Reproducibility Report for ACM SIGMOD 2022 Paper: "AutoMon: Automatic Distributed Monitoring for Arbitrary Multivariate Functions"

MATTHIAS WEIDLICH, Humboldt-Universität zu Berlin, Germany

The authors provide a comprehensive reproducibility package. The reproducibility instructions point to a public repository that contains a single script that executes all experiments from the paper. Using this package, we have been able to reproduce all results and plots that run locally (scripts for cloud-based experiments have been ignored due to the expected costs). In general, the execution of the experimental pipeline was straightforward and fully automated.

1 INTRODUCTION

This reproducibility report concerns the paper "AutoMon: Automatic Distributed Monitoring for Arbitrary Multivariate Functions" [1], co-authored by Hadar Sivan, Moshe Gabel, and Assaf Schuster, who are all with the Technion – Israel Institute of Technology. The paper was published at SIGMOD 2022 and introduced a scheme for the distributed evaluation of functions over data streams. The authors provide a reproducibility package including detailed instructions and a public repository. The latter includes all the code required to reproduce all experiments of the original paper and recompile the paper with the new plots. We tested all experiments that run locally and could confirm the results reported in the original paper. A few experiments require significant cloud resources (AWS) and, therefore, have not been considered.

2 SUBMISSION

The reproducibility package is available in a public git repository at `https://github.com/hsivan/automon`. This package includes the code required to reproduce all results from the paper and, in particular, the following important files:

- `README.md`, the entry point of the repository that summarizes the approach and gives an overview of the usage of the implementation.
- `docs/AutoMon_Availability_and_Reproducibility.pdf`, a concise summary of the reproducibility setup.
- `experiments/README.md` and `aws_experiments/README.md`, technical descriptions of the local and cloud-based experiments, respectively.

The reproducibility instructions point to a single main script, `reproduce_experiments.py`, to run all experiments. The cloud-based experiments are not executed by default but can be activated with a dedicated flag, `--aws`. We verified that the main script indeed executes the following steps, as detailed in the reproducibility instructions:

- It fetches the source code from the GitHub repository and potentially merges updates.
- It downloads external datasets.
- It executes all local experiments and, if activated by the aforementioned flag, the cloud-based experiments.
- It generates all figures and re-compiles the paper with the new plots.

It is worth mentioning that the script builds Docker images to run the experiments, which requires root privileges. Moreover, the configuration for each experiment is available in a dedicated file in JSON format, which makes it easy to run the experiments with changed settings.

3 HARDWARE AND SOFTWARE ENVIRONMENT

We summarize the hardware specifications used in the original paper as well as those used in the reproducibility assessment in Table 1.

Table 1. Hardware & Software environment

	Paper	Reproducibility Review
CPU	Intel i9-7900X	Intel i7-12700T
cores	10	8+4
GHz	3.3GHz - 4.3GHz	1.4GHz/1.0GHz - 4.6GHz/3.4GHz
RAM	64GB	32GB
OS	Ubuntu 18.04	Debian 11

4 REPRODUCIBILITY EVALUATION

4.1 Process

The main script, `reproduce_experiments.py`, worked flawlessly. It executed all the local experiments, generated the figures, and re-compiled the paper with the new plots. Also, the script is robust and can be stopped at any time. Once it is restarted, it skips over steps that have been done already (such as downloading the data or building a Docker image). Experiments that have been finished earlier are also skipped automatically. This way, we could split up the total runtime (roughly 2.5 days, as also reported by the authors) over several days. In general, the execution went smoothly.

After reproducing all results from the original paper, as a sanity check, we also explored the impact of changes in the configuration of some selected experiments. For instance, for the experiments on the error-communication trade-off (Figure 5), we increased the number of distributed streams.

4.2 Results

We successfully reproduced the results for all local experiments. This includes the results shown in Figures 4-9 in the original paper [1]. Our results have been virtually equivalent. Not even minor deviations could be observed for most experiments. Only in Figure 6, in the part on the DNN intrusion detection, the data points for the max error showed a very minor difference. However, this does not affect the conclusions drawn in the paper, and our data points for the 99th percentile error correspond to those reported in the original paper.

Considering changes in the parameter configurations of some experiments, the obtained results met our expectations of the impact of these changes.

5 SUMMARY

The reproducibility package provided by the authors made it straightforward to reproduce the results for the (local) experiments reported in the paper. In particular, the provided instructions have been very useful in understanding the implementation of the experimental workflow.

REFERENCES

[1] Hadar Sivan, Moshe Gabel, and Assaf Schuster. 2022. AutoMon: Automatic Distributed Monitoring for Arbitrary Multivariate Functions. In *SIGMOD '22: International Conference on Management of Data, Philadelphia, PA, USA, June 12 - 17, 2022*, Zachary Ives, Angela Bonifati, and Amr El Abbadi (Eds.). ACM, 310–324. https://doi.org/10.1145/3514221.3517866