

URL Rot in Software Engineering Scholarly Works

APOSTOLOS GAROS, Athens University of Economics and Business, Greece

DIOMIDIS SPINELLIS, Athens University of Economics and Business, Greece

STEFANO ZACCHIROLI, Institut Polytechnique de Paris, France

Context: Citations play an important role in scholarly publishing. They increasingly reference Web-based resources via Uniform Resource Locators (URLs). However, URLs are prone to decay over time, threatening the value and integrity of scientific works.

Objective: We investigate the persistence of URLs in software engineering publications, analyzing their stability over time, by hosting purpose, protocol, venue, document placement, and other attributes.

Method: We extract nearly 144 thousand URLs from over 50 thousand papers published in major software engineering venues indexed in DBLP, spanning the period 1971–2024. Using automated tools and manual verification, we assess URL accessibility, classify hosts, and conduct regression analyses on several potential correlates of URL stability.

Results: Overall, 71% of URLs remain active, with an estimated half-life of 11 years—substantially higher than reported in prior studies. URLs in version control systems (89% active) and archival platforms (94% active) exhibit significantly higher persistence than other Web links (71% active). HTTPS URLs (82% active) are more stable than HTTP ones (60% active), possibly due to their relative recency. Variations in URL stability exist across venues, document sections, and domains; however, regression analysis shows no strong correlation between stability and factors such as venue prestige.

Conclusion: URL stability in software engineering literature is better than that reported in earlier studies. However, substantial URL decay persists. Increased reliance on archival services can further mitigate this threat. Periodic large-scale studies remain necessary to monitor evolving patterns of URL persistence in scholarly publishing.

CCS Concepts: • **Information systems** → **Information retrieval**; **Evaluation of retrieval results**; • **Software and its engineering** → *Backup procedures*.

Additional Key Words and Phrases: URL, digital preservation, scientific articles, replicability, reproducibility, software engineering

ACM Reference Format:

Apostolos Garos, Diomidis Spinellis, and Stefano Zacchiroli. 2026. URL Rot in Software Engineering Scholarly Works. *ACM Trans. Softw. Eng. Methodol.* X, Y, Article Z (October 2026), 29 pages. <https://doi.org/10.1145/3849476>

1 INTRODUCTION

Citations are a key element of scholarly publishing, used, among other purposes, to derive operational information, to enhance persuasiveness, to provide positive or negative credit, to establish a work's currency, to alert readers, and to establish social consensus [1, 6]. Increasingly scholarly publications reference internet resources, which are typically available via the Web [9]. As Web

Authors' addresses: Apostolos Garos, apostolisgaros@gmail.com, Department of Management Science and Technology, Athens University of Economics and Business, Athens, Greece; Diomidis Spinellis, dds@aub.gr, Department of Management Science and Technology, Athens University of Economics and Business, Athens, Greece; Stefano Zacchiroli, stefano.zacchiroli@telecom-paris.fr, Institut Polytechnique de Paris, LTCI, Télécom Paris, Palaiseau, France.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

© 2026 Copyright held by the owner/author(s).

ACM 1049-331X/2026/10-ARTZ

<https://doi.org/10.1145/3849476>

URLs decay over time [17], publications end up citing information that is no longer easily available, if at all [4, 5, 7, 14, 16, 23, 24, 26, 28, 30–32, 34–37]. URL decay is a problem, because it compromises the original publication’s integrity: readers may not be able to compare the publication against the state of the art, to read complementary information, or to put the work in its originally intended context. At worst, the published research may no longer be reproducible or repeatable. It is therefore important to understand the decay’s quantitative and qualitative characteristics. These involve the phenomenon’s extent and evolution as well as the characteristics of failing and surviving URLs and the venues that publish them. Corresponding data allow us as a community to design and deploy policies and technologies that increase the cited URLs’ robustness.

With this context in mind, we formulated the following research questions:

RQ 1. Temporal evolution. How has the share of *active* URLs in software-engineering papers changed over time, and what half-life best describes that decay?

RQ 2. Hosting purpose. How does URL accessibility differ among three host classes—(i) general Web hosts, (ii) development forges (e.g., GitHub, GitLab), and (iii) long-term archival services (e.g., Zenodo, arXiv, Software Heritage)?

RQ 3. Protocol impact. Does using the more brittle https scheme (vs. http) decrease the likelihood that a referenced URL is still reachable?

RQ 4. Venue & placement. How does URL accessibility vary (a) across publication venues (conferences vs. journals) and (b) across document sections (abstract, body, references)?

RQ 5. Technical correlates. Which intrinsic URL features (length, number of path elements, top-level domain, scheme) and extrinsic paper features (CORE ranking, journal impact factor) correlate with URL persistence?

Exploratory nature. The nature of this study is descriptive and exploratory. Our goal is to provide a large-scale empirical characterization of URL persistence in software engineering publications and to examine associations between URL survival and observable attributes such as time, hosting purpose, protocol, venue, document placement, and basic URL features. We did not formulate or preregister *a priori* hypotheses. The analyses presented in this paper are, unless otherwise stated, correlational rather than causal. The results should therefore be interpreted as mapping patterns in the data and as generating hypotheses for future, confirmatory work.

Methodologically, this work is a critical review of software engineering publications focused on the persistence of Web-based references. In terms of URL sources, we focus on major software engineering publication venues (both journals and conferences), without selecting specific subfields; this way our findings provide a broad overview of URL rot in software engineering publishing.

The remainder of the paper details the employed methods, presents empirical results, discusses their implications, and summarizes threats to validity before concluding.

2 RELATED WORK

The persistence of URLs used in research papers is the subject of several prior studies that examined their stability and permanence over time. These studies provide valuable insights into the factors that can affect the half-life of URLs and the implications of using them in scholarly works. We summarize the findings of previous studies in Table 1 and discuss them more in detail in the remainder of this section.

Most previous studies were conducted and published much earlier than this one. The state of the Web and its use in scholarly work has evolved significantly since. It is hence interesting and

Table 1. Prior studies on URL persistence: characteristics and findings

Authors	Year	Papers	URLs	Research field	Half-life (years)	Inactive URLs (%)
Koehler [23]	1999	-	361	Cross-field	2	14.8%
Lawrence et al. [26]	2000	100 826	67 577	Computer science	-	23–53%
Germain [16]	2000	31	64	Cross-field	3	51.5%
Casserly and Bird [7]	2003	1425	500	Library and information sciences	-	18.6%
Spinellis [32]	2003	2471	4224	Computer science	4	28%
Wren [35]	2004	-	1630	Biomedicine	-	22%
Koehler [24]	2004	-	361	Cross-field	-	66.2%
McCown et al. [30]	2005	453	4387	Cross-field	10	30%
Wren et al. [37]	2006	7337	1113	Dermatology	-	18.3%
Wren [36]	2008	-	7462	Biomedicine	-	20%
Wagner et al. [34]	2009	-	2011	Healthcare management	-	49.3%
Saberi and Abedi [31]	2012	748	3734	Social sciences	8.9	27%
Loan and Shah [28]	2020	221	358	Library and information sciences	-	32.1%
Bansal and Parmar [5]	2020	1564	1724	Cross-field	1.76	43.3%
Bansal [4]	2021	273	1921	Library and information sciences	6.5	23.31%
Escamilla et al. [14]	2023	2 641 041	253 590	Cross-field	-	6.02%
<i>This study</i>	2025	51 116	144 403	Software engineering	10.66	29.36%

important to look at previous works and consider them as baselines to compare against, in order to understand how URL rot is evolving over time and across fields.

Koehler [23, 24] was one of the earliest authors to perform an extensive URL persistence study, measuring a URL half-life of approximately 2 years. The studies by Koehler have been cited by numerous researchers in the field and have contributed significantly to the understanding of URL half-life.

In 2000, Lawrence et al. [26] checked 67 577 URLs extracted from 100 826 computer science research articles from the CiteSeer database. The percentage of invalid URLs differs based on the year of article publication, peaking at 53% for papers published in 1994. After applying a number of correction techniques, including using search engines and manually fixing URLs, they were able to find the new locations of broken references, resulting in around 2.9% manually fixable URLs. A similar approach was used by Casserly and Bird [7], who categorize Web references as “permanent” and “impermanent”, based on whether URLs led to the cited content (or a page that redirects to that content). After further investigating and reevaluating “impermanent” resources, they obtain an increase of resolvable references from 56.4% to 81.4%.

Germain [16] had a different view on whether a reference is considered persistent or not. Only minimal corrections in URLs syntax were allowed and, contrary to the studies mentioned before, only links to the referenced files (not host sites) were considered valid. Their study reports a half-life of 3 years, having tested 64 Web references collected from 31 randomly chosen academic journal articles.

An even stricter approach was used by Spinellis [32] in 2003. He identified 4224 URLs, extracted from computer science research papers obtained from the IEEE Computer Society and the ACM digital libraries, as incorrect due to syntax or lexicographical errors. Additionally, the only criterion to classify them as valid or not was their ability to be used to download the referenced resource. The content of the downloaded material was not further inspected. In the end, the study reports an average URL half-life of 4 years.

In this work we adopt the approach of Spinellis [32], where URL checking is performed automatically at the Internet protocol level, without attempting to manually fix invalid URLs, or to conceptually verify if they point to the same artifact intended by the original authors. This allows scaling the study to a large number of papers/URLs and observing the evolution over a long period.

In 2005, McCown et al. [30] focused their research on articles published in D-Lib Magazine from July 1995 to August 2004. From 453 articles, they extracted 4387 Web references; after extensive and frequent trials through 25 weeks, they concluded that around 30% were unreachable.

Wren [35] in 2004 conducted an extensive analysis of URLs found in biomedicine articles. Using abstracts from MEDLINE, 1630 unique URLs were identified and tested, concluding that approximately 22% were no longer available. In 2006, Wren et al. [37] conducted similar research on URLs found in dermatology journals, reporting that 18.3% of the cited resources were not available. The study of 2004 was replicated in 2008 [36], utilizing a body of 7462 URLs, obtaining analogous outcomes. Wagner et al. [34] in 2009 tested URLs in health care management journals published between 2002 and 2004. Of the original 2011 cited resources, 49.3% were not reachable at the provided URL and stated that “*URL decay is a serious problem in health care management journals*”. Taking stock of these results, we can conclude that this situation shows little improvement as time passes—in health related fields and during the first decade of 2000, at least.

In 2020, Loan and Shah [28] tested 358 Web citations present in articles published between 2007 and 2011 in the Journal of Informetrics and found that 32.12% URLs were unavailable. The same year, Bansal and Parmar [5] studied 1724 URLs, cited in papers published during 2015–2016 in the Current Science Journal. They report a 43.33% inaccessibility rate and an estimated half-life of 1.76 years. Bansal [4] in 2021, found a longer half-life of 6.5 years, having studied 1921 URLs from articles included in the DESIDOC Journal of Library & Information Technology between 2014–2018. What these three studies have in common is that they tested Web references from more recent publications (about one decade fresher), compared to previous works. After examining their findings, it emerges that there has been progress throughout the years: Loan and Shah [28] found a 67.88% accessibility rate 9–13 years after the articles were published and Bansal [4] found a significant increase in half-life with respect to older studies.

Extending this investigation to the realm of software and code references, Escamilla et al. [14] analyzed in 2023 the status of Git Hosting Platforms (GHP) URIs in scholarly articles, analyzing over 250 thousand URLs from more than 2.6 million papers. They discovered that 93.98% of GHP URIs remained active, indicating better results than other (possibly older) kinds of Web references.

Comparison with previous work. In this work we focus on the field of software engineering, analyzing more than 140 thousand URLs extracted from more than 50 thousand scholarly papers from journals and conferences in the field, over a period of more than 50 years. In comparison to previous works our study is field-specific (narrower than some, analogous to others) and large-scale—larger than most previous works in terms of both papers and URLs analyzed, with the exception of two large-scale studies [14, 26]. We also analyze more thoroughly the causes of URL rot, looking at network, protocol, domain, and venue-related causes.

3 METHODS

Figure 1 shows the experimental method followed to obtain papers, URLs in them, URL liveness, and subsequent analyses. We detail below its main steps. Where relevant, the figure reports the number of items obtained after each step, e.g., number of article entries extracted from DBLP, papers downloaded in PDF form, etc.

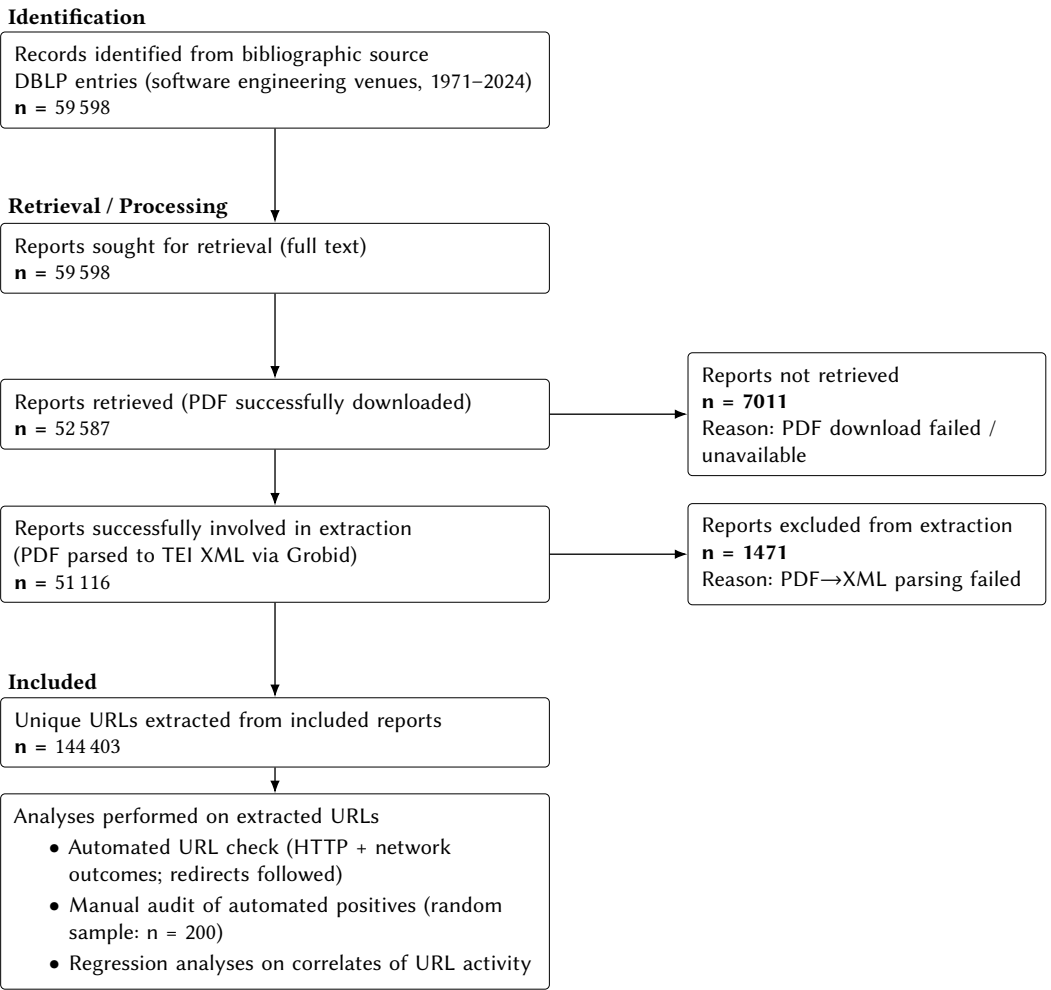


Fig. 1. Experimental method

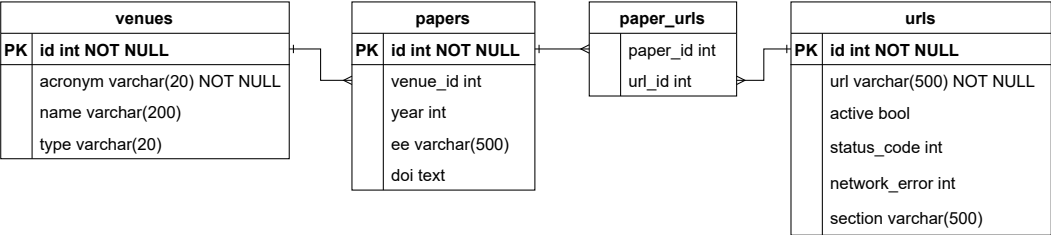


Fig. 2. Database schema

3.1 Data storage

We used a relational database—specifically PostgreSQL—as main data storage for conducting this study. The database schema is shown in Figure 2. The replication package [2] of this study contains a complete database dump.

The main table is `papers`, with one entry for each paper considered. The `paper_urls` table associates URLs to papers they were found in. `urls` contains one entry for each URL we encountered, together with status information about its verification: whether it was active or not and relevant errors if not. Note that we also store the specific paper section a URL was found in, allowing to include/exclude URLs based on that. `venues` contains information about the scientific venue (conference or journal) in which papers were published.

3.2 Data gathering

The data-gathering process for this study was structured into several steps, inspired by the methods adopted by Khalil and Zacchiroli [22]. Each step relies on specific tools and technologies, detailed in the following.

DBLP extraction. We rely on DBLP [27], the reference bibliographic database for computer science publications, as initial data source. We retrieve from it the most recent data dump available at the time of analysis (2025-02-01). The dump comes in the form of a `dblp.xml` file, an XML document containing papers metadata such as titles, authors, publication venues, and electronic edition (“ee”) links associated with computer science publications. As our study focuses on software engineering venues, we selected relevant venues indexed by DBLP following the selection established by Khalil and Zacchiroli [22], as updated in its currently released code.

Table 2 summarizes the chosen venues, detailing their acronyms, full names, and type (conference or journal). The papers analyzed were published between 1971 (the earliest year with publications in the field) and 2024 (the latest complete year at the time of data collection).

In total we identified 59 598 software engineering papers to analyze further.

PDF download. We use a customized version of the `doi2pdf` tool [11] to download all papers from selected venues and years, in PDF format. `doi2pdf` is a Python-based tool for automatically retrieving academic papers, starting from DOI identifiers. We modified `doi2pdf` to enhance its download capabilities, specifically by supporting trying several mirrors in case of failure of the primary source, as well as using arXiv to find “preprint” versions of papers in case they cannot be retrieved from the primary source. The customized version of `doi2pdf` is included in the replication package [2].

We managed to retrieve 52 587 papers (90% of the papers selected from DBLP).

XML parsing. We then parse the retrieved PDF files into XML using Grobid (GeneRation Of Bibliographic Data) [29]. Grobid produces a TEI [33]-based XML output, where relevant document sectioning information is preserved, allowing us to later populate the database’s `urls.section` field (see Figure 2). In a detailed evaluation [33] of tools for extracting content from scientific publications, Grobid outperformed all other solutions. Our choice of tool is hence motivated by the need for a sectioning output, its performances, and the fact it is open source, hence easier to replicate in the future.

We were able to parse 51 116 (97% of the retrieved PDF files).

URL extraction. We extract URLs from the XML documents via a custom Python script. The script uses regular expressions to identify and extract URL patterns. (Note that it is not possible to rely only on XML markup to identify URLs, e.g., in hyperlinks, because many PDF papers contain

Table 2. Analyzed venues: conferences (C) and journals (J) in the field of software engineering, indexed by DBLP.

Acronym	Full name	Type
ASE	IEEE/ACM International Conference on Automated Software Engineering	C
ESEM	International Symposium on Empirical Software Engineering and Measurement	C
FASE	Fundamental Approaches to Software Engineering	C
FSE	ACM SIGSOFT Symposium on the Foundations of Software Engineering	C
GPCE	Generative Programming and Component Engineering	C
ICPC	IEEE International Conference on Program Comprehension	C
ICSE	International Conference on Software Engineering	C
ICSM	IEEE International Conference on Software Maintenance	C
ICSME	International Conference on Software Maintenance and Evolution	C
ICST	IEEE International Conference on Software Testing, Verification and Validation	C
ISSTA	International Symposium on Software Testing and Analysis	C
MODELS	International Conference On Model Driven Engineering Languages And Systems	C
MSR	Working Conference on Mining Software Repositories	C
RE	IEEE International Requirements Engineering Conference	C
SANER	IEEE International Conference on Software Analysis, Evolution and Reengineering	C
SCAM	International Working Conference on Source Code Analysis & Manipulation	C
SSBSE	International Symposium on Search Based Software Engineering	C
WCRE	Working Conference on Reverse Engineering	C
ASEJ	Automated Software Engineering	J
ESE	Springer - Empirical Software Engineering	J
IJSEKE	International Journal of Software Engineering and Knowledge Engineering	J
ISSE	Innovations in Systems and Software Engineering	J
IST	Information and Software Technology	J
JSS	Elsevier - Journal of Systems and Software	J
REJ	Requirements Engineering Journal	J
NOTES	ACM SIGSOFT Software Engineering Notes	J
SMR	Journal of Software: Evolution and Process	J
SOSYM	Software and System Modeling	J
SPE	Software: Practice and Experience	J
SQJ	Software Quality Journal	J
STVR	Software Testing, Verification and Reliability	J
SW	IEEE Software	J
TOSEM	ACM - Transactions on Software Engineering Methodology	J
TSE	IEEE - Transactions on Software Engineering	J

URLs in plain text, without any associated markup.) During extraction, we paid special attention to correctly extracting URLs that might be broken for typographic reasons irrelevant to the study goals, such as line breaks or immediately adjacent punctuation characters.

To assess the accuracy of the regular expressions used for URL extraction, we conducted a manual evaluation on a random sample of 100 extracted URLs. Of these, 95 were correctly extracted,

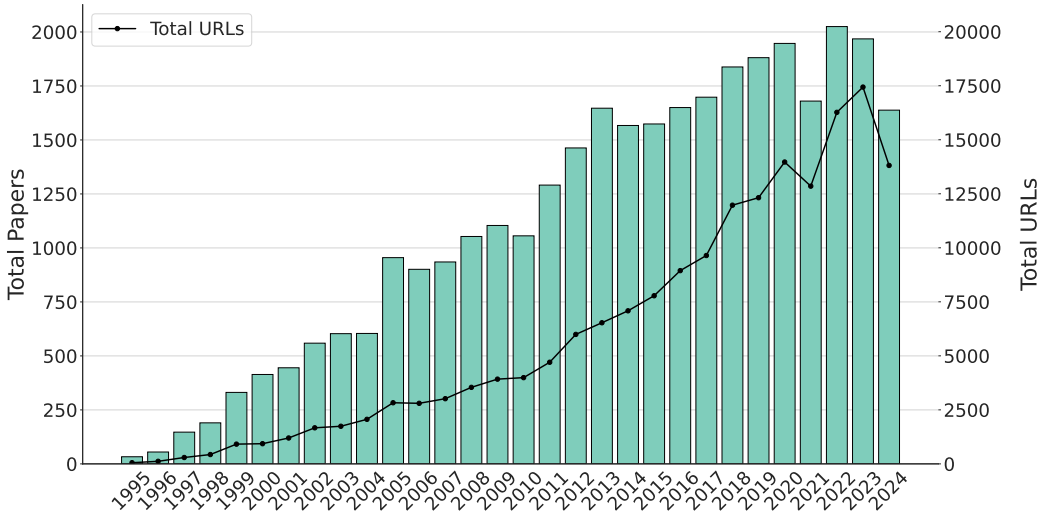


Fig. 3. Papers and URLs over time (1995–2024)

yielding an extraction accuracy of 95%, with the remaining failing mainly due to edge cases involving complex punctuation or formatting artifacts.

Final Dataset. Based on the data collected, we visualize the number of published papers in the field of Software Engineering along with the number of URLs that are referenced in them. In Figure 3 we can see an upward trend in both paper and URL metrics. (The last year is an exception, attributable to the fact that publication databases can lag behind present time and will see the addition of papers for a given year in the near-ish future.) The increase in publications can be an indication of the growing interest on the domain or more generous page budgets as publications abandoned expensive printing in favor of digital distribution. A comparison between the graph’s first and last years indicates a significant growth, with publications increasing by a factor of eight. Simultaneously, the number of URL references exhibits a comparable growth trend, marking a seven-fold increase.

3.3 URL check — automatic verification

We perform two subsequent phases of URL checking: (1) automated verification of all URLs, (2) manual verification of a randomly sampled URL subset. The first phase is a fully automated URL verification aimed at detecting network errors at different protocol levels.

URL parsing. We parse all extracted URLs to ensure that they are syntactically correct, according to the relevant Internet standards (RFC 1738 and successors). This step allows to filter out incorrect URLs that have no chance of being active anyway, before the actual crawling.

Network error detection. All syntactically correct URLs are crawled using the open source tool `curl` and the Python `requests` library (which operated as a fallback).¹ For each URL we try to access it, transparently and recursively following HTTP redirections.

Several unsuccessful outcomes are possible. First, a network error can be encountered at a protocol level lower than HTTP, e.g., failure to resolve a host name or to connect to the relevant

¹<https://curl.se/>, <https://requests.readthedocs.io/>, accessed 2024-02-09.

IP address after resolution. Second, an erroneous HTTP status code, i.e., other than 200, can be obtained (after redirections). We note down the various possibilities in the `urls` table, via the `status_code` and `network_error` columns.

To make the URL verification process robust we incorporate a retry mechanism in the process. For URLs that either do not respond within a 10-second window or encounter other errors, our protocol entails a five-second wait before retrying, a process that can be repeated up to three times. Should these attempts result in three consecutive failures, the URL is then classified as inactive. This avoids reporting as invalid URLs that were only temporarily inaccessible, e.g., due to transient network errors.

Ultimately, we record as *active* any URL for which we could obtain an HTTP 200 response, after all redirections and retries; as *inactive* any other URL.

3.4 Retrieval of inactive URLs from Web archives

Some of the inactive URLs can, in theory, still be available from digital preservation archives. To quantify the potential of this approach to counter URL rot, we verify for presence of archived copies of all inactive URLs in the Wayback Machine service provided by the Internet Archive [21].

For each inactive URL, we query the CDX index API [19] of the Wayback Machine to determine whether the URL has any archived copy (or “snapshot”). When snapshots exist, we identify the earliest capture that corresponds to a successful retrieval (i.e., HTTP status code 200, 301, or 302). To reduce false negatives due to common canonicalization differences, we also probe normalized URL variants (i.e., HTTP vs. HTTPS, with/without `www` prefix, and with/without trailing slashes).

If at least one successful capture is found, we record the URL as having an archived copy. This step refines the characterization of inactive references by distinguishing between URLs whose content may have disappeared entirely and those for which an archived representation remains available. (Note that we do not verify, and hence make no claim about, whether the archived copy actually corresponds to the resource referenced *by paper authors at the time of writing*.)

3.5 URL check — manual verification

The second phase of URL checking is a manual verification of a random sample of 200 URLs, to validate the results obtained by phase 1 (automatic verification). To that end, we manually audit a random sample of 200 URLs. Given that URLs can only be accessed programmatically over the network, we review only those URLs reported as active by our automated checks. The rationale is that if `curl` cannot access a URL, neither can a human, so verifying false negatives is unnecessary; only false positives can occur. The primary goal of this audit is to ensure that URLs marked as active by our automated verification process are indeed considered active by human evaluators. In this context, a URL is considered active if it redirects to content that we deem a plausible destination.

2.5% of the URLs (5 out of 200) audited are inactive—most likely having become inaccessible between our automated verification and the manual audit. Approximately 6.5% of the URLs (13 out of 200) redirected to generic pages or content that did not seem pertinent to the initial URL; these cases represent legitimate false positives that could not be detected automatically at the network protocol level. Overall, about 91% of the URLs we audited were indeed active, which makes us confident in the results obtained by the automated URL verification process.

3.6 Regression analysis

As part of our analysis of the results of URL validity, we conduct several regression analyses to verify if, and which, factors correlate with URL active/inactive status.

Regression analysis is driven by the hypothesis that specific attributes of research papers and URLs in them could significantly influence their longevity and accessibility. All models are *linear*

regressions, fit with ordinary least squares (OLS). For the citation-level model described below, whose outcome is the binary indicator $is_active \in \{0, 1\}$ (whether a cited URL is reachable), this amounts to a linear probability model: we prefer it over logistic regression because its coefficients are directly interpretable as changes in the probability of a URL being active, without transformation through odds ratios, which suits the descriptive and exploratory aims of this study. The model's simplicity is also well-suited for isolating and understanding the impact of individual variables. The statistical implications of applying OLS to a binary outcome are disclosed and assessed together with the regression results (Section 4.7), where we also base inference on heteroskedasticity-robust and clustered standard errors and report a logistic-regression robustness analysis. For the venue-level models described below, the outcome is a continuous quantity, the mean URL success rate per venue, for which linear regression is the standard choice.

We consider three models.

The first operates at the *citation level* and includes the following predictors: URL length in characters, number of "/"-separated path components, paper age (years elapsed since publication), whether the URL uses HTTPS, venue type (conference vs. journal), and the document section in which the URL appears (abstract, citations, or body). We examined variance inflation factors to assess multicollinearity; all predictors exhibited VIF values below 3, indicating no severe collinearity.

The second and third models operate at the *venue level* and examine, respectively, CORE conference rankings² and journal impact factors as predictors of the mean URL success rate per venue. CORE rankings and impact factors are assumed to reflect the quality and prestige of publication venues, possibly inducing better practices in terms of Web references and archival practices. Note that these two models are fit on disjoint subsets of venues, as a venue is either a conference or a journal.

Venue-level models are estimated separately from the citation-level model because CORE ranking and journal impact factor are defined only for mutually exclusive venue types (conferences and journals, respectively) and are measured at the venue level rather than the citation level. Incorporating them directly into the citation-level regression would either require extensive missing-value handling or restrict the analysis to a reduced subset of observations, thereby altering the estimand and conflating levels of aggregation. Accordingly, the citation-level model and the two venue-level models address distinct analytical questions and operate on different units of analysis.

3.7 URL content classification

In addition to measuring whether a URL is reachable, we also characterize what kind of resource each URL references. This distinction allows us to assess whether different classes of scholarly Web references (e.g., papers, software repositories, datasets, documentation) exhibit different stability profiles.

The classification step is applied to all extracted URLs as part of the data processing pipeline. It is performed independently of URL liveness and relies only on information available in the URL string (e.g., host name and path patterns), without crawling the Web or inspecting retrieved content.

Categories. Each URL is assigned to exactly one of the following categories: *PAPER*, *DATASET*, *ARTIFACT*, *SOFTWARE*, *HOME PAGE*, *DOCUMENTATION*, *OTHER*. This taxonomy is designed to separate traditional scholarly references from research artifacts and supporting resources as these are commonly used as stepping stones for further studies [25] and in reproducibility-oriented work.

Rule-based, ordered classification. Classification is performed using a deterministic set of rule-based heuristics. Rules are evaluated in a fixed priority order, and the first matching rule determines

²<https://www.core.edu.au>, accessed 2024-02-09

the assigned category. At a high level, the rules rely on characteristic URL patterns associated with well-known platforms and resource types, such as publisher and preprint servers (for *PAPER*), code hosting and package registries (for *SOFTWARE*), dedicated dataset portals (for *DATASET*), and documentation hosts or documentation paths (for *DOCUMENTATION*). URLs that point to high-level project or resource entry points (such as project homepages, overview pages, or umbrella Web sites that describe a tool, dataset, or initiative without directly providing downloadable artifacts) are classified as *HOMEPAGE*.

Ambiguous cases, particularly those involving mixed repositories that host multiple asset types, are conservatively handled by assigning *ARTIFACT* only when strong bundle-related cues are present. If a URL is not assigned to any of the preceding categories, it defaults to the *OTHER* category.

Persistent identifier URLs and URL shorteners. In addition to semantic content classification, we perform a complementary analysis that captures citation mechanisms rather than resource type. This analysis is orthogonal to the taxonomy above: instead of identifying *what* a URL references, it detects whether authors rely on indirection or preservation infrastructures intended to improve reference longevity.

As with the content classification step, detection relies exclusively on URL-string information (hostname and path structure), without crawling or inspecting retrieved content, and is applied uniformly to the full corpus independently of URL liveness.

URL shorteners. URLs are classified as shortened when their normalized hostname (lowercased and stripped of a leading `www.`) matches a curated list of widely used public shortening services (e.g., `bit.ly`, `tinyurl.com`, `t.co`, `goo.gl`, etc.). This provides a conservative estimate of shortening usage in scholarly references, capturing cases where the cited destination is obscured behind a redirection layer.

Persistent identifier and archival resolvers. We additionally identify URLs that correspond to persistent identifier or preservation-resolution infrastructures. A URL is marked as persistent when it matches resolver patterns associated with established long-term identification or archival systems. The detected categories include DOI resolvers (`doi.org`, `dx.doi.org`), the Handle.Net registry (`hdl.handle.net`), ARK identifiers (dedicated ARK resolvers or paths containing `/ark:/`), Persistent URLs (`pur1.org`), w3id identifiers, Perma.cc archived references, Internet Archive Wayback snapshots, Zenodo record pages, and Software Heritage identifier gateways.

Unlike the semantic taxonomy, these classifications are not mutually exclusive with content categories: a URL may simultaneously belong to a semantic class (e.g., *PAPER* or *SOFTWARE*) and be resolved through a persistent identifier infrastructure. This separation allows us to analyze preservation-oriented citation practices independently from resource semantics.

4 RESULTS

We present in this section the empirical results we obtained about the stability of URLs found in software engineering scholarly works. We discuss them further in Section 5.

Before presenting the results, we briefly discuss data quality, which has two facets in this study. The first is measurement correctness, i.e., whether each URL is accurately classified as active or inactive; we assessed this risk through a manual audit of a random URL sample (Section 3), which indicates a low false-positive rate. The second concerns the assumptions underlying our regression models: because the claims of statistical significance reported below rest on the assumptions of the underlying linear model, we disclose those assumptions and evaluate how well the data meet them where the regression analysis is presented (Section 4.7). Most analyses in this section are

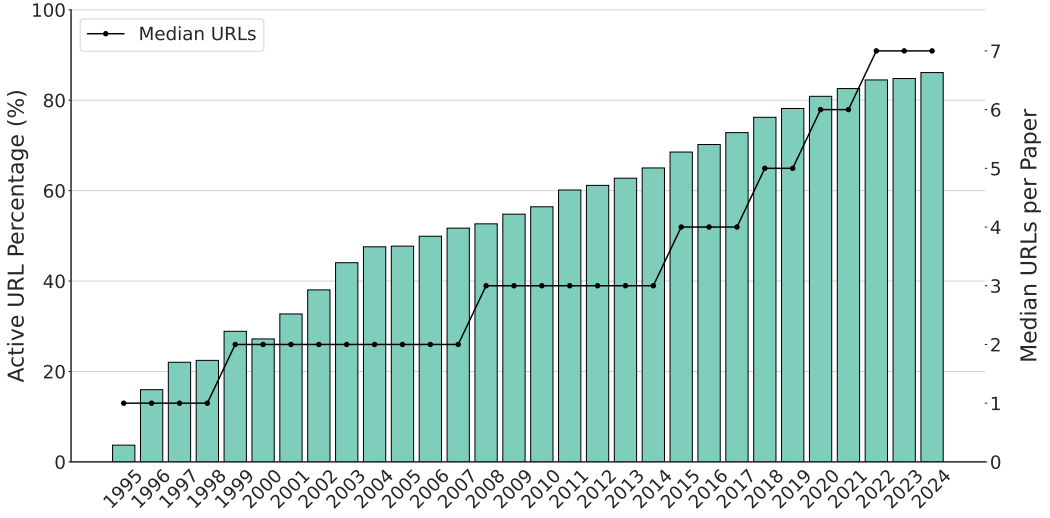


Fig. 4. Active URLs over time (1995–2024)

descriptive; the regression analysis, in particular, is exploratory and interpreted without causal claims.

4.1 URL stability over time

We start by presenting a temporal overview of URL rot, focusing on the 1995–2024 period. Papers from earlier years (1971–1994) reference fewer than 50 URLs/year in total; we remove them from the visualizations of this section to reduce visual clutter, as they do not add much to the analysis of our main findings. Data about those early years in our corpus is available in the replication package [2].

Figure 4 shows the evolution of active URLs over time, as well as the median number of URL references in papers for that year. The number of URLs referenced by software engineering papers is growing over time, together with a more general increase of papers published over time (see Figure 3). The growth of URL references over time is understandable, due to the increasing relevance of Web resources in most human activities, including scientific discovery. This increase creates additional opportunities for breaking the scholarly record, as URLs tend to be less persistent than more traditional scholarly artifacts such as journal articles, which are preserved through long-established archiving practices.

The percentage of active URLs varies across years, with a clear trend towards URLs in more recent papers being more likely to be active than in older papers. For instance, the year 2023 had 84.8% active URLs, indicating a relatively high level of URL accessibility, while 1996 exhibited 15.97% active URLs. A common measure for active URLs decay over time is *half-life*, i.e., the time required for half of the URLs published in a given year to become inactive; as anticipated in Table 1 we observe a half-life of 10.66 years, which is the longest among those reported in related studies.

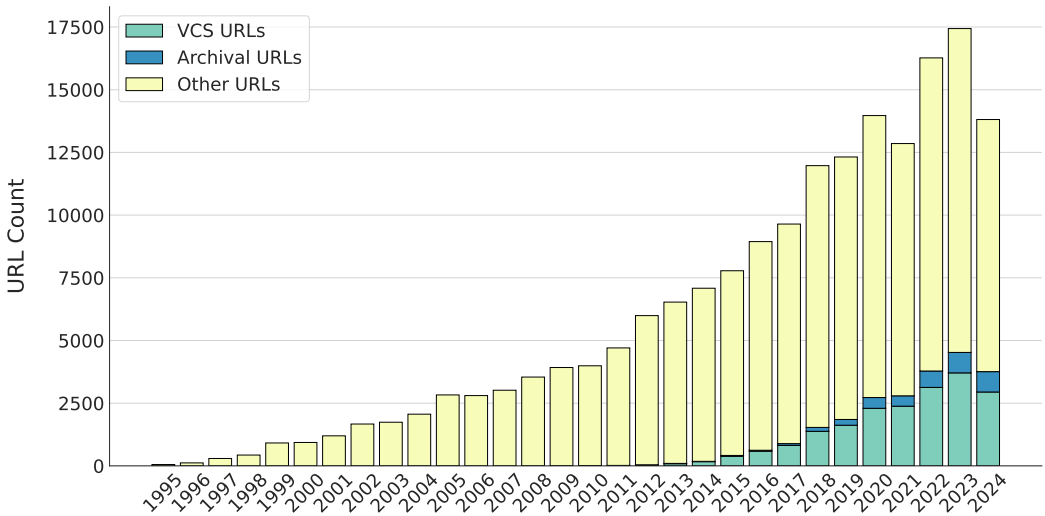


Fig. 5. Evolution of total URLs by hosting purpose through the years: all, version control systems, archival platforms.

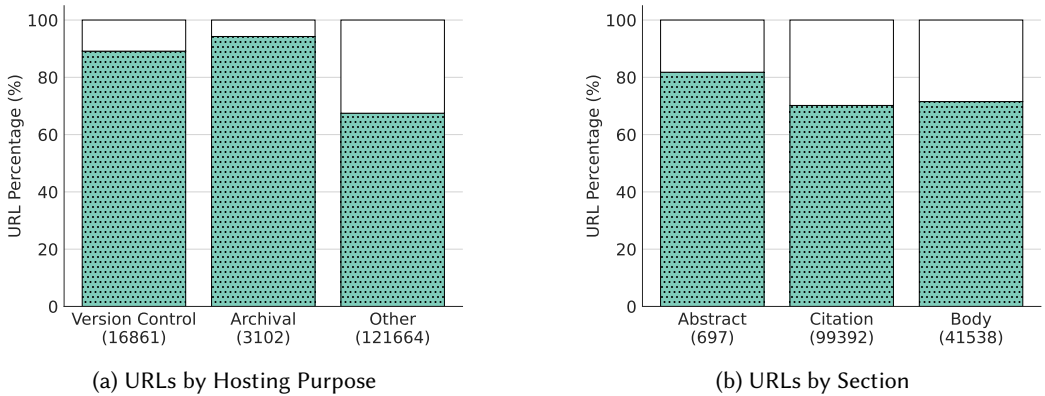


Fig. 6. Active URL ratios: By hosting purpose and document section

4.2 URL stability by hosting purpose

With increased awareness of the problem of URL instability, the use of archival services for Web references is also increasing over time. The increase in popularity of version control systems—which are not archival services, but do provide versioning—is also contributing to the kind of URLs scientists use in their papers. Both patterns are illustrated in Figure 5.

To investigate how these practices contribute to the persistence of URL references in software engineering papers, we classify URLs as: version control URLs, archival URLs, and all other URLs. To be classified as a version control URL, it must point to popular version control platforms (specifically GitHub, Bitbucket, and GitLab). These platforms are typically used for hosting and managing collaborative projects (most often for software source code, but also for data and Web sites), involving regular updates and revisions.

Table 3. Publication venues by active URL ratio, highest ratios first.

Venue	Active URL (%)	Venue	Active URL (%)
ICSME	82.30%	SOSYM	69.57%
TOSEM	81.86%	RE	68.94%
SANER	81.28%	SMR	67.05%
ISSTA	80.96%	ASEJ	66.96%
MSR	80.84%	SQJ	66.94%
FSE	80.73%	GPCE	66.83%
ICPC	80.39%	IST	65.63%
ASE	79.20%	JSS	64.78%
SCAM	77.95%	SPE	64.50%
ICSE	77.64%	ISSE	63.19%
ESEM	77.24%	SSBSE	62.29%
ESE	77.16%	ICSM	57.55%
ICST	76.57%	NOTES	56.67%
TSE	75.01%	SW	56.06%
STVR	72.79%	IJSEKE	55.78%
MODELS	71.47%	WCRE	55.44%
FASE	71.42%	REJ	55.22%

Archival URLs, on the other hand, are those that lead to digital archives or repositories with the stated mission of long-term preservation of digital assets. We classify as archival URLs those that point to the following platforms: Zenodo, arXiv, Internet Archive, Software Heritage, and Figshare.

Figure 6a shows a breakdown of URL stability by hosting type. Both version control and archival URLs tend to be more stable than the rest of the URL corpus, and by a significant margin. Also, as one would hope, archival URLs are more stable than VCS URLs although, somewhat unexpectedly, only by a small margin. The use of archival URLs remains overall scarce though: about 3102 URLs for archival versus 16 861 for version control URLs.

4.3 URL stability by venue

Table 3 provides a breakdown of the ratio of active URLs by venue (see Table 2 for an expansion of the venue acronyms). We could not identify any clear pattern although, anecdotally, the top venues in the field (e.g., FSE, TOSEM, ASE, ICSE, and TSE) are all in the top-half of the ranking by active URL ratio. The spread in the ratios is very high (55–82%) and does not appear to be induced by the journal/conference age.

4.4 URL stability by document placement

URLs are used in papers for different purposes and can hence appear in different document sections. We analyze three classes of document placements, based on the section a URL was found in: abstract, citation (“References” sections), body (everywhere else in the document). A breakdown of active URL ratios by placement is shown in Figure 6b.

URLs are not used much in abstracts, but when they are, they appear to be very stable (81.78% of active ratio). URLs are used the most in citations (e.g., DOI URLs or Web site for non-paper references), but there they show the worst active ratio: 70.18%. In between the two placements in terms of frequency of use, but much closer to citation URLs in terms of liveness, we find URLs located elsewhere in the paper with an active ratio of 71.54%.

Table 4. Comparison of top and bottom performing TLDs by active URL ratio

TLD	Top 10			TLD	Bottom 10		
	Total URLs	Active URL (%)	Introduced		Total URLs	Active URL (%)	Introduced
.press	202	99.50	2014	.no	207	39.61	1987
.science	80	95.00	2014	.es	692	39.02	1988
.dev	233	90.56	2014	.fi	250	38.40	1986
.ai	161	87.58	1995	.br	362	38.12	1989
.co	453	87.20	1991	.ca	1385	36.25	1987
.blog	135	85.93	2016	.nl	842	35.75	1986
.to	65	84.62	1995	.edu	6536	35.39	1985
.be	484	84.30	1988	.it	853	35.05	1987
.cc	433	83.83	1997	.ie	115	30.43	1988
.me	56	80.36	2007	.mil	226	23.89	1985

4.5 URL stability by URL scheme, top-level domain, domain

URLs are composed of various parts, the initial ones being URI scheme (e.g., `http`) and host name (e.g., `acm.org`), with the latter being further decomposable into a top level domain in the rightmost position (e.g., `.com`) and the rest of it. In this subsection we examine the impact of the various URL parts on URL rot in scientific papers.

URI scheme. Two main URI schemes are used these days for Web references: `http` and `https`, the latter denoting Web accesses cryptographically protected against eavesdropping and man-in-the-middle attacks by the TLS protocol.

We initially expected `https` URIs to be more fragile than `http` ones, because more “moving parts” can fail with the former: a certificate can expire, client and host can disagree on acceptable cryptographic algorithms, etc. To our surprise, the reverse is true. Active URL percentages by protocol are as follows:

- `http`: 59.90%
- `https`: 82.43%

This can be explained by the fact that, examined over a sufficiently long observation period (around 50 years in our case), `https` URLs tend to be younger (due to protocol adoption delays) and hence more likely to be still active to this day. But it is also possible that `https` URLs correlated with better URL referencing practices, which were not followed decades ago.

Top-Level Domains (TLDs). The analysis of retrieval rates across different top-level domains (TLDs), shown in Table 4, demonstrates significant variations in URL stability. Note that only TLDs with more than 50 URLs were included in the analysis, to avoid skewed results due to exceedingly small sample sizes.

Domains. Moving past *top* level domains, we looked at second-level domains (e.g., from `.gov` to `nasa.gov`), aggregating together all further sub-domains into second-level domains. Table 5 shows inactive URL ratios for the top domains in our corpus, as well as the corresponding active URLs per domain. The Global Rank information was retrieved from <https://pro.similarweb.com/>, accessed on 2025-07-11.

4.6 Network errors

We now look into the technical reasons why URLs fail, focusing on network and HTTP errors.

Table 5. Comparison of top and bottom performing domains by active URL ratio

Domain	Top 10			Domain	Bottom 10		
	Active URLs	Active URL (%)	Global rank		Active URLs	Active URL (%)	Global rank
orcid.org	204	100.00%	5 710	gatech.edu	26	27.66%	12 249
alturl.com	70	100.00%	9 291 360	gmu.edu	25	24.75%	13 526
4open.science	54	100.00%	269 540	codeplex.com	16	16.16%	709,339
youtu.be	326	100.00%	-	uiuc.edu	9	9.47%	159 417
youtube.com	421	99.53%	2	jstor.org	14	8.00%	3 932
mlr.press	201	99.50%	-	usc.edu	8	7.34%	9 500
git-scm.com	58	98.31%	19 194	hp.com	3	2.80%	702
php.net	51	98.08%	27 059	etherscan.io	3	2.33%	10 355
aclweb.org	143	97.95%	182 288	bham.ac.uk	1	1.11%	39 434
aclanthology.org	181	97.31%	66 254	tigris.org	0	0.00%	-

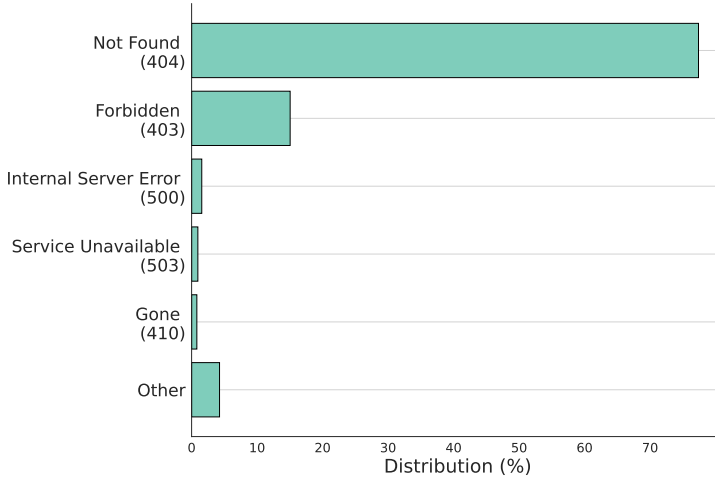


Fig. 7. HTTP error distribution

4.6.1 HTTP error distribution. Figure 7 shows the distribution of higher-level protocol (HTTP) errors, categorized by the HTTP status code obtained during URL accessibility checks. The distribution is dominated by, as most Internet users would have guessed, the infamous 404 error, denoting the nonexistence of a Web resource, followed by 403, denoting that the resource is not accessible without further authorization.

4.6.2 Network error analysis. URLs can also fail at a lower network level, before reaching the HTTP protocol level: a host name can be not resolvable at the DNS level, or the Internet server behind it can be unreachable over the network. Figure 8 shows the distribution of lower level network errors, as observed on a UNIX system using the curl tool (see Section 3 for details).

Non-resolvable host names dominate the distribution, followed by various sorts of connectivity errors, including some of the cryptographic certificate issues we anticipated for https URLs.

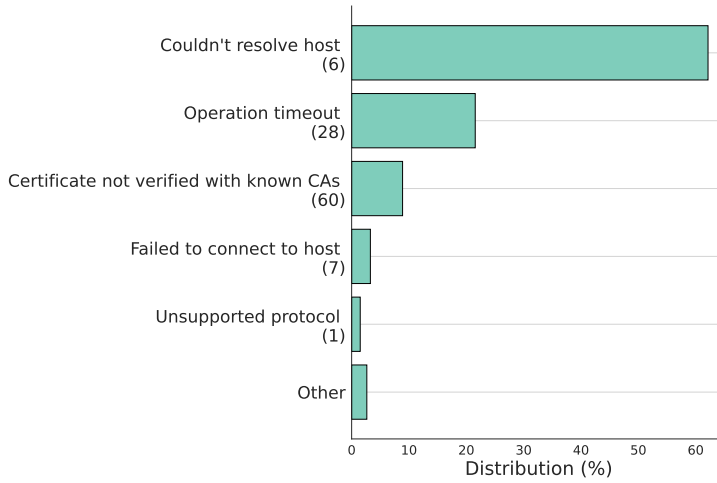


Fig. 8. Network error distribution

4.7 Regression analysis

We conducted regression analyses to examine the association between URL persistence and a range of URL, paper, and venue-level features.

Outcome (likelihood of URL persistence). For citation-level analyses, the dependent variable is a binary indicator *is_active* $\in \{0, 1\}$ (derived from the *active* field), indicating whether a cited URL is reachable. For venue-level analyses (CORE ranking and journal impact factor), the outcome is the mean success rate per venue.

Model specification. We fit ordinary least squares (OLS) linear regressions. The primary model operates at the citation level and includes the following predictors:

- URL length (#characters),
- URL length (#path steps),
- Paper age (reference year 2024 minus publication year),
- HTTPS usage (binary),
- Venue type (journal vs. conference),
- Document section (abstract, citations, other; dummy-encoded with “other” as reference).

Two additional venue-level models examine: (i) conference CORE ranking versus mean URL success rate, and (ii) journal impact factor versus mean URL success rate. CORE ranking levels (A*, A, B, C) are mapped to ordinal numerical values.

Regression assumptions and inference. Ordinary least squares assumes a correctly specified linear conditional mean, independent observations, homoscedastic residuals, and, for exact finite-sample inference, normally distributed residuals; it further requires the absence of severe multicollinearity, which we verified through variance inflation factors (all below 3). Because the citation-level outcome *is_active* is binary, the fitted model is a *linear probability model* and two of these assumptions fail by construction: the residual variance equals $p(1 - p)$, and residuals take only two values for any predictor vector. Fitted probabilities can also fall outside the $[0, 1]$ interval. Observations are not independent either, as multiple URLs originate from the same paper and URLs on the same domain share unobserved characteristics.

Table 6. Citation-level linear regression results (dependent variable: *is_active*), with 95% CIs and *p*-values from paper-clustered standard errors.

Variable	Coef.	95% CI	<i>p</i> -value
Intercept	0.9558	[0.9464, 0.9651]	0
URL length (#characters)	−0.00262	[−0.00279, −0.00245]	0
URL length (#path steps)	−0.00131	[−0.00359, 0.00097]	0.261
Paper age	−0.01787	[−0.01844, −0.01731]	0
HTTPS usage	0.08496	[0.07811, 0.09182]	0
Journal venue (vs. conf.)	−0.04744	[−0.05282, −0.04206]	0
Section: abstract	−0.02118	[−0.04618, 0.00382]	0.097
Section: citations	0.02386	[0.01875, 0.02897]	0

Table 7. Citation-level *p*-values across standard-error specifications and logistic-regression average marginal effects (AME), with OLS coefficients identical throughout and reported in Table 6.

Variable	Classical	HC3	Clustered by			Logistic	
			paper	domain	both	AME	<i>p</i>
URL length (#characters)	0	0	0	0	0	−0.00259	0
URL length (#path steps)	0.109	0.137	0.261	0.855	0.855	−0.00069	0.523
Paper age	0	0	0	0	0	−0.01481	0
HTTPS usage	0	0	0	0.006	0.006	0.10280	0
Journal venue (vs. conf.)	0	0	0	0	0	−0.04578	0
Section: abstract	0.145	0.093	0.097	0.588	0.588	−0.01644	0.277
Section: citations	0	0	0	0.240	0.240	0.03017	0

We retain the linear specification for its direct interpretability, but base inference on standard errors that do not rely on these assumptions: heteroskedasticity-robust (HC3), clustered by paper, clustered by domain, and two-way clustered by paper and domain. Statistical significance is accordingly interpreted using the robust and clustered estimates rather than conventional OLS standard errors. Table 6 reports the paper-clustered estimates, which address the strongest source of non-independence in our data, and Table 7 compares all specifications; we discuss the implications for conclusion validity in Section 6.4. For the two venue-level models, which regress a continuous per-venue success rate on a single predictor, the small number of units ($n = 17$ and $n = 14$) limits statistical power and precludes a reliable check of distributional assumptions; we likewise treat their estimates as exploratory.

Citation-level model. The citation-level model includes $n = 204\,383$ observations ($R^2 = 0.113$, adjusted $R^2 = 0.113$). Under paper-clustered standard errors, URL length (#characters), paper age, HTTPS usage, journal venue type, and citation-section placement are statistically significant associations at the 5% level; path depth and abstract placement are not. These results hold across all specifications (Table 7), with one exception: citation-section placement loses significance under domain clustering ($p = 0.240$), suggesting the association is driven by a few heavily cited domains rather than by placement itself.

Logistic-regression robustness analysis. Because the outcome is binary, we also fit a logistic regression (which bounds fitted probabilities within $[0, 1]$) using the same observations, predictors, and reference categories, again clustering standard errors by paper. We report average marginal

Table 8. CORE ranking (conference venues) and mean URL success rate.

Variable	Coef.	95% CI	<i>p</i> -value
Intercept	0.7732	[0.7148, 0.8316]	0
CORE ranking (numeric)	−0.00087	[−0.00318, 0.00143]	0.433

Table 9. Journal impact factor and mean URL success rate.

Variable	Coef.	95% CI	<i>p</i> -value
Intercept	0.6276	[0.5511, 0.7040]	0
Journal impact factor	0.01752	[−0.00352, 0.03855]	0.095

Table 10. Distribution of URLs by content category, with per-category active URL ratios.

Category	URLs	Share	Active URL (%)
PAPER	15 514	10.74%	85.49%
DATASET	141	0.10%	78.01%
ARTIFACT	1415	0.98%	79.51%
SOFTWARE	19 478	13.49%	88.12%
HOMEPAGE	26 347	18.25%	70.32%
DOCUMENTATION	9354	6.48%	72.36%
UNCLASSIFIED	72 154	49.97%	62.30%
Total	144 403	100.00%	70.64%

effects, i.e., the sample-average change in predicted probability, so that they are directly comparable to the linear-probability coefficients (Table 7). Their direction, magnitude, and significance match the linear model throughout. Refitting with B-spline terms for URL length and paper age leaves the significance pattern of the remaining predictors unchanged, indicating that our conclusions do not depend on assuming linear effects for these two variables.

Venue-level models. The CORE ranking model includes $n = 17$ venues ($R^2 = 0.042$, adjusted $R^2 = -0.022$). No statistically significant association is observed.

The journal impact factor model includes $n = 14$ venues ($R^2 = 0.215$, adjusted $R^2 = 0.150$). The estimated association is positive but not statistically significant at the 5% level.

4.8 URL content categories

Table 10 summarizes the distribution of URLs by the category introduced in Section 3.7, also showing the corresponding active URL ratio. The largest categories are *HOMEPAGE* URLs (26 347, 18.25%) and *SOFTWARE* URLs (19 478, 13.49%), reflecting the prominence of project homepages, tool entry points, and code-related resources in software engineering publications. *PAPER* URLs account for 10.74% (15 514), capturing links to publisher and preprint servers that can be identified based on URL patterns. *DOCUMENTATION* URLs (6.48%, 9354) are also common, consistent with the practice of referencing by URLs user guides, API references, and project documentation sites. In contrast, *DATASET* URLs (0.10%, 141) and *ARTIFACT* bundles (0.98% (1415)) are rare in our corpus. Finally, nearly half of all URLs (49.97%, 72 154) remain *UNCLASSIFIED* by URL-string heuristics

Table 11. Persistent identifier and archival resolver URLs with activity ratios.

Category	URLs	Share	Active URL (%)
DOI resolver	1643	1.14%	76.93%
Zenodo record	246	0.17%	92.68%
Handle System	182	0.13%	77.47%
Internet Archive	173	0.12%	78.61%
Perma.cc	22	0.02%	95.45%
PURL	15	0.01%	80.00%
ARK	7	0.01%	71.43%
w3id	7	0.01%	71.43%
Software Heritage	3	0.01%	33.33%
Total persistent	2298	1.59%	78.89%

alone, indicating that many cited Web resources do not expose sufficiently distinctive host/path patterns to be mapped to a semantic category without further crawling or content inspection.

Active URL ratios differ substantially across categories. *SOFTWARE* (88.12%) and *PAPER* (85.49%) URLs exhibit the highest retrieval rates. *ARTIFACT* (79.51%) and *DATASET* (78.01%) URLs follow with moderately high levels of persistence. In contrast, *DOCUMENTATION* (72.36%) and *HOME PAGE* (70.32%) URLs display lower retrieval rates, suggesting greater susceptibility to decay. *UNCLASSIFIED* URLs show the lowest proportion of active links (62.30%), representing the least stable category in the corpus.

Persistent identifier URLs and URL shorteners. Complementing the semantic content analysis, we also examine the extent to which software engineering publications rely on URL indirection mechanisms, namely URL shorteners and persistent identifier or archival resolvers. These dimensions are orthogonal to the content categories described above and capture citation practices rather than the referenced resource type.

URL shorteners account for 2445 URLs (1.69% of the corpus). Despite introducing an additional redirection layer, shortened URLs exhibit an activity rate of 75.87%, slightly higher than the overall corpus average. The majority of shortened references originate from a small number of widely used services, most notably `bit.ly`, `goo.gl`, and `tinyurl.com`.

Persistent identifier and archival resolver URLs are summarized in Table 11. Overall, such references account for a relatively small fraction of the corpus (2298 URLs, 1.59% of all links), yet they exhibit higher availability than the overall average.

DOI resolver URLs constitute by far the most prominent persistent identifier mechanism, representing the majority of persistence-oriented references by a large margin. All remaining identifier systems individually account for well below 0.2% of the corpus. This indicates that, when authors explicitly rely on persistent identifiers, they overwhelmingly adopt the DOI infrastructure. Note that these counts only capture DOIs written out as full resolver URLs: DOIs cited as bare identifiers (e.g., `doi:10...`), as is common in bibliographies, are not URLs and hence fall outside this study's scope and employed corpus. The same holds for other resources cited via non-URL identifiers, so these figures are a lower bound on identifier usage.

Because of this finding, we additionally examine how frequently DOI resolver URLs appear over time, measured as the percentage of all URLs cited in papers published in a given year. DOI usage is virtually absent in early decades of the corpus and begins to appear only after the late 2000s. From

2014 onward, DOI resolver links consistently represent approximately 1–1.4% of all cited URLs per year, indicating a gradual but sustained adoption of identifier-based referencing practices.

4.9 Archival availability of inactive URLs

We further examine URLs that were classified as inactive by the automated verification step, to assess whether they remain accessible through Web archives. Using the Wayback Machine as described in Section 3.4, we find that 62.48% of inactive URLs have at least one archived snapshot that could still be retrieved successfully.

This result shows that a non-negligible share of references that are no longer reachable in their original locations remain accessible through an archival service. Accordingly, archive availability constitutes a complementary dimension of URL persistence, distinct from network-level reachability at the originally cited URL.

5 DISCUSSION

This section interprets our empirical findings in light of prior work and examines the structural, temporal, and infrastructural factors associated with URL persistence. We discuss how ageing effects, hosting choices, content categories, and technical characteristics jointly shape the long-term accessibility of Web references in software engineering literature.

5.1 Comparison with previous studies

Drawing upon a substantial dataset, this study contributes novel empirical evidence to the substantial body of scientific literature on the persistence of URLs referenced from scholarly works. We focus on the field of software engineering and, in that field, analyze the largest body of papers (more than 50 thousand). The decision to automatically classify URLs as active in the absence of error responses aligns with the large scale of the dataset, making this approach both pragmatic and necessary—similarly to what was done in previous large-scale empirical studies.

The evolution of URL retrieval rates over time reveals significant advancements in digital resource availability. For instance, we observe a 84.8% active URL rate in 2023, much higher than 15.97% in 1996.

Interestingly, our observed overall active URL rate of 70.64% aligns closely with some previous findings, such as the 56.4%–81.4% range reported by Casserly and Bird [7], after reevaluation. This is also broadly consistent with findings from other fields discussed in Section 2 (e.g., biomedicine and library/information science), where substantial URL decay is likewise reported over comparable time horizons, despite differences in corpora, URL-validation criteria, and publication venues.

However, it also highlights a significant increase from the much lower retrieval rates observed in earlier years, as evidenced by the 23–53% range found in 2000 by Lawrence et al. [26].

A significant divergence with respect to previous studies is in URLs half-life. We measured a half-life of 10.66 years, a substantial increase compared to the latest half-life of 6.5 years reported by Bansal [4] in 2021. The most likely reason for this is an age effect. Our study includes more and more recent papers, which contain more recent URLs that have not yet had the time to decay.

This study, much larger and 25 years more recent than the last one in the overlapping broader field of computer science [26], contributes new empirical insights on the persistence of URLs in scholarly works, in the context of recent technological advancements and changes in digital archiving practices. The evolution in URL persistence, as shown by the comparison with earlier studies, underscores the dynamic nature of digital resource management in scholarly communication and suggests the need for periodically monitoring URL rot.

5.2 Temporal dynamics and cohort effects

Figure 4 reports the proportion of URLs that remain reachable for every publication year between 1995 and 2024. As expected, newer papers show higher active URL ratios than older ones: 84.8% of the URLs cited in 2023 were still available at the time of verification, whereas only 15.97% survive from 1996. Fitting an exponential decay curve yields a half-life of 10.66 years, meaning the community loses roughly half of its URL-based citations every decade.

This pattern is best interpreted as a consequence of ageing rather than evidence of URL-citation practices that improve over time. Given a decay rate on the order of a decade, a URL that has been online for only a few years is still very likely to resolve, whereas one that has lingered for multiple decades has a markedly lower chance. The upward trend on the right-hand side of the plot simply reflects the shorter exposure time of recent URLs; they have not yet had the opportunity to decay.

Because the characteristic decay is measured in years rather than months, URLs remain practical for short-term artifact sharing, e.g., over media used for day-to-day collaboration within research groups. For resources expected to matter beyond a single research cycle, as is the case for research papers that are routinely read decades later, URLs pointing to archival services—such as Zenodo [8], Software Heritage [12], or the Internet Archive [21], to name just a few—should always be preferred.

5.3 Venue, domain, and placement effects

Publication venue. The data reveal sizeable variation in URL longevity across publication venues. Conferences are, on average, a few percentage points ahead (i.e., higher URL survival rate) of journals, and the spread between the best- and worst-performing venues is substantial (more than twenty percentage points separate the extremes listed in Table 3). Recency might explain part of that gap: flagship conferences in computer science publish larger proportions of freshly minted papers, whereas journals publish “older” material to be stably incorporated into the scholarly record. The irony that scholarly material meant to be more stable ends up containing *less* stable URLs is not lost on us.

Regression analysis tells a different story about venue prestige. As shown in Tables 8 and 9, neither the CORE ranking ($\beta = -0.00087$, $p = 0.433$, 95% CI $[-0.00318, 0.00143]$, $R^2 = 0.041$) nor the journal impact factor ($\beta = 0.0175$, $p = 0.095$, 95% CI $[-0.00352, 0.03855]$, $R^2 = 0.215$) show a significant association with URL success. Reputation therefore does not guarantee URL survival; the observed differences might simply stem from the age profile of each venue’s corpus and other confounding factors.

URL placement in documents. URL placement inside the paper displays a clearer pattern. Links in abstracts are the most stable, those in the main body sit in the middle, and citations in the references section fare worst (Figure 6b). Abstracts often contain high-level project or dataset homepages that change little over time. This is supported by the fact that URL retrieval rate in abstracts (81.78%) is very similar to that of Version Control URLs (89.14%). A large share of reference-section links resolve to publisher HOMEPAGES, which are more prone to break during platform migrations.

Internet domains and hosts. Tables 4 and 5 show clear differences in URL survival by Internet domain and host. Notably, the most stable TLDs are ones relatively recently introduced, so the observed differences are most likely to be due to URL age. Also, service-oriented TLDs such as .press, .science, and .dev retain most links. Country-code TLDs (e.g., .no, .es) retain fewer.

A similar pattern appears at the host level: community and commercial platforms such as youtube.com and orcid.org lose fewer links than institutional servers such as gmu.edu or usc.edu,

which likely refer to student and faculty personal pages, that easily disappear when someone graduates or leaves. The host `codeplex.com` has a particularly low retrieval rate, which is not surprising given that the platform has been shut down.

Web site popularity does not seem to play a role, as observed by the Global Ranks of the domains.

Our findings therefore support the idea that two factors promote link longevity: a broad user base, which generates constant traffic and community maintenance, and explicit preservation mechanisms—version control mirrors, automated backups, or long-term-archive snapshots—that keep content reachable even when the original page changes or disappears.

5.4 Hosting purpose: forges vs archival services

Figure 6a partitions the corpus into three groups: general Web links, version-control-system (VCS) links, and links that target explicit archival services. Although VCS links already show a high live ratio of 89.14%, archival links are higher still, reaching 94.26%. Yet archival links are in total an order of magnitude fewer (3102 versus 16 861).

The difference is easy to explain. Public forges such as GitHub benefit from replication, automated redirects, and a large maintenance community. Archival platforms add a further layer: stable identifiers, fixity checks [3], and preservation mandates. The result is a modest but measurable gain in survival probability (with respect to simple VCS URLs; the gap with general Web links is more significant).

When authors must choose a single URL to reference a piece of software, a live VCS link, while not permanent by design, is better than an ordinary Web link, and an archival snapshot is better still. Ideally the two should be paired: referencing an archival source such as Software Heritage or Zenodo for long-term access, together with a reference to the current VCS hosting location (e.g., GitHub or a GitLab instance) for fetching the most recent version of the software and potentially contributing to it. Recent developments, such as the SWHID standard (ISO/IEC 18670) [10, 20], allow to pair the two together nicely; it will be interesting to monitor the standard’s adoption over time in future empirical studies on URL rot.

5.5 Technical challenges in URL accessibility

Our results demonstrate the technical aspects of URLs that impact their long-term availability when used in the academic literature. Among HTTP errors, “Not Found (404)” and “Forbidden (403)” emerge quantitatively as the primary culprits. Not Found errors, accounting for a significant 77.39%, often reflect inactive links or moved content. On the other hand, Forbidden errors, constituting 15.04%, hint at access control issues, perhaps signaling shifting permissions on Web resources [18], e.g., content that used to be freely available but no longer is.

Network errors further compound these challenges, with “Couldn’t resolve host” and “Operation timeout” standing out. These errors, which occurred in 62.15% and 21.55% of cases, respectively, underscore the impact of hosting and connectivity quality on the reliability of scholarly records.

Addressing these issues calls for changes in scientific publishing practices. On the one hand, favoring *public* archives (like those already mentioned before) and links to them over general Web links will go a long way to address both Not Found and Forbidden errors. Furthermore, for “Not Found” errors, publishers can integrate link-checking tools within the publication process, offering real-time alerts to authors and editors about inactive links [13].

For “Forbidden” errors, a more transparent and consistent policy regarding digital rights and access permissions could be beneficial. This might include clearer guidelines for authors on the appropriate places to link to from their academic publications—this policy too, however, should be verified throughout the publication process to be effective, short of a major shift in community norms in the field.

The frequency of network errors suggests the need for a robust infrastructure. Academic platforms could explore the use of redundant hosting or content delivery networks to minimize downtime and enhance content availability [15]. Archives should also take this into account, as they become more and more used as alternative link destinations from scholarly works.

To our surprise, protocol aspects (HTTP vs HTTPS in our experiments) strike in the opposite directions than what we expected: the HTTPS protocol, potentially more fragile due to certificate expiration and cipher negotiation issues, shows a noticeably higher rate (82.43%) than HTTP links (59.90%). As already mentioned, it is likely that part of this gap reflects age: HTTPS became common only in the late 2000s, so most secure links are younger. Authors should prefer linking to HTTPS resources rather than HTTP, for reasons other than this gap: HTTPS links provide a better guarantee that whoever is following them will actually arrive at the intended place, rather than being deflected to something else by an attacker. It would be interesting to monitor the long-term evolution of HTTPS URL-rot, to see if the causes of fragility that we expected to encounter (but did not, overall) surface in the near future.

5.6 Insights from regression analysis

The regression analysis undertaken in this study could offer insights into the factors influencing URL persistence in scientific publications. At the citation level, paper age and URL length (in characters) show statistically significant negative associations with persistence, while HTTPS usage is positively associated. These findings align with the expectation that older references are more exposed to link decay over time, and that secure protocol adoption may reflect more stable hosting practices. Venue type and document section also exhibit measurable differences, whereas the number of path elements does not appear to meaningfully affect persistence.

In contrast, the venue-level analyses reveal weaker patterns. Neither CORE ranking nor journal impact factor shows a statistically significant association with mean URL success rate. This suggests that formal indicators of venue prestige do not necessarily translate into improved long-term accessibility of cited web resources.

Overall, the results indicate that temporal and technical characteristics play a more prominent role in URL persistence than publication prestige, although the moderate explanatory power of the models implies that additional factors likely contribute to long-term link stability.

5.7 URL content categories

We have compared the stability of URLs across different categories of what they point to (see Table 10).

Links to *SOFTWARE* and *PAPER* resources exhibit the highest retrieval rates. Both categories commonly correspond to infrastructures characterized by versioning, persistent identifiers, or institutional stewardship. In contrast, *HOME PAGE* and *DOCUMENTATION* URLs display lower stability, which is consistent with the evolving nature of project websites and documentation portals. Such resources are frequently reorganized, renamed, or migrated as projects mature.

The comparatively low stability of *UNCLASSIFIED* URLs suggests that general-purpose or structurally indistinct Web resources are more vulnerable to long-term decay. This category aggregates a heterogeneous set of domains and hosting environments, many of which lack explicit preservation mechanisms or long-term institutional guarantees.

We also observe comparatively few URLs that can be attributed to *DATASET* and *ARTIFACT* resources, albeit for different reasons. For *ARTIFACT* URLs, this pattern is consistent with an interpretation in which packaged research artifacts are frequently referenced indirectly through higher-level project or software entry points, rather than through URLs that explicitly and unambiguously denote bundled artifact releases. In contrast, the scarcity of URLs classified as *DATASET*

should be interpreted more cautiously: datasets are still too often hosted within general-purpose repositories, project websites, or version control platforms whose URL structures do not reliably expose dataset-specific semantics. As a result, dataset references may be present in the corpus, but not identifiable as such using URL-string heuristics alone.

A key contribution of our classification pipeline is that it achieves substantial semantic coverage while relying exclusively on URL strings, without crawling or content inspection. This allows scalability to a large corpus and reproducibility of the method, while keeping the analysis aligned with the broader study design, which emphasizes automated, protocol-level and metadata-level signals. In turn, this enables our downstream analyses that tie URL intent to persistence-related outcomes (differences in liveness by category), and supports context-sensitive interpretation of URL rot: the loss of a link does not carry uniform implications, but depends on the type of referenced resource and its role in the scholarly record.

Taken together, the content-category analysis suggests that URL persistence is influenced not only by hosting platform, but also by the semantic role of the referenced resource. Concrete, versioned, or institutionally curated artifacts tend to exhibit higher stability than navigational or descriptive entry points. This complements the findings on hosting purpose (Section 5.4) and reinforces the interpretation that preservation practices and infrastructure choices—rather than venue-level prestige alone—play a central role in shaping the long-term accessibility of Web-based scholarly references.

Persistent identifiers and citation practices. The analysis of URL shorteners and persistent identifier infrastructures offers a complementary perspective on URL persistence, capturing citation practices rather than resource semantics. Overall adoption of persistence-oriented mechanisms remains limited, as only a small fraction of URLs rely on dedicated resolver or archival systems, despite their generally higher availability compared to typical Web references.

DOI resolver URLs dominate this space by a large margin, reflecting the strong integration of the DOI ecosystem within scholarly publishing workflows. Other identifier and archival infrastructures appear only sporadically, suggesting that their uptake within software engineering publications is still relatively modest. Notably, several persistence-oriented mechanisms that appear only rarely in our corpus, such as archival or repository-backed identifiers, exhibit very high activity ratios, indicating promising preservation performance when adopted. However, given the comparatively small number of observed instances, these results should be interpreted cautiously and do not allow strong conclusions regarding their long-term effectiveness at scale yet.

The temporal evolution of DOI usage indicates a gradual shift toward identifier-based referencing. While essentially absent in earlier decades, DOI links grew in popularity over time, though they still represent a small share of all cited Web resources. This suggests that most references continue to depend on location-based URLs rather than persistent identifiers.

Perhaps surprisingly, URL shorteners do not appear to constitute a major source of link decay. Their retrieval rates outperform the corpus average, suggesting that long-term accessibility depends primarily on the persistence of the target resource rather than the presence of an intermediate redirection service.

6 THREATS TO VALIDITY

We look in this section at potential limitations of this study, following the usual classification of threats to validity: internal validity, external validity, construct validity, conclusion validity, and reliability.

6.1 Internal validity

Internal validity concerns the extent to which observed associations can be interpreted as reflecting causal relationships. Because this study is cross-sectional and observational, internal validity is inherently limited: the reported associations do not support causal inference. Establishing causality would require a longitudinal design that tracks URL status changes over time or experimentally controls relevant factors. Accordingly, all results in this paper are meant to be interpreted as correlational rather than causal.

6.2 External validity

We do not claim generality outside the stated focus of this study: software engineering papers. Within that field, we rely on DBLP as bibliographic index and, as researchers ourselves in that field, we are confident it is a good choice of index. It is still possible that more niche venues in the field exhibit different URL stability patterns, but it would be quite surprising.

6.3 Construct validity

While the study aims to measure URL persistence accurately, the operational definitions of “active” and “inactive” URLs depend heavily on automated criteria such as HTTP status codes and network errors. This approach may not fully capture all the workings of URL functionality, particularly in cases where URLs redirect to content that is relevant for the domain, but not what the original authors meant to reference.

An additional threat stems from the exploratory character of several analyses presented in this work. We examine multiple potential correlates of URL persistence across venues, domains, URL features, and publication characteristics. Exploring many associations increases the risk of false positives, particularly when results are interpreted in isolation. For this reason, we avoid causal claims and treat statistically significant associations as descriptive signals rather than definitive effects.

6.4 Conclusion validity

Conclusion validity concerns the degree to which our statistical conclusions about associations are warranted. As detailed in Section 4.7, the citation-level model applies ordinary least squares to a binary outcome and is therefore a linear probability model. Such model’s residuals are heteroscedastic and non-normal by construction. In addition, URLs extracted from the same paper or URLs from the same domain are not strictly independent. These departures affect the standard errors rather than the estimated coefficients. We therefore base significance claims on heteroskedasticity-robust (HC3) and clustered standard errors. To verify that the linearity assumption is likewise not skewing our conclusions, we fit a logistic regression model on the same data and compare average marginal effects. The principal associations hold throughout; citation-section placement does not, as it loses significance under domain clustering, and we accordingly do not treat it as an established result. Coefficients should be read as associations conditional on the remaining predictors.

6.5 Reliability

The study relies on a multi-step process to both gather data (bibliographic information, papers, etc.) and verify URL stability (parsing, URL extraction, URL checks, etc.). Even though each step in isolation is fairly simple, it is always possible that specific bugs crept into our implementation of any of them. To mitigate this, we make available a complete replication package [2] for our implementation of the methods and the study results.

7 CONCLUSION

This large-scale study provides a new empirical view of URL persistence in software engineering publications. Analyzing nearly 144 thousand published URLs from more than 50 thousand papers, we find that URL stability has improved compared to earlier studies, with 71% of URLs remaining active and an estimated URL half-life of 11 years. This represents meaningful progress, yet the decay rate remains non-negligible, threatening the value and integrity of software engineering publications.

Our results underscore both technical and community aspects of URL rot. On the technical side, archival services and version control system platforms show markedly higher retrieval rates than other Web links, demonstrating the value of stable hosting infrastructures and long-term preservation mandates. Surprisingly, regression analyses reveal no significant association between URL stability and venue prestige, and the URL and paper characteristics we examined account for only a modest share of the variation in URL survival, suggesting that link decay is shaped more by hosting choices and preservation practices than by publication policies.

From a community perspective, our findings point to specific actionable steps. Based on our results, by using the Internet Archive, we were able to access a significant fraction of inactive URLs. Authors should try to leverage archival services — such as the Internet Archive, Software Heritage, or Zenodo — when referencing digital artifacts, particularly for datasets, tools, and software that underpin their research claims. This is supported by the fact that such services offer higher successful retrieval rates for URLs, as was shown by our results. Conference organizers, journal editors, and publishers may consider encouraging or even requiring archival deposits to improve long-term access and reproducibility. Finally, given the ongoing evolution of digital publishing practices, periodic large-scale studies should monitor URL rot and guide preservation efforts that safeguard the scholarly record for future generations.

DATA AVAILABILITY

A full replication package for the experiments described in this paper is available from Zenodo [2].

REFERENCES

- [1] Dag W. Aksnes, Liv Langfeldt, and Paul Wouters. 2019. Citations, Citation Indicators, and Research Quality: An Overview of Basic Concepts and Theories. *SAGE Open* 9, 1 (Jan. 2019), 215824401982957. <https://doi.org/10.1177/2158244019829575>
- [2] Stefano Zacchiroli Apostolos Garos, Diomidis Spinellis. 2025. Replication package for: URL Rot in Software Engineering Scholarly Works. <https://doi.org/10.5281/zenodo.16058275>
- [3] Mohamed Aturban, Sawood Alam, Michael L. Nelson, and Michele C. Weigle. 2019. Archive Assisted Archival Fixity Verification Framework. In *2019 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*. IEEE, 162–171. <https://doi.org/10.1109/jcdl.2019.00032>
- [4] Sonia Bansal. 2021. Decay of URL References cited in DESIDOC Journal of Library & Information Technology. (07 2021).
- [5] Sonia Bansal and Seema Parmar. 2020. Decay of URLs Citation: A Case Study of Current Science. https://www.researchgate.net/publication/338990081_Decay_of_URLs_Citation_A_Case_Study_of_Current_Science
- [6] Terrence A. Brooks. 1985. Private acts and public objects: An investigation of citer motivations. *Journal of the American Society for Information Science* 36, 4 (July 1985), 223–229. <https://doi.org/10.1002/asi.4630360402>
- [7] Mary F. Casserly and James E. Bird. 2003. Web Citation Availability: Analysis and Implications for Scholarship. *College & Research Libraries* 64, 4 (July 2003), 300–317. <https://doi.org/10.5860/crl.64.4.300>
- [8] CERN. 2013. Zenodo. <https://en.wikipedia.org/wiki/Zenodo>. Accessed: 2025-06-16.
- [9] Chuanfu Chen, Kai Sun, Gang Wu, Qiong Tang, Jian Qin, Kuei Chiu, Yushuang Fu, Xiaofang Wang, and Jing Liu. 2009. The impact of internet resources on scholarly communication: A citation analysis. *Scientometrics* 81, 2 (April 2009), 459–474. <https://doi.org/10.1007/s11192-008-2180-y>
- [10] Roberto Di Cosmo, Morane Gruenpeter, and Stefano Zacchiroli. 2020. Referencing Source Code Artifacts: A Separate Concern in Software Citation. *Comput. Sci. Eng.* 22, 2 (2020), 33–43. <https://doi.org/10.1109/MCSE.2019.2963148>
- [11] croumgeous. 2023. doi2pdf: A Command Line Tool to Download PDFs of Research Paper from DOI, Name or URL. <https://pypi.org/project/doi2pdf/>. Accessed: 2024-02-03.
- [12] Roberto Di Cosmo and Stefano Zacchiroli. 2017. Software Heritage: Why and How to Preserve Software Source Code. In *Proceedings of the 14th International Conference on Digital Preservation, iPRES 2017, Kyoto, Japan, September 25-29, 2017*, Shoichiro Hara, Shigeo Sugimoto, and Makoto Goto (Eds.). <https://hdl.handle.net/11353/10.931064>
- [13] Stephan Druskat, Oliver Bertuch, Guido Juckeland, Oliver Knodel, and Tobias Schlauch. 2022. Software publications with rich metadata: state of the art, automated workflows and HERMES concept. <https://doi.org/10.48550/ARXIV.2201.09015>
- [14] Emily Escamilla, Martin Klein, Talya Cooper, Vicky Rampin, Michele C. Weigle, and Michael L. Nelson. 2023. *Cited But Not Archived: Analyzing the Status of Code References in Scholarly Articles*. Springer Nature Singapore, 194–207. https://doi.org/10.1007/978-981-99-8088-8_17
- [15] Jose Luis Garcia-Dorado. 2015. Bandwidth in the Cloud. (2015). <https://doi.org/10.48550/ARXIV.1512.01129>
- [16] Carol Anne Germain. 2000. URLs: Uniform Resource Locators or Unreliable Resource Locators. *College & Research Libraries* 61, 4 (July 2000), 359–365. <https://doi.org/10.5860/crl.61.4.359>
- [17] Daniel Gomes and Mário J. Silva. 2006. Modelling information persistence on the web. In *Proceedings of the 6th international conference on Web engineering - ICWE '06 (ICWE '06)*. ACM Press. <https://doi.org/10.1145/1145581.1145623>
- [18] Mandeep Kaur Gondara. 2011. Access Control Mechanisms for Semantic Web services-A Discussion on Requirements & Future Directions. (2011). <https://doi.org/10.48550/ARXIV.1105.0141>
- [19] Internet Archive. 2024. Wayback Machine APIs. https://archive.org/help/wayback_api.php. Accessed: 2025-01-20.
- [20] ISO/IEC 18670:2025 2025. *SoftWare Hash Identifier (SWHID) Specification V1.2*. Standard. International Organization for Standardization, Geneva, Switzerland.
- [21] Brewster Kahle. 1997. Archiving the internet. *Scientific American* 3 (1997).
- [22] Zeinab Abou Khalil and Stefano Zacchiroli. 2022. The General Index of Software Engineering Papers. In *19th IEEE/ACM International Conference on Mining Software Repositories, MSR 2022, Pittsburgh, PA, USA, May 23-24, 2022*. ACM, 98–102. <https://doi.org/10.1145/3524842.3528494>
- [23] Wallace Koehler. 1999. An analysis of web page and web site constancy and permanence. *Journal of the American Society for Information Science* 50, 2 (1999), 162–180. [https://doi.org/10.1002/\(SICI\)1097-4571\(1999\)50:2%3C162::AID-ASI7%3E3.0.CO;2-B](https://doi.org/10.1002/(SICI)1097-4571(1999)50:2%3C162::AID-ASI7%3E3.0.CO;2-B)
- [24] Wallace Koehler. 2004. A longitudinal study of Web pages continued: a consideration of document persistence. *Information Research* 9, 2 (2004), 9–2. <http://informationr.net/ir/9-2/paper174.html> Paper no. 174.
- [25] Zoe Kotti, Konstantinos Kravvaritis, Konstantina Dritsa, and Diomidis Spinellis. 2020. Standing on Shoulders or Feet? An Extended Study on the Usage of the MSR Data Papers. *Empirical Software Engineering* 25, 5 (Sept. 2020), 3288–3322. <https://doi.org/10.1007/s10664-020-09834-7>
- [26] Steve Lawrence, Frans Coetzee, Eric Glover, Gary Flake, David Pennock, Bob Krovetz, Finn Nielsen, Andries Kruger, and Lee Giles. 2000. Persistence of information on the web. In *Proceedings of the ninth international conference on*

- Information and knowledge management - CIKM '00*. ACM Press. <https://doi.org/10.1145/354756.354824>
- [27] Michael Ley. 2002. The DBLP Computer Science Bibliography: Evolution, Research Issues, Perspectives. In *String Processing and Information Retrieval, 9th International Symposium, SPIRE 2002, Lisbon, Portugal, September 11-13, 2002, Proceedings (Lecture Notes in Computer Science, Vol. 2476)*, Alberto H. F. Laender and Arlindo L. Oliveira (Eds.). Springer, 1–10. https://doi.org/10.1007/3-540-45735-6_1
 - [28] Fayaz Ahmad Loan and Ufaira Yaseen Shah. 2020. The decay and persistence of web references. *Digital Library Perspectives* 36 (5 2020), 157–166. Issue 2. <https://doi.org/10.1108/DLP-02-2020-0013>
 - [29] Patrice Lopez. 2008–2021. GROBID - GeneRation Of Bibliographic Data. <https://grobid.readthedocs.io/> Accessed 2022-01-25.
 - [30] Frank McCown, Sheffan Chan, Michael L. Nelson, and Johan Bollen. 2005. The Availability and Persistence of Web References in D-Lib Magazine. (2005). <https://doi.org/10.48550/ARXIV.CS/0511077>
 - [31] M.K. Saberi and H. Abedi. 2012. Accessibility and decay of web citations in five open access ISI journals. *Internet Research* 22, 2 (March 2012), 234–247. <https://doi.org/10.1108/10662241211214584>
 - [32] Diomidis Spinellis. 2003. The decay and failures of web references. *Commun. ACM* 46, 1 (Jan. 2003), 71–77. <https://doi.org/10.1145/602421.602422>
 - [33] Dominika Tkaczyk, Andrew Collins, Paraic Sheridan, and Joeran Beel. 2018. Machine Learning vs. Rules and Out-of-the-Box vs. Retrained: An Evaluation of Open-Source Bibliographic Reference and Citation Parsers. <https://doi.org/10.48550/ARXIV.1802.01168>
 - [34] Cassie Wagner, Meseret D. Gebremichael, Mary K. Taylor, and Michael J. Soltys. 2009. Disappearing act: decay of uniform resource locators in health care management journals. *Journal of the Medical Library Association : JMLA* 97, 2 (April 2009), 122–130. <https://doi.org/10.3163/1536-5050.97.2.009>
 - [35] J. D. Wren. 2004. 404 not found: the stability and persistence of URLs published in MEDLINE. *Bioinformatics* 20, 5 (Jan. 2004), 668–672. <https://doi.org/10.1093/bioinformatics/btg465>
 - [36] J. D. Wren. 2008. URL decay in MEDLINE—a 4-year follow-up study. *Bioinformatics* 24, 11 (April 2008), 1381–1385. <https://doi.org/10.1093/bioinformatics/btn127>
 - [37] Jonathan D. Wren, Kathryn R. Johnson, David M. Crockett, Lauren F. Heilig, Lisa M. Schilling, and Robert P. Dellavalle. 2006. Uniform Resource Locator Decay in Dermatology Journals. *Archives of Dermatology* 142, 9 (Sept. 2006). <https://doi.org/10.1001/archderm.142.9.1147>