Reproducibility Report for ACM SIGMOD 2023 Paper: “ML2DAC: Meta-learning to Democratize AutoML for Clustering Analyses”

ZICHEN ZHU, Boston University, USA
ANDY HUYNH, Boston University, USA
CHUNWEI LIU, Massachusetts Institute of Technology, USA

The authors have made a commendable effort to ensure the reproducibility of this project. They have assembled an extensive reproducibility package complete with clear instructions. This package links to a public Git repository that contains a single script capable of running all the paper's experiments. By utilizing this package, we were able to successfully replicate most results and generate plots on our local system. While it is impossible to reproduce exactly the same results due to some randomness in the algorithm and variations in hardware environment, the experimental results we produced closely match those presented in the original paper. This consistency with the core idea conveyed in the paper is noteworthy. Overall, the process of executing the experimental pipeline was smooth and entirely automated.

1 INTRODUCTION

This reproducibility report pertains to the paper titled "ML2DAC: Meta-learning to Democratize AutoML for Clustering Analyses" [1], authored by Dennis Treder-Tschechlov, Manuel Fritz, Holger Schwarz, and Bernhard Mitschang from the University of Stuttgart. The paper was featured in the Proceedings of the 2023 ACM SIGMOD International Conference on Management of Data (SIGMOD'23). In this paper, the authors introduce a novel meta-learning approach, ML2DAC, for addressing the CASH (combined algorithm selection and hyperparameter) problem in clustering analyses. The experiment compares ML2DAC with baseline methods such as AutoML4Clust, AutoClust, and AutoCluster.

The authors have thoughtfully included a comprehensive reproducibility package in their submission. This package comprises detailed instructions and a public repository containing all the necessary code and datasets for replicating the experiments presented in the original paper, along with the plotting script. We conducted tests on all experiments that can run within a Docker container, which significantly streamlined the process and eliminated the need for manual setup. In summary, the submitted experiments validate the central results and claims of the paper. The key figures have been faithfully reproduced, and the reproducibility scripts are user-friendly and well-documented.

2 SUBMISSION

The source code and scripts for reproducing the paper's results are accessible on GitHub. The submission includes:

- GitHub repository with code and scripts: https://github.com/tschechlovdev/ml2dac
- A detailed readme file for experiment setup and execution is available in the reproducibility branch: https://github.com/tschechlovdev/ml2dac/blob/reproducibility/reproducibility.md
- The dataset is provided along with the code repository under the “real_world_data” directory.
- Within the reproducibility branch of the given git repository, you will find a “Dockerfile” to build the environment, and a “reproducibility.py” is included to execute all the experiments. This script is invoked within the Docker build file, so a simple “docker run” command with different parameters is sufficient to perform all the experiments and generate the corresponding figures.

3 HARDWARE AND SOFTWARE ENVIRONMENT

We summarize the hardware specifications from the original paper and the reproducibility assessment in Table 1.

Table 1. Hardware & Software environment

	Paper	Repro Review
CPU	Intel Xeon E5 2683 (v3)	Intel Xeon Gold 6230
cores	16	40
GHz	2.6	2.1
RAM	32GB	375GB

4 REPRODUCIBILITY EVALUATION

4.1 Process

We started by following the instructions to download Git, Git LFS, and Docker. Next, we downloaded the repository. Due to the large dataset in this repository, Git LFS is used, which may require authors' quotas to maintain its availability. After downloading the repository locally, we executed all the experiments and generated plots by running "docker build -t ml2dac ." and "docker run -v local/dir/output:/app/evaluation_results/output -d ml2dac." This process was facilitated by a single main script, "reproducibility.py". Please note that running this script can take more than 12 hours but fewer than 24 hours to complete. In addition, the script is designed with Docker input support, offering a user-friendly interface that allows users to specify which experiments to run and whether to generate plots.

4.2 Results

We have successfully reproduced the following figures and tables: Figure 1 (Figure 4 in [1]), Figure 2 (Figure 5 in [1]), Figure 3 (Figure 6 in [1]), Figure 4 (Figure 7 in [1]), Table 2 (Table 3 in [1]),Table 3 (Table 4 in [1]), Table 4 (Table 5 in [1]) and Table 5 (Table 6 in [1]). The numerical values and visual plots closely align with the paper's reported results, with the exception of Figure 5(b). The observed deviations can be attributed to differences in hardware environments and inherent algorithmic randomness. Notably, although Figure 5(b) shows a slightly lower ARI with a larger deviation,

0	Meta-Feature Set	CVI Selection	ARI (w=25)	Runtime Meta-Feature Extraction
0	General	0.46153846153846156	0.8935	0.0669
1	Stats	0.6923076923076923	0.9086	0.1915
2	Info	0.6025641025641025	0.8738	1.1187
3	Stats+Info	0.6794871794871795	0.8929	1.1887
4	Stats+General	0.6923076923076923	0.9173	0.0659
5	Info+General	0.5769230769230769	0.8803	1.1199
6	Stats+Info+General	0.717948717948718	0.9236	0.8372
7	MF - AutoClust	0.5512820512820513	0.7624	82.845
8	MF - AutoCluster	0.66789	0.9147	35.6888

Table 2. Reproduction of Table 3 in [1]

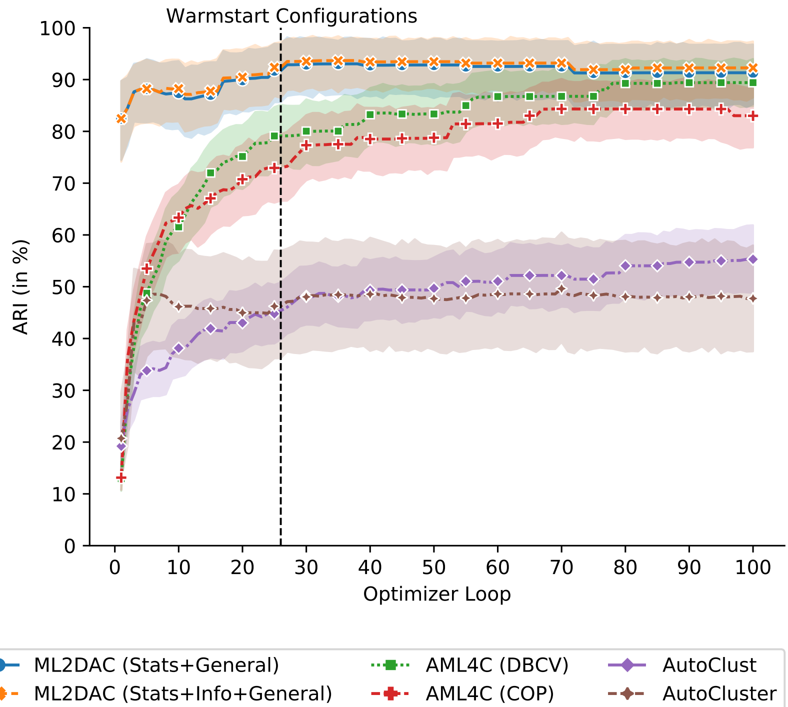
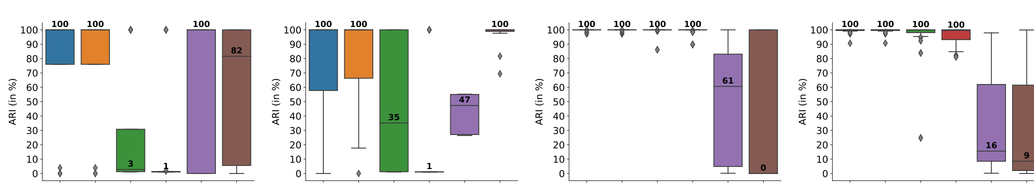


Fig. 1. Reproduction of Figure 4 in [1]



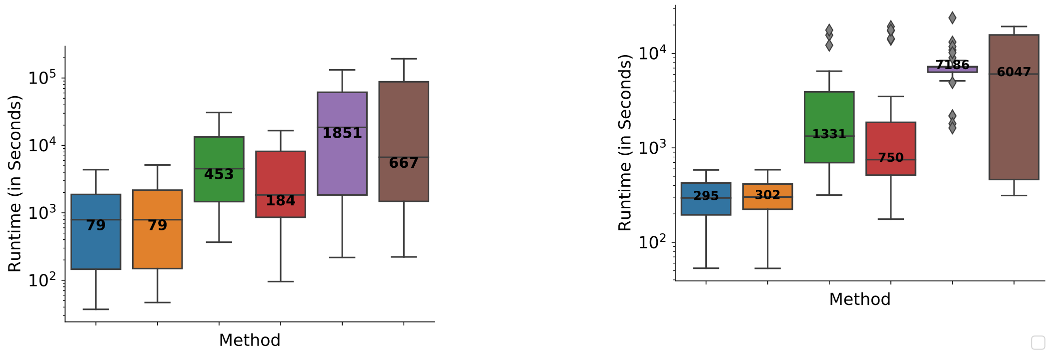
(a) Circles (b) **Moons** (c) Moons (d) Varied Gaussian

Fig. 2. Reproduction (excerpt) of Figure 5 in [1]

0	1	2	3	4	5	6	7
	2	12	22	32	42	52	62
ARI	0.8253	0.8166	0.9138	0.9212	0.9182	0.9274	0.9768

Table 3. Reproduction of Table 4 in [1]

the relative performance of ML2DAC in comparison to the other baselines is consistent with the paper’s findings and discussions. This consistency holds for all other figures and tables.



(a) All datasets.

(b) Large datasets ($n = 10,000$).

Fig. 3. Reproduction (excerpt) of Figure 6 in [1]

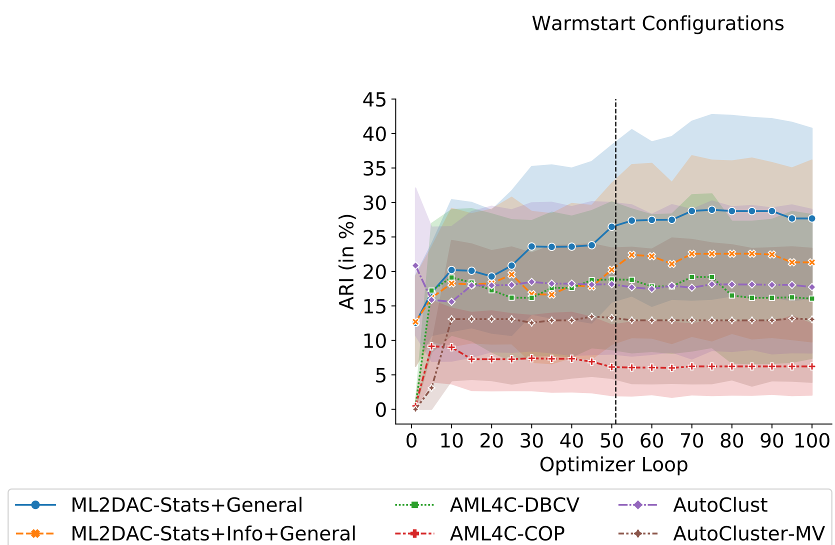


Fig. 4. Reproduction of Figure 7 in [1]

0	Method	Runtime (s) Median	Runtime (s) Mean
0	ML2DAC-Stats+General	162.16	25.95
1	ML2DAC-Stats+Info+General	73.97	27.73
2	AML4C-DBCV	314.42	124.42
3	AML4C-COP	244.82	100.19
4	AutoClust	540.39	331.51
5	AutoCluster-MV	122.01	25.19

Table 4. Reproduction of Table 5 in [1]

component	ARI
all	0.2563
no_algo_reduction	0.2636
no_cvi_selection	0.1769
no_warmstart	0.1989

Table 5. Reproduction of Table 6 in [1]

5 SUMMARY

All of the paper's claims have been successfully reproduced with minimal effort. The authors deserve commendation for their meticulous preparation, packaging, and documentation of the reproducibility package. This made replicating the local experiments reported in the paper a straightforward task.

REFERENCES

[1] Dennis Treder-Tschechlov, Manuel Fritz, Holger Schwarz, and Bernhard Mitschang. 2023. ML2DAC: Meta-Learning to Democratize AutoML for Clustering Analysis. *Proceedings of the ACM on Management of Data* 1, 2 (2023), 1–26.