Repeatability, Reproducibility, Replicability, Reusability (4R) in Journals' Policies and Software/Data Management in Scientific Publications: A Survey, Discussion, and Perspectives

José Armando Hernández, Miguel Colom

Repeatability, Reproducibility, Replicability, Reusability (4R) in Journals' Policies and Software/Data Management in Scientific Publications: A Survey, Discussion, and Perspectives

José Armando Hernández González
jose.hernandez_gonzalez@ens-paris-saclay.fr
ORCID 0000-0002-6692-8640

Miguel Colom
mcolomba@ens-paris-saclay.fr
ORCID 0000-0003-2636-0656

Université Paris Saclay, Université Paris Cité, ENS Paris Saclay, CNRS, SSA, INSERM, Centre Borelli. France.

December 7, 2023

Abstract

With the recognized crisis of credibility in scientific research, there is a growth of reproducibility studies in computer science, and although existing surveys have reviewed reproducibility from various perspectives, especially very specific technological issues, they do not address the author-publisher relationship in the publication of reproducible computational scientific articles. This aspect requires significant attention because it is the basis for reliable research. We have found a large gap between the reproducibility-oriented practices, journal policies, recommendations, publisher artifact Description/Evaluation guidelines, submission guides, technological reproducibility evolution, and its effective adoption to contribute to tackling the crisis.

We conducted a narrative survey, a comprehensive overview and discussion identifying the mutual efforts required from Authors, Journals, and Technological actors to achieve reproducibility research. The relationship between authors and scientific journals in their mutual efforts to jointly improve the reproducibility of scientific results is analyzed. Eventually, we propose recommendations for the journal policies, as well as a unified and standardized Reproducibility Guide for the submission of scientific articles for authors.

The main objective of this work is to analyze the implementation and experiences of reproducibility policies, techniques and technologies, standards, methodologies, software, and data management tools required for scientific reproducible publications. Also, the benefits and drawbacks of such an adoption, as well as open challenges and promising trends, to propose possible strategies and efforts to mitigate the identified gaps. To this purpose, we analyzed 200 scientific articles, surveyed 16 Computer Science journals, and systematically classified them according to reproducibility strategies, technologies, policies, code citation, and editorial business.

We conclude there is still a reproducibility gap in scientific publications, although at the same time also the opportunity to reduce this gap with the joint effort of authors, publishers, and technological providers.

Keywords: Repeatability, Reproducibility, Replicability, Reusability, Data Science AI/ML, RaaS, Scientific journal, Trustworthy, Data Citation, Rewarding Research, Reproducible Research.

1 Introduction

Reproducibility is a broad and complex topic strongly related to the history of science and knowledge [1] reflected in the cumulative technological and scientific development of humanity [2]. Such development has been based on the evolutionary capacity of human beings to build new knowledge from previous discoveries and achievements, transmitting this knowledge to new generations in a continuous cycle of improvement. The evolution of Science through the reproducibility of knowledge could be metaphorically compared to the natural mechanisms of DNA replication [3] transmitted from generation to generation in a continuous cycle of refinement. Within these reproducible mechanisms, scientific journals play a significant role in the communication, divulgation, corroboration, validation, and acceptance of reliable and trustworthy knowledge.

The reproducibility of knowledge has recently become relevant to the scientific community given that there is a growing concern for ethics and transparency in the research results in scientific publications in the so-called *reproducibility crisis* [4, 5]. In addition, with the boom in artificial intelligence/machine learning (AI/ML), publications have evolved towards data-centric and model-centric developments that force journals to adapt their publishing business models to new dynamics accelerated by technological changes [6].

In response to these developments, several recent articles have hypothesized what the future of academic publishing will be like [7, 8] [9], analyzing important changes, proposing technological tools [10, 11] and identifying significant gaps in publishing policies [12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24]. These articles account for policies implemented by publishers and their evolution, which are crucial for understanding their evaluation processes oriented to the Reproducibility of knowledge.

This paper analyzes the journal policies concerning the reproducibility of knowledge addressed to trustworthiness and transparency through a survey of computer science journals indexed in SCOPUS& WoS. This survey identifies recent advancements in computational reproducibility and policies, develop strategies that engage with the reproducibility crisis, and analyze future perspectives in computer base scientific publications. In particular, the role of stakeholders [25] in the reproducibility of computational scientific articles, especially ML and AI-related projects, is explored to understand better the relationship between authors and scientific journals driven by advanced cloud-computing technologies. By reviewing the most promising trends, this paper broadens the landscape [8] of possibilities for advances in reproducibility from both technological and methodological points of view, addressing the gap in effective journal policies that guarantee the reproducibility of computational works.

A wide variety of scientific journals deal with different topics or specialize in various disciplines and fields of knowledge. In several cases, these are subjects based completely on theory or physical experiments. Therefore, this article delimits its scope to specifically analyze scientific journals of computer science, especially the management and reproducibility of scientific articles based on software and data, where the objects to be replicated are information (*bits*), highlighting that many scientific disciplines converge in the computational field to solve their problems and build up their knowledge base.

Figure 1 summarizes the structure of this paper, where three aspects are analyzed: efforts required by authors (Section 5.1) and by publishers (Section 5.2), and the technological evolution required to reproduce results (Section 2).

The combined efforts of these three actors are required to close the reproducibility gap. Our systematic literature is a PRISMA-based review (see Appendix A.1). We discuss the corresponding definitions, difficulties, and measures related to Reproducibility, which allows us to define the technological evolution as necessary to reproduce results and the fundamental strategies of reproducibility in Section 2. Section 2.4 presents a landscape of existing tools, data management platforms, and techniques that are helpful in reproducible research. Section 2.6 introduces the role of publishers in reproducible research, including new types of publications with code, and discusses the problem of evaluating research artifacts. Section 3 surveys 16 computer science journals and provides insights about experiences implementing data-code sharing policies based on the reviewed reproducibility platforms and technologies. The methodology that we followed is described in Appendix A.2.

Section 4 includes our technological discussion of the topics presented before. Section 4.2 focuses on and highlights the shared responsibility between authors and publishers supported by technological evolution. Section 5 asses and analyzes the combined efforts of these three actors required to close the reproducibility gap. The efforts required by authors in Section 5.1 and by publishers in section5.2. Important dilemmas that emerge are addressed in sections 4.1 5.3. Under the possibility of regarding reproducibility as a service provided by trusted third party, considering software as valuable research artifacts, and how to reward authors.

Section 6 concludes the paper.

1.1 Definitions

Several works [26, 27, 28, 29, 30, 31] have addressed reproducibility from different points of view, as [32] reproducibility is considered a fundamental part of the scientific method. However, to our knowledge, no works have holistically reviewed the different dimensions and strategies of reproducibility in computer science, i.e., to consider their essential participation within an end-to-end data science project/experiment life cycle. This life cycle begins from scientific research and ends in mass industrial production for final customers. The life cycle also incorporates the responsibilities of the main stakeholders [25, 33, 34] in this process (e.g., journals, authors, industry, and the scientific community).

The report [35] from the National Academies of Sciences, Engineering, and Medicine (NASEM) is a reference reproducibility study that gathers contributions from relevant specialized researchers. It focuses on strategies for obtaining consistent computational results using the same input data, computational steps, methods, code, analysis conditions, and replicability to get consistent results across studies. In NASEM's definitions, *reproducibility* involves the original data and code, whereas *replicability* is related to the collection of new data and similar methods used in previous studies.

The simplest definition of reproducibility extended and used in the different works is the one proposed by ACM in version 1.1 of their Artifact Review and Badging report[1], as shown in Figure 2.

Figure 4 shows the four definitions concerning the team, method, code, metadata, and setup elements.

[1] `https://www.acm.org/publications/policies/artifact-review-and-badging-current`

Reproducibility (different team, same experimental setup): the experiment is done with different equipment, different environment, same code/algorithm. **Repeatability** (same team, same experimental setup): the experiment is done by the same team, same environment (software/hardware), same code/algorithm. **Replicability** (different team, different experimental setup): the experiment is done with different equipment, different environment, different code, same algorithm. **Reusability** (different equipment, different and partial experimental configuration): the experiment is carried out with different equipment, different environments, different codes, and the algorithm partially implemented.

There is still some discussion, in some cases even confusion, about the definitions [36] even from a taxonomic point of view [37, 38]. A very different interpretetion of reproducibility is presented in [39] where it is a continuous improvement process rather than an achievable objective. However, following the discussions with the National Information Standards Organization (NISO), ACM accepted the recommendation to harmonize its terminology and definitions with those widely used in the community of scientific research. In this way, it interchanged the terms *reproducibility* and *replicability* with the existing definitions proposed by ACM to ensure consistency.

In this article, we specifically discuss the reproducibility of complex ML/AI data science projects. A scientific publication in ML/AI can range from effectively a model developed in an experiment by a single researcher for a tiny device to large implementations of Distributed Big Data supercomputing developed by large consortiums of universities, governmental, or research institutions.

1.2 Types of Reproducibility

Defining what reproducibility is as important as determining the types of reproducibility, considering the nuances that conceptually appear when studying the various cases and possibilities. The term reproducibility is acceptable in the case where the same input can lead to statically equivalent same results. It is also important to note that reproducibility in data science does not necessarily imply obtaining the same numerical result from the same numerical input.

Previous works [40, 31] have defined three degrees of reproducibility: R1 (Experiment, Data, Method), R2 (Data, Method), and R3 (Method). It is only sometimes possible to obtain the same numerical result from different realizations of an experiment. In that case, one can consider the following definitions [41] :

- **Experimental** reproducibility: similar input (data) + similar experimental protocol → similar results
- **Statistical** reproducibility: same input (data) + same analysis → same conclusions (independently from (random) sampling variability)
- **Computational** reproducibility: similar input (data) + same code/software + same software environment → exact same bit-wise results

1.3 Side difficulties to achieve reproducibility

Several difficulties identified in our review are not related to the actual shared source code used to reproduce an experiment. Rather, they involve external considerations such as dependency on 3rd-party libraries, complex and uncontrolled software dependencies, the quality of the writing in scientific articles, the documentation of the software, sustainability for the long term [42], reproducibility, and the reward for the career advancement of the researchers [43].

Complex and uncontrolled software dependencies With the increasing demand for functionalities, the code has become more complex, relying on several 3rd party software dependencies. These include libraries, packages, or complete frameworks. Changes from different releases (or even from version to version) and irregular maintenance or support lead to what has been described as *dependency hell*. This irregular maintenance represents one significant contemporary challenge for obtaining reproducible results from software made from different components that are sometimes obsolete. Using package managers, virtual environments, and container tools, keep an up-to-date list of dependencies, and perform continuous testing [44], are recommended to mitigate this problem. With these strategies, the management of dependencies can be simplified, and projects can then be run in a reproducible manner across different configurations and systems.

Low writing quality The impact of the writing quality in scientific articles and the associated documentation of the software has already been studied [45]. In practice, however, these aspects are sometimes overlooked. One can easily identify articles with confusing writing that is unnecessarily overloaded with complex academic jargon. Such writing is very difficult to interpret and, consequently, very hard or even impossible to reproduce.

Compilation and infrastructure setup. In applied computer science, significant time is generally spent compiling source code, debugging it, and configuring the running platform. The time spent configuring is comparable to the effort to solve the scientific problem. If building and running the program is too time-consuming, the software could be *de facto* considered as nonreproducible.

Float point operations. Given their deterministic nature, computer systems should theoretically reproduce any numerical result *per se*. However, in practice, point floating operations can give slightly different numerical results in different systems [46, 47]. This aspect needs to be taken into account in the reproducibility assessment. Float point operations errors are also analyzed by Jezequel [48] on numerical reproducibility with the IEEE 754 encoding.

Adapted operating systems. Replicated environments require installing the exact version and its respective dependencies to obtain the equivalent result, and doing so is very difficult with no isolated systems. Conda resolves this difficulty at the Python level by creating isolated environments (env); Equally, the same mechanism is applied at the operating system level in "NixOS/Guix: *a Purely Functional Linux Distribution* [49]. One benefit to reproducibility is that Nix creates packages that are isolated from each other. These packages ensure that the environments are reproducible and have no undeclared dependencies. For example, BioNix [50] is based on the characteristics of Guix.

Lack of academic reward. One major problem in reproducibility concerns the career advancement [51] of researchers in academia and how their work is acknowledged and eventually rewarded [52, 53]. Traditionally, recognition and prestige have been associated with the number of publications and citations [54] in high-impact factor publications and have been decided according to metrics such as the h-index-h, Altmetrics, CiteScore, the Clarivate Analytic Journal Impact Factor (JIF), the Source Normalized Impact per Paper (SNIP), the SCImago Journal Rank (SJR), and the proposed Scientific Impact Factor (SIF) [55]. The abuse of these metrics to decide career advancement promotes behaviors in scientists and journals that are detrimental to reproducibility, such as reticence or reluctance to publish code and data in articles or to split a single research into several non-significant articles because of the high pressure to publish. Much of the effort required to perform quality and reproducible research is not usually reflected in traditional metrics [56]. In particular, the rewarding of computer science publications is limited to granting reproducibility badges [57, 58]. Currently, at least 138 computer science journals award reproducibility badges [2].

[2] `https://cos.io/our-services/open-science-badges/`

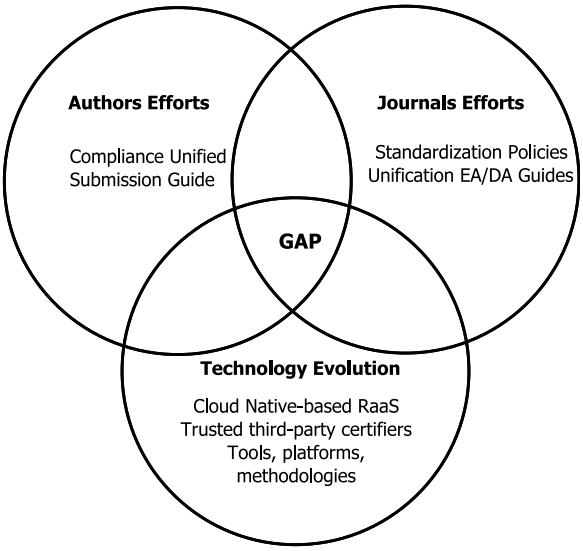


Figure 1: This diagram summarizes the structure of this paper, where three aspects are analyzed: efforts required by authors and publishers and the technological evolution required to reproduce results. The combined efforts of these three actors are required to close the reproducibility gap.

Moreover, some of the current incentives produce perverse behaviors in a hyper-competitive environment [59] which certainly goes against ethics and scientific transparency. They have also promoted the rise of *predatory journals* [60] based solely on Article Processing Charges (APC), articles of low-quality review standards, uncorroborated, or even false claims.

1.4 Measure and Evaluate Reproducibility

Reproducibility can be measured from different points of view, including the type of reproducibility which is evaluated (bitwise or statistical), the type of data [62] and field of the research [15]. It has already been shown in multiple works [24] that executing the same code on a different machine might not necessarily produce the same numerical results, but one can establish that a result is statistically equivalent to other [63, 64].

The survival analysis proposed in [24] permits to extract new insights that better explain past longitudinal data and extend a recent data set with *reproduction times*, taking into account the number of days it took to reproduce an article [65].

This point is certainly important because it is imperative to measure reproducibility to evaluate the degree and percentage of reproducibility of an article. As will be seen in the artifact evaluation section 2.8, there is a wide disparity among journals/conferences in the criteria and policies for describing and evaluating artifacts partly as a result of the difficulty in measuring reproducibility.

2 Reproducibility Strategies and Technological evolution

Motivated by the great reproducibility challenges [66, 67], there is extensive literature on data science projects, including current approaches for executing big data science projects [68], and the

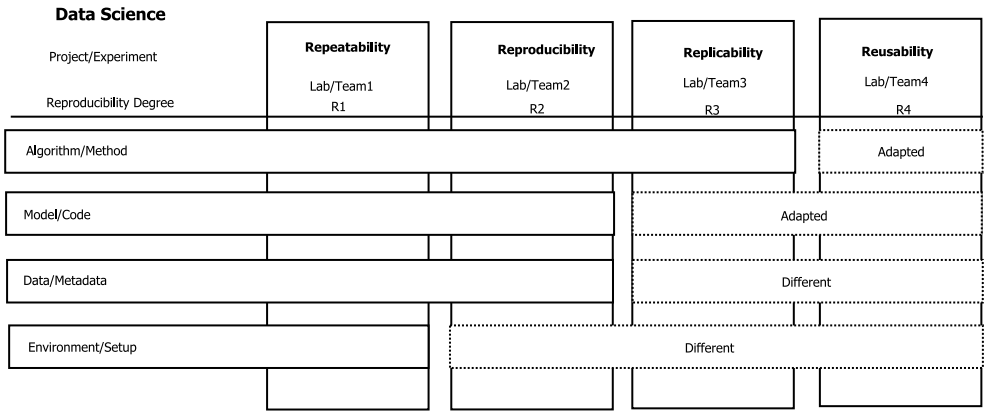
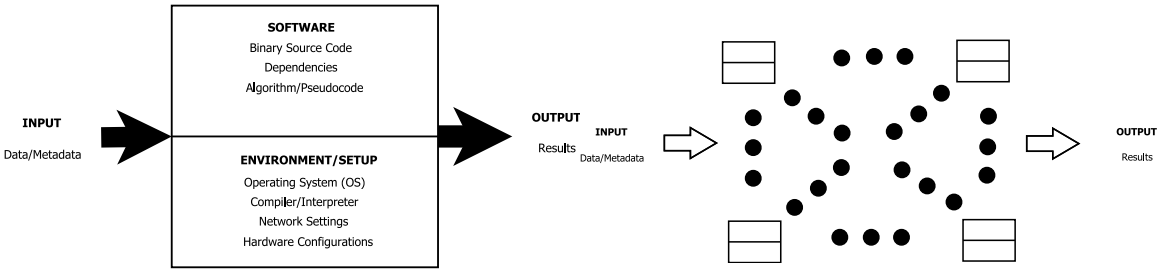


Figure 2: Definition of Reproducibility, Replicability, Repeatability, and Reusability (4R) [61]. Different degrees of reproducibility can be considered according to the characteristics of the particular experiment or project.



(a) Basic block Software/Data and Hardware/Environment Reproducibility.

(b) System/Flow Reproducibility.

Figure 3: Generalization of an architecture allowing for reproducible projects or experiments. It is made of basic blocks, interconnected to build complex systems, applications, and workflows.

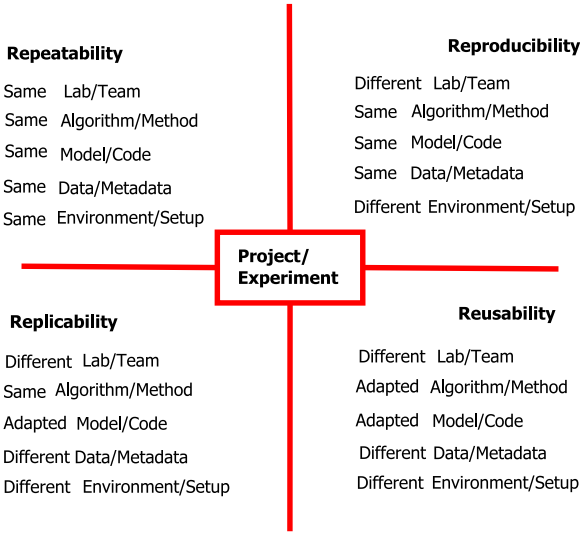


Figure 4: A classification of Repeatability, Reproducibility, Replicability, and Reusability (4R) [61] according to the characteristics a project or experiment.

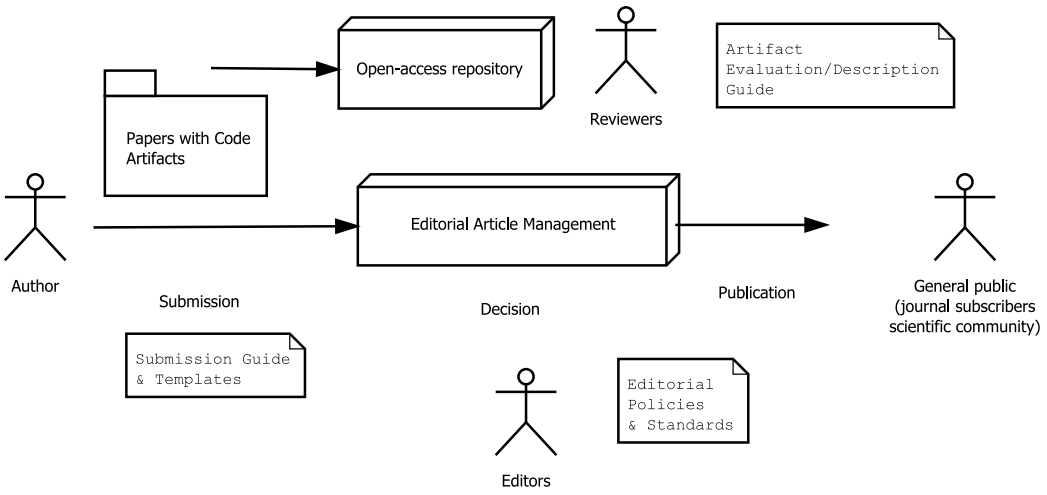


Figure 5: General description of an editorial process for the publication of a scientific article with code and the involved actors: authors, editors, reviewers, and readers.

best coding practices to ensure reproducibility [69]. Different strategies have been proposed [70] to tackle the problem of reproducibility of scientific works, specifically in ML/AI.

Given that the size and scope of data science projects that can range from small projects of the Internet of Things (IoT) [71] to very large high-end distributed HPC [72] as complex infrastructure required for example for the recently popular Large Language Models (LLM), different strategies are required to address the complexity of each project/experiment specifically.

In this section, we survey the most relevant characteristics that can be considered as general reproducibility strategies [32], such as the use of open source software, open repositories, and open data formats, the use of well-established methodologies and following good practices, or system architectures which are typically used in systems dedicated to run ML/AI applications.

We classify the strategies identified in the literature into four main classes: software and data, environment, system data management and workflows, and methods. Each strategy can be part of a more complex one. For example Workflows can be themselves made of Containers, and Code publications strategies require the Open code/data repositories strategy.

1. **Software and data reproducibility**
    - Adoption of free and open source software
    - Tools with the potential to be used as reproducibility tools. For example, notebooks.
    - Standardized automation benchmarks, open dataset formats, state of the art model baselines.
2. **Environment reproducibility**
    - Software reproducibility: containers and virtualization
    - System architecture: monolithic, microservices, server-less functions
    - Hardware reproducibility, including ambient configuration. For example, Infrastructure as Code.
3. **System and workflow reproducibility**
    - Data science project-life cycle management tools
    - Metadata and provenance (lineage and traceability)
    - Reproducibility as a Service. This includes 3rd-party specialized and trusted entities that certify reproducibility. They typically also offer services for the execution of algorithms on the infrastructure they provide.
4. **Methodological reproducibility**
    - Adaption of good practices and methodologies
    - Teaching and reproducibility culture
    - Performing evaluation specifically for research code and data artifacts
    - Publications with code (journals and conferences)

2.1 Open Source software and open repositories

One could think that uploading code and data to a public repository and labeling it as open-source software could be a sufficient guarantee of reproducibility and transparency in research [34, 73]. There have been objections to this approach [74], as well as proposals for evaluating the reproducibility level [75]. Others have proposed concrete solutions to the problem is reproducibility and transparency of scientific software [76, 77, 78]. It cannot be ensured that code will not be modified after publication[3], or that the code is executed in exactly the same environment, dependencies, and parameters. In many cases, the full reproduction of the work cannot be achieved and often requires contacting the authors to obtain detailed information. It might happen that even the authors themselves cannot replicate the experiment due to changes in their own research infrastructure, lack of documentation, or being outdated as the project evolves [79].

As significant examples of these repositories and open source communities, we could cite, for example Github, Bitbucket, Gitlab, Zenodo, the Open Science Framework [4], OpenAIRE (Open Access Infrastructure for Research in Europe), COAR (Confederation of Open Access Repositories), the French open document repository HAL, EOSC (European Open Science Cloud), HuggingFace, the Harvard DLhub [5], Dataverse [6], among many others.

2.1.1 FAIR data

Data in reproducible research should be Findable, Accessible, Interoperable, and Reusable (FAIR). Indeed, an open-source code can very well end up being non-reproducible without proper access to the data.

Findable: Metadata is assigned a globally unique and persistent identifier. For example, the Minimal Viable Identifiers (minids), or the Software Heritage SWHIDs; **Accessible**: The metadata is retrievable by its identifier using a standardized communication protocol; **Interoperable**: Metadata uses a formal, accessible, shared, and widely applicable specification for knowledge representation; **Reusable**: metadata is described in detail with a plurality of precise and relevant attributes.

The FAIR principles [80] applied to data allow promoting the reproducibility of the scientific publications focusing specifically on its R (Reuse) aspect. However, each research work is a particular case. Compliance with the FAIR principles is a real challenge that is usually only partially achieved [81]. These principles aim at categorizing the data in a more extensive and systematic way [80], as a mean to improve research data services. Also, they promote a convenient tripartite categorization of research data artifacts.

Many data science projects and research labs have started to adopt the FAIR principles, but there are discrepancies in how to implement it, from considering how to handle big data and using cloud-native repositories [74], as well as smaller scale data science projects that require an affordable means of sharing [82]. Each team or laboratory ends up establishing its own means to comply with the guiding principles. Therefore, it becomes a challenging task to determine the degree of FAIR compliance and, to a certain extent, audit it.

[3] Indeed, the history of a repository in Github can be altered with a *hard push* command or using the corresponding tools provided by GitHub.

[4] `https://osf.io`

[5] `https://www.dlhub.org/`

[6] `https://dataverse.org/`

2.2 Open data formats and benchmarking

There are cases in which it is not viable to publish datasets and codes because they contain sensitive data (for example, medical data corresponding to individuals) or simply because it is under industrial secret. In these cases, the assessment of the reproducibility of the methods is compromised [62]. However, is in developing the concept of Federated Learning [83, 84] as a novel paradigm based on decentralized and private data for the shared training of models.

However, these exceptions are scarce and, in general, it is possible to approach open science by the use of open datasets, standardized formats, baselines, and benchmarks [85], allowing the scientific community to check the results published methods reliably.

Even when data can not be made available because of confidentiality reasons, one can rely on benchmarking and comparing results without necessarily accessing the data itself. There are several recent tools for this purpose [85], such as DataPerf, Mlperf, Collective Mind [86], ReQuEST [87] or MLCommons [7] with MLcube[8] among others. They attempt to determine the state of the art in certain disciplines by comparing the performances. Competitions such as Kaggle or BRATS (brain tumor segmentation) [88] challenge and others publish open datasets and have become a reference for the industry to evaluate and compare models.

2.3 System architecture

In terms of architecture for ML/AI systems, one can find two major trends: the deployment of microservices and the use of serverless functions.

Microservices allow the building of scalable and flexible software systems for which each component works independently and can be reused in different contexts. Because many applications of ML/AI require large resources in terms of computations and storage, they are usually deployed as a distributed system. The fact that different modules can work autonomously contributes to the reproducibility and understanding of the system, compared to monolithic ones [89].

In particular, to reproduce scientific experiments, microservices can help improve the portability and reusability of the software. By dividing the components of an experiment into microservices, one can increase the flexibility and modularity of the software, making it easier to adapt the code for new tests or experiments and lessening the dependency on software specific to a development environment.

Also related to isolation, Serverless Computing is a popular cloud-based computing model [90] where the cloud provider manages the server infrastructure and platform resources, allowing developers to focus on application logic. Depending on the provider, they can also be referred to as lambda-functions[9].

The use of these serverless functions is beneficial for reproducibility in computer science as it reduces the complexity and variability of the underlying infrastructure, and it enables greater modularity and automation when developing applications and services.

[7] `https://mlcommons.org`

[8] `https://mlcommons.org/en/mlcube/`

[9] Note that, despite their name, they are totally unrelated to lambda calculus!

2.4 Tools and platforms

This section surveys tools and platforms that are commonly used in ML/AI applications and how they contribute to reproducibility. Specifically, we focus on containers and cloud computing and the Infrastructure as Code (IaC) technique.

2.4.1 Notebooks

In data science, the use of notebooks has been popularized because of the opportunity to incorporate executable code, rich visualization, and documentation in the same document. It has become a common practice to publish and share work and a step forward for reproducibility. However, it has been shown [91] that this approach has some deficiencies, such as the lack of version control. Very recent studies [92] have also studied the low degree of reproducibility of Jupyter notebooks in biomedical publications.

Several solutions have been proposed to address these challenges [93], including the use of Python scripts and the adoption of best practices for documentation, version control or additional packages as ReproduceMeGit tool to analyze the reproducibility of ML pipelines in Notebooks [94] and Osiris [95].

2.4.2 Containers and cloud computing

Advances in cloud computing and containerization have undoubtedly contributed to the reproducibility of large distributed systems.

These systems are complex, made of several interacting components [96, 97] along a pipeline. It is required to have control over the execution environment in order to not only reproduce the experiments but even trust them at all.

Given a code, the associated data, and the execution pipeline, one should be able to obtain the same results over and over again. To achieve this, the pipelines, dependencies of the software, and the environment need to be perfectly defined. Virtual machines and lightweight containers such as Docker help by defining and fixing the execution environment [98]. One could summarize these two concepts as

- Virtualization = data + code + environment,
- Cloudcomputing = data + code + environment + resources + services.

We address the topics on lightweight containers such as Docker, the MLOps methodology, the management of scientific workflows, and techniques such as IaC in the following.

2.4.3 Docker containers

Since its appearance in 2007, Docker containers have quickly become popular in computer systems and have become a fundamental tool for reproducibility. Its lightweight nature allows having several containers dedicated to small microservices on the same machine, with limited consumption and sharing of resources. This is a major advantage with respect to full virtual machines such as VMWare or Hyper-V. The light containers eventually allow for better reuse, and many infrastructures are nowadays migrating to containers, e.g. RE3 [99].

Docker is one of the most efficient and widely used tools with applications for reproducibility nowadays, but, at the same time, we identify some of its limitations [100] to this purpose compared to container alternatives such as for example, Singularity Containers for HPC.

The emergence of containerization technologies such as Docker and orchestrators such as Kubernetes [101] has allowed the rapid development and automation [99, 102] of pipelines of experiments, thus making the reproduction of complex and computationally intensive experiments possible. Indeed, they can be divided into different functional blocks.

We can cite Repo2Docker [103], which with Binder, can fetch a notebook for a given repository, create a proper execution environment, and run it inside a container. This makes the code publicly available for anyone to reproduce the results.

Specifically for HPC, there are initiatives such as The Extreme-Scale Scientific Software Stack (E4S), a community effort to provide open-source software packages for developing, deploying, and running scientific applications on High-Performance Computing (HPC) platforms. As an important contribution to the reproducibility of such a complex, E4S builds from source and provides containers of a wide collection of HPC software packages.

Many scientific experiments are made up of pipelines that concatenate several processes [104]. In terms of reproducibility over time, highly specialized platforms have been developed to manage these complex workflow management systems [105] (e.g. watchdog [106]), tools [107], roadmaps [108] and general frameworks are proposed [109]. They allow researchers can focus on solving their specific scientific problems rather than the underlying infrastructure, networking, or other technical specifics [110]. Despite the great step forward, many interoperability and reproducibility difficulties still persist [111, 112] considering the immense possibility of languages, open or private infrastructures that are currently available or under development in the ecosystem of ML/AI data science technologies.

2.4.4 Workflow management systems

Scientific workflow management systems are useful for managing complex, cloud-distributed workflows [113] and automating repetitive processes [114]. They also enable detailed documentation and workflow sharing with other researchers, thus helping improve the reproducibility of results and speeding up scientific research [115].

Formally, the workflows are represented as Direct Acyclic Graphs (DAG) [116], where a task starts in a particular node to be processed, and then transferred to the next one in the chain until the final result is available in the last node. As pointed out in Section 2, it is required that the pipeline of the workshop and the node themselves follow well-established reproducibility principles to obtain reliable results, including access to the source code running the computations, along with an accurate description of the environment, the use of FAIR data, and the use of open data formats for interoperability, among others.

Each scientific community has developed its own workflow managers. We can cite some well-known ones, such as Taverna (bioinformatics, cheminformatics, and ever social sciences) or the Galaxy project (bioinformatics), OpenAlea (Botanics), Chimera (cheminformatics), or Pegasus (physics and bioinformatics), Knime(semantic workflow), Wings(graphical workbench). Pegasus was the workflow management system used by LIGO for the first detection of gravitational waves, certainly a paramount hit in physics.

The criteria to establish the reproducibility of a given pipeline can vary much from community to community. Although the basic principles remain (see Section 2), there are specificities depending on the field. We invite the reader to check the work of Cohen-Boulakia and co-authors, who conducted a study [114] analyzing three cases of use of in-silico experiments in the domain of biological sciences

with Taverna, Galaxy, OpenAlea, VisTrails, and Nextflow, proposing different criteria and discussing about these reproducible environments based on docker, Vagrant, Conda, and ReproZip.

The significant increase in articles on AI/ML inevitably forces an adaptation towards the management of both data and software because both are a source and contribution to knowledge. Therefore it is necessary to analyze tools, infrastructures, and technologies that have evolved to support these requirements. In this sense, AIOps/MLops evolves from the DevOps/DevSecOps (Development - Operations) concept to cover several of these aspects of reproducibility management infrastructures for computer-based scientific articles.

Transferring knowledge and prototypes from the academy to the industry is, most of the time, challenging [117]. There are very well-specified methodologies for the development of software in the industry, such as DevOps, which include CI/CD (Continuous Integration/Continuous Delivery). However, in the academic environment, these practices are not necessarily followed. In part, this is explained by the lack of career reward pointed out in Section 1.3.

MLOps [118] can be considered the natural evolution of the DevOps best practices components adapted to the particular needs of ML-based software development [119]. In general terms, within data science projects MLOps tries to harmonize the practices of two environments with very different characteristics, such as academic/research environments with ML production environments for a final client, where reproducibility plays a very important role. It is an end-to-end process from the research model to the final model, exploited by the end customer or reproducibility reviewer.

There are few works that deal with MLops from the point of view of reproducibility. Among these, [120] does an excellent analysis of the reproducibility of various MLops tools.

Other articles made a benchmark for different MLops features [121] and products available in the open source such as private code [122], which is equally important when data and software management is required by a journal. Let us mention here the most relevant ones, from our review of the literature:

- Neptune. A metadata store for any MLOps workflow. It was built for both research and production teams that run a lot of experiments [10].
- Weights&Biases. A machine learning platform built for experiment tracking, dataset versioning, and model management [11].
- Comet. An ML platform that helps data scientists track, compare, explain and optimize experiments and models across the model's entire lifecycle [12].
- Sacred + Omniboard. Open-source software that allows machine learning researchers to configure, organize, log, and reproduce experiments [13].
- Tensorboard. A visualization toolkit for TensorFlow [14].
- Polyaxon. A platform for reproducible and scalable machine learning and deep learning applications [15].
- ClearML. An open-source platform, a suite of tools to streamline your ML workflow [16].

[10] `https://neptune.ai/`
[11] `https://wandb.ai/`
[12] `https://www.comet.com/`
[13] `https://github.com/IDSIA/sacred`
[14] `https://www.tensorflow.org/tensorboard`
[15] `https://polyaxon.com/`
[16] `https://clear.ml/`

- Pachyderm. An enterprise-grade, open-source data science platform that makes it possible for its users to control an end-to-end machine learning cycle [17].
- MLflow. An open-source platform that helps manage the whole machine learning lifecycle. This includes experimentation but also model storage, reproducibility, and deployment [18].
- DVC (Data Version Control). It is a very popular tool in MLops and in the data science environment because it allows the versioning of training, testing and validation datasets in a very simple format [19].
- NextFlow. In terms of reproducibility, it allows Docker and Singularity containers technology for the creation of workflows. [20]
- Collective Knowledge [123] is an initiative that, based on its experience trying to reproduce hundreds of experiments, came to identify common patterns that are repeated from project to project. In a certain sense, it is a unifying proposal within the wide variety of existing MLOps solutions and seeks to resolve persistent integration issues.

With the emerging Internet of Things (IoT) technology and the advances in smaller devices with significant computing power, simplified ML models at the edge are possible with TinyMLOps [71]. Significant reproducibility challenges appear considering the strong restrictions of energy consumption, limited computing capacity, and heterogeneity between different devices and technologies. Also considering that you can no longer containerize/virtualize with Docker.

2.4.5 Workflow languages

Despite the efforts to unify existing workflows, each community has kept its own particularities, including the language to define the pipelines [114]. This fragmentation [124] makes it harder for integration and interoperability between different academic groups. Indeed, some of the groups use a very particular language for their workflows.

There are initiatives such as SHIWA (SHaring Interoperable Workflows for Large-Scale Scientific Simulations on Available DCIs) [125] which try to provide a solution to this problem of interoperability. Multiple organizations and providers of workflow systems have also jointly worked to propose the Common Workflow Language (CWL) [126] with the aim of standardizing the pipelines around a common language.

Those specifications propose a conceptual workflow language to describe high-level scientific tasks, with the aim of promoting workflow specification portability and reusability and addressing the heterogeneity of workflow languages.

2.4.6 Infrastructure as code (IaC)

Much attention is paid to source code and containerization in order to address reproducibility, but unfortunately, not that much to hardware [127]. With the rise of cloud computing technologies, the possibility of replicating the exact execution environment for an experiment is viable. Indeed, for reproducibility purposes, it is a requirement to define the characteristics of the hardware, such as the type of CPU, TPU, GPU, memory amount, or network architecture. This is especially important for a large distributed system as, for example, HPC applications.

In this respect, IaC provides several advantages towards reproducibility in computer science. One of the main benefits is that IaC allows researchers to accurately define and control their infrastructure in

[17] `https://www.pachyderm.com/`
[18] `https://mlflow.org/`
[19] `https://dvc.org/`
[20] `https://www.nextflow.io/`

a format that can be easily stored, versioned, and shared, making it easy to reproduce experiments and obtain the same results at each execution. Defining infrastructure as code discharges from manually configuring infrastructure resources and allows researchers to easily version and share the infrastructure configuration with other colleagues. According to the Octave 2022 report [128], the Hashicorp Configuration Language (HCL) programming Terraform languages were widely used by developers in 2022, indicating that IaC practices are becoming quite popular for Github projects.

Additionally, IaC can help improve consistency and accuracy by ensuring that all instances of the infrastructure are created and configured identically. This helps ensure that the test conditions are the same each time an experiment is performed. IaC in the academic environment can significantly help in many aspects, such as the quality of the software developed, and is a step forward in the reproducibility of scientific research. As a recent example, Adorno-Gomes and Serodio [129] managed to define a complete experiment with IaC from a unique high-level code with Pulumi [130].

2.4.7 Provenance and Metadata Traceability of Artifacts

Provenance refers to the way in which the origin [131] of the artifacts of an experiment is documented in metadata. Provenance documentation is a commonly used technique to improve the reproducibility of scientific workflows and research artifacts. There are numerous articles proposing tools such as ProvStore [132], ReproZip [133], MERIT [134], CAESAR [135], Provbook [107] in several different disciplines and research areas [136], demonstrating how it can help improve traceability, linage and transparency of results.

The PROV standards allow the task to be carried out (see Openprovenance[21], for example. However, it is not yet complete and does not allow it to be generalized to multiple cases and languages. The foregoing requires the use of permanent, Unique Identifiers and tools that manage this aspect in order to have correct traceability of data sources and artifacts, even using new technologies such as blockchain [137] InterPlanetary File System (IPFS) [138] to achieve traceability and lineage of Software or code snippets.

2.4.8 Reproducibility as a Service (RaaS)

The *Reproducibility as a Service* (RaaS) concept was proposed in 2021 by Wolsin [139]. An strategy based on RaaS takes advantage of the availability of cloud computing technology to offer reproducibility services. This include the reproduction and research artifacts after the execution of the software in the controlled environment and its evaluation, validation, and certification (related to this, see Section 2.8 about code review). Also, granting reproducibility badges, tracking the provenance of software, or assigning persistent identifiers to the software at different granularity levels. Another responsibility of RaaS is to manage the underlying architecture if a way that makes it easier for authors to share and execute their code depending on the chosen complexity, from baremetal infrastructure to fully managed services. Figure 6 shows how a SaaS architecture is organized in a complex system.

Crick and co-authors proposed [140] to make a first approach to the offer of reproducibility services for journals/conferences from an empirical and quantitative point of view. They presented a *cyber-infrastructure* and the associated workflow for a reproducibility service as a high-level technical specification without delving into technical details. On the other hand, the work of Demchenko [126] addresses the topic of provisioning on demand of research environments and introduces the concept

[21] `https://openprovenance.org/store/`)

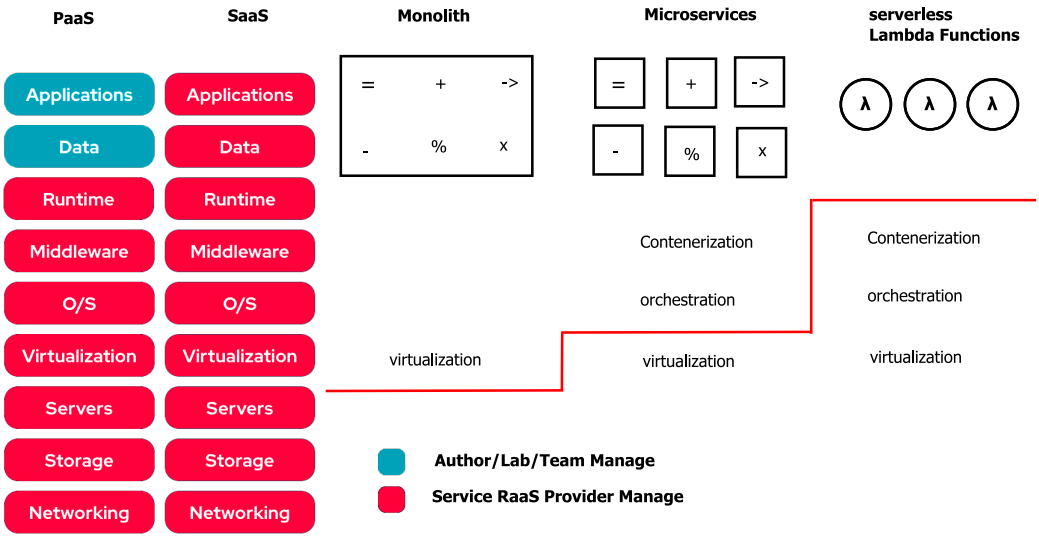


Figure 6: Representation of a RaaS-managed cloud infrastructure. The description layers, microservices, the serverless approach, and taking care of the granularity of the software help the reproducibility of a complex system.

of Platform Research Infrastructure as a Service (PRIaaS) with the aim to ensure data quality and support effective data sharing.

For example, the IPOL journal [141] also partially meets the attributes of what can be considered as a RaaS tool, together with the article, makes available a technological platform for the creation and execution of online demos (simplified demonstrations of algorithms).

Equally, among other existing reference platforms, we could mention CodeOcean, Chameleon, and Whole Tale. They allow code to be executed in a wide range of languages, but they are still maintained at the demo level with certain technical restrictions to offer the mentioned features. They start to be actively taken into account by publishers.

2.5 Good practices and data management methodologies

Agility and security are among the many quality attributes of software [142], even though the majority of them are not specifically designed for reproducibility in computer science. However, data project management methodologies and well-known best practice guides are applied widely across the AI/ML industry to improve reproducibility.

Several studies [66] and best practices guides [143] have proposed different tools for the management of data science project artifacts [121] as well as methodologies. For example, Goodman et al. propose ten simple rules to achieve reproducibility, and *The Turing Way* handbook also provides a relevant compilation of good practices [144] to reproducible, ethical, and collaborative data science projects.

There is a consensus that one of the main factors limiting the success of data science projects is the lack of reproducibility in the management platforms [8, 145, 146]. From the many methodologies available, the most popular are CRISP-DM [147], KDD, SEMMA, Microsoft TDSP, Agile DS Lifecycle, Domino, DS Lifecycle, IBM FMDS, RAMSYS, and MIDST, among others. These are widely

used in the industry, especially CRISP-DM.

Certainly, as observed in the scarce literature, there is not a standard or unified methodology that is focused on reproducibility for data management itself. So far, only good practices, recommendations [148, 149, 150, 151], and guides from different fields of computer science of different needs are available.

2.6 Scientific publishers and reproducible research

Historically one of the main forms of communication, recognition, socialization, and validation before the scientific community are the articles published in journals and conferences [79]. Publishers *de facto* become auditors of the scientific activity, and indeed, the metrics (Impact Factor and others) that they have established are the typical indicators that are used to evaluate researchers in their career and their advancement. Publishers have, therefore, a responsibility to assure the scientific integrity of the work they make public, along with their own interest in maintaining their own reputation. This includes not only avoiding fraud but also establishing clear quality criteria. In scientific publications, reproducibility is fundamental since it allows others to verify if the same or equivalent results are obtained when repeating the experiment, thus allowing them to potentially refuse a paper containing wrong or inaccurate claims. As pointed out by Heesen [152], *the work that is not widely shared is not really scientific work.*

In the following, we discuss two significant initiatives which have been put into practice by publishers: the possibility of associating code with the publications and the proper evaluation of software artifacts.

2.7 Publications with code

Associating source code with a particular publication is gaining great popularity in the scientific and technical community [153]. It allows for greater transparency and reproducibility, essential to guarantee the quality and reliability of the results [154]. However, the reproducibility aspects of this practice are evaluated in the Dataverse repository [155], Figure 5 describes a typical editorial structure for publishing articles with code, which, as will be seen later, is not so easy to implement by the Journals considering technical and economic aspects.

Many conferences have started to request that the source code be given and made public. Others go one step further and perform an exhaustive evaluation of the artifacts. For example, Checklist NeuroIPS[156]: It is a widely recognized checklist for the reproducibility assessment of conference papers.

From many examples, one might include here **Code Ocean**, used by IEEE's publishers after the integration of the CodeOcean's platform as a Computational Research Platform, **Whole Tale** [157] allowing researchers to create and share scientific narratives that include data, code, and runtime environments [158, 159], **Binder** as a platform that allows users to create and share code execution environments online, making it easy to reproduce and distribute results, **PapersWithCode** with open resources on ML, **ReproducedPapers** with open teaching and structuring machine learning reproducibility [160], or the **ReScience Journal** [161] which replicates computations from independent open-source implementations of original research and the advanced Chameleon[22] large-scale edge to cloud tool [162].

[22] `https://www.chameleoncloud.org/`

Unfortunately in many cases, this is limited to providing a non-persistent link [163, 164] to the source code repository in public platforms (see Section 2.1). Moreover, each journal sets its own strict criteria, formats, and procedures for authors. Aspects such as consistency, reproducibility, and reusability cannot be properly tracked or audited by other teams and research over time, thus limiting their impact [165].

2.8 Review of Research Artifacts

To begin with, it must be understood that for different reasons [166] an article is not 100% reproducible, but rather certain elements (e.g. computational artifacts, pseudocode, algorithms, demos) that the author decides to share and considers sufficient grounds to legitimize his results.

The evaluation criteria to accept articles for publication is traditionally well defined for scientific journals. They are based typically on originality, novelty, or overall scientific interest. However, when considering a publication not only as the article but also all major research artifacts, including source code, the criteria is relaxed, if considered at all. When the evaluation takes into account the associated source code, it is required to establish the proper evaluation criteria for peer review [167].

Conferences have started to publish guides containing checklists for the evaluation of artifacts and to grant the so-called *reproducibility badges* [57, 168] if the conditions are met. Among the most important conferences, we can cite the checklist of NeurIPS 2019 [23], the ACM reproducibility badges [24], as well as other initiatives such as the Unified Artifact Appendix and the Reproducibility Checklist [25], the CTuning artifact evaluate [26] or the Empirical Evaluation Guidelines SIGPLAN NISO RP-31-2021 [27], among others.

Following several of the published guides, recently, the SC23 supercomputing conference (one of the most important conferences in HPC) [115] adopted the Reproducibility Initiative where *accepted papers with available artifacts* were acknowledged with the corresponding ACM badges. The use of blockchain technology for artifact traceability has also been proposed [169, 138].

CTuning has participated in the artifact evaluation task for different ACM conferences [170] and has defined a more detailed Unified Artifact Appendix and the Reproducibility Checklist based on the previous evaluation experience in ACM ASPLOS, MLSys, MICRO, and SCC'23 conferences.

Other specialized scientific journals have already implemented specific criteria to a greater or lesser degree. For example, Table 6 in the Appendix summarizes the checklist for Artifacts Description/Artifacts Evaluation (AD/AE) reproducibility [171, 172] for a data science experiments and projects of different publishers.

Finally, reproducibility-certifying agencies are starting to offer their evaluation as a service in different disciplines working with sensitive or confidential data, outsourcing this function as a trusted third party. Recently Cascad [173] has been proposed in the field of Economics and Management [169].

From our review of the data above, we observe that the existing criteria are still quite varied, not standardized, complex for the authors to fulfill, and time-consuming on the reviewer's side.

[23] `https://nips.cc/Conferences/2019/CallForPapers`
[24] `https://www.acm.org/publications/policies/artifact-review-badging`
[25] `https://ctuning.org/ae/checklist.html`
[26] `https://ctuning.org/ae/reviewing.html`
[27] `https://www.sigplan.org/Resources/EmpiricalEvaluation/`

Table 4 is an extensive summary of the reproducibility strategies and technologies that have been reviewed in this work. However, it needs to be analyzed how they are implemented according to the reproducibility policies of the different scientific journals. Our survey tries, from an empirical point of view, to provide insights on the application of these strategies directly from participating journals.

3 Survey in Computer Science Journals

The increasing number of published articles in computer sciences, as well as and the fast development of new and innovative AI/ML methods, pose several challenges to publishers. The reproducibility, legitimacy of the works, and adapting the policy of the journal and the procedures of evaluation of the works is challenging [174].

Several papers [175] have explored the effectiveness of journal policies regarding open source code and data sharing to validate the research procedures in an attempt to mitigate the *reproducibility crisis*. Likewise, other studies address solutions, platforms, technologies, mechanisms, and procedures for the reproducibility of scientific articles that have been proposed to deal with the problem from the perspectives of the different actors involved: authors, publishers, industry, and scientific community. For example, the work of Gomes et al. [166], as well as Baker et al. [43], focus on the barriers why authors might reluctant to share code and data in their publications and why that would be pertinent.

From another point of view, the *reproducibility culture* has also been analyzed in previous works [176, 58, 177, 178, 179], and along with the culture, it has been discussed how to teach reproducibility in academic environments to young students and as seen in previous sections incentives [59], Massive Open Online Courses (MOOC)[28][29], good practices in the way of measuring and rewarding reproducibility, such as novel badging mechanisms, new measurement indices of valuation and code/data citation [30] [180], the Scientist Impact Factor (SIF) for reputation and impact of researchers [55], or applying statistical methods a mean to measure impact [181]. However, all these elements have not been analyzed as a whole as part of an articulated, agreed-upon *Reproducibility policy* in publishers.

Moreover, the experience, opinions, and results of journals implementing and adapting their policies with a strong focus on reproducibility have not yet been surveyed. Therefore, we surveyed SCOPUS-indexed journals specialized in computer science to know from them as a primary source of information about their experience in the application of reproducibility policies, insights, and the difficulties and successes derived from their policies.

In subsection 3.1 we explore which aspects were previously surveyed by other authors, including especially relevant questions. Subsection 3.2 presents our survey, along with the answers, which eventually leverages the discussion at Section 4.

3.1 Previous work

From the existing literature one can conclude that there is still incipient and timid progress toward implementing sharing and open science policies in scientific works [182, 79]. The traditional peer review scheme is maintained, with slight variations, and it is, in general, limited to encouraging the publication of the source code and data in software repositories [183, 184]

[28] https://www.fun-mooc.fr/en/courses/reproducible-research-methodological-principles-transparent-scie/
[29] https://www.coursera.org/learn/reproducible-research
[30] https://datacite.org/

For example, The Diamond OA Journals Study [185] makes a general survey; in our case, the results of its question 41 are highlighted. To the question "*Do you have any policy or practice to stimulate open sharing of research data?*" 42% of the respondents declared to have policy or practice to stimulate open sharing of research data. The study finds that an equal number of respondents who did not have an established policy and an additional 15% answered "*Unknown*". However, the factors that explain the adoption of open-data policies are not analyzed and it only focuses on other aspects of the publishing business.

Question 54 asked "*Does the journal require linking to data, code, and other research results?*". Although there is not much information available from journals about requiring links to data, code, and other research outputs in DOAJ, from the survey data the study found that nearly half of respondents reported not requiring this, against 24.8% who do. For more than 25% the answer was "*No*" or "*Unknown*".

The above questions are certainly limited to code-sharing policies in journals, but do not delve into actual reproducibility policies through article automation, evaluation, and preservation of reproducibility technologies. This represents a dilemma that is discussed in Section 5.3.

In the article [186], 318 biomedical journals were manually reviewed to analyze the journal's data sharing requirements and characteristics A total of 11.9% of journals analyzed explicitly stated that data share a total of 11.9% of journals analyzed explicitly stated that data sharing was required as a condition of publication. A total of 9.1% of journals required data sharing but did not state that it would affect publication decisions. 23.3% of journals had a statement encouraging authors to share their data but did not require it. A total of 9.1% of journals mentioned data sharing indirectly, and only 14.8% addressed protein, proteomic, or genomic data sharing. There was no mention of data sharing in 31.8% of journals. Impact factors were significantly higher for journals with the strongest data-sharing policies compared to all other data-sharing criteria. Open-access journals were not more likely to require data sharing than subscription journals.

Another contribution by Konkol et al. from the point of view of the analysis of reproducibility technologies for Publishing computational research [187] concludes that still, publishing reproducible articles is a demanding task and not achieved simply by providing access to code scripts and data files. Several platforms were analyzed, including Whole Tale, ReproZip, REANA, o2r, Manuscripts, Gigantum, Galaxy, eLife RDS, Code Ocean, Binder and its limitations as well, the facilities it offers for authors. The previous article is complemented by the work of Willis [188], who made an analysis of technical aspects and use of some of these technological infrastructures and repositories around seven reproducibility initiatives designed by journals to improve computational reproducibility.

In the work of Malik [172], the technical difficulties are discussed, but also the benefits of implementing Artifact Description and evaluation policies for presenting scientific articles to journals and conferences.

The percentages of implementation of concrete reproducibility policies remain low. However, there is ongoing open discussion on the efforts and contributions that can be made by each of the actors in the reproducibility research ecosystem. In our case, this work analyzes the problem from the point of view of practical implementation of policies by publishers, based on your opinion and experiences with the following research questions: *What is the best way reproducibility policy mandatory, or instead an incentive policy for authors and reviewers, to allow publishers improve the quality and impact of their publications? What type of technological infrastructure best supports these types of reproducibility policies?*

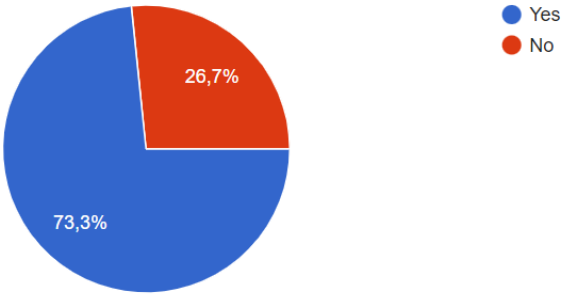


Figure 7: Do you want to be mentioned in the acknowledgment section as a Survey participant?

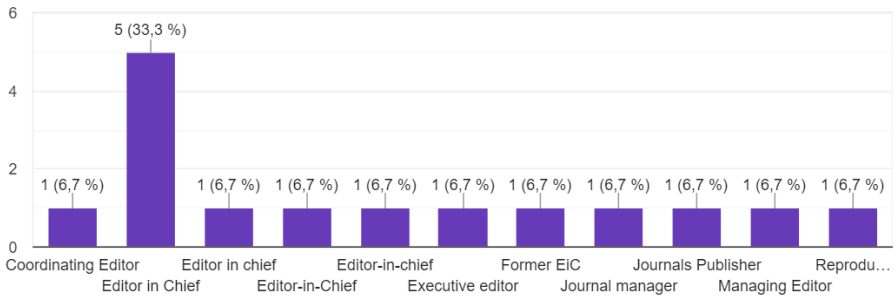


Figure 8: Respondent's role in the scientific journal.

3.2 Survey and Results

Following the methodology described in section A.2 and considering the literature mentioned above and the technologies discussed in Section 2.4, a series of reproducibility-oriented questions were carefully designed for the evaluation of reproducibility policies and their implementation. This is the base of the brief gap analysis we do in Section 5.4, where the answers are analyzed and discussed.

In the following we present the questions of the survey and the results.

Question 1. Do you want to be mentioned in the acknowledgment section as a Survey participant?

Despite having a policy of sharing and publishing code and data implemented at some level, some publishers refrained from being mentioned, probably due to not being able to match several items. Indeed, the survey asked for very specific questions about the implementation of infrastructures and technical details. Some publishers requested to be considered as anonymous in this question. Figure 7 shows the results.

Question 2. Respondent's role in the scientific journal

The answers came from a variety of different roles, with a slight predominance of Editors in Chief, Figure 8.

Question 3. Do you have a Reproducibility policy or similar in your guidelines for authors?

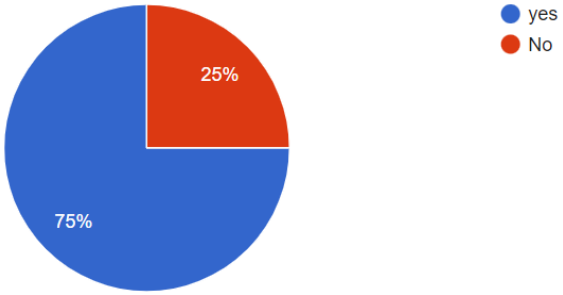


Figure 9: Do you have a Reproducibility policy or similar in your guidelines for authors?

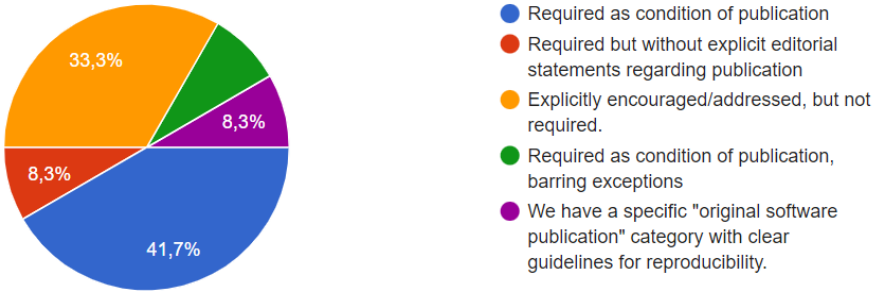


Figure 10: How do you think the reproducibility policy requirements should be?

A large majority (75 %) of the respondents indicate that they do have a reproducibility policy, as shows in Figure 9.

Related to this, in Figure 10 we can observe that there is a significant percentage (41.7%) of journals which request reproducibility as an essential condition for publication and thus make it mandatory. This decision has important consequences and, in general, it is counterproductive for almost all journals to add extra requirements for the publication because it reduced the publication rate[31]). On the other hand, it improves the overall quality of the publications.

Question 4. If you wish, you can indicate the link to the policy of the scientific journal or guides for authors.

Nine journals provided a link to their reproducibility policy, Table 1.

Question 5. How do you think the reproducibility policy requirements should be?

In this question we asked about what should be the most significant requirements for a reproducibility policy, regardless whether the journal actually implemented them or not. The results are given in Figure 11, with a variety of different preferences and showing, in any case, gradual interest towards making them mandatory.

Question 6. Do you follow any guide or checklist for the evaluation of research artifacts?

[31] See `https://scholarlykitchen.sspnet.org/2018/09/25/does-adopting-a-strict-data-sharing-policy-affect-submissions/`

Journal	Policy Link
IPOL	https://tools.ipol.im/wiki/ref/software_guidelines/
ACM Transactions on Graphics	https://www.replicabilitystamp.org
GigaScience	https://academic.oup.com/gigascience/pages/editorial_policies_and_reporting_standards?login=false#Reporting%20Standards
Anonymous	https://www.springer.com/journal/12532/submission-guidelines#Instructions%20for%20Authors_MPC%20Reviewing%20Guidelines
Optical Memory and Neural Networks (Information Optics)	https://www.pleiades.online/en/journal/optmem/authors-instructions/
Information Systems (Elsevier)	https://www.elsevier.com/journals/information-systems/0306-4379/guide-for-authors http://doi.org/10.13140/RG.2.2.34277.22243/1
Science of Computer Programming	https://www.journals.elsevier.com/science-of-computer-programming/call-for-software/a-new-software-track-on-original-software-public
INFORMS Journal on Computing	https://pubsonline.informs.org/page/ijoc/datapolicy; https://pubsonline.informs.org/page/ijoc/softwarepolicy
JETAI	https://authorservices.taylorandfrancis.com/data-sharing-policies/open-data/

Table 1: The nine journals who answered question #4 about their policy, and the links they provided.

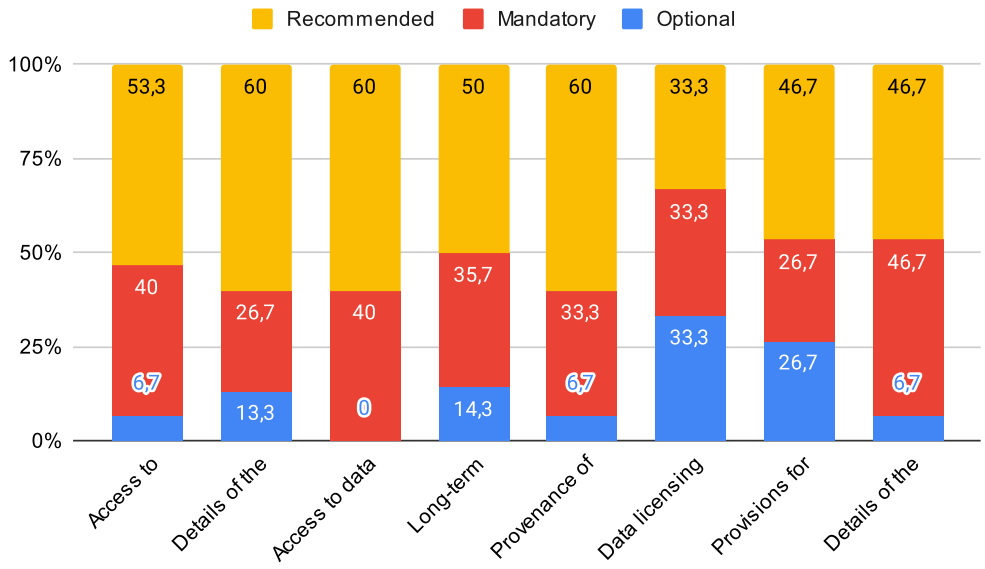


Figure 11: How do you think the reproducibility policy requirements should be?

If so, which one?

The responses were very varied, which shows the lack of standardization in this matter. The problem of the evaluation fo the research artifacts has been extensively studied, yet without much agreement or formalization. See Figure 12.

Question 7. Journal access modality

Most of the journals answered that their publication modality was open-access, Figure 13.

Question 8. What is the range of your APC (Article Publication Charges)?

The ACP are very relevant for the discussion about how the reproducibility costs are shared between authors, publishers, and technology providers (see Section 5.1). Free publication costs predominate in the responses. In addition to question #7, it is an indicator that the business model of these journals is based on open platforms and repositories.

Question 9. Preferred sharing method

This question confirms that free open platforms are used to share the code, and the number of journals that owe third parties, or have their own technological storage infrastructure, is very low. See Figure 15 for the results.

Question 10. How compliant is your publication of software and data policy with FAIR-TLC?

The answers indicate the increasing level of implementation of the reproducibility policies, considering that most of the articles are accessible and reusable, but still low in the other attributes.

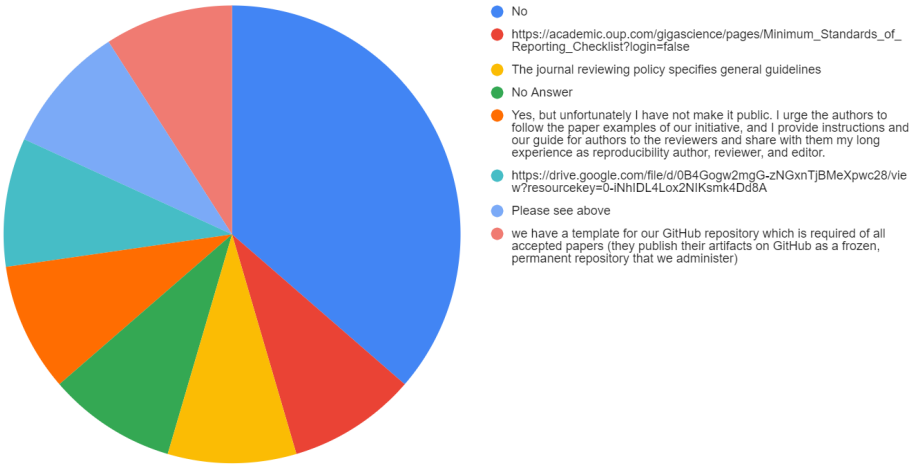


Figure 12: Do you follow any guide or checklist for the evaluation of research artifacts? If so, which one?

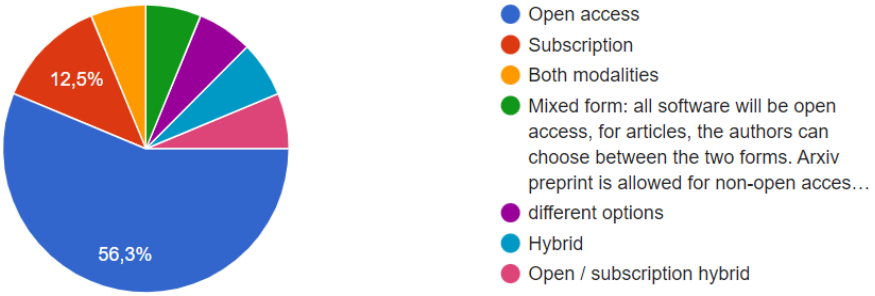


Figure 13: Journal access modality.

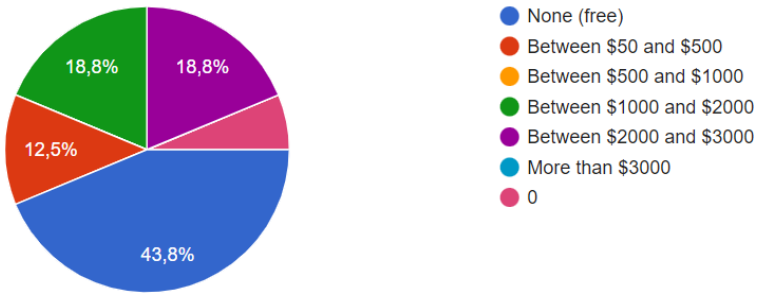


Figure 14: Which is the range of your APC (Article Publication Charges)?

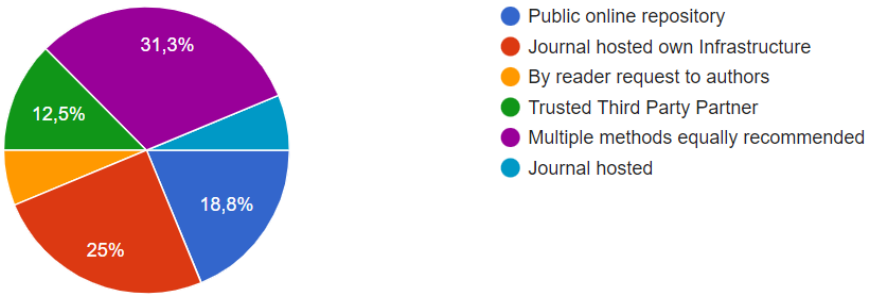


Figure 15: Preferred method to share research artifacts.

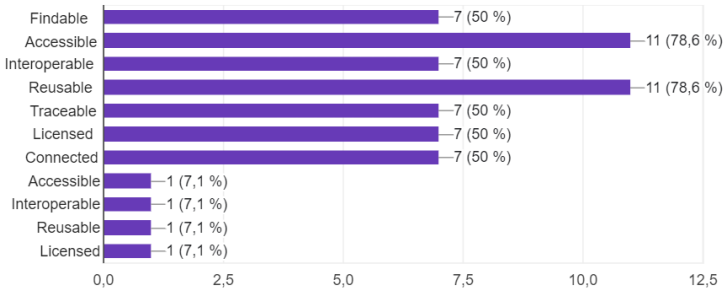


Figure 16: How compliant is your publication of software and data policy with FAIR-TLC?

This could be explained because the use of open repositories limits the journals to offer the other attributes satisfactorily. Figure 16 provides the results.

Question 11. Reproducibility validation method

The results (Figure 17) show that the traditional peer review model for article validation and acceptance is maintained, compared to other more automated forms of reproducibility validation. Therefore, validating the legitimacy of an article rests on one or two experts as well as their own available testing resources.

Question 12. If you request to share the source code. What platforms or repositories do you recommend for sharing code? If others, you can write those you recommend

The results (Figure 18) describe show that Github if the preferred specialized platform, although more for developers than for publishing research results. Zenodo, on the other hand, allows the citation of code and data through its identifiers, but remains a simple non peer-reviewed repository. There is therefore still a significant lack of automation in the policies of code and data for reproducibility purposes, to validate the legitimacy and quality of the articles.

Question 13. In the case you request reproducible research artifacts, which format is preferred?

The majority of the journals indicated that indeed they request the software and data artifacts, but as supplementary material (58.3%), with a large majority (50%) that request a link to the source

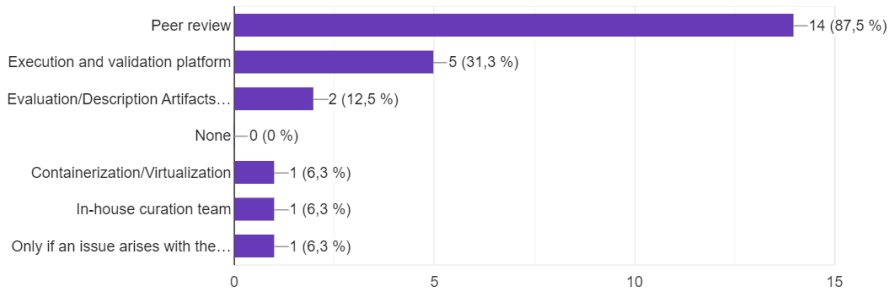


Figure 17: Reproducibility validation method.

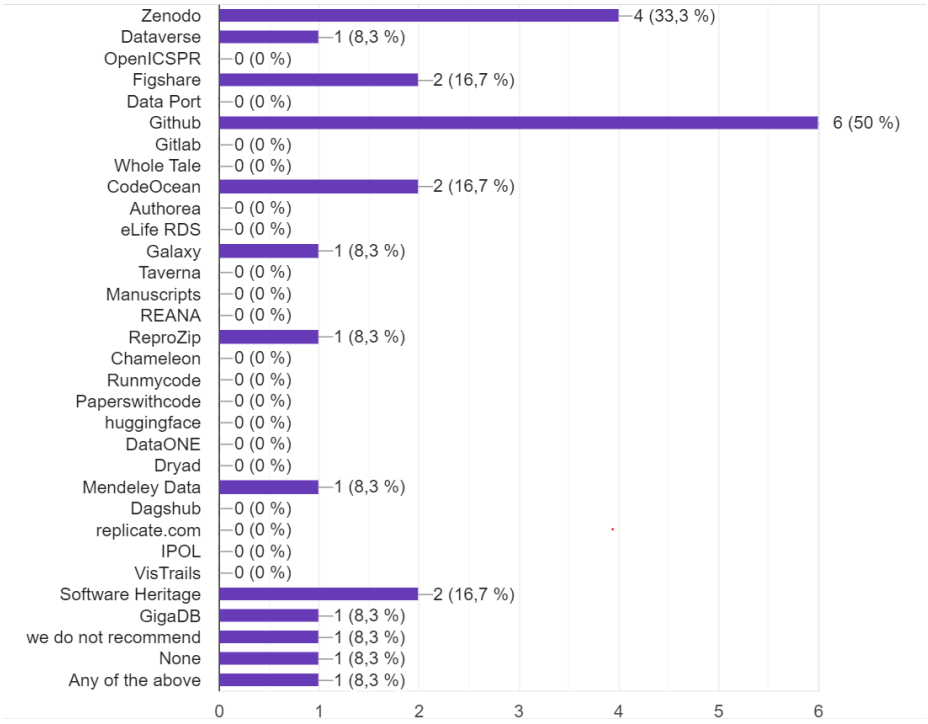


Figure 18: If you request to share source code, what platforms or repositories do you recommend for sharing code?

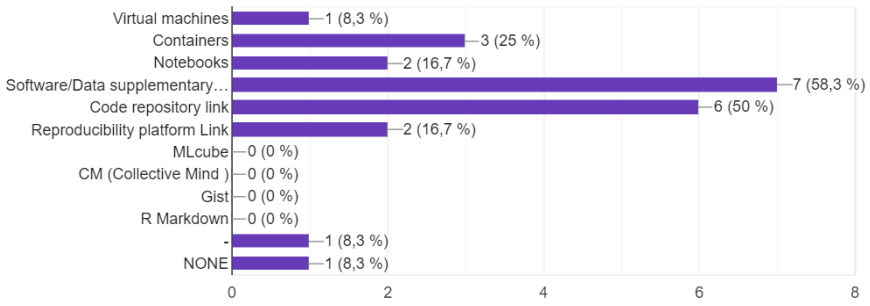


Figure 19: In the case you request reproducible research artifacts, which format is preferred?

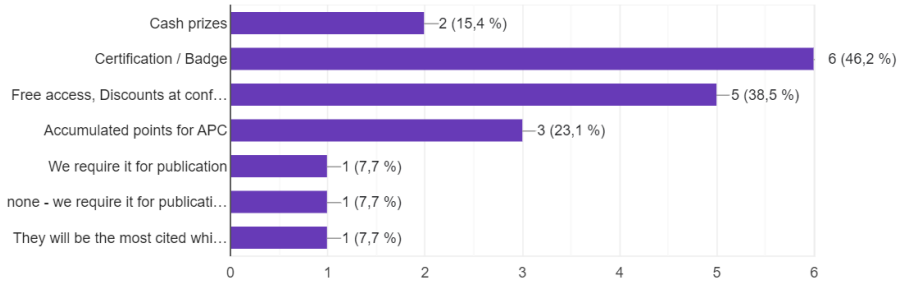


Figure 20: Which do you think would be the best way to reward the authors to submit reproducible articles?

code. Figure 19 shows the results.

Question 14. Which do you think would be the best way to reward the authors to submit reproducible articles?

Most of the answers (Figure 20) indicate that for most of the journals (46.2%) the preferred way to reward authors is to grant reproducibility badges, followed by a 38.5% of the answers which claim that the best way would be to offer free access to the journal or discounts.

Question 15. Which do you think would be the best way to reward the reviewers of reproducible articles?

The answers to this question (Figure 21) do not show a strong preference for any of the options, being the most preferred to offer free access to the journal or discounts, offering the reviewers being part of the editorial board of the journals, or even considering that it should be a voluntary task.

Question 16. Would you like to share briefly your view on the impact of implementing a reproducibility policy in the journal? For example, in terms of APC or other costs, the quality, citation impact, credibility of articles, or any other topic you would like to address.

This was an open questions to obtain from direct insights from the publishers. In the following we present their answers:

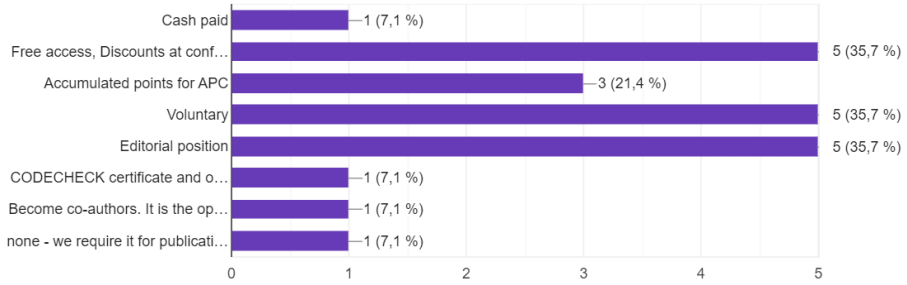


Figure 21: Which do you think would be the best way to reward the Reviewers of reproducible articles?

- *Observed average 15% increase in citations after one year, 33% after two years.*
- *It's a great selling point and definitely has increased our visibility and content reuse.*
- *We strongly encourage the authors to use Zenodo/Github as the repository for the code used in their experiments. We feel that both the reproducibility of the work and the ability of other to build upon it has a positive impact on science as a whole.*
- *Credibility of the results is the main impact. However, software evaluation adds time and complexity to our reviewing procedure.*
- *It has a deep impact on the credibility and citation of any article. I consider that our reproducibility initiative should be encouraged by most of journals. However, the most important issue is to encourage the authors to adopt a reproducibility-centered research methodology from the very beginning of their research.*
- *As an author, I always submit my articles with a reproducibility appendix, and I always design my experiments to be fully automatic and reproducible from scratch, including the automatic generation of all data tables and figures in the articles. On the other hand, as editor, I am encouraging this same approach among our authors, and most authors who have published reproducible papers in our reproducibility section are adopting this approach. However, it demands extra effort and a lot of discipline, and for this reason, not all authors are probed to do it. For this reason, our reproducibility initiative rewards the authors with a second article (reproducible paper).*

  Finally, there is a long walk to achieve most of the authors adopt a reproducibility-centered approach. It is absolutely necessary that PhD supervisors adopt and encourage this approach among their students. A very good way of achieving it is teaching all MSc and PhD students to design and implement reproducibility protocols and submitting reproducibility appendixes with all their papers.
- *We do not have APC. In most cases, as much of the generated information has normally already been paid by public institutions.*
- *Higher Impact Factor since the policy has been in place*
- *Unfortunately, we do not implement a reproducibility policy in the BSR journal, but I would like to know more about it. On the other hand, I am not fully for the transparent publication*

of the data and software since it may violate privacy and licensing.

- *APC for original software publications is less than half of the APC for research papers. Original Software Publications are currently the strategic focus of our journal.*
- *Surprisingly, authors are fine with our repository and requirements. We have not had any pushback. It does add editorial work and slows down the final acceptance. See the Github IJOC platform for how it works. https://github.com/INFORMSJoC".*
- *This is very new to us - we have only introduced it this year - to start on Sept 1st. We expect it to improve the reputation of the Journal and thus its attractiveness to researchers.*

From this answers and insights, we shall in the following discuss on the technological possibilities available to address the reproducibility problem, and about the shared responsibility between authors, publishers and technology providers, including a brief gap analysis.

4 Reproducibility technological Discussion

As pointed out by several of the works we have reviewed here, reproducibility comes with great benefits for both authors and journals. Let us briefly summarize them.

- **Greater credibility and recognition** [52]: reproducibility increases the credibility of the research and, therefore, can increase the recognition of the authors and the influence of their work in the scientific community.
- **Research results are accurate and reliable** to the extent that reproducibility is guaranteed. This is fundamental in the Scientific Method.
- **Increased visibility and impact** [189]: the attention of the scientific community is attracted when articles are reproducible, thus increasing the visibility and the impact of the research.
- **Facilitate collaboration and reuse** [145]: results obtained from reproducible experiments allow other researchers to build on them, which promotes collaboration and reuse of findings. Eventually, it allows for faster scientific advancement.
- **Increase credibility and confidence in the results** [150]: reproducibility allows the results obtained in an investigation to be verified and validated, increasing the confidence in its validity, following ethic directives [190], and transparency.

In the following, we shall discuss from the point of view of technological evolution based on the strategies presented as a reproducibility fundamental lever and support, how these interrelate into different challenges, problems, and solutions that have been proposed, and how they relate to these benefits. In particular, we discuss the problem of the responsibility of authors and publishers, such as their efforts towards reproducibility, the possibility of understanding reproducibility as a service, and finally, the impact of considering software as an important research artifact and the reward to researchers.

4.1 Dilemma: virtualization solution or dependency

As described in the technological evolution subsection 2.4, many of the reproducibility tools and platforms (eg, Workflows) proposed so far are completely based on container technology; this strong trend leads us to discuss and draw attention to the benefits and drawbacks of relying exclusively on this technology.

It could be stated that Docker or, in general, lightweight virtualization, is the *holy grail* of reproducibility, and, as will be seen, many solutions are based on this technology. With the popularization of agile methodologies and as shown by the landscape and the containerization strategy, many of the reproducibility problems try to be solved with docker [191, 100], which leads us to question if there is an abuse of lightweight virtualization. As suggested, Docker is practical light and facilitates many processes that in the past were tedious; however, it is not a tool specifically designed for reproducibility, and it cannot be used indiscriminately to hide bad practices.

The possibility of packaging, freezing, and porting a code to any infrastructure and maintaining stable functionality over time make it attractive in the scientific world; however, as stated in [192], this indiscriminate use brings great inconveniences, we will discuss in the following.

At this point, it is necessary to analyze in depth the problem of reproducibility, repeatability, containers, and development. The problem is that two characteristics are desirable in systems, but actually, they are antagonists. On the one hand, we want robust systems that will not break after an update. The classic example is a Python program that uses PyPI packages that, even if the user sets the versions in a virtual environment, the libraries may not be available in a particular version of Python. In that case, many system designers opt for virtualization.

The containers ensure reproducibility given that the complete environment is fixed. However, if proper attention is not paid to the maintenance of the container, it might end up facing security problems given that if the environment is simply fixed and not updated, the libraries will stop receiving bug fixes and security updates.

Docker is certainly a useful tool that allows to fix the execution environment, but still maintenance is required. Regular automatic testing is recommended.

4.2 The shared responsibility between authors and publishers

As presented in Sections 2.4 and 2.6, complying with the criteria for reproducibility implies some costs, as well as the shared responsibility and pooling between authors and scientific journals. It also needs a commitment to transparency and reproducibility [17]. Despite the analysis of the reproducibility stakeholders in [25, 33, 34], the roles and relationship between authors and publishers are still diffuse and at least questionable: indeed much of the reproducibility burden relies on the authors. Authors have a responsibility to provide detailed information about the methods and techniques used in their research, as well as to make public the data and codes that were used to generate the results. They must also ensure that their results are replicable and thus they can be verified by peers.

On the other hand, publishers have the responsibility to establish clear policies and guidelines [79] for the submission of scientific articles, as well as looking for the transparency and reproducibility of the results.

Indeed, guaranteeing and legitimizing the reproducibility of scientific work in ML/AI implies assuming significant economic and time costs [193] depending on the size and complexity of the research project. These cannot be assumed only by the researcher.

4.2.1 Reproducibility Cost

Estimating the cost of reproducibility is not easy because it can be considered from the execution of a simple container on a personal laptop to a distributed execution of software in the cloud, with the

market costs per hour of CPU, GPUs and storage depending on each provider and their business model (for example, GCP, Amazon, Azure, Oracle, and others).

Existing virtualization and containerization techniques and cloud computing infrastructure are key elements in this problem [98]. Therefore, the costs associated with cloud computing become relevant concerning reproducibility.

It is essential to highlight these associated costs [194] and the implications for the scientific parties that have a role in the reproducibility of the scientific work. One can describe the main technological costs for the reproducibility of computational projects/experiments as follows:

$$C_R = C_{HD} + C_{HC} + C_{RC},$$

where C_R is the total reproducibility cost, C_{HD} the cost of hosting data, C_{HC} the cost of hosting code, and C_{RC} the cost of running code.

The problem of increasingly complex research projects is not specific to computer science but common to other disciplines, especially when they combine different fields. Let's mention briefly the case of bioinformatics as an example. The researchers need to have not only knowledge of Biology but also the skills to operate the software and the data formats of the research artifacts, as well as the running environment. Typically, specialized expertise is required in Python, R [195], diverse operating systems, and database management and complex platforms such as, for example, Galaxy [196].

IaC, virtualization and containerization, and cloud computing approach help address this divisional responsibility in a simplified manner. They allow to track the steps followed by the author so other researchers can repeat and reproduce the experiment in the same environment. However, this still requires a high level of computing skills, which should not necessarily be assumed only by the authors. In Section 5.2.1, we discuss the Reproducibility as a Service (RaaS) strategy, which could be applied to manage this shared responsibility.

5 Reproducibility efforts of authors and publishers

5.1 Effort of authors

In the case of the authors, it should be understood that for several of the reasons discussed, in most cases, a scientific article cannot be reproduced in its entirety (100%). The authors generally choose to reproduce only parts of the algorithms, demos, or data, which is essential to support the conclusions.

Therefore, it is the authors' effort to seek 100% reproducibility of their work or to fully clarify the reasons that prevented reaching this objective, complying with the policies and requirements of the journals/conferences.

5.1.1 Articles Submission Reproducibility Guide for Authors

From our analysis of the shared responsibility between authors and journals and the most recent technological advances in computing, we shall discuss the efforts required by each of these two actors.

It is still very difficult for journals and authors to close the gap in a mutual effort, and it is even harder when the authors must comply with article submission guidelines between different journals.

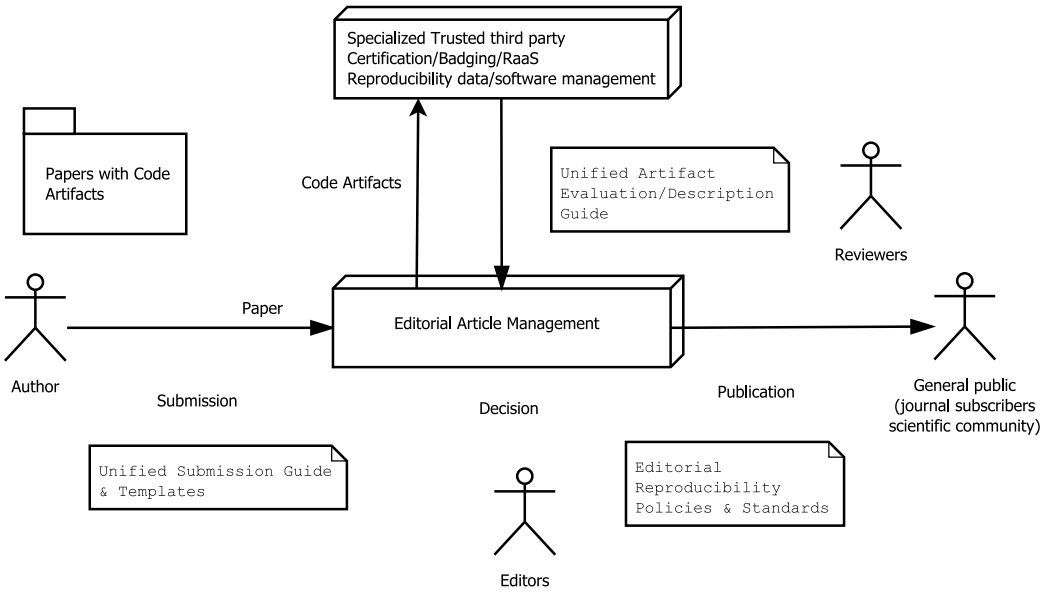


Figure 22: Proposal for a future editorial process on which the article and code are published as a whole, and third-parties certify reproducibility.

An article composed of theoretical and computational parts can only be reproduced in a certain percentage and certain components that only the author is responsible for defining and specifying with the greatest of details and following a standard guide that avoids reprocessing between publishers.

MICRO2023 is a recent experience towards unified EA (artifact evaluation) guides and procedures[32], which allow speed up the AE process. A conference where artifacts can be complex and time-consuming to evaluate and 25% of the submitted artifacts were awarded the artifact reusable badge. In this context, practices were developed such as Reviewers performed an initial 'smoke-test' (for example, installing the artifact, or resolving access/environment/setup issues) and also reviewed the key claims of the paper and the artifact. Likewise, two surveys were carried out consulting authors and evaluators to seek feedback on the AE process. Important insights are derived from this survey, especially in enabling authors and reviewers to faster iterate on artifacts efficiently, seamlessly, in reasonable time. For example Reviewers provided suggest that requesting authors to prepare a subset of simulations (and/or representative checkpoints) would be a good practice. Results "*will appear in the ACM/IEEE MICRO 2023 conference front-matter*"[33] and support a trend towards improvements to the process and clearer and standardized instructions preferable to most subjective assessment of other experiences.

Therefore, in addition to standardizing the different evaluation and description guides of Artifacts (see Table 6), we propose to incorporate a mandatory and standardized unified guide between journals where the author contributes the *effort to comply and assess* the level of reproducibility of their scientific article (see Table 2).

[32] `https://ctuning.org/ae/micro2023.html`
[33] `https://www.linkedin.com/pulse/micro-2023-artifact-evaluation-report-56th-ieeeacm-symposium-fursin-bsgwe/`

ITEM	**OPTIONS**
Article based on Software/Data?	yes/no
Programming languages used	(e.g., Python, C++)
Contains instructions for reproducibility	(e.g., complete, verified)
Badges, Certified third-party Reproducibility Evaluators	(e.g., ACM badge, Ctuning)
Infrastructure Reproducibility Required/Trusted third-party RaaS Operator	(e.g., Docker containers, MLflow, CodeOcean, Chameleon)
Repository	(e.g., Zenodo, Software Heritage)
Unique persistent citable identifiers of Software/Data Artifacts	(DOI, SWID, BlockchainID)
Percentage of reproducibility of the Article	(%)
Reproducible components	(e.g. DEMO,virtual infrastructure, figures, tables, Backend, Frontend, Microservices, Lambda functions)
Component Reproducibility degree	(R1,R2,R3,R4)
Non-reproducible components (Why)	(e.g., proprietary software, sensitive data, distributed project)

Table 2: Proposal of a *reproducibility checklist* guide for authors. We propose to incorporate a mandatory and standardized unified guide between journals where the author contributes the effort to comply and assess the level of reproducibility of the scientific article.

5.2 Efforts of publishers

In the case of journals, given the wide typology of submitted articles and the reproducibility costs, it is economically unfeasible that they have their own reproducibility infrastructure, which explains the current tendency to rely on third trusted parties. In the following, we shall discuss how the Reproducibility as a Service (RaaS) methodology could help discharge authors from the burden that implies running code and maintaining a complex reproducibility infrastructure, as well as the opportunity of considering a software more valued research artifacts and thus properly rewarding authors. Finally, we provide a brief gap analysis from the results of our survey in Section 3.2.

5.2.1 Reproducibility as a Service

As pointed out in Section 4.2, reaching reproducibility might require for some projects a large technological investment, which should not be assumed only by the authors but shared with publishers and offered by specialized third parties. Here, we will focus on a particular strategy, Reproducibility as a Service (RaaS) [139], which might be helpful to this purpose.

As introduced in Section 2, RaaS is an approach to address non-reproducibility in scientific research by providing access to tools and resources that researchers and industrial actors to replicate experiments and data science projects. Also, to facilitate, manage, or overcome many of the limitations and barriers that we have identified in our review of the literature.

According to Brundage and co-authors [197], one could label as RaaS any 3rd-party service made of tools that allow the reproducibility of scientific work. Their proposal is to use the existing cloud computing tools to offer a service that fills the gap between two major requirements to achieve reproducibility. On one hand, the actions taken by researchers who want to facilitate reproducibility. They provide a detailed procedure that allows to obtain the result artifacts, as well as the exact execution environment. On the other hand, the actions taken by publishers or the industry validate reproducibility.

Moreover, there are trusted third parties that deal with big data projects and confidentiality issues of sensitive datasets. They aim to reduce the need for the strong computing skills typically required to work in complex AI/ML data science projects. Cloud Native-based RaaS [139] adds an additional standardized layer with simplified interfaces for IaC, virtualization, and cloud computing (see Figure 6). As examples of these tools, we can cite Invenio [34], Eprints [35], or DSpace [36], among others.

Although not particularly adapted to complex workflow systems and user interactivity, a good example of the relationship between a journal and a third party that offers reproducibility services in the cloud is the partnership between IEEE and Code Ocean[37]. The code from IEEE articles can be browsed, discovered (assigned a DOI), run, modified, and eventually built the researcher's work on the cloud without any complex setup.

5.2.2 Software as a valuable research artifact and reward to authors

As presented in Section 1.3, traditionally, the published article has been considered the most important and rewarded [180] research artifact, leaving aside Software production. Universities, research

[34] `http://invenio-software.org`
[35] `http://www.eprints.org`
[36] `http://www.dspace.org`
[37] `https://codeocean.com/signup/ieee`

centers, and evaluation committees usually consider the number of articles published in high-impact factor journals and the number of citations and sometimes as the major criterion to hire a researcher for increasing the salary and career evolution, among other incentives. Consequently, researchers typically do not invest large resources in the reproducibility of the results, the quality of the produced software, or even the possibility of publishing the software itself.

Many research projects are based on software contributed by others, including libraries, applications, or complete frameworks, and in many cases, there is no explicit recognition of the authors of the third-party software. The recent incident about the vulnerability in the log4j library [198] is a good example of a widely-spread software used by hundreds of companies, not necessarily acknowledging the library's authors.

The lack of incentives for researchers and software developers to produce quality and reproducible software has a clear negative impact [199] on the development of Open Science. Fortunately, the criteria to evaluate researchers are evolving in parallel in the right direction. For example, CNRS (the Center for National Scientific Research) in France announced in 2022 changes in their evaluation policy to include SW projects as an equally valid element to evaluate scientific production. The reproducibility tools provided by journals and conferences are fundamental for traceability, thus allowing the proposal of new metrics specific to software.

Different works [180, 200] have focused on analyzing the citation of scientific software and data publishers [174] as a natural need to implement FAIR and paper-with-code strategies. To this purpose, FORCE11 (The Future of Research Communications and e-Scholarship) [200] provides guidelines for the citation of software and data.

Software needs to be properly cited and preserved. These two requirements are certainly not easy to fulfill, given their dynamic and changing nature. Indeed, millions of software repositories are constantly being updated at every instant in Github and other repositories.

The Software Heritage project, supported by UNESCO, is one important step forward in both citation and perpetual preservation of software via proper identifiers, such as the SWHID [201]. Zenodo also provides a Digital Object Identifier (DOI) and Chameleon a QR-code to reference the code. It is also a great source of information to determine the provenance of software contributions.

Regarding using badges as an incentive for authors reproducibility, it must be observed that they really impact the researcher's reputation in the same way as the popularized and mature badge system awarded in e-learning by important companies, academies, to certify technical skills[202] published on reputable platforms such as Credly[38] and easily shared on Linkedin[39], which allow the candidate to reinforce their CV and demonstrate to the employers in a competitive labor market.

As pointed out by Dozmorov at al. [181], Github is, at the moment, the most complete database to measure the impact of software. Interestingly, they concluded that the number of *forks* as a measure of software impact is not correlated with the number of citations associated with a scientific paper. This finding, at first counter-intuitive, shows that citation indices (such as the h-index and others) do not fully explain the true impact of the scientific work and the associated software. The consequence is, therefore, that producing quality software is neither properly promoted nor taken into account for the career advancement of researchers.

There is, therefore, a need for metrics that are specific to software, beyond indirect measures such as the number of *forks* or *stars* in public repositories. Strategies such as RaaS and more adapted

[38] `https://info.credly.com/`
[39] `https://www.linkedin.com/`

metrics such as the Scientific Impact Factor (SIF) [55] could be of great help rather than the H-index or impact factor of the journal (FIJ) discussed in Section 1.3.

Finally, our recommendation is depicted in Figure 22, where there are incentives for all the actors, including 3rd-parties that implement permanent and long-term reproducibility infrastructures that support the publishing business.

5.3 Dilemma: reproducibility sharing policies

At this point an important clarification must be made, *journal reproducibility policies* should not be confused with traditional open access and open science initiatives. It could even be considered as an open topic that requires standardization. In [79] Stodden makes a first approximation from the analysis of Data and Code Policy Adoption by Journals, and then in [182] [184] analysis of journal policy implementation and effectiveness for computational reproducibility, however a clear concept of "reproducibility policies" is not consolidated. This leads us to consider that journals face an important dilemma in defining their internal policy of just limiting themselves to a code and data sharing policy or going further in defining veritable and strict automation tools and reproducibility evaluation article reproducibility policy.

5.4 Brief gap analysis

We provide a small gap analysis of the level of implementation of reproducibility policies that we observed from our survey. We intended to bring together all the elements of analysis. We include technological aspects, as well as the efforts required by both authors and publishers to help close or, at least, reduce the reproducibility gap. Aspects such as the standardization and implementation of reproducibility policies, adaptation of business models, and association with specialized third parties are considered. These recommendations come from analyzing the answers in our survey (Sec. 3.2).

Table 3 shows the *journal policy evaluation gap in identified key aspects*, indicating the survey question that helps evaluate the percentage of implementation of the reproducibility policies. With this table, each journal is evaluated in terms of its reproducibility policies and the effort it must make in the key aspects identified.

It can observed from the answers that there is a low percentage of implementation of reproducibility policies, as well as the low use of technological tools for automation, validation and sustainability of reproducibility in the long term (longevity of reproducibility).

This is explained because there is still no consensus and standardization on what should be a good reproducibility policy for journals, as well as the lack of a developed and mature market of trusted specialized RaaS services.

To determine the gap, we used the following qualitative ranking:

- **High**: when there is a complete lack of accomplishment or implementation of the criterion
- **Intermediate**: when there is the presence of an initiative with immature development of the criterion
- **Low**: when there is a complete and functional implementation of the criterion

Unfortunately there is a large gap in the implementation of some aspects, mainly those related to automation, establishing reproducibility policies, management of repositories, and the use of

Journals Reproducibility Features	Survey questions	Gap level
Automatic Validation and Execution Tool	11, 12, 13	High
Author Incentives	14	Intermediate
Reviewer Incentives	15	Intermediate
Reproducibility Policy	3, 4, 5, 6, 16	High
Managed Repository	9	High
Article/Data/Software Persistent Unique Identifier	9	High
Business Model Oriented to Reproducibility	7, 8	Intermediate
FAIR-TLC	10	Intermediate

Table 3: The gap in the implementation of the journal policies policies, along with the related survey's questions.

persistent identifiers for the research artifacts, including software. Other aspects such as orienting the business model towards reproducibility itself or the use of FAIR data seem to be more developed.

Despite the observed gap, there is an opportunity to reduce the reproducibility gap with the common efforts of authors, publishers, and technological providers. See tables 2, 3 and 4 for more details.

6 Conclusion

We have presented a PRISMA-based systematic review of the existing literature on techniques and platforms which can be used in computer science and machine learning projects, putting the focus on reproducible research. In order to clarify what *reproducibility* is, we have also reviewed the different definitions in the literature, which after the NASEM report have been *de facto* standardized.

The main difficulties reported by researchers when trying to reproduce the work of their peers have been enumerated. We analyzed also the problem of how to measure objectively the reproducibility, especially in the case where the experiment not necessarily gives the same results each time it is repeated, but statistically equivalent.

From our discussion we have identified what we consider are the most important reproducibility strategies, such as the use of open source repositories or the use of FAIR data, or following methodologies which have been proven to be relevant to achieve reproducibility.

In this work, we have intended to address the problem of the credibility crisis specifically in computer science including ML/AI projects, from diverse reproducibility stakeholder points of view. We establish insights from the best practices, frameworks, methodologies, and technologies available at the moment.

In computer science, the variety of languages, new developments, platforms, frameworks, hardware, and architectures on which the code of scientific articles can be run are vast. In some cases, it requires third-party proprietary software and data, which is why means a significant challenge for publishers. Implementing reproducibility policies supported by RaaS or similar methodologies can certainly help reduce the reproducibility gap on publications.

Summary of strategies for reproducibility			
Type	Strategy	Papers	Examples
(1)Sof	Open Source Software, Open science, repositories, FAIR	[80, 165, 34, 74, 76, 77, 75, 73, 78]	Github, Gitlab, bitbucket, Zenodo, softwareheritage, dataverse, Huggingface
	CSharing/Documentation tools	[91, 93, 94, 95]	Reprozip, Notebooks, CRAN, Rmarkdown,
	Open data formats, Baselines, SOTA Benchmarks	[203, 88, 86]	JSON, XML, MLperf, Dataperf, Kaggel, Brats,CM, MLcube, MLdev
(2)Env	Container/ virtualization/ Cloud	[98, 100, 191, 204]	Docker, Vmware, singularity, AWS, GCP, AZURE, ORACLE, BioNix/ Guix
	Architectures	[89, 90]	monolithic/ microservice/ serverless/cloud/hybrid
	IaC-Infrastructure as a Code	[127, 128, 129, 130, 101]	Terraform, pulumi, kubernetes CloudFormation, Ansible, puppet
(3)Sys	Cientific Workflows and MLOps tools BPML CWL languages	[118, 126, 114, 113, 120, 99, 112, 106, 125]	Taverna, Galaxy, VisTrails, Nextflow, Neptune, Weight, Comet, Omniboard, Mlflow, TensorBoard, Polyaxon, ClearML, Valohai, Pachyderm, Kubeflow, Verta.ai, SageMaker, DVC, kheOps, RE3, Hyperflow, watchdog ,SHIWA
	Metadata and Provenance (Traceability Lineage Logging Monitoring)	[132, 131, 134, 136, 135, 84, 138, 137, 107]	MERIT, ROVPY, PROV-NEO4J, PROV-DB, CONNECTOR, NOWORKFLOW, GIT2PROV,Provbook, blockchain, SWHID, DOI
	RaaS-Reproducibility as a Service	[139, 126, 157]	Whole Tale, chameleon, CodeOcean, IPOL
(4)Met	AE/AD Peer Code Reviews	[167, 171, 172, 156, 168, 115, 23]	reviewcommons, ArVix, Peer Community In (PCI), SIGPLAN, Ctuning, NeuroIPS, Badging
	Publications with code	[154, 153, 163, 155]	Some Journals(nature), Conferences(ACM,IEEE), runmycode
	Policies, Good Practices, General frameworks, Recommendations, Methodologies Life Cycle Management	[125, 147, 142, 144, 144, 109, 58, 179, 42, 149, 148, 177, 150, 151]	NASEM, DevSecOps, AIOps, MLops, CRISP-DM, KDD, SEMMA, Turing way, Teaching Culture Reproducibility, Journal Reproducibility Policies

Table 4: Strategies for Reproducibility.

We conclude that the high cost of guaranteeing the reproducibility of a software project is not properly rewarded at this moment to the reproducibility stakeholders. Considering the costs in own infrastructure that would be required, one still needs to take into consideration business models that encourage investment by third parties in infrastructure and thus guarantee longevity and perennity in the reproducibility of scientific publications.

It is, therefore, necessary to share the efforts among the different actors. A mutually-beneficial relationship must be established between authors, reviewers, and publishers to balance the benefits and costs that authors and scientific publishers eventually assume. There are not standardized Description/Evaluation standards in this regard and, in many cases, authors reported that it is very tedious to meet the requirements of each different publisher. Therefore, some authors are discouraged from improving the quality of their papers in the so-called *publish or perish* race. Moreover, some publishers (excluding those under the diamond model) are in a position of power over the authors who are charged significant article processing charges.

It is convenient to define a new metric equivalent of the Impact Factor, which could be used specifically for software. This could help properly reward the effort of software developers by acknowledging them clearly as co-authors of the scientific work and measuring the real impact of their contributions in reproducible computer science projects.

Our survey shows a growing concern about the reproducibility and how their policies can get into it. This reflected in the comments provided by the journals and the initiatives they want to push forward. Therefore, clearly defined and long-term plans are required to achieve a sustainable reproducibility model where each stakeholder obtains a benefit. For the moment, the traditional peer review evaluation methodologies are preferred. The responsibility of the validation is mainly on the expertise of the reviewers chosen by the editors and the few functional tests to the artifacts that they can do with their limited testing infrastructure. It also indicates that, in computer science journals indexed in SCOPUS, there is a low level of formal implementation of reproducibility policies supported by their own reproducibility platforms.

We conclude that it is imperative to bring together coordinated efforts to agree on standardized guides for authors for the submission of articles, unified reproducibility policies and artifact evaluation criteria from editors, supported by the reproducibility strategies and technological evolution discussed in this article. Consequently, there is a promising future with opportunities and potential to reduce the reproducibility gap identified with the joint effort of all actors involved to ensure reliability and trustworthiness in the knowledge conveyed by computer science-based publications.

Acknowledgements

The authors would like to thank the financial support to the SESAME's OVD-SaaS project from Région Île de France and BPI France. Also, to the Ministry of Science, Technology and Innovation of Colombia (Minciencias), within the framework of Call 885 of 2020 for the financing of the Doctorates of Excellence Program.

References

[1] Peter Ivie and Douglas Thain. Reproducibility in Scientific Computing. *ACM Computing Surveys*, 51(3):63:1–63:36, July 2018.

[2] T.P. Hughes. History of Technology. In *International Encyclopedia of the Social & Behavioral Sciences*, pages 6852–6857. Elsevier, 2001.

[3] Y. Hu, A. Tareen, Y-J. Sheu, W. T. Ireland, C. Speck, H. Li, L. Joshua-Tor, J. B. Kinney, and B. Stillman. Evolution of DNA replication origin specification and gene silencing mechanisms. *Nature Communications*, 11(1):5175, October 2020.

[4] Pontus Plavén-Sigray, Granville James Matheson, Björn Christian Schiffler, and William Hedley Thompson. The readability of scientific texts is decreasing over time. *eLife*, 6:e27725, September 2017.

[5] Odd Erik Gundersen. The Reproducibility Crisis Is Real. *AI Magazine*, 41(3):103–106, September 2020.

[6] Matthew Hutson. Artificial intelligence faces reproducibility crisis. *Science*, 359(6377):725–726, February 2018.

[7] Francis Dodds. The future of academic publishing: Revolution or evolution revisited. *Learned Publishing*, 32(4):345–354, October 2019.

[8] John Baillieul, Gerry Grenier, and Gianluca Setti. Reflections on the future of research curation and research reproducibility [point of view]. 106(5):779–783.

[9] Abubakari Ahmed, Aceil Al-Khatib, Yap Boum, Humberto Debat, Alonso Gurmendi Dunkelberg, Lisa Janicke Hinchliffe, Frith Jarrad, Adam Mastroianni, Patrick Mineault, Charlotte R. Pennington, and J. Andrew Pruszynski. The future of academic publishing. *Nature Human Behaviour*, 7(7):1021–1026, July 2023.

[10] Barbara Kitchenham, Lech Madeyski, and Pearl Brereton. Meta-analysis for families of experiments in software engineering: a systematic review and reproducibility and validity assessment. *Empirical Software Engineering*, 25(1):353–401, January 2020.

[11] Carlos E. Anchundia and Efrain R. Fonseca C. Resources for reproducibility of experiments in empirical software engineering: Topics derived from a secondary study. 8:8992–9004.

[12] Charlotte Stoddart. Is there a reproducibility crisis in science? *Nature*, pages d41586–019–00067–3, May 2016.

[13] Sayash Kapoor and Arvind Narayanan. Leakage and the Reproducibility Crisis in ML-based Science. 2022.

[14] Ana Lucic, Maurits Bleeker, Sami Jullien, Samarth Bhargav, and Maarten De Rijke. Reproducibility as a Mechanism for Teaching Fairness, Accountability, Confidentiality, and Transparency in Artificial Intelligence. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(11):12792–12800, June 2022.

[15] Hana Ahmed, Roselyne Tchoua, and Jay Lofstead. Measuring Reproduciblity of Machine Learning Methods for Medical Diagnosis. In *2022 Fourth International Conference on Transdisciplinary AI (TransAI)*, pages 9–16, Laguna Hills, CA, USA, September 2022. IEEE.

[16] Amin Moradi and Alexandru Uta. Reproducible Model Sharing for AI Practitioners. In *Proceedings of the Fifth Workshop on Distributed Infrastructures for Deep Learning (DIDL) 2021*, pages 1–6, Virtual Event Canada, December 2021. ACM.

[17] Benjamin Haibe-Kains, George Alexandru Adam, Ahmed Hosny, Farnoosh Khodakarami, Massive Analysis Quality Control (MAQC) Society Board of Directors, Thakkar Shraddha, Rebecca Kusko, Susanna-Assunta Sansone, Weida Tong, Russ D. Wolfinger, Christopher E. Mason, Wendell Jones, Joaquin Dopazo, Cesare Furlanello, Levi Waldron, Bo Wang, Chris McIntosh, Anna Goldenberg, Anshul Kundaje, Casey S. Greene, Tamara Broderick, Michael M. Hoffman, Jeffrey T. Leek, Keegan Korthauer, Wolfgang Huber, Alvis Brazma, Joelle Pineau, Robert Tibshirani, Trevor Hastie, John P. A. Ioannidis, John Quackenbush, and Hugo J. W. L. Aerts. Transparency and reproducibility in artificial intelligence. *Nature*, 586(7829):E14–E16, October 2020.

[18] Scott Mayer McKinney, Alan Karthikesalingam, Daniel Tse, Christopher J. Kelly, Yun Liu, Greg S. Corrado, and Shravya Shetty. Reply to: Transparency and reproducibility in artificial intelligence. *Nature*, 586(7829):E17–E18, October 2020.

[19] Felipe C. Kitamura, Ian Pan, and Timothy L. Kline. Reproducible Artificial Intelligence Research Requires Open Communication of Complete Source Code. *Radiology: Artificial Intelligence*, 2(4):e200060, July 2020.

[20] Elizabeth Gibney. This AI researcher is trying to ward off a reproducibility crisis. *Nature*, 577(7788):14–14, January 2020.

[21] Sindhu Ghanta, Sriram Subramanian, Swaminathan Sundararaman, Lior Khermosh, Vinay Sridhar, Dulcardo Arteaga, Qianmei Luo, Dhananjoy Das, and Nisha Talagala. Interpretability and Reproducability in Production Machine Learning Applications. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 658–664, Orlando, FL, December 2018. IEEE.

[22] Selin Aviyente and Abdullah Karaaslanli. Explainability in Graph Data Science: Interpretability, replicability, and reproducibility of community detection. *IEEE Signal Processing Magazine*, 39(4):25–39, July 2022.

[23] Daniel Lopresti and George Nagy. Reproducibility: Evaluating the Evaluations. In Bertrand Kerautret, Miguel Colom, Adrien Krähenbühl, Daniel Lopresti, Pascal Monasse, and Hugues Talbot, editors, *Reproducible Research in Pattern Recognition*, volume 12636, pages 12–23. Springer International Publishing, Cham, 2021.

[24] Edward Raff. Research Reproducibility as a Survival Analysis. *arXiv:2012.09932 [cs, stat]*, December 2020. arXiv: 2012.09932.

[25] Patrick Diaba-Nuhoho and Michael Amponsah-Offeh. Reproducibility and research integrity: the role of scientists and institutions. 14(1):451.

[26] D. Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips, Dietmar Ebner, Vinay Chaudhary, Michael Young, Jean-François Crespo, and Dan Dennison. Hidden technical debt in machine learning systems. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

[27] Justin Kitzes, Daniel Turek, and Fatma Deniz, editors. *The practice of reproducible research: case studies and lessons from the data-intensive sciences.* University of California Press, Oakland, California, 2018.

[28] Nathan Baker, Frank Alexander, Timo Bremer, Aric Hagberg, Yannis Kevrekidis, Habib Najm, Manish Parashar, Abani Patra, James Sethian, Stefan Wild, Karen Willcox, and Steven Lee. Workshop Report on Basic Research Needs for Scientific Machine Learning: Core Technologies for Artificial Intelligence. Technical Report 1478744, February 2019.

[29] Manish Parashar, Michael A. Heroux, and Victoria Stodden. Research Reproducibility. *Computer*, 55(8):16–18, August 2022.

[30] Paul Thompson and Andrew Burnett. Reproducible research. *CORE Issues in Professional and Research Ethics*, 1, 01 2012.

[31] Wullianallur Raghupathi, Viju Raghupathi, and Jie Ren. Reproducibility in Computing Research: An Empirical Study. *IEEE Access*, 10:29207–29223, 2022.

[32] Odd Erik Gundersen. The Fundamental Principles of Reproducibility. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 379(2197):20200210, May 2021. arXiv:2011.10098 [cs].

[33] Sebastian Stefan Feger and Paweł W. Woźniak. Reproducibility: A Researcher-Centered Definition. *Multimodal Technologies and Interaction*, 6(2):17, February 2022.

[34] Malcolm Macleod and the University of Edinburgh Research Strategy Group. Improving the reproducibility and integrity of research: what can different stakeholders contribute? *BMC Research Notes*, 15(1):146, April 2022.

[35] Committee on Reproducibility and Replicability in Science, Board on Behavioral, Cognitive, and Sensory Sciences, Committee on National Statistics, Division of Behavioral and Social Sciences and Education, Nuclear and Radiation Studies Board, Division on Earth and Life Studies, Board on Mathematical Sciences and Analytics, Committee on Applied and Theoretical Statistics, Division on Engineering and Physical Sciences, Board on Research Data and Information, Committee on Science, Engineering, Medicine, and Public Policy, Policy and Global Affairs, and National Academies of Sciences, Engineering, and Medicine. *Reproducibility and Replicability in Science*. National Academies Press, Washington, D.C., September 2019.

[36] Hans E. Plesser. Reproducibility vs. Replicability: A Brief History of a Confused Terminology. *Frontiers in Neuroinformatics*, 11:76, January 2018.

[37] Bakinam T. Essawy, Jonathan L. Goodall, Daniel Voce, Mohamed M. Morsy, Jeffrey M. Sadler, Young Don Choi, David G. Tarboton, and Tanu Malik. A taxonomy for reproducible and replicable research in environmental modelling. *Environmental Modelling & Software*, 134:104753, December 2020.

[38] Michael A. Heroux, Lorena Barba, Manish Parashar, Victoria Stodden, and Michela Taufer. Toward a Compatible Reproducibility Taxonomy for Computational and Computing Sciences. Technical Report SAND2018-11186, 1481626, October 2018.

[39] Jimmy Lin and Qian Zhang. Reproducibility is a Process, Not an Achievement: The Replicability of IR Reproducibility Experiments. In Joemon M. Jose, Emine Yilmaz, João Magalhães, Pablo Castells, Nicola Ferro, Mário J. Silva, and Flávio Martins, editors, *Advances in Information Retrieval*, volume 12036, pages 43–49. Springer International Publishing, Cham, 2020.

[40] Odd Erik Gundersen and Sigbjørn Kjensmo. State of the Art: Reproducibility in Artificial Intelligence. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), April 2018.

[41] Russell Impagliazzo, Rex Lei, Toniann Pitassi, and Jessica Sorrell. Reproducibility in learning. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, pages 818–831, Rome Italy, June 2022. ACM.

[42] Mohammad Akhlaghi, Raul Infante-Sainz, Boudewijn F. Roukema, Mohammadreza Khellat, David Valls-Gabaud, and Roberto Baena-Galle. Toward Long-Term and Archivable Reproducibility. *Computing in Science & Engineering*, 23(3):82–91, May 2021.

[43] Monya Baker. Why scientists must share their research code.

[44] Gang Fan, Chengpeng Wang, Rongxin Wu, Xiao Xiao, Qingkai Shi, and Charles Zhang. Escaping dependency hell: finding build dependency errors with the unified dependency graph. In *Proceedings of the 29th ACM SIGSOFT International Symposium on Software Testing and Analysis*, ISSTA 2020, pages 463–474, New York, NY, USA, July 2020. Association for Computing Machinery.

[45] Chris A. Mack. *How to write a good scientific paper*. SPIE Press, Bellingham, Washington, 2018. OCLC: on1019885580.

[46] David H Bailey. Reproducibility and variable precision computing. *The International Journal of High Performance Computing Applications*, 34(5):483–490, September 2020.

[47] Sara Faraji Jalal Apostal, David Apostal, and Ronald Marsh. Improving Numerical Reproducibility of Scientific Software in Parallel Systems. In *2020 IEEE International Conference on Electro Information Technology (EIT)*, pages 066–074, Chicago, IL, USA, July 2020. IEEE.

[48] Fabienne Jézéquel, Philippe Langlois, and Nathalie Revol. First steps towards more numerical reproducibility. *ESAIM: Proceedings and Surveys*, 45:229, September 2014.

[49] Eelco Dolstra, Andres Löh, and Nicolas Pierron. NixOS: A purely functional Linux distribution. *Journal of Functional Programming*, 20(5-6):577–615, November 2010.

[50] Nicolas Vallet, David Michonneau, and Simon Tournier. Toward practical transparent verifiable and long-term reproducible research using Guix. *Scientific Data*, 9(1):597, October 2022.

[51] Jing Liu, Jacob Carlson, Josh Pasek, Brian Puchala, Arvind Rao, and H. V. Jagadish. Promoting and Enabling Reproducible Data Science Through a Reproducibility Challenge. *Harvard Data Science Review*, July 2022.

[52] Valentina Ghimpau. Incentives, rewards, and recognition - what really motivates a researcher? In *Judging Research*. MDPI.

[53] Hugh Desmond. Incentivizing replication is insufficient to safeguard default trust. *Philosophy of Science*, 88, 03 2020.

[54] Dag W. Aksnes, Liv Langfeldt, and Paul Wouters. Citations, Citation Indicators, and Research Quality: An Overview of Basic Concepts and Theories. *SAGE Open*, 9(1):215824401982957, January 2019.

[55] Giuseppe Lippi and Camilla Mattiuzzi. Scientist impact factor (sif): a new metric for improving scientists' evaluation? *Annals of Translational Medicine*, 5(15), 2017.

[56] Edward Raff. Does the Market of Citations Reward Reproducible Work? *arXiv:2204.03829 [cs]*, April 2022. arXiv: 2204.03829.

[57] Alejandro C. Frery, Luis Gomez, and Antonio C. Medeiros. A Badging System for Reproducibility and Replicability in Remote Sensing Research. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 13:4988–4995, 2020.

[58] Wolfgang Mauerer, Stefan Klessinger, and Stefanie Scherzinger. Beyond the badge: reproducibility engineering as a lifetime skill. In *Proceedings of the 4th International Workshop on Software Engineering Education for the Next Generation*, pages 1–4, Pittsburgh Pennsylvania, May 2022. ACM.

[59] Marc A. Edwards and Siddhartha Roy. Academic Research in the 21st Century: Maintaining Scientific Integrity in a Climate of Perverse Incentives and Hypercompetition. *Environmental Engineering Science*, 34(1):51–61, January 2017.

[60] Samantha Cukier, Lucas Helal, Danielle B. Rice, Justina Pupkaite, Nadera Ahmadzai, Mitchell Wilson, Becky Skidmore, Manoj M. Lalu, and David Moher. Checklists to detect potential predatory biomedical journals: a systematic review. *BMC Medicine*, 18(1):104, December 2020.

[61] Fabien C. Y. Benureau and Nicolas P. Rougier. Re-run, repeat, reproduce, reuse, replicate: Transforming code into scientific contributions. 11:69.

[62] Lucas Rosenblatt, Bernease Herman, Anastasia Holovenko, Wonkwon Lee, Joshua Loftus, Elizabeth McKinnie, Taras Rumezhak, Andrii Stadnik, Bill Howe, and Julia Stoyanovich. Epistemic Parity: Reproducibility as an Evaluation Metric for Differential Privacy. *Proceedings of the VLDB Endowment*, 16(11):3178–3191, July 2023.

[63] Edward Raff. A Step Toward Quantifying Independently Reproducible Machine Learning Research. 2019.

[64] Torbjörn Nordling and Tomas Melo Peralta. A literature review of methods for assessment of reproducibility in science. preprint, In Review, November 2022.

[65] Christian S. Collberg. Measuring reproducibility in computer systems research. 2014.

[66] Sebastian Schelter, Felix Biessmann, Tim Januschowski, David Salinas, Stephan Seufert, and Gyuri Szarvas. On challenges in machine learning model management. *IEEE Data Engineering Bulletin*, 2015.

[67] Juliana Freire, Philippe Bonnet, and Dennis Shasha. Computational reproducibility: state-of-the-art, challenges, and database research opportunities. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, pages 593–596, Scottsdale Arizona USA, May 2012. ACM.

[68] Jeffrey S. Saltz and Iva Krasteva. Current approaches for executing big data science projects—a systematic literature review. *PeerJ Computer Science*, 8:e862, February 2022.

[69] Jiating Chen Gonzalo Rivero. Best coding practices to ensure reproducibility. 2020.

[70] Randall J. LeVeque, Ian M. Mitchell, and Victoria Stodden. Reproducible research for scientific computing: Tools and strategies for changing the culture. *Computing in Science & Engineering*, 14(4):13–17, 2012.

[71] Partha Pratim Ray. A review on TinyML: State-of-the-art and prospects. *Journal of King Saud University - Computer and Information Sciences*, 34(4):1595–1623, April 2022.

[72] Line Pouchard, Sterling Baldwin, Todd Elsethagen, Shantenu Jha, Bibi Raju, Eric Stephan, Li Tang, and Kerstin Kleese Van Dam. Computational reproducibility of scientific workflows at extreme scales. *The International Journal of High Performance Computing Applications*, 33(5):763–776, September 2019.

[73] Lorena A. Barba. Defining the Role of Open Source Software in Research Reproducibility. *Computer*, 55(8):40–48, August 2022.

[74] Ryan P. Abernathey, Tom Augspurger, Anderson Banihirwe, Charles C. Blackmon-Luca, Timothy J. Crone, Chelle L. Gentemann, Joseph J. Hamman, Naomi Henderson, Chiara Lepore, Theo A. McCaie, Niall H. Robinson, and Richard P. Signell. Cloud-native repositories for big scientific data. *Computing in Science & Engineering*, 23(2):26–35, 2021.

[75] Jesus M. Gonzalez-Barahona and Gregorio Robles. Revisiting the reproducibility of empirical software engineering studies based on data retrieved from development repositories. *Information and Software Technology*, 164:107318, December 2023.

[76] Aaron Haim, Stacy T. Shaw, and Neil T. Heffernan. How to Open Science: Promoting Principles and Reproducibility Practices Within the Artificial Intelligence in Education Community. In Ning Wang, Genaro Rebolledo-Mendez, Vania Dimitrova, Noboru Matsuda, and Olga C. Santos, editors, *Artificial Intelligence in Education. Posters and Late Breaking Results, Workshops and Tutorials, Industry and Innovation Tracks, Practitioners, Doctoral Consortium and Blue Sky*, volume 1831, pages 74–78. Springer Nature Switzerland, Cham, 2023.

[77] Aaron Haim, Stacy Shaw, and Neil Heffernan. How to Open Science: A Principle and Reproducibility Review of the Learning Analytics and Knowledge Conference. In *LAK23: 13th International Learning Analytics and Knowledge Conference*, pages 156–164, Arlington TX USA, March 2023. ACM.

[78] Victoria Stodden. Beyond Open Data: A Model for Linking Digital Artifacts to Enable Reproducibility of Scientific Claims. In *Proceedings of the 3rd International Workshop on Practical Reproducible Evaluation of Computer Systems*, pages 9–14, Stockholm Sweden, June 2020. ACM.

[79] Victoria Stodden, Jennifer Seiler, and Zhaokun Ma. An empirical analysis of journal policy effectiveness for computational reproducibility. *Proceedings of the National Academy of Sciences*, 115(11):2584–2589, March 2018.

[80] Jessica Parland-von Essen, Katja Fält, Zubair Maalick, Miika Alonen, and Eduardo Gonzalez. Supporting FAIR data: categorization of research data as a tool in data management. *Informaatiotutkimus*, 37(4), December 2018.

[81] Riccardo Albertoni, Sara Colantonio, Piotr Skrzypczyński, and Jerzy Stefanowski. Reproducibility of Machine Learning: Terminology, Recommendations and Open Issues, February 2023. arXiv:2302.12691 [cs].

[82] Joaquin Vanschoren, Mikio Braun, and Cheng Soon Ong. Open science in machine learning. *Implementing Reproducible Research*, 02 2014.

[83] Nathalie Baracaldo, Ali Anwar, Mark Purcell, Ambrish Rawat, Mathieu Sinn, Bashar Altakrouri, Dian Balta, Mahdi Sellami, Peter Kuhn, Ulrich Schopp, and Matthias Buchinger. Towards an Accountable and Reproducible Federated Learning: A FactSheets Approach. 2022.

[84] José A. Peregrina, Guadalupe Ortiz, and Christian Zirpins. Towards a Metadata Management System for Provenance, Reproducibility and Accountability in Federated Machine Learning. In Christian Zirpins, Guadalupe Ortiz, Zoltan Nochta, Oliver Waldhorst, Jacopo Soldani, Massimo Villari, and Damian Tamburri, editors, *Advances in Service-Oriented and Cloud Computing*, volume 1617, pages 5–18. Springer Nature Switzerland, Cham, 2022.

[85] Jan Vitek and Tomas Kalibera. Repeatability, reproducibility, and rigor in systems research. In *Proceedings of the ninth ACM international conference on Embedded software*, pages 33–38, Taipei Taiwan, October 2011. ACM.

[86] Grigori Fursin, Renato Miceli, Anton Lokhmotov, Michael Gerndt, Marc Baboulin, Allen D. Malony, Zbigniew Chamski, Diego Novillo, and Davide Del Vento. Collective mind: Towards practical and collaborative auto-tuning. *Sci. Program.*, 22:309–329, 2014.

[87] Grigori Fursin. Invited Talk Abstract: Introducing ReQuEST: An Open Platform for Reproducible and Quality-Efficient Systems-ML Tournaments. In *2018 1st Workshop on Energy Efficient Machine Learning and Cognitive Computing for Embedded Applications (EMC2)*, pages 3–3, Williamsburg, VA, March 2018. IEEE.

[88] Anahita Fathi Kazerooni, Nastaran Khalili, Xinyang Liu, Debanjan Haldar, Zhifan Jiang, Syed Muhammed Anwar, Jake Albrecht, Maruf Adewole, Udunna Anazodo, Hannah Anderson, Sina Bagheri, Ujjwal Baid, Timothy Bergquist, Austin J. Borja, Evan Calabrese, Verena Chung, Gian-Marco Conte, Farouk Dako, James Eddy, Ivan Ezhov, Ariana Familiar, Keyvan Farahani, Shuvanjan Haldar, Juan Eugenio Iglesias, Anastasia Janas, Elaine Johansen, Blaise V Jones, Florian Kofler, Dominic LaBella, Hollie Anne Lai, Koen Van Leemput, Hongwei Bran Li, Nazanin Maleki, Aaron S McAllister, Zeke Meier, Bjoern Menze, Ahmed W Moawad, Khanak K Nandolia, Julija Pavaine, Marie Piraud, Tina Poussaint, Sanjay P Prabhu, Zachary Reitman, Andres Rodriguez, Jeffrey D Rudie, Ibraheem Salman Shaikh, Lubdha M. Shah, Nakul Sheth, Russel Taki Shinohara, Wenxin Tu, Karthik Viswanathan, Chunhao Wang, Jeffrey B Ware, Benedikt Wiestler, Walter Wiggins, Anna Zapaishchykova, Mariam Aboian, Miriam Bornhorst, Peter de Blank, Michelle Deutsch, Maryam Fouladi, Lindsey Hoffman, Benjamin Kann, Margot Lazow, Leonie Mikael, Ali Nabavizadeh, Roger Packer, Adam Resnick, Brian Rood, Arastoo Vossough, Spyridon Bakas, and Marius George Linguraru. The Brain Tumor Segmentation (BraTS) Challenge 2023: Focus on Pediatrics (CBTN-CONNECT-DIPGR-ASNR-MICCAI BraTS-PEDs). 2023.

[89] Jonas Fritzsch, Justus Bogner, Markus Haug, Ana Cristina Franco da Silva, Carolin Rubner, Matthias Saft, Horst Sauer, and Stefan Wagner. Adopting Microservices and DevOps in the Cyber-Physical Systems Domain: A Rapid Review and Case Study. *Software: Practice and Experience*, 53(3):790–810, March 2023. arXiv:2210.06858 [cs].

[90] Eric Jonas, Johann Schleier-Smith, Vikram Sreekanti, Chia-Che Tsai, Anurag Khandelwal, Qifan Pu, Vaishaal Shankar, Joao Carreira, Karl Krauth, Neeraja Yadwadkar, Joseph E. Gonzalez, Raluca Ada Popa, Ion Stoica, and David A. Patterson. Cloud Programming Simplified: A Berkeley View on Serverless Computing. 2019.

[91] João Felipe Pimentel, Leonardo Murta, Vanessa Braganholo, and Juliana Freire. A Large-Scale Study About Quality and Reproducibility of Jupyter Notebooks. In *2019 IEEE/ACM 16th*

International Conference on Mining Software Repositories (MSR), pages 507–517, May 2019. ISSN: 2574-3864.

[92] Sheeba Samuel and Daniel Mietchen. Computational reproducibility of jupyter notebooks from biomedical publications, 2023.

[93] João Felipe Pimentel, Leonardo Murta, Vanessa Braganholo, and Juliana Freire. Understanding and improving the quality and reproducibility of Jupyter notebooks. *Empirical Software Engineering*, 26(4):65, July 2021.

[94] Sheeba Samuel and Birgitta König-Ries. ReproduceMeGit: A Visualization Tool for Analyzing Reproducibility of Jupyter Notebooks. In Boris Glavic, Vanessa Braganholo, and David Koop, editors, *Provenance and Annotation of Data and Processes*, volume 12839, pages 201–206. Springer International Publishing, Cham, 2021.

[95] Jiawei Wang, Tzu-yang Kuo, Li Li, and Andreas Zeller. Restoring reproducibility of Jupyter notebooks. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: Companion Proceedings*, pages 288–289, Seoul South Korea, June 2020. ACM.

[96] Andreas Wolke, Martin Bichler, Fernando Chirigati, and Victoria Steeves. Reproducible experiments on dynamic resource allocation in cloud data centers. *Information Systems*, 59:98–101, July 2016.

[97] Faical Congo. Building a Cloud Service for Reproducible Simulation Management. pages 187–193, Austin, Texas, 2015.

[98] Bill Howe. Virtual appliances, cloud computing, and reproducible research. *Computing in Science & Engineering*, 14(4):36–41, 2012.

[99] Layan Bahaidarah, Ethan Hung, Andreas F. De Melo Oliveira, Jyotsna Penumaka, Lukas Rosario, and Ana Trisovic. Toward Reusable Science with Readable Code and Reproducibility. *arXiv:2109.10387 [cs]*, September 2021. arXiv: 2109.10387.

[100] R. Shane Canon. The Role of Containers in Reproducibility. In *2020 2nd International Workshop on Containers and New Orchestration Paradigms for Isolated Environments in HPC (CANOPIE-HPC)*, pages 19–25, Atlanta, GA, USA, November 2020. IEEE.

[101] Michał Orzechowski, Bartosz Baliś, Renata G. Słota, and Jacek Kitowski. Reproducibility of Computational Experiments on Kubernetes-Managed Container Clouds with HyperFlow. In Valeria V. Krzhizhanovskaya, Gábor Závodszky, Michael H. Lees, Jack J. Dongarra, Peter M. A. Sloot, Sérgio Brissos, and João Teixeira, editors, *Computational Science – ICCS 2020*, volume 12137, pages 220–233. Springer International Publishing, Cham, 2020.

[102] Alexey Vasyukov and Igor Petrov. Using Computing Containers and Continuous Integration to Improve Numerical Research Reproducibility. *International Journal of Computer (IJC)*, 30(1):27–33, July 2018.

[103] Jessica Forde, Tim Head, Chris Holdgraf, Yuvi Panda, Gladys Nalvarete, Benjamin Ragan-Kelley, and Erik Sundell. Reproducible Research Environments with Repo2Docker. June 2018.

[104] Peter Sugimura and Florian Hartl. Building a Reproducible Machine Learning Pipeline. *arXiv:1810.04570 [cs, stat]*, October 2018. arXiv: 1810.04570.

[105] Monika Steidl, Michael Felderer, and Rudolf Ramler. The pipeline for the continuous development of artificial intelligence models—Current state of research and practice. *Journal of Systems and Software*, 199:111615, May 2023.

[106] Michael Kluge, Marie-Sophie Friedl, Amrei L Menzel, and Caroline C Friedel. Watchdog 2.0: New developments for reusability, reproducibility, and workflow execution. *GigaScience*, 9(6):giaa068, June 2020.

[107] Sheeba Samuel, Frank Löffler, and Birgitta König-Ries. Machine Learning Pipelines: Provenance, Reproducibility and FAIR Data Principles. In Boris Glavic, Vanessa Braganholo, and David Koop, editors, *Provenance and Annotation of Data and Processes*, volume 12839, pages 226–230. Springer International Publishing, Cham, 2021.

[108] Rafael Ferreira Da Silva, Henri Casanova, Kyle Chard, Ilkay Altintas, Rosa M Badia, Bartosz Balis, Taina Coleman, Frederik Coppens, Frank Di Natale, Bjoern Enders, Thomas Fahringer, Rosa Filgueira, Grigori Fursin, Daniel Garijo, Carole Goble, Dorran Howell, Shantenu Jha, Daniel S. Katz, Daniel Laney, Ulf Leser, Maciej Malawski, Kshitij Mehta, Loic Pottier, Jonathan Ozik, J. Luc Peterson, Lavanya Ramakrishnan, Stian Soiland-Reyes, Douglas Thain, and Matthew Wolf. A Community Roadmap for Scientific Workflows Research and Development. In *2021 IEEE Workshop on Workflows in Support of Large-Scale Science (WORKS)*, pages 81–90, St. Louis, MO, USA, November 2021. IEEE.

[109] Fran Melchor, Roberto Rodriguez-Echeverria, Jose M. Conejero, and Juan Prieto, Alvaro amd Gutierrez. A Model-Driven Approach for Systematic Reproducibility and Replicability of Data Science Projects. In Xavier Franch, Geert Poels, Frederik Gailly, and Monique Snoeck, editors, *Advanced Information Systems Engineering*, volume 13295, pages 147–163. Springer International Publishing, Cham, 2022.

[110] Jules Françoise, Baptiste Caramiaux, and Téo Sanchez. Marcelle: Composing Interactive Machine Learning Workflows and Interfaces. In *The 34th Annual ACM Symposium on User Interface Software and Technology*, pages 39–53, Virtual Event USA, October 2021. ACM.

[111] Anirudh Prabhu and Peter Fox. Reproducible Workflow. 2020.

[112] Devarshi Ghoshal, Drew Paine, Gilberto Pastorello, Abdelrahman Elbashandy, Dan Gunter, Oluwamayowa Amusat, and Lavanya Ramakrishnan. Experiences with Reproducibility: Case Studies from Scientific Workflows. In *Proceedings of the 4th International Workshop on Practical Reproducible Evaluation of Computer Systems*, pages 3–8, Virtual Event Sweden, June 2020. ACM.

[113] Daniel Rosendo, Kate Keahey, Alexandru Costan, Matthieu Simonin, Patrick Valduriez, and Gabriel Antoniu. KheOps: Cost-effective Repeatability, Reproducibility, and Replicability of Edge-to-Cloud Experiments. In *Proceedings of the 2023 ACM Conference on Reproducibility and Replicability*, pages 62–73, Santa Cruz CA USA, June 2023. ACM.

[114] Sarah Cohen-Boulakia, Khalid Belhajjame, Olivier Collin, Jérôme Chopard, Christine Froidevaux, Alban Gaignard, Konrad Hinsen, Pierre Larmande, Yvan Le Bras, Frédéric Lemoine, Fabien Mareuil, Hervé Ménager, Christophe Pradal, and Christophe Blanchet. Scientific workflows for computational reproducibility in the life sciences: Status, challenges and opportunities. *Future Generation Computer Systems*, 75:284–298, October 2017.

[115] Beth A. Plale, Tanu Malik, and Line C. Pouchard. Reproducibility Practice in High-Performance Computing: Community Survey Results. *Computing in Science & Engineering*, 23(5):55–60, September 2021.

[116] Idafen Santana-Perez and María S. Pérez-Hernández. Towards Reproducibility in Scientific Workflows: An Infrastructure-Based Approach. *Scientific Programming*, 2015:e243180, February 2015.

[117] Eric Breck, Shanqing Cai, Eric Nielsen, Michael Salib, and D. Sculley. The ml test score: A rubric for ml production readiness and technical debt reduction. In *2017 IEEE International Conference on Big Data (Big Data)*, pages 1123–1132, 2017.

[118] Noah Gift and Alfredo Deza. *Practical MLOps: operationalizing machine learning models.* O'Reilly Media Inc, Sebastopol, CA, first edition edition, 2021. OCLC: on1249501065.

[119] Saleema Amershi, Andrew Begel, Christian Bird, Robert DeLine, Harald Gall, Ece Kamar, Nachiappan Nagappan, Besmira Nushi, and Thomas Zimmermann. Software engineering for machine learning: A case study. In *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*, pages 291–300, 2019.

[120] Odd Erik Gundersen, Saeid Shamsaliei, and Richard Juul Isdahl. Do machine learning platforms provide out-of-the-box reproducibility? *Future Generation Computer Systems*, 126:34–47, January 2022.

[121] Marius Schlegel and Kai-Uwe Sattler. Management of Machine Learning Lifecycle Artifacts: A Survey. 2022.

[122] Preprint. Ml reproducibility systems: Status and research agenda. 2021.

[123] Grigori Fursin. Collective Knowledge: organizing research projects as a database of reusable components and portable workflows with common APIs. 2020.

[124] Michael Adams, Andreas V. Hense, and Arthur H.M. ter Hofstede. YAWL: An open source Business Process Management System from science for science. *SoftwareX*, 12:100576, July 2020.

[125] Vladimir Korkhov, Dagmar Krefting, Johan Montagnat, Tram Truong-Huu, Tamas Kukla, Gabor Terstyanszky, David Manset, Matthan Caan, and Silvia Olabarriaga. Shiwa workflow interoperability solutions for neuroimaging data analysis. *Studies in health technology and informatics*, 175:109–10, 09 2012.

[126] Yuri Demchenko, Sebastian Gallenmuller, Serge Fdida, Panayiotis Andreou, Cedric Crettaz, and Mathias Kirkeng. Experimental Research Reproducibility and Experiment Workflow Management. In *2023 15th International Conference on COMmunication Systems & NETworkS (COMSNETS)*, pages 835–840, Bangalore, India, January 2023. IEEE.

[127] Richard W. Bowman. Improving instrument reproducibility with open source hardware. 3(1):27.

[128] https://octoverse.github.com/. State of the octoverse 2022. 2023.

[129] Pedro Mestre Daniel Adorno Gomes and Carlos Serodio. Infrastructure-as-code for scientific computing environments. CENTRIC 2019.

[130] pulumi.com. Delivering cloud native infrastructure as code - pulumi. 2019.

[131] Daniel Silva Junior, Esther Pacitti, Aline Paes, and Daniel De Oliveira. Provenance-and machine learning-based recommendation of parameter values in scientific workflows. *PeerJ Computer Science*, 7:e606, July 2021.

[132] Trung Dong Huynh and Luc Moreau. ProvStore: A Public Provenance Repository. In Bertram Ludäscher and Beth Plale, editors, *Provenance and Annotation of Data and Processes*, volume 8628, pages 275–277. Springer International Publishing, Cham, 2015.

[133] Fernando Chirigati, Dennis Shasha, and Juliana Freire. Reprozip: Using provenance to support computational reproducibility. 01 2013.

[134] Joseph Wonsil, Jack Sullivan, Margo Seltzer, and Adam Pocock. Integrated Reproducibility with Self-describing Machine Learning Models. In *Proceedings of the 2023 ACM Conference on Reproducibility and Replicability*, pages 1–14, Santa Cruz CA USA, June 2023. ACM.

[135] Sheeba Samuel and Birgitta König-Ries. A collaborative semantic-based provenance management platform for reproducibility. *PeerJ Computer Science*, 8:e921, March 2022.

[136] Sheeba Samuel and Birgitta König-Ries. End-to-End provenance representation for the understandability and reproducibility of scientific experiments using a semantic approach. *Journal of Biomedical Semantics*, 13(1):1, December 2022.

[137] Kevin Wittek, Neslihan Wittek, James Lawton, Iryna Dohndorf, Alexander Weinert, and Andrei Ionita. A Blockchain-Based Approach to Provenance and Reproducibility in Research Workflows. In *2021 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pages 1–6, Sydney, Australia, May 2021. IEEE.

[138] Yasutaka Kawamoto and Akihiro Kobayashi. AI pedigree verification platform using blockchain. In *2020 2nd Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS)*, pages 204–205, Paris, France, September 2020. IEEE.

[139] Joseph Wonsil. Reproducibility as a service.

[140] Tom Crick, Benjamin A. Hall, and Samin Ishtiaq. Reproducibility as a Technical Specification. 2015.

[141] Miguel Colom, Bertrand Kerautret, Nicolas Limare, Pascal Monasse, and Jean-Michel Morel. IPOL: a new journal for fully reproducible research; analysis of four years development. July 2015.

[142] Reed Milewicz and Miranda Mundt. Towards Evidence-Based Software Quality Practices for Reproducibility: Preliminary Results and Research Directions. In *Proceedings of the 2023 ACM Conference on Reproducibility and Replicability*, pages 85–88, Santa Cruz CA USA, June 2023. ACM.

[143] Victoria Stodden and Sheila Miguez. Best practices for computational science: Software infrastructure and environments for reproducible and extensible research. 2(1):e21.

[144] The Turing Way Community and Scriberia. Illustrations from The Turing Way: Shared under CC-BY 4.0 for reuse, May 2022.

[145] Yasmin AlNoamany and John A. Borghi. Towards computational reproducibility: researcher perspectives on the use and sharing of software. *PeerJ Computer Science*, 4:e163, September 2018.

[146] Inigo Martinez, Elisabeth Viles, and Igor G Olaizola. A survey study of success factors in data science projects. In *2021 IEEE International Conference on Big Data (Big Data)*, pages 2313–2318, Orlando, FL, USA, December 2021. IEEE.

[147] Christoph Schröer, Felix Kruse, and Jorge Marx Gómez. A systematic literature review on applying crisp-dm process model. *Procedia Computer Science*, 181:526–534, 01 2021.

[148] Yasemin Turkyilmaz-van der Velden, Nicolas Dintzner, and Marta Teperek. Reproducibility Starts from You Today. *Patterns*, 1(6):100099, September 2020.

[149] Kenneth M. Merz, Rommie Amaro, Zoe Cournia, Matthias Rarey, Thereza Soares, Alexander Tropsha, Habibah A. Wahab, and Renxiao Wang. Editorial: Method and data sharing and reproducibility of scientific results. 60(12):5868–5869.

[150] Sheeba Samuel and Birgitta König-Ries. Understanding experiments and research practices for reproducibility: an exploratory study. *PeerJ*, 9:e11140, April 2021.

[151] James D. Nichols, Madan K. Oli, William. L. Kendall, and G. Scott Boomer. A better approach for dealing with reproducibility and replicability in science. 118(7):e2100769118.

[152] Remco Heesen. Communism and the incentive to share in science. *Philosophy of Science*, 84(4):698–716, 2017.

[153] Fabio Bonsignorio. A New Kind of Article for Reproducible Research in Intelligent Robotics [From the Field]. *IEEE Robotics & Automation Magazine*, 24(3):178–182, September 2017.

[154] Hans De Sterck, Chi-Wang Shu, and Rémi Abgrall. Enhancing Reproducibility of Research Papers in SISC, JSC and JCP. *Journal of Scientific Computing*, 95(3):77, s10915–023–02193–7, June 2023.

[155] Ana Trisovic, Philip Durbin, Tania Schlatter, Gustavo Durand, Sonia Barbosa, Danny Brooke, and Mercè Crosas. Advancing Computational Reproducibility in the Dataverse Data Repository Platform. In *Proceedings of the 3rd International Workshop on Practical Reproducible Evaluation of Computer Systems*, pages 15–20, Stockholm Sweden, June 2020. ACM.

[156] Joelle Pineau, Philippe Vincent-Lamarre, Koustuv Sinha, Vincent Larivière, Alina Beygelzimer, Florence d'Alché Buc, Emily Fox, and Hugo Larochelle. Improving Reproducibility in Machine Learning Research (A Report from the NeurIPS 2019 Reproducibility Program). *arXiv:2003.12206 [cs, stat]*, December 2020. arXiv: 2003.12206.

[157] Kyle Chard, Niall Gaffney, Mihael Hategan, Kacper Kowalik, Bertram Ludäscher, Timothy McPhillips, Jarek Nabrzyski, Victoria Stodden, Ian Taylor, Thomas Thelen, Matthew J. Turk, and Craig Willis. Toward enabling reproducibility for data-intensive research using the whole tale platform. In Ian Foster, Gerhard R. Joubert, Luděk Kučera, Wolfgang E. Nagel, and Frans Peters, editors, *Advances in Parallel Computing*. IOS Press.

[158] Adam Brinckman, Kyle Chard, Niall Gaffney, Mihael Hategan, Matthew B. Jones, Kacper Kowalik, Sivakumar Kulasekaran, Bertram Ludäscher, Bryce D. Mecum, Jarek Nabrzyski, Victoria Stodden, Ian J. Taylor, Matthew J. Turk, and Kandace Turner. Computing environments for reproducibility: Capturing the "Whole Tale". *Future Generation Computer Systems*, 94:854–867, May 2019.

[159] Kyle Chard, Niall Gaffney, Matthew B. Jones, Kacper Kowalik, Bertram Ludäscher, Jarek Nabrzyski, Victoria Stodden, Ian Taylor, Matthew J. Turk, and Craig Willis. Implementing

Computational Reproducibility in the Whole Tale Environment. In *Proceedings of the 2nd International Workshop on Practical Reproducible Evaluation of Computer Systems*, pages 17–22, Phoenix AZ USA, June 2019. ACM.

[160] Burak Yildiz, Hayley Hung, Jesse H. Krijthe, Cynthia C. S. Liem, Marco Loog, Gosia Migut, Frans Oliehoek, Annibale Panichella, Przemyslaw Pawelczak, Stjepan Picek, Mathijs de Weerdt, and Jan van Gemert. ReproducedPapers.org: Openly teaching and structuring machine learning reproducibility. *arXiv:2012.01172 [cs]*, 12636:3–11, 2021. arXiv: 2012.01172.

[161] Nicolas P. Rougier and Konrad Hinsen. ReScience C: A Journal for Reproducible Replications in Computational Science. In Bertrand Kerautret, Miguel Colom, Daniel Lopresti, Pascal Monasse, and Hugues Talbot, editors, *Reproducible Research in Pattern Recognition*, volume 11455, pages 150–156. Springer International Publishing, Cham, 2019.

[162] Kate Keahey, Pierre Riteau, Dan Stanzione, Tim Cockerill, Joe Mambretti, Paul Rad, and Paul Ruth. *Chameleon: A Scalable Production Testbed for Computer Science Research*, pages 123–148. 05 2019.

[163] Lamia Salsabil, Jian Wu, Muntabir Hasan Choudhury, William A. Ingram, Edward A. Fox, Sarah M. Rajtmajer, and C. Lee Giles. A Study of Computational Reproducibility using URLs Linking to Open Access Datasets and Software. In *Companion Proceedings of the Web Conference 2022*, pages 784–788, Virtual Event, Lyon France, April 2022. ACM.

[164] Al Idrissou, Veruska Zamborlini, and Tobias Kuhn. Documenting the Creation, Manipulation and Evaluation of Links for Reuse and Reproducibility. In Oscar Corcho, Laura Hollink, Oliver Kutz, Nicolas Troquard, and Fajar J. Ekaputra, editors, *Knowledge Engineering and Knowledge Management*, volume 13514, pages 81–96. Springer International Publishing, Cham, 2022.

[165] Edward Raff and Andrew L. Farris. A Siren Song of Open Source Reproducibility. *arXiv:2204.04372 [cs]*, April 2022. arXiv: 2204.04372.

[166] Dylan G. E. Gomes, Patrice Pottier, Robert Crystal-Ornelas, Emma J. Hudgins, Vivienne Foroughirad, Luna L. Sánchez-Reyes, Rachel Turba, Paula Andrea Martinez, David Moreau, Michael G. Bertram, Cooper A. Smout, and Kaitlyn M. Gaynor. Why don't we share data and code? Perceived barriers and benefits to public archiving practices. *Proceedings of the Royal Society B: Biological Sciences*, 289(1987):20221113, November 2022.

[167] Supporting computational reproducibility through code review. *Nature Human Behaviour*, 5(8):965–966, August 2021.

[168] Manos Athanassoulis, Peter Triantafillou, Raja Appuswamy, Rajesh Bordawekar, Badrish Chandramouli, Xuntao Cheng, Ioana Manolescu, Yannis Papakonstantinou, and Nesime Tatbul. Artifacts Availability & Reproducibility (VLDB 2021 Round Table). *ACM SIGMOD Record*, 51(2):74–77, July 2022.

[169] Swapna Krishnakumar Radha, Ian Taylor, Jarek Nabrzyski, and Iain Barclay. Verifiable Badging System for scientific data reproducibility. *Blockchain: Research and Applications*, 2(2):100015, June 2021.

[170] Grigori Fursin. Enabling reproducible ML and Systems research: the good, the bad, and the ugly. August 2020.

[171] Iordanis Fostiropoulos, Bowman Brown, and Laurent Itti. Reproducibility Requires Consolidated Artifacts. In *2023 IEEE/ACM 2nd International Conference on AI Engineering – Software Engineering for AI (CAIN)*, pages 100–101, Melbourne, Australia, May 2023. IEEE.

[172] Tanu Malik. Artifact Description/Artifact Evaluation: A Reproducibility Bane or a Boon. In *Proceedings of the 4th International Workshop on Practical Reproducible Evaluation of Computer Systems*, pages 1–1, Virtual Event Sweden, June 2020. ACM.

[173] Christophe Pérignon, Kamel Gadouche, Christophe Hurlin, Roxane Silberman, and Eric Debonnel. Certify reproducibility with confidential data. *Science*, 365(6449):127–128, July 2019.

[174] Helena Cousijn, Amye Kenall, Emma Ganley, Melissa Harrison, David Kernohan, Thomas Lemberger, Fiona Murphy, Patrick Polischuk, Simone Taylor, Maryann Martone, and Tim Clark. A data citation roadmap for scientific publishers. *Scientific Data*, 5(1):180259, November 2018.

[175] Heidi Seibold, Severin Czerny, Siona Decke, Roman Dieterle, Thomas Eder, Steffen Fohr, Nico Hahn, Rabea Hartmann, Christoph Heindl, Philipp Kopper, Dario Lepke, Verena Loidl, Maximilian Mandl, Sarah Musiol, Jessica Peter, Alexander Piehler, Elio Rojas, Stefanie Schmid, Hannah Schmidt, Melissa Schmoll, Lennart Schneider, Xiao-Yin To, Viet Tran, Antje Völker, Moritz Wagner, Joshua Wagner, Maria Waize, Hannah Wecker, Rui Yang, Simone Zellner, and Malte Nalenz. A computational reproducibility study of PLOS ONE articles featuring longitudinal data analyses. 16(6):e0251194.

[176] Nestoras Karathanasis, Daniel Hwang, Vibol Heng, Rimal Abhimannyu, Phillip Slogoff-Sevilla, Gina Buchel, Victoria Frisbie, Peiyao Li, Dafni Kryoneriti, and Isidore Rigoutsos. Reproducibility efforts as a teaching tool: A pilot study. *PLOS Computational Biology*, 18(11):e1010615, November 2022.

[177] Jake M. Hofman, Daniel G. Goldstein, Siddhartha Sen, and Forough Poursabzi-Sandegh. Expanding the Scope of Reproducibility Research Through Data Analysis Replications. In *Companion Proceedings of the Web Conference 2020*, pages 567–571, Taipei Taiwan, April 2020. ACM.

[178] Fraida Fund. We Need More Reproducibility Content Across the Computer Science Curriculum. In *Proceedings of the 2023 ACM Conference on Reproducibility and Replicability*, pages 97–101, Santa Cruz CA USA, June 2023. ACM.

[179] Jimmy Lin. Building a Culture of Reproducibility in Academic Research. 2022.

[180] Mark A. Parsons, Ruth E. Duerr, and Matthew B. Jones. The History and Future of Data Citation in Practice. *Data Science Journal*, 18:52, November 2019.

[181] Mikhail G. Dozmorov. GitHub Statistics as a Measure of the Impact of Open-Source Bioinformatics Software. *Frontiers in Bioengineering and Biotechnology*, 6:198, December 2018.

[182] Victoria Stodden, Peixuan Guo, and Zhaokun Ma. Toward Reproducible Computational Research: An Empirical Analysis of Data and Code Policy Adoption by Journals. *PLoS ONE*, 8(6):e67111, June 2013.

[183] Thu-Mai Lewis. *From policy to practice: How journal-based data policies encourage scientists' adoption of reproducible research practices.* PhD thesis, The University of North Carolina at Chapel Hill University Libraries.

[184] Zhaokun Stodden Victoria, Guo Peixuan. How journals are adopting open data and code policies.

[185] Jeroen Bosman, Jan Erik Frantsvåg, Bianca Kramer, Pierre-Carl Langlais, and Vanessa Proudman. OA Diamond Journals Study. Part 1: Findings. Technical report, Zenodo, March 2021.

[186] Nicole A. Vasilevsky, Jessica Minnier, Melissa A. Haendel, and Robin E. Champieux. Reproducible and reusable research: are journal data sharing policies meeting the mark? *PeerJ*, 5:e3208, April 2017.

[187] Markus Konkol, Daniel Nüst, and Laura Goulier. Publishing computational research - a review of infrastructures for reproducible and transparent scholarly communication. *Research Integrity and Peer Review*, 5(1):10, December 2020.

[188] Craig Willis and Victoria Stodden. Trust but Verify: How to Leverage Policies, Workflows, and Infrastructure to Ensure Computational Reproducibility in Publication. *Harvard Data Science Review*, 2(4), December 2020.

[189] Delphine R. Boulbes, Tracy Costello, Keith Baggerly, Fan Fan, Rui Wang, Rajat Bhattacharya, Xiangcang Ye, and Lee M. Ellis. A Survey on Data Reproducibility and the Effect of Publication Process on the Ethical Reporting of Laboratory Research. *Clinical Cancer Research*, 24(14):3447–3455, July 2018.

[190] Abhishek Gupta, Connor Wright, Marianna Bergamaschi Ganapini, Masa Sweidan, and Renjie Butalid. State of AI Ethics Report (Volume 6, February 2022), February 2022. arXiv:2202.07435 [cs].

[191] David Moreau, Kristina Wiebels, and Carl Boettiger. Containers for computational reproducibility. *Nature Reviews Methods Primers*, 3(1):1–16, July 2023.

[192] Grigori Fursin. The Collective Knowledge project: making ML models more portable and reproducible with open APIs, reusable best practices and MLOps. 2020.

[193] Russell A. Poldrack. The Costs of Reproducibility. *Neuron*, 101(1):11–14, January 2019.

[194] Armbrust. Above the clouds: A berkeley view of cloud computing. 01 2009.

[195] Christopher Gandrud. *Reproducible research with R and RStudio*. The R series. CRC Press, Boca Raton, FL, third edition edition, 2020.

[196] Enis Afgan, Dannon Baker, Bérénice Batut, Marius Van Den Beek, Dave Bouvier, Martin Čech, John Chilton, Dave Clements, Nate Coraor, Björn A Grüning, et al. The galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2018 update. *Nucleic acids research*, 46(W1):W537–W544, 2018.

[197] Miles Brundage, Shahar Avin, Jasmine Wang, Haydn Belfield, Gretchen Krueger, Gillian Hadfield, Heidy Khlaaf, Jingying Yang, Helen Toner, Ruth Fong, Tegan Maharaj, Pang Wei Koh, Sara Hooker, Jade Leung, Andrew Trask, Emma Bluemke, Jonathan Lebensold, Cullen O'Keefe, Mark Koren, Théo Ryffel, JB Rubinovitz, Tamay Besiroglu, Federica Carugati, Jack Clark, Peter Eckersley, Sarah de Haas, Maritza Johnson, Ben Laurie, Alex Ingerman, Igor Krawczuk, Amanda Askell, Rosario Cammarota, Andrew Lohn, David Krueger, Charlotte Stix, Peter Henderson, Logan Graham, Carina Prunkl, Bianca Martin, Elizabeth Seger, Noa Zilberman, Seán hÉigeartaigh, Frens Kroeger, Girish Sastry, Rebecca Kagan, Adrian Weller, Brian Tse, Elizabeth Barnes, Allan Dafoe, Paul Scharre, Ariel Herbert-Voss, Martijn Rasser,

Shagun Sodhani, Carrick Flynn, Thomas Krendl Gilbert, Lisa Dyer, Saif Khan, Yoshua Bengio, and Markus Anderljung. Toward Trustworthy AI Development: Mechanisms for Supporting Verifiable Claims. 2020.

[198] Raphael Hiesgen, Marcin Nawrocki, Thomas C Schmidt, and Matthias Wählisch. The race to the vulnerable: Measuring the log4j shell incident. *arXiv preprint arXiv:2205.02544*, 2022.

[199] Qing Ke, Alexander J. Gates, and Albert-László Barabási. A network-based normalized impact measure reveals successful periods of scientific discovery across disciplines. *Proceedings of the National Academy of Sciences*, 120(48):e2309378120, November 2023.

[200] Arfon M. Smith, Daniel S. Katz, Kyle E. Niemeyer, and FORCE11 Software Citation Working Group. Software citation principles. *PeerJ Computer Science*, 2:e86, September 2016.

[201] Roberto Di Cosmo, Morane Gruenpeter, and Stefano Zacchiroli. 204.4 Identifiers for Digital Objects: The case of software source code preservation. August 2022.

[202] Jill Stefaniak and Kimberly Carey. Instilling purpose and value in the implementation of digital badges in higher education. *International Journal of Educational Technology in Higher Education*, 16, 12 2019.

[203] Anton Khritankov, Nikita Pershin, Nikita Ukhov, and Artem Ukhov. MLDev: Data Science Experiment Automation and Reproducibility Software. *arXiv:2107.12322 [cs]*, July 2021. arXiv: 2107.12322.

[204] Justin Bedő, Leon Di Stefano, and Anthony T Papenfuss. Unifying package managers, workflow engines, and containers: Computational reproducibility with BioNix. 9(11):giaa121.

[205] José Antonio Salvador Oliván, Gonzalo Marco Cuenca, and Rosario Arquero Avilés. reproducibilidad de las estrategias de búsqueda en revisiones sistemáticas publicadas en revistas españolas de Biblioteconomía y Documentación. *Ibersid: revista de sistemas de información y documentación*, 17(1):129–137, June 2023.

[206] Matthew J Page, Joanne E McKenzie, Patrick M Bossuyt, Isabelle Boutron, Tammy C Hoffmann, Cynthia D Mulrow, Larissa Shamseer, Jennifer M Tetzlaff, Elie A Akl, Sue E Brennan, Roger Chou, Julie Glanville, Jeremy M Grimshaw, Asbjørn Hróbjartsson, Manoj M Lalu, Tianjing Li, Elizabeth W Loder, Evan Mayo-Wilson, Steve McDonald, Luke A McGuinness, Lesley A Stewart, James Thomas, Andrea C Tricco, Vivian A Welch, Penny Whiting, and David Moher. The PRISMA 2020 statement: an updated guideline for reporting systematic reviews. *BMJ*, page n71, March 2021.

A Research Methods

A.1 PRISMA Literature Review

To perform our review two strategies [205] were adopted. On one hand, active search for articles highly cited reports, and follow-up of citation threads on computational reproducibility. On the other hand, we applied the PRISMA 2020 [206] methodology with the SCOPUS & WoS databases between 2020 to the present day (Table 5). Eventually, we reduced our analysis to 100 representative works.

PRISMA		
Step	Num items	Condition
Identification	SCOPUS (413)	TITLE (reproducibility) AND PUBYEAR > 2019 AND PUBYEAR < 2024 AND (LIMIT-TO (SUBJAREA , "COMP"))
Identification	WoS (371)	TI=(reproducibility) and 2023 or 2022 or 2021 or 2020 (Publication Years) and Multidisciplinary Sciences or Computer Science Interdisciplinary Applications or Computer Science Theory Methods or Computer Science Information Systems (Web of Science Categories)
Deleting repeated	(144)	Repeated articles
Screening	(80)	Main theme reproducibility
Included	(60)	Systematically classified

Table 5: The PRISMA screening was used in this review to select recent relevant articles related to reproducibility in scientific research.

A.2 Journals Survey

To avoid bias in the research and not only address journals that are known to apply reproducibility policies, e.g. (IEEE with CodeOcean [40], ACM with reproducibility Badges [41], Journal Nature[42]), a request for participation was sent to several journals specialized in computer science.

Information was voluntarily requested through a 16-question form from a list of 500 journals specialized in computer science, scopus indexed from Q1 to Q4 in the period from July 1, 2023 to September 1, 2023. Although the question was very precise, they were left open to comments.

A.3 Classification of reproducibility criteria.

Table 6 shows the classification of reproducibility criteria for Artifacts Description (AD) and Artifacts Evaluation (AE).

Criterion	**Description**	**Type**
Results	Documented result and analysis	Experiment
Analysis	Supported claims	Experiment
Justification	Justified method, metrics, datasets	Experiment
Workflow	Summarized experiment execution and configurations	Experiment
Workflow execution	Tracked execution with configuration	Experiment

[40] https://innovate.ieee.org/ieee-code-ocean/
[41] https://www.acm.org/publications/policies/artifact-review-badging
[42] https://www.nature.com/nature-portfolio/editorial-policies/reporting-standards

Hardware	Specified hardware	Experiment
Software	Documented software dependencies.	Experiment
Citation Export	Reference automatically generated	Experiment
Code repository	Shared code in repository	Experiment
Code metadata	Code metadata included	Experiment
Code license	Code license included	Experiment
Code citeable	Code (DOI) or (PURL) assigned.	Experiment
Hypothesis	Documented hypothesis	Method
Prediction	Documented predictions	Method
Setup	Documented parameters and conditions, statistical significance of results.	Method
Problem description	Clearly described problem	Method
Outline	Conceptually described method	Method
Pseudo code	Documented pseudo code	Method
Data repository	Data shared in accessible repository	Data
Data metadata	Metadata included in the datasets	Data
Data license	Licensed data	Data
Data citeable	DOI or P-URL of data assigned	Data
NEUROips Checklist [156]		
Model and algorithms	Clarified mathematical models, algorithms, settings - assumptions explained - algorithm complexity analyzed	Experiment
Theoretical claim	Clarified claim statements - fully proven claims	Method
Datasets	Relevant statistics, details of train/validation/test split, explanation of excluded data and preprocessing, link to downloadable version of environment and dataset, description of quality Control methods	Data
Code	Specified dependencies, Training code, evaluation code, README file with results table, pre trained models	Experiment
Rxperimental result	Selection method, range, specification of best hyperparameters, exact number of training and evaluation runs, clear definition of metrics and results statistics, description of results with trend and central variation, average energy cost, results runtime.	Method
SIGPLAN		
Clearly stated claims	Explicit claims, appropriately scoped, recognize limitations	Method
Suitable comparison	Compare with the appropriate baseline, comparison is fair	Method
Principled benchmark choice	Appropriate and fair use of non-standard suit, using applications instead of kernels	Method
Adequate data analysis	Sufficient number of trials, appropriate summary of statistics, data distribution reported	Method

Relevant Metrics	Direct and appropriate proxy metrics, successful in measuring all effects	Method
Appropriate and clear Experimental Design	Enough information to repeat, reasonable platform, Consider all key design parameters, open workload generator, evaluated in test set	Method
Appropriate presentation of results	Clear summary of results, appropriate truncate axes, ratios plotted correctly, appropriate level of precision	Method
Ctuning		
Abstract	Clearly stated the problem, solution and supporting results?	Method
Algorithm	Is it a new algorithm?	Experiment
Program	Are any benchmarks used?	Method
Compilation	Does it require a specific compiler?	Experiment
Transformations	Does it require a program transformation tool?	Experiment
Binary	Are binaries included?	Experiment
Model	Are specific models used?	Experiment
Data set	Are specific data sets used ?	Experiment
Run-time environment	Are there any OS-specific artifact?	Experiment
Hardware	Is specific hardware required?	Experiment
Run-time state	Is the state sensitive to run-time?	Experiment
Execution	Is the software running under specific conditions?	Experiment
Metrics	How are the metrics evaluated?	Experiment
Output	What is the output?	Experiment
Experiments	How to prepare and reproduce results?	Method
Disk space	How much disk space is required?	Experiment
Workflow	How much time is needed to prepare the workflow?	Experiment
Time evaluation	How much time is needed to complete the experiments?	Experiment
Publicly available		Data
Code licenses	Is the software under any licenses?	Data
Workflow frameworks	Are workflow framework used for automation?	Method
Archived	Is the software archived and make it public?	Data
Description		
Access	Does the system describe how reviewers will access the research artifacts?	Data
Hardware dependencies	Does the system describe any specific hardware and specific features?	Experiment
Software dependencies	Does the system describe OS and software packages required to evaluate your artifacts?	Experiment
Data sets	Are there any third-party data sets used in your packages?	Data
Installation	Does the system describe the setup procedures?	Method
Experiment workflow	Does the system describe how the workflow is implemented and executed?	Experiment
Evaluation and expected result	Does the system describe how to reproduce the key results from the paper?	Method

Experiment customization Notes	Does the system provide special instructions to customize and tune the experiments?	Method
NISO		
Artifact Available	A DOI or URL link to the repository along with a unique identifier for the object is provided	Data
Artifacts Evaluated-Functional	the artifacts are documented, consistent, complete, exercisable, and include appropriate evidence of verification and validation.	Method
Artifacts Evaluate-Reusable	Artifacts associated with the paper are of a quality, documented and well-structured that significantly exceeds minimal functionality to the extent that reuse and re-purposing is facilitated.	Method
Open Research Objects (ORO)	A DOI or URL link to the repository along with a unique identifier for the object is provided for Functional + placed on a publicly accessible archival repository.	Method
Research Object Reviewed (ROR)	The results of the article have been obtained independently in a study by a team or reviewer other than the authors, without the use of artifacts provided by the authors.	Method
Results Replicated (RER)	ROR + ORO + without the use of artifacts provided by the authors, the main results of the article were obtained independently in a subsequent evaluation by a reviewer or team other than the authors,	Method
Results Reproduced (ROR-R)	ROR + ORO + the main results of the article have been obtained in a subsequent evaluation carried out by a reviewer or team other than the authors, using, in part, artifacts provided by the author.	Method
FAIR-TLC[80]		
Findable	Data must have rich and accurate metadata that allows for easy discovery and identification	Data
Accessible	Data must be available and accessible for free or through clear and well-defined mechanisms	Data
Interoperable	Data must be structured and organized in a coherent way so that it can be combined, integrated, and used in conjunction with other data	Data
Reusable	The data must be made available under a clearly specified open license that allows its use and reuse by other users	Data
Traceable	The data must be accompanied by information that makes it possible to trace its origin and the way in which it has been modified or processed. This includes provenance information	Data

Licensed	The data must have a license or a legal agreement that specifies the terms and conditions for its use and reuse	Data
Connected	The data must be linked to other related data sets and relevant resources, such as scientific publications, source codes, and documentation, among others	Data

Table 6: Classification of reproducibility criteria for Artifacts Description (AD) and Artifacts Evaluation (AE).