929	0.904	1.517	1.029	$\times 10^{4}$
forest_fires	4.830	5.037	4.883	4.601	6.112	$\times 10^{1}$
fps_benchmark	0.051	3.363	1.166	1.189	2.339	$\times 10^{1}$
geographical_origin_of_music	1.519	1.567	1.733	1.711	1.809	$\times 10^{1}$
grid_stability	0.744	1.280	1.711	2.212	2.678	$\times 10^{-2}$
health_insurance	1.439	1.452	1.465	1.503	1.523	$\times 10^{1}$
kin8nm	1.092	1.452	1.974	2.034	2.160	$\times 10^{-1}$
kings_county	1.144	1.314	1.560	1.651	2.050	$\times 10^{5}$
miami_housing	0.815	0.925	1.328	1.803	1.726	$\times 10^{5}$
Moneyball	2.218	2.428	2.090	2.265	3.640	$\times 10^{1}$
naval_propulsion_plant	0.078	0.112	0.013	1.080	0.773	$\times 10^{-2}$
physiochemical_protein	3.326	3.456	4.951	5.232	5.422	$\times 10^{0}$
pumadyn32nh	2.176	2.621	3.306	3.322	2.424	$\times 10^{-2}$
QSAR_fish_toxicity	0.864	0.861	0.923	0.986	1.028	$\times 10^{0}$
red_wine	5.473	5.614	6.508	6.647	6.828	$\times 10^{-1}$
sarcos	0.214	0.292	0.472	0.628	1.122	$\times 10^{1}$
socmob	1.246	1.902	2.119	2.904	2.273	$\times 10^{1}$
solar_flare	7.627	8.004	7.664	8.106	7.921	$\times 10^{-1}$
space_ga	1.049	1.151	1.053	1.535	1.400	$\times 10^{-1}$
student_performance_por	2.675	2.638	2.749	2.844	2.889	$\times 10^{0}$
superconductivity	0.901	0.914	1.414	1.901	1.796	$\times 10^{1}$
video_transcoding	0.078	0.337	1.092	1.115	0.706	$\times 10^{1}$
wave_energy	0.497	4.536	0.009	0.420	9.226	$\times 10^{4}$
white_wine	5.693	5.937	7.183	7.639	7.613	$\times 10^{-1}$

Table 3: The root mean-square error of all five models (XGBoost, Random Forest, GAM, Ridge Regression, and Regression Tree) on all 35 datasets of the CTR23. To obtain the actual RMSE score, each value must be multiplied by the factor in the rightmost column.

Task	XGBoost	RF	GAM	Ridge	Tree
abalone	1	3	2	4	5
airfoil_self_noise	1	2	4	5	3
auction_verification	1	2	4	5	3
brazilian_houses	3	4	1	2	5
california_housing	1	2	3	4	5
cars	1	2	3	4	5
concrete_compressive_strength	1	2	4	5	3
cps88wages	1	2	3	5	4
cpu_activity	1	2	3	5	4
diamonds	1	2	5	4	3
energy_efficiency	1	2	4	5	3
fifa	1	3	2	5	4
forest_fires	3	4	2	1	5
fps_benchmark	1	5	2	3	4
geographical_origin_of_music	1	2	4	3	5
grid_stability	1	2	3	4	5
health_insurance	1	2	3	4	5
kin8nm	1	2	3	4	5
kings_county	1	2	3	4	5
miami_housing	1	2	3	5	4
Moneyball	2	4	1	3	5
naval_propulsion_plant	2	3	1	5	4
physiochemical_protein	1	2	3	4	5
pumadyn32nh	1	3	4	5	2
QSAR_fish_toxicity	2	1	3	4	5
red_wine	1	2	3	4	5
sarcos	1	2	3	4	5
socmob	1	2	3	5	4
solar_flare	1	4	2	5	3
space_ga	2	3	1	5	4
student_performance_por	2	1	3	4	5
superconductivity	1	2	3	5	4
video_transcoding	1	2	4	5	3
wave_energy	3	4	1	2	5
white_wine	1	2	3	5	4

Table 4: The ranks of all five models (XGBoost, Random Forest (RF), GAM, Ridge Regression, and Regression Tree) on all 35 tasks from the OpenML-CT323. Lower ranks indicate a smaller root mean-square error.

C Discussion of Datasets

While we compared the design criteria of existing benchmarking suites in section 3.2, here we go one step further and comment on some of the datasets included in other benchmarking suites.

C.1 Usage Restrictions

Both the AMLB and the GOVB include datasets from Kaggle challenges that can only be used for the purpose and the duration of the challenge. These include the *Mercedes-Beng Greener Manufacturing* challenge (OpenML dataset ID 42570) and the *Santander Customer Transaction Prediction* challenge (ID 42395).

C.2 Ethical Considerations

The *boston housing* (Dataset ID 531) is ethically questionable, as one of its features encodes racist assumptions (Fairlearn, 2016). As a precaution, we also remove the *us crimes* dataset (ID 42730), as the goal is to predict crime rates based on ethnical demographics, for which we do not have enough information about its context.

C.3 Dataset Description

For both the *yprop_4_1* and the *topo_2_1* datasets (IDs 416, 550) we could not match the dataset description from the associated paper (Feng et al., 2003) with the dimensions of the data on OpenML and therefore exclude them. The paper associated with *Buzzinsocialmedia_Twitter* (ID 4549) is in French and therefore inaccessible to non-French speakers. Other datasets such as *delays_zurich_transport* (ID 40753), *pol* (ID 201), *Yolanda* (ID 42705), or *quake* have rather sparse descriptions, which make it impossible to judge the quality of the data.

C.4 I.I.D. Data

Some of the datasets in the AMLB and and GOVB have time dependencies. The *OnlineNewsPopularity* (ID 42724) should be treated with a rolling window cross-validation as described in the associated article (Fernandes et al., 2015). Another example is *Airlines_DepDelay_10M* (ID 42728), as the delays of different aircraft are inherently time related and the dataset is treated accordingly in other research papers (Bayle et al., 2020). The *Bike Sharing Demand* data (ID 44142) comes from a Kaggle forecasting challenge (Kaggle, 2015) and should therefore also be treated with a rolling window cross-validation. The *particulate-matter-ukair-2017* (ID 42207) is a data stream collected continuously over time.

In addition to time dependencies, other datasets also require custom resampling splits. These include the *ailerons* and *elevators* datasets (IDs 296 and 216, Michie and Camacho (1994)) and the *YearPredictionMSD* dataset (ID 44027, Bertin-Mahieux et al. (2011)).