Reproducibility Report for the Paper: "A Toolset for Predicting Performance of Legacy Real-Time Software Based on the RAST Approach"

EMILIO INCERTO, IMT School for Advanced Studies Lucca, Lucca, Italy
ROMOLO MAROTTA, DICII, University of Rome Tor Vergata, Roma, Italy

The examined paper introduces a tool for predicting performance of legacy real-time software based on the RAST Approach. The artifact evaluated in this report is well documented and allows to easily reproduce the computational results presented in the article. Additionally, the artifact is hosted on permanent repositories, ensuring a long-term retention. This paper can thus receive the *Artifacts Available*, *Artifacts Evaluated–Functional*, and *Artifact Validated–Results Reproduced* badges.

CCS Concepts: • **General and reference** → Empirical studies; *Experimentation*; **Validation**; **Verification;**

Additional Key Words and Phrases: Replication of computational results

ACM Reference Format:
Emilio Incerto and Romolo Marotta. 2025. Reproducibility Report for the Paper: "A Toolset for Predicting Performance of Legacy Real-Time Software Based on the RAST Approach". *ACM Trans. Model. Comput. Simul.* 35, 3, Article 21 (May 2025), 8 pages. https://doi.org/10.1145/3715113

1 Introduction

In Reference [3], the authors present a **Regression Analysis, Simulation, and Load Testing (RAST)** approach for predicting the performance of legacy software systems by exploiting only access logs. The authors show the feasibility and effectiveness of their approach by using the TeaStore [4] application as use-case by training two Machine Learning models, based on decision trees and ridge regressors, and using them to build simulators. Finally, each simulator can be configured to apply or not a correction to the predicted waiting times.

In this report, we discuss the evaluation of the artifacts referenced by the authors to reproduce the results reported in Reference [3], namely, Figures 8–11 and Tables 3–5, supporting the major claims listed in the following:

- **C1:** RAST is able to produce ridge and decision tree regression models for TeaStore with R^2 scores higher than 0.9.
- **C2:** Regardless of the model and correction value used, the similarity values generally decrease as the load intensity increases.
- **C3:** Performing the correction loop in the Simulator leads to lower similarity values.

Authors' Contact Information: Emilio Incerto, IMT School for Advanced Studies Lucca, Lucca, Italy; e-mail: emilio.incerto@imtlucca.it; Romolo Marotta, DICII, University of Rome Tor Vergata, Roma, Italy; e-mail: romolo.marotta@gmail.com.

ACM 1558-1195/2025/05-ART21
https://doi.org/10.1145/3715113

C4: The ridge regression model outperforms the decision tree model, on average, in terms of similarity values across all request types.

2 Artifact Evaluation

The artifact can be downloaded from GitHub[1] and Zenodo [1]. Given that the artifact [1]:

— has a proper license (BSD 3-Clause);
— has an associated DOI and permanent availability;
— contains data and source code relevant to the article;

the **Artifacts Available** badge can be assigned.

The evaluation discussed in the following is related to the revision 1.5.0 of the artifact [2].

2.1 Software Installation

The dependencies of the artifact are well-known software packages (e.g., GCC, CMake, Curl, Screen, Python, Docker) that are publicly available and can easily be installed on all major Linux distributions.

2.2 Quality of the Artifact

The artifact includes a file (`/docs/SIGSIM_PADS_2024/README.md`) providing details about:

— software dependencies;
— artifact inventory;
— instructions to run experiments and to reproduce tables and figures.

In addition, it provides:

— expected running times;
— list of software/hardware used by the authors,
— summary of the major claims of the article.

Finally, the artifact contains data used to generate figures and tables contained in the article.

The instructions allow to launch experiments both manually (in a step-by-step fashion) or fully automated. Generating figures and tables requires minimal human interaction.

Given that:

— the artifact is well-documented;
— the artifact is relevant to the associated article;
— the provided scripts and instructions allows to run smoothly the experiments (within the expected running times) and generate all figures and tables without errors;

the **Artifacts Evaluated—Functional** badge can be assigned.

2.3 Replicating the Experiments

The article embeds 5 tables and 11 figures, but only the ones listed in the following report experimental results:

— Figures 8–11;
— Tables 3–5.

The test-bed platforms for both the original and reproduced experimental evaluation are described in the following table. Furthermore, to stress the experimental infrastructure under different hardware/software configurations, we repeated the experiments on two completely different execution environments, as detailed in Table 1.

[1] https://github.com/jtpgames/RAST

Table 1. Hardware and Software Specifications of the Original and Artifact Evaluation Environments

		Article	Artifact Evaluation (A)	Artifact Evaluation (B)
Host	CPU	Intel Core i5-1135G7	Intel Core i7-13700	AWS EC2 c5a.4xlarge
	RAM	16 GB	32 GB	32 GB
	OS	Ubuntu 22.04 X64	Ubuntu 22.04 X64	
	Hypervisor	QEMU	VirtualBox V7.0.18	Xen
Guest	CPU	6 vCPUs	8 vCPUs	16 vCPUs
	RAM	8 GB	16 GB	32 GB
	OS	Ubuntu 22.04 X64	Ubuntu 22.04 X64	Ubuntu 24.04 X64

Table 3. R^2 Scores of Ridge Regression and Decision Tree Estimators for Both the Reproducing Environments (A) and (B) as Described in Table 1

Estimator	R^2 Score – Original	R^2 Score – (A)	R^2 Score – (B)
Ridge Regression	0.9642	0.9689	0.9287
DecisionTree Regression	0.9658	0.9814	0.9502

The scripts contained in the artifact allow to run three experiments. The first two allow to obtain training and validation data. The third experiment runs the simulators built on top of ML models trained by using the results from the first experiment. A final processing step produces results for the similarity analysis between prediction of the generated simulators and validation data. During this evaluation, we used the following script:

— `/Automation/setup_and_create_all_data_for_similarity_comparison.sh`

which executes all the aforementioned steps after ensuring that all (non-system) dependencies have been installed. Once the script completes, the results can be easily used to reproduce figures and tables by using the spreadsheet provided by the authors.

The reproduced Table 3 fully supports Claim C1, namely, the ML-models achieve R^2 scores higher than 0.9. Similarly, Tables 5 and 5 are coherent with Claim C2, showing that the Ridge model outperforms the **Decision Tree (DT)** model, on average.

Figures 8–11 show that the similarity decreases with increased load intensity for most of the cases when no correction is applied ($corr_{max} = 0$) and for all cases when correction is applied ($corr_{max} = 1$). This verifies Claim C3.

Finally, Figures 8–11 and Tables 5–5 show that adopting the correction ($corr_{max} = 1$) leads to lower similarity values between predicted and validation data, supporting Claim C4.

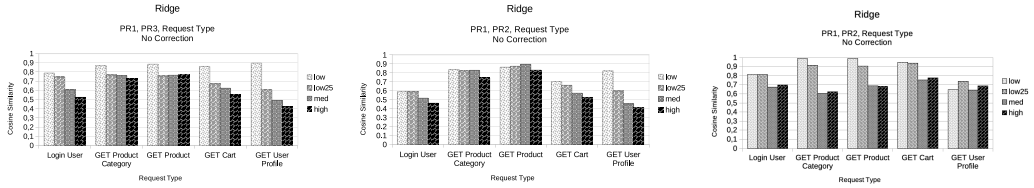
Considering that:

— all the major claims of the article have been supported by reproduced results;
— both reproduced figures and tables show trends comparable w.r.t. original results;

the **Results Validated—Results Reproduced** badge can be assigned.

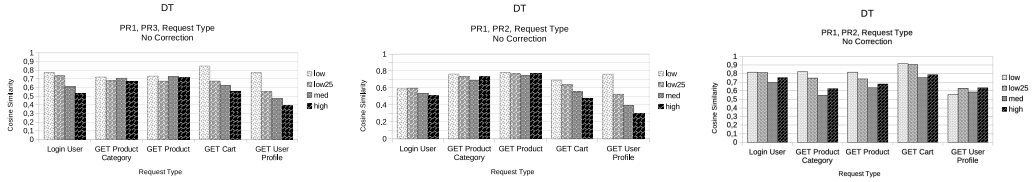
3 Conclusions

Given that we were successful in reproducing the results from the article [3], the badge *Results Validated - Results Reproduced* can be assigned. The code is documented, well structured, and distributed with multiple use cases, justifying the badge *Artifacts Evaluated - Functional*. While the package provides guidelines for application to other systems, the process remains complex, challenging broader applicability. Therefore, the *Artifacts Evaluated - Reusable* badge cannot be assigned. Finally, since the main contribution of the article is available in permanent repositories [1, 2], the *Artifacts Available* badge can also be assigned.



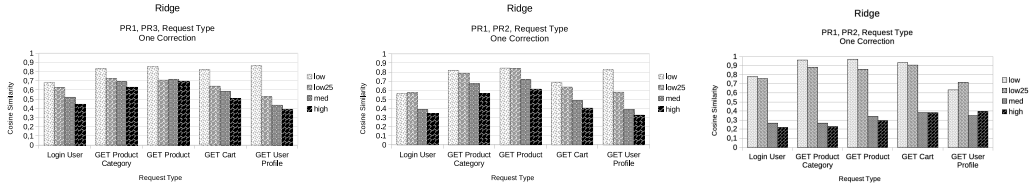
(a) Original. (b) Reproduced (A) (c) Reproduced (B)

Fig. 8. Similarity values with $corr_{max} = 0$ for Ridge Regression ranges: 0.7881–0.8962 for low, 0.6066–0.7669 for low25, 0.4898–0.7640 for medium, and 0.4271–0.7772 for high intensity profile.



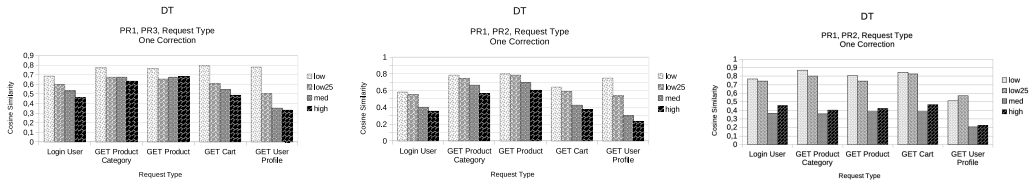
(a) Original. (b) Reproduced (A). (c) Reproduced (B).

Fig. 9. Similarity values with $corr_{max} = 0$ for Decision Tree (DT) Regression ranges: 0.7199–0.8469 for low, 0.5479–0.7349 for low25, 0.4680–0.7223 for medium, and 0.3888–0.7139 for high intensity profile.



(a) Original. (b) Reproduced (A). (c) Reproduced (B).

Fig. 10. Similarity values with $corr_{max} = 1$ for Ridge Regression ranges: 0.6817–0.8649 for low, 0.5247–0.7233 for low25, 0.4332–0.7146 for medium, and 0.3889–0.6956 for high intensity profile.



(a) Original. (b) Reproduced (A). (c) Reproduced (B).

Fig. 11. Similarity values with $corr_{max} = 1$ for Decision Tree (DT) Regression ranges: 0.6847–0.7950 for low, 0.5000–0.6731 for low25, 0.3527–0.6740 for medium, and 0.3318–0.6786 for high intensity profile.

Table 4. Comparison between Ridge and DT Similarity for All Request Types ($corr_{max} = 0$)

Original:				
Load Intensity Profile	Request Type	Similarity Ridge	Similarity DT	Difference
Low	Login User	0.7881	0.7725	2.02%
	GET Product Category	0.8677	0.7199	20.53%
	GET Product	0.8834	0.7302	20.98%
	GET Cart	0.8598	0.8469	1.52%
	GET USER Profile	0.8962	0.7709	16.25%
Low25	Login User	0.7457	0.7349	1.47%
	GET Product Category	0.7669	0.6752	13.58%
	GET Product	0.7565	0.6719	12.59%
	GET Cart	0.6721	0.6731	−0.15%
	GET USER Profile	0.6066	0.5479	10.71%
Med	Login User	0.6069	0.6049	0.33%
	GET Product Category	0.7600	0.7059	7.66%
	GET Product	0.7640	0.7223	5.77%
	GET Cart	0.6230	0.6214	0.26%
	GET USER Profile	0.4898	0.4680	4.66%
High	Login User	0.5231	0.5257	−0.49%
	GET Product Category	0.7297	0.6708	8.78%
	GET Product	0.7772	0.7139	8.87%
	GET Cart	0.5580	0.5557	0.41%
	GET USER Profile	0.4271	0.3888	9.85%
Average		0.7051	0.6560	7.48%
Reproduced (A):				
Load Intensity Profile	Request Type	Similarity Ridge	Similarity DT	Difference
Low	Login User	0.5898	0.5897	0.02%
	GET Product Category	0.8376	0.7643	9.59%
	GET Product	0.8625	0.7799	10.59%
	GET Cart	0.7033	0.6927	1.53%
	GET USER Profile	0.8217	0.7620	7.83%
Low25	Login User	0.5895	0.5975	−1.34%
	GET Product Category	0.8234	0.7337	12.23%
	GET Product	0.8731	0.7665	13.90%
	GET Cart	0.6606	0.6361	3.84%
	GET USER Profile	0.5979	0.5249	13.91%
Med	Login User	0.5174	0.5345	−3.20%
	GET Product Category	0.8277	0.6900	19.96%
	GET Product	0.8949	0.7424	20.54%
	GET Cart	0.5728	0.5529	3.60%
	GET USER Profile	0.4529	0.3929	15.29%
High	Login User	0.4622	0.5108	−9.52%
	GET Product Category	0.7431	0.7361	0.96%
	GET Product	0.8314	0.7716	7.75%
	GET Cart	0.5252	0.4782	9.82%
	GET USER Profile	0.4128	0.2941	40.40%
Average		0.6800	0.6275	8.36%

(Continued)

Table 4. Continued

Reproduced (B):				
Load Intensity Profile	Request Type	Similarity Ridge	Similarity DT	Difference
Low	Login User	0.8132	0.8168	−0.43%
	GET Product Category	0.9873	0.8187	20.59%
	GET Product	0.9878	0.8170	20.91%
	GET Cart	0.9480	0.9160	3.5%
	GET USER Profile	0.6476	0.5542	16.85%
Low25	Login User	0.8104	0.8121	−0.20%
	GET Product Category	0.9122	0.7451	22.42%
	GET Product	0.9032	0.7399	22.06%
	GET Cart	0.9358	0.9045	3.4%
	GET USER Profile	0.7400	0.6300	17.46%
Med	Login User	0.6744	0.6931	−2.69%
	GET Product Category	0.6039	0.5466	10.48%
	GET Product	0.6888	0.6326	8.88%
	GET Cart	0.7514	0.7501	0.18%
	GET USER Profile	0.6404	0.5843	9.59%
High	Login User	0.6985	0.7508	−6.96%
	GET Product Category	0.6219	0.6254	−0.57%
	GET Product	0.6810	0.6760	0.74%
	GET Cart	0.7755	0.7879	−1.57%
	GET USER Profile	0.6865	0.6329	8.46%
Average		0.7754	0.7217	7.44%

Table 5. Comparison between Ridge and DT Similarity for All Request Types ($corr_{max} = 1$)

Original:				
Load Intensity Profile	Request Type	Similarity Ridge	Similarity DT	Difference
Low	Login User	0.6817	0.6847	−0.44%
	GET Product Category	0.8336	0.7703	8.22%
	GET Product	0.8545	0.7616	12.20%
	GET Cart	0.8229	0.7950	3.51%
	GET USER Profile	0.8649	0.7785	11.10%
Low25	Login User	0.6266	0.5970	4.96%
	GET Product Category	0.7233	0.6731	7.46%
	GET Product	0.7064	0.6550	7.85%
	GET Cart	0.6403	0.6041	5.99%
	GET USER Profile	0.5247	0.5000	4.94%
Med	Login User	0.5188	0.5310	−2.30%
	GET Product Category	0.6900	0.6740	2.37%
	GET Product	0.7146	0.6731	6.17%
	GET Cart	0.5843	0.5458	7.05%
	GET USER Profile	0.4332	0.3527	22.82%
High	Login User	0.4433	0.4578	−3.17%
	GET Product Category	0.6294	0.6329	−0.55%
	GET Product	0.6956	0.6786	2.51%
	GET Cart	0.5074	0.4858	4.45%
	GET USER Profile	0.3889	0.3318	17.21%
Average		0.6442	0.6091	5.76%

(Continued)

Table 5. Continued

Reproduced (A):				
Load Intensity Profile	Request Type	Similarity Ridge	Similarity DT	Difference
Low	Login User	0.5641	0.5829	−3.22%
	GET Product Category	0.8174	0.7838	4.30%
	GET Product	0.8429	0.7987	5.53%
	GET Cart	0.6860	0.6415	6.94%
	GET USER Profile	0.8240	0.7485	10.09%
Low25	Login User	0.5715	0.5529	3.36%
	GET Product Category	0.7858	0.7450	5.46%
	GET Product	0.8383	0.7844	6.87%
	GET Cart	0.6358	0.5880	8.13%
	GET USER Profile	0.5784	0.5362	7.87%
Med	Login User	0.3882	0.4012	−3.26%
	GET Product Category	0.6702	0.6657	0.67%
	GET Product	0.7190	0.6966	3.22%
	GET Cart	0.4844	0.4262	13.65%
	GET USER Profile	0.3852	0.3001	28.35%
High	Login User	0.3425	0.3557	−3.72%
	GET Product Category	0.5642	0.5611	0.56%
	GET Product	0.6136	0.6048	1.44%
	GET Cart	0.4048	0.3742	8.18%
	GET USER Profile	0.3278	0.2338	40.24%
Average		0.6022	0.5691	5.82%
Reproduced (B):				
Load Intensity Profile	Request Type	Similarity Ridge	Similarity DT	Difference
Low	Login User	0.7816	0.7690	1.63%
	GET Product Category	0.9622	0.8715	10.41%
	GET Product	0.9702	0.8073	20.17%
	GET Cart	0.9315	0.8463	10.07%
	GET USER Profile	0.6311	0.5107	23.58%
Low25	Login User	0.7549	0.7439	1.49%
	GET Product Category	0.8817	0.8016	9.99%
	GET Product	0.8605	0.7435	15.74%
	GET Cart	0.9062	0.8266	9.63%
	GET USER Profile	0.7149	0.5701	25.40 %
Med	Login User	0.2657	0.3663	−27.46%
	GET Product Category	0.2655	0.3581	−25.86%
	GET Product	0.3412	0.3842	−11.19%
	GET Cart	0.3845	0.3826	0.52%
	GET USER Profile	0.3487	0.2083	67.43%
High	Login User	0.2189	0.4581	−52.21%
	GET Product Category	0.2316	0.4037	−42.63%
	GET Product	0.2958	0.4234	−30.15%
	GET Cart	0.3800	0.4683	−18.85%
	GET USER Profile	0.3978	0.2276	74.75%
Average		0.5762	0.5585	3.16%

Acknowledgments

The authors would like to thank Juri Tomak for his help in resolving all the issues encountered during the evaluation and for his willingness and dedication to further improve the originally submitted artifact.

References

[1] Juri Tomak. 2024. *A Toolset for Predicting Performance of Legacy Real-Time Software Based on the RAST Approach - Artefact Submission.* DOI: https://doi.org/10.5281/zenodo.11266812

[2] Juri Tomak. 2024. *A Toolset for Predicting Performance of Legacy Real-Time Software Based on the RAST Approach - Artefact Submission.* DOI: https://doi.org/10.5281/zenodo.14145393

[3] Juri Tomak and Sergei Gorlatch. 2024. A toolset for predicting performance of legacy real-time software based on the RAST approach. *ACM Trans. Model. Comput. Simul.* (June 2024). DOI: https://doi.org/10.1145/3673897 Just Accepted.

[4] Jóakim von Kistowski, Simon Eismann, Norbert Schmitt, André Bauer, Johannes Grohmann, and Samuel Kounev. 2018. TeaStore: A micro-service reference application for benchmarking, modeling and resource management research. In *IEEE 26th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS'18).* IEEE Computer Society, 223–236. DOI: https://doi.org/10.1109/MASCOTS.2018.00030

Received 29 November 2024; revised 29 November 2024; accepted 14 January 2025