Toward Long-term Computational Reproducibility: Assessing the Archival Rate of URLs to Git Hosting Platforms in Scholarly Publications

Emily Escamilla, Vicky Rampin, Jian Wu, Martin Klein, Michele C. Weigle, and Michael L. Nelson

Keywords: reproducibility, open source software, digital preservation

Extended Abstract

Reproducibility is a foundational principle of scientific research and it is contingent on the availability of the original methodology, including software products and data. Recent studies indicated that a significant fraction of Uniform Resource Locators (URLs) linking to open-access datasets or software that claimed to be available are no longer accessible, e.g., [1]. This situation undermines the FAIR (findable, accessible, interoperable, reusable) Guiding Principles for scientific data management and stewardship and has become a hurdle in verifying published results. While major archiving efforts to preserve conventional scholarly content, typically in PDFs are underway, few analogous efforts have yet emerged to preserve the open-source software references in those PDFs, particularly the code hosted on Git Hosting Platforms (GHPs), such as GitHub, GitLab, Bitbucket, and SourceForge. This project focuses on studying the current archival efforts to preserve software products in scholarly publications and reveal the technical and policy gaps to better preserve software products using archival services toward long-term computational reproducibility.

This project answers the following two research questions (RQs). RQ1: For GHP URLs identified in scholarly articles, what is the prevalence of these GHP URLs on the live Web and archival services? RQ2: Does popularity as measured by user engagement affect the likelihood of a GitHub repository being archived?

Existing work focused on *general* URLs used in scholarly publications. For example, Klein et al. [2] found that the number of general URLs used in scholarly publications rapidly increased from 1997 to 2012, but nearly 20% of Science, Technology, and Medicine publications contain reference rot, i.e., the link is no longer alive or the content has drifted away from the original content. Understanding the scope of how scholarly source code is represented in scholarly publications is vital to strengthening efforts to preserve the code and make it available for the long term as part of the scholarly record.

To this end, we analyzed a total of about 2.64 million scholarly publications collected from arXiv (1.56 million) and PubMed Central (PMC; 1.08 million) corpora, published between 2007 and 2021, as a representative sample of scholarly publications across Science, Technology, Engineering, and Math disciplines. Our data preprocessing includes extracting URLs from PDF text and using regular expressions to filter and categorize URLs that reference one of the four GHPs. This step resulted in 253,590 URLs to one of the four GHPs (231,206 URLs from arXiv and 22,384 URLs from PMC).

The web archival services we used include Software Heritage (SH) and Web archive. SH is a non-profit organization that provides a service for archiving and referencing historical and contemporary software. However, in the current implementations source codes and their ephemera, such as issues, pull requests, and wikis are not preserved. These ephemera are outside the scope of the GIT version control system but provide important context to the code, which presents a problem for scholarly projects where reproducibility matters. For the Web archives, instead of using a single archival service such as Internet Archive, we use MemGator, an API that requests a given URL from 12 distinct Web archives, such as the Internet Archive, Library of Congress, and Stanford Web Archive. MemGator compiles all the archives' responses into a TimeMap that includes mementos at each datetime. Because Web archives capture more than just source code, mementos can be created for both the software product and the surrounding ephemera to allow users a complete picture of the hosted repository.

To answer RQ1, we tested whether the URLs are available on the live Web, in SH, or in Web archives. We define a GHP URL resource as *vulnerable* if it is publicly available on the live Web but has not been archived. We define a GHP URL resource as *unrecoverable* if it is no longer publicly available on the live Web and has not been archived. Across all four GHPs, 93.98% of all GHP URLs were on the live Web, and 6.02% unique GHP URLs were rotten. We found 32.48% of all repository-level GHP URLs do not have any snapshots in SH and 19.57% of GHP URLs do not have any snapshots in Web archives. We found that 13.26% (16,456) active URLs in our corpus were not captured by either SH or the Web archives making their resources *vulnerable*. For rotten URLs, 33.35% have not been archived by SH or Web archives, making their resources *unrecoverable*.

To answer RQ2, we analyzed the relationship between user engagement and archival. We focus on GHP URLs to GitHub alone because they take the majority (94%) of all GHP URLs in our corpus. We used GitHub API to extract engagement metrics, including *forks*, *subscribers*, and *stargazers*. Our analysis used the Mann-Whitney U test, which compares whether the distributions of two populations are different, and Cliff's Delta, which measures the effect size for each of the three engagement metrics and both archives. We found a statistically significant difference between the engagement metrics for GitHub repositories that have and have not been archived in Web archives and SH. We also found that for GitHub repositories in SH, there is a *stronger* correlation between the popularity of the repository and its archival than for Web archives. The results reveal that smaller, less popular repositories are at increasing risk of being vulnerable, and eventually, unrecoverable.

The White House Office of Science and Technology Policy issued a guidance in 2022 to make federally funded research freely available without delay. Our research envisions the importance of future policies on preserving software products for long-term computational reproducibility. Ideally, the archival of URLs to software products included in scholarly publications would be required by institutions and publishers to expand upon current open-access requirements. Future work should create a workflow that extracts software URLs and archives the software product and surrounding ephemera upon submission to guarantee the long-term preservation of the software products used and created by authors.

References

[1] K. Ajayi, M. H. Choudhury, S. M. Rajtmajer, and J. Wu. A study on reproducibility and replicability of table structure recognition methods. In *Document Analysis and Recognition - ICDAR 2023*, pages 3–19, Cham, 2023. Springer Nature Switzerland.

[2] M. Klein, H. Van de Sompel, R. Sanderson, H. Shankar, L. Balakireva, K. Zhou, and R. Tobin. Scholarly context not found: One in five articles suffers from reference rot. *PLOS ONE*, 9(12):1–39, 12 2014.