Empirical Software Engineering (2026) 31:83
https://doi.org/10.1007/s10664-025-10797-w

An audit of machine learning experiments on software defect prediction

Giuseppe Destefanis[1] · Leila Yousefi[2] · Martin Shepperd[2] · Allan Tucker[2] · Stephen Swift[2] · Steve Counsell[2] · Mahir Arzoky[2]

Received: 18 October 2024 / Accepted: 18 December 2025

Abstract

Background Machine learning algorithms are increasingly being proposed to solve the problem of predicting defect-prone software components. In this literature, computational experiments are the primary means of evaluating and comparing learners and the credibility of findings depends critically on their experimental design and reporting.

Objective This paper audits recent software defect prediction (SDP) experiments by assessing their experimental design, analysis and reporting practices against widely accepted norms from statistics, machine learning and empirical software engineering. Our aim is to characterise the current state of practice and evaluate the reproducibility of published findings.

Method We undertook an audit of relevant studies published from the SCOPUS database (2019-2023) focusing on their experimental design and analysis choices e.g., the outcome variables such as F-measure and the type of out of sample (OOS) validation regime, e.g., cross-validation, plus the statistical analysis and inference mechanisms. In all, we evaluated nine different study issues. This was complemented by an assessment of reproducibility using the instrument proposed by González-Barahona and Robles.

Results Our search located approximately 1,585 experiments in SDP (2019-2023), a substantial body of work. From this, we randomly sampled 101 ($\approx 6.4\%$) papers, 61 journal and 40 conference papers. Almost 50% are behind 'paywalls'. We found considerable divergence in research practice. The number of datasets used ranged 1-365, the number of learners or learner variants evaluated from 1-34 and the number of performance metrics from 1 to 9. Approximately 45% of papers made use of formal statistical inference. We detected a total of 427 issues distributed across 101 papers (median=4) with only one paper being entirely issue-free. In terms of reproducibility, experiments ranged from near perfect to lacking almost all required information. We also found two examples of tortured phrases and potential "paper mill" activity.

Conclusions Approaches to designing and reporting computational experiments varied greatly, but almost half the studies provided insufficient information such that reproduction would be challenging. Overall, our audit suggests that as a research community, we have considerable scope for improvement. Fortunately, many improvements should be neither difficult nor costly to achieve.

Communicated by: Leandro L. Minku.

Extended author information available on the last page of the article

Published online: 20 February 2026

Keywords Software defect · Machine learning · Audit · Research review

1 Introduction

> "A variety of recent studies, primarily in the biomedical field, have revealed that an uncomfortably large number of research results found in the literature fail this [quality] test, because of sloppy experimental methods, flawed statistical analyses, or in rare cases, fraud." ACM Artifact Review and Badging (2020) (ACM 2020).

Sadly, there is mounting evidence that software engineering is not immune from such quality problems of questionable research methods and poor reporting. This is compounded by growing activities of "paper mills" and other sources of fake papers (Candal-Pedreira et al. 2022). While it might seem somewhat negative to focus on problems, we argue that this is an essential step in the journey to improve research practice.

Although machine learning algorithms are being widely touted as an effective means of classifying software components into those that are likely to be defect-prone and those that are not, it is proving difficult to obtain an overall picture of what this large and rapidly growing body of research is actually telling us. For example, in 2012 Menzies and Shepperd published an editorial on the lack of "conclusion stability" (Menzies and Shepperd 2012) in the field. A subsequent meta-analysis indicated that who conducted the research, i.e., the team, was a major source of variability in results (Shepperd et al. 2014). Similarly, Li et al. (2019) found that "predictive power is heavily influenced by the evaluation metrics and testing procedure".

Baltes and Ralph (2022) undertook a more general review of empirical software engineering studies and exposed widespread problems with the way samples are constructed and interpreted. More specifically to software defect prediction (SDP), Liu et al. (2021) in their systematic review of deep learning algorithms, showed cause for considerable concern regarding both the reproducibility and the replicability of such research. Related questions are being asked about the underlying research quality, experimental design and reporting of much of this growing body of work, e.g., Shepperd et al. (2019); Liem and Panichella (2020). These concerns are likely contributors to the low take up by practitioners (Rana et al. 2014; Stradowski and Madeyski 2022).

So why is this audit called for? First, SDP is an extremely active and growing research area. Second, combining these many studies into a coherent body of knowledge is proving quite challenging, Mohammadi et al. (2023). Third, much of the methodological research literature focuses on experiments using human participants so we thought it would be interesting to seek some contrasts with computational experiments.

This paper makes the following contributions in that we:

1. Conduct an in depth audit (i.e., a systematic technical review against established methodological standards) of computational experiments in SDP over the past five years (2019-2023),
2. Examine factors associated with (i) study reproducibility, (ii) experimental and reporting issues,[1]

[1] We prefer the less emotive term of 'issue' to 'problem' because we do not wish this audit to be construed as some kind of "witch hunt".

3. Make a set of recommendations as to how we as an SDP research community might improve our practice.

Please note that in the spirit of reproducibility, our raw data, the meta-data and analysis code (R embedded in an RMarkdown notebook) can be found in our Zenodo repository (Destefanis et al. 2024).

The remainder of the paper is organised as follows. In the next section, we review approaches to scrutiny and audit in science and particularly software engineering. In Section 3, we describe the conduct of our audit and the data collected. Next, in Section 4 we present and analyse the results from the 101 audited papers, followed by a discussion in Section 5 of the significance of our findings. Finally, we conclude with a summary of the key points arising from this analysis, various recommendations to the community along with consideration of its significance, weaknesses and areas for further research.

2 Background

In this section, we discuss the context of our SDP audit drawing from both software engineering and more widely in order to understand the backdrop to our audit and the motivation to undertake it.

2.1 Problems in Scientific Research

Back in 2005, in a highly controversial paper, Ioannidis (2005) suggested that most scientific results were wrong. Subsequent studies have lent some support to this viewpoint, e.g., Earp and Trafimow (2015); Nuijten et al. (2016); Héroux et al. (2023). Diong et al. (2018) conducted an audit of physiology and pharmacology papers comparing two time periods (2007-2010 and 2012-2015) and found extensive statistical errors and poor reporting practices with little evidence of improvement. More recently (2023), from an analysis of papers published in psychology, Brewin (2023) reported that they were “marred by multiple errors and inaccuracies and often fail to reflect the changing nature of the knowledge base”.

Closer to our focus of machine learning applied to SDP, Kapoor and Narayanan (2023) report there are many methodological pitfalls, including data leakage, essentially where the separation between training data and unseen validation data is violated. The consequence is over-fitting leading to frequent over-optimism and over-claiming by researchers. Once these problems are accounted for, they suggest that modern, complex methods frequently fail to out-perform simple benchmarks such as logistic regression. Also addressing machine learning experiments—though focused on the domain of medical imaging—is the critique (rather than formal audit) by Varoquaux and Cheplygina (2022) who also note widespread methodological problems and sources of bias, in part arising from the desire to publish and the quest for novelty.

Fazekas and Kovács (2024) examinined results from machine learning experiments in an audit that looked at the consistency of reported classification performance metrics in medical image processing. Typically, they found researchers report multiple metrics, many of which have structural relationships thus enabling consistency checks. Notably, they identified inconsistencies in $\sim 30\%$ of papers from an audit of 100 highly-cited scientific papers.

2.2 Problems in Software Engineering

Within software engineering, we can go back to a seminal paper in 2002 by Kitchenham et al. (2002) who undertook, to the best of our knowledge, the first critical review or audit of the empirical software engineering literature with a goal to assess statistical practice and make recommendations. They concluded, that "software researchers often make statistical mistakes"; however, the analysis was based on only eight, "non-randomly sampled" papers. Nevertheless, this is a highly influential paper in addressing methodological quality in software engineering research.

Specifically, within SDP, pioneering work was undertaken by Bowes et al. (2014) who developed a checking tool DConfusion to assess the consistency of a confusion matrix which is a 2×2 matrix of true-positive, false-positive, false-negative and true-negative counts. This forms the basis for the majority of classification metrics and thus the basis for consistency checking.[2] They illustrated the utility of DConfusion on a set of example papers and revealed consistency problems that were subsequently confirmed by the authors as indeed arising from a typographical error.

These ideas were then deployed in a previous audit[3] by ourselves examining 49 papers drawn from a systematic review of unsupervised learning algorithms applied to SDP (Shepperd et al. 2019). Disturbing findings included that $\sim 45\%$ (22/49) papers contained demonstrable errors. In addition, incomplete reporting on many occasions made it difficult to determine whether an error was present or not.

So to summarise, problems in scientific research papers abound, as has been revealed by multiple studies and audits. Preliminary evidence suggests that the use of machine learning techniques for software defect prediction are not immune; consequently, we decided to undertake a larger and more extensive audit of recent (2019-2023) experiments.

2.3 Reproducibility and Replicability

At this juncture, we need to be clear that reproducibility is distinct from replicability, since assessing study reproducibility is one of the motivations for this audit. RQ3 explores how well SDP studies report on their methods and results in order to support reproducibility.

Reproducibility: is the ability to recreate the results of a study by using the same data and following the exact same procedures as the original experiment. In terms of the ACM Badges definition (ACM 2020), we mean both repeatable (by the same team) *and* reproducible (by another team).

Replicability is the ability to obtain consistent results across different studies, potentially by different researchers and under potentially different conditions. This may involve using different samples, locations or even slight variations in methods.

Reproducibility is therefore a necessary precursor to replicability (Madeyski and Kitchenham 2017). It is an important and often overlooked concept in empirical research that serves at least two purposes. Firstly, it promotes a high level of confidence in the results, that they are demonstrably correct. Secondly, reproducibility facilitates future replication.

[2] The work by Bowes et al. (2014) underpins many of the ideas of the more recent audit by Fazekas and Kovács (2024).

[3] Note that our audit extends the previous findings with an entirely new, larger sample, broadened scope and also examines the question of study reproducibility

Specifically for software engineering, are assessment proposals from González-Barahona and Robles (2012, 2023) and also Liu et al. (2021). We largely follow the original approach of González-Barahona and Robles (2012) for reasons of simplicity and to not “reinvent the wheel” (see Section 3.3 for a more detailed description of our approach).

Related is a move towards Open Science or Open Research. Clearly, undertaking such activities as sharing data, analysis methods and other research artefacts is likely to underpin reproducibility and replicability. Initial calls came from the wider scientific research community, e.g., Munafò et al. (2017) and were in part triggered by the widespread inability to replicate results ind disciplines such as psychology (Open Science Collaboration 2015). This led computer scientists and software engineering researchers to explore whether similar problems existed in our research fields. An investigation into Open Science and the lack of shared artefacts in software engineering by Heumüller et al. (2020) of 789 ICSE research track papers between 2007 and 2017 showed a positive trend towards artefact availability, but even by 2017, only 58.5% of studies made their research artefacts available.

To summarise, there is a growing sense within the software engineering community that Open Science and reproducibility of a study are important but not complete consensus as to what this entails and, clearly, still some way to go in achieving this.

3 Method

Our goal is to evaluate the quality of published computational experiments in the domain of software defect prediction. Since we want to consider recent and current practice, we focused on the past five years, i.e., 2019-2023.

To achieve this, we undertook an audit, which Ralph and Baltes (2022) refer to as a critical review.[4] This is an evaluative process that assesses the quality, reliability and validity of existing primary studies in a specific field. It entails a detailed examination of the research design, methods, data analysis and findings of individual studies with the goal of identifying strengths, weaknesses, biases and gaps. So the purpose is to provide a comprehensive understanding of the state of research on a particular topic and to offer constructive feedback for improvement. In software engineering, they have been deployed to investigate methodological topics. For example, Baltes and Ralph’s critical review/audit (Baltes and Ralph 2022) evaluates how software engineering researchers use and report on the representativeness of the sample of the target population.

Note that we are conducting a systematic technical review against established methodological standards rather than an audit against a particular pre-registered or published standard. Note also that this is distinct from a traditional systematic review or meta-analysis, where the emphasis on evidence generated by primary studies; the goal in that case would be to synthesise all relevant studies on a specific research question to provide a comprehensive summary of the evidence. Low quality studies are typically excluded (or at least downweighted). This contrasts with an audit, where we explore the quality of the studies, not the evidence. We summarise the process diagrammatically in Fig. 1.

[4] We prefer the term audit, as it is more widely used beyond software engineering, e.g., in the context of statistical methods.

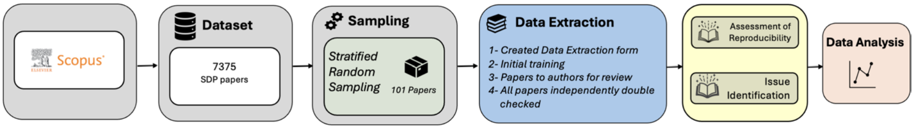


Fig. 1 Overall audit process

Specifically, we use the audit to answer the following questions:

1. Using bibliometric data, how can we characterise our sample of SDP studies?
2. What experimental design approaches are used in SDP research?
3. How reproducible are SDP studies?
4. What kinds of quality issues are found in SDP studies?

3.1 Search

We chose Elsevier's Scopus database (Baas et al. 2020) for two reasons. First, like Web of Science (WoS), it is subject to various quality requirements such as ensuring that all papers are subject to at least some minimal peer review process and has active policing of fake journals and so-called predatory publishers. Second, it has a broader scope than WoS and, in particular, better coverage of conferences. As a result, our population excludes the so-called grey literature (Kamei et al. 2021) and covers fewer venues than compared to a database such as Google Scholar. This was intentional, since our focus is on mature, refereed studies and what is hopefully, state-of-the-art practice.

For the actual search, we used the following, covering all fields, but restricted to: (1) articles and conference items, (ii) computer science, (iii) English items and (iv) years 2019-2023.

```
( "software defect" OR bug OR "software fault" )
AND prediction
AND "machine learning"
```

This retrieved 7,375 documents (09/04/2024). Given the large numbers, we undertook a stratified random sample to select 20 papers from each year to achieve a target of 100 papers.[5] Our rationale was that we were interested in changes over time and in particular whether there were any improvements. To achieve this, we randomised papers within year and searched until a target of 20 relevant papers satisfying all inclusion criteria was achieved. These were:

1. A focus on predicting software defects
2. Presenting a new experiment with results
3. Based on real-world data as opposed to simulated data or student experiments
4. Availability of the full content (14 articles excluded)
5. A minimum length of 3 pages

[5] Our total sample size was 101 due to the additional inclusion of an initial training paper.

A common concern is whether 100 papers can 'represent' all SDP studies (2019-2023). However, our sample was probabilistically drawn from a fully enumerated sampling frame, ensuring that each article had an equal probability of selection. This allowed us to draw unbiased estimates of population-level properties with calculable margins of error.

To estimate the proportion of the total number of relevant articles sampled from, we made the following calculations given in Table 1. Not all the 7,375 articles retrieved by the search were relevant. In finding the required number (101) for the audit sample, we discarded 354 from which we could extrapolate to estimate the total number of relevant articles. From this, we suggest that between 2019 and 2023 an *estimated* 1,585 primary studies on experiments into SDP have been published and indexed by SCOPUS. Since SCOPUS does not cover all software engineering conferences, the true figure is likely to be greater than this. Our audit sample reviewed 6.4% of these articles.

3.2 Outcome Variables

Our audit addresses two outcome variables: reproducibility and issues (which are split into experimental design plus implementation issues and reporting issues).

3.2.1 Reproducibility Metrics

To assess the reproducibility of each paper, we adapted the Instrument proposed by González-Barahona and Robles (2012) and revisited in 2023 (González-Barahona and Robles 2023). Our adaptation was to simplify that aspect of the instrument related to the raw data collection due to all experiments in our audit being based on secondary data. González-Barahona and Robles' approach has the merit of being quite generic and simple to apply since the questions are all either yes or no. We took the view that all questions should contribute equally and then normalised the total score to give a range between zero and one.

There are 27 basic indicators scored (0 or 1) e.g., indicating how easily the element, e.g., a data set, can be identified for replication (see the Appendix for full details). These fall into five categories:

1. Identification: is the element actually specified or identified?
2. Description: The detail level of the published information about the element.
3. Availability: Indicating how easily the element can be obtained.
4. Persistence: Indicating the likelihood of the element being available in the future.
5. Flexibility: How easily the element can be adapted to new environments.

Table 1 Sample calculations

Articles	Count or proportion
Total from SCOPUS	7,375
Articles manually searched	469
Irrelevant articles rejected	354
Unavailable articles rejected	14
Articles in final sample	101
Hit proportion (101/469)	0.215
Estimated total relevant articles (7375 × 0.215)	1,585
Sample proportion (101/1585)	6.37%

Of course, there is some subjectivity in scoring, especially for borderline cases. We believe the use of two independent reviewers and the fact that are 27 items ameliorates this problem, (revisited in Section 6 Threats to Validity). These categories are then applied to five aspects of a study, namely:

1. Raw data
2. Extraction methodology/tools
3. Processed dataset
4. Analysis methodology/tools
5. Results dataset

The remaining two indicators relate to the specification of relevant hyperparameters. This results in an overall Reproducibility Score out of 27 then normalised to 0-1.

3.2.2 Study Issues Assessed

This section defines and explains the nine issues we checked for in each article. These fall into two groups. Issues 1–5 concern the design and conduct of the experiment, while Issues 6–9 concern reporting. We selected issues that are widely relevant, detectable, non-trivial and, as far as possible, uncontentious. For this reason we avoided more debatable topics such as the use of p-values in NHST (Benavoli et al. 2017). We also aimed to be sensitive to context, since different studies have different goals that may call for different design choices. At the same time we recognise that, although a range of practices persists in SDP, the broader statistics and machine learning communities have shifted toward more robust and transparent approaches. In short, this is an audit rather than a critique, and our aim is to be flexible and only flag issues that threaten the validity of an SDP experiment or limit its scientific value.

1. Lack of Out of Sample (OOS) validation: Distinct from model-fitting (using all available data), SDP studies seek to evaluate the predictive performance of the learner where it deployed upon unseen data. This is accomplished by simulating the process of using a trained learner to predict the target variable of a new and unseen instance. OOS validation strategies commonly used in machine learning experiments include:

    1. Holdout / train–test split: The dataset is randomly split into separate training and testing sets and the performance of the held out test set reported. Typically this is in a fixed ratio (e.g., 70% training, 30% testing) (Provost and Fawcett 2001) Clearly such a procedure is vulnerable to the chance allocation of individual instances.
    2. k-fold Cross-Validation (CV): Since the 1990s the standard approach for OOS validation has been cross-validation (Kohavi 1995). This works by randomly allocating instances to k approximately equal-sized folds. One fold is then used as the validation or ‘unseen’ set and the remaining $k - 1$ folds are used for training. This procedure is repeated so that after k iterations every instance has been entered once into the validation fold. Kohavi recommends $k = 10$, although there appears some flexibility in practice. Some researchers advocate the use of $j \times k$ cross validation

where the whole procedure is repeated *j* times to reduce errors in the estimate of the mean prediction metrics.

3. Leave-One-Out Cross-Validation (LOOCV): Each instance in the dataset is used once as a test set while the remaining instances form the training set (Wong and Yeh 2020). This approach is deterministic but can be computationally very demanding.
4. Bootstrap Resampling: This involves repeatedly sampling with replacement of the dataset to create training and testing sets. It provides robust estimates of model performance (Kohavi 1995).
5. Cross-Project Validation: Training is performed on one project and testing is done on a different project in order to evaluate generalisability across projects (Menzies and Shepperd 2012).
6. Time-Based Validation: Used in scenarios like Just-in-Time (JIT) defect prediction, where the training and testing sets are divided based on temporal order to simulate real-world deployment (Stapor et al. 2021).

Clearly some strategies are more appropriate/realistic for SDP such as cross-project and time-based, however, for the purposes of our audit we merely determine whether OOS validation has been undertaken.

7. Using problematic metrics: this is particularly relevant for classifiers where a number of researchers in multiple fields have convincingly demonstrated that some widely used performance metrics such as F1 and accuracy are biased and unreliable for two-class classification problems (see, for instance Powers 2011; Chicco and Jurman 2020; Yao and Shepperd 2021; Lavazza and Morasca 2022). Specifically, Accuracy is inadequate under skew as majority class dominance can yield deceptively high scores even for trivial predictors. F1 was proposed for information retrieval problems (van Rijsbergen 1979) which are generally 1-class, e.g., the number of relevant pages correctly retrieved whilst the number of irrelevant pages not retrieved is both unknowable and uninteresting. For 2-class problems it is vulnerable to bias and is difficult to interpret. Correlation metrics, e.g., MCC uses all four cells (TP, TN, FP, and FN) and are typically less biased than Accuracy or F1 on imbalanced data. However, MCC is *not completely unbiased* and can degrade under *extreme* skew (Zhu 2020). To continue to use such metrics is problematic. We did however, check the specific use and context of problematic metrics for each paper so, for instance, where a metric was reported but flagged as potentially misleading, this was *not* recorded as an issue.
8. Benchmarks and better than random: another problem with predictive performance metrics is the need for some benchmark to establish that the learner is doing better than guessing. Metrics such as Area Under the Curve (AUC) and Matthew's correlation coefficient (MCC) are chance-anchored (Chicco et al. 2021). Youden's *J* (Bookmaker odds / Informedness) provides another chance-anchored view at a chosen threshold and is a principled alternative when an operating point must be fixed, e.g., when using logistic regression. Failure to include such a benchmark is again problematic. For the record, Yao and Shepperd (2021) showed that 16 out of the 33 reviewed studies contained at least one result worse than random, that is, it was a perverse predictor.

9. Multiple statistical tests without correction: computational experiments such as SDP studies typically generate many results ($\#Learners \times \#Datasets \times \#Metrics$ and these are often evaluated using statistical inference tests. Again, it is well known that without making some kind of correction, the false positive rate will be greatly inflated (Midway et al. 2020). In the spirit of only focusing on clearly problematic issues/methods we accept *any* method to correct alpha as being acceptable although obviously a more modern approach is preferable e.g., Benjamini and Hochberg (1995).
10. Addressing spread as well location: typically results are given as a single statistic which only summarises the central tendency of the mean predictive performance. This reduction of many results to a single summary statistic has some ramifications, such as the extent to which results may vary due to the stochastic nature of both algorithms and validation procedures being unknown. Boxplots of the performance metrics are an effective, and quite widely used graphical method to indicate dispersion.
11. No link to data: the data used is not fully indicated e.g., version or location.
12. No link to code: the relevant code is not shared.
13. Low reproducibility: the level of reporting for reproducibility is unacceptably low, which we define as a score of less than 50% for the González-Barahona and Robles (2012) instrument meaning that 14+ out of 27 questions are answered 'No'.
14. Article behind a 'paywall': Given the widespread opportunities to provide a final post-print version of the paper in a publicly accessible, not-for-profit archive, not doing so seems both perverse and harmful to the progress of research.

3.3 Data Collection and Analysis

For the audit, all authors independently reviewed one paper which we then collectively discussed and resolved disagreements. The guidance notes and definitions were then refined. Each author was then allocated two blocks of 14-15 papers, meaning that all papers were independently read and reviewed by two authors and again disagreements resolved. This process enabled us to extract the following data (for full details refer to the Appendix).

This information covered the following areas:

1. Bibliographical details: publication venue, year, page count and citation information of the research paper being analysed.
2. Open Access: indicates the type of access e.g., Gold, Green, etc.
3. Document Type: specifies if the publication is a journal or conference article.
4. Experimental design information including number of datasets, learners and choice of performance metrics and out of sample validation methods.
5. Choices of benchmarks e.g., comparison with a random predictor.
6. Statistical methods including how the results are summarised, type of inference and statistical tests.
7. Reporting details and reproducibility score.

Note that the data analysis was performed using R embedded in an RMarkdown Notebook which is shared on zenodo along with our data.

4 Results

In this Section we present the results of our audit in order of the four research questions.

4.1 RQ1: Using Bibliometric Data how can we Characterise our Sample of SDP Studies?

There are 40 conference and 61 journal papers in our sample from 2019-2023. A conspicuous feature of our sample is the diversity of sources, i.e., distinct journals and conferences. The 101 papers cover 74 unique sources or venues. This underscores the scale of research activity, the distributed nature of publications and the challenges of maintaining a comprehensive and up to date understanding of the field. Next, we examine paper length which ranges from 3 to 46 pages. The minimum is an result of our decision to exclude any paper shorter than 3 pages from the audit, on the grounds of it not having sufficient space to fulfil the norms of scientific reporting. The median length is 12 pages, reflecting the page restrictions typically imposed by conference publication (See Fig. 2).

Then we looked at whether papers were open access or not. Aside from moral or ethical considerations, there is also growing evidence that this has a marked impact on the diversity and volume of citations from other researchers (Huang et al. 2024). From our analysis, we see that 50/101 i.e., just under 50% of papers are behind paywalls and therefore open access (see Table 2).

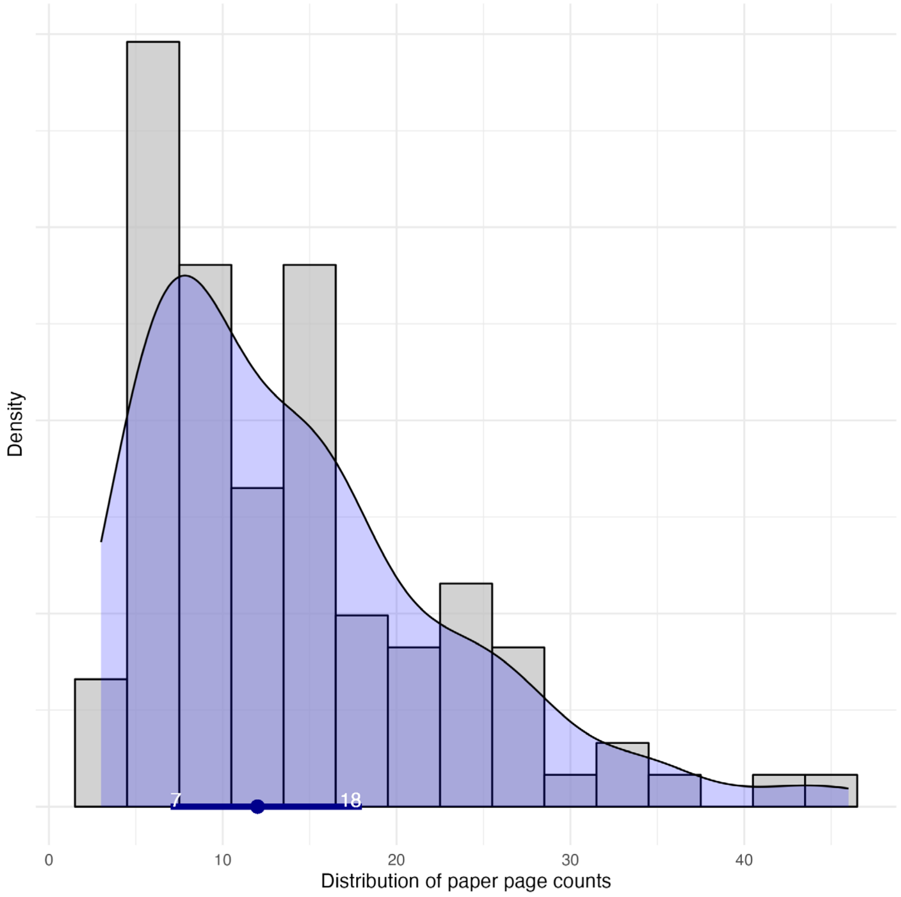
All the mainstream software engineering publishers e.g., Elsevier, IEEE and Springer allow the author to post a postprint, meaning Green Access. What is frustrating is that for most Green published papers (41/64), authors have chosen *not* to make a postprint available (for example on their own institution website or a public, not-for-profit archive such as arXiv or zenodo).

Gold or Diamond Access is when either the author's institution pays the publisher either as optional charges (e.g., Elsevier) or sometimes as a compulsory article processing fee (e.g., PLOS ONE, Frontiers, etc.); this allows the article to be found on the publisher's website and freely downloaded by anyone. For more detail and explanation see Harnad et al. (2004); Meagher (2021).

We also investigated paper citation information (see Fig. 3). This reveals a strong positive skew (not least because negative values are impossible for count data) ranging from 0 to 120, with a median of 4. The IQR of 1-13 is highlighted on the x-axis of the histogram as a dark blue bar. The high variance was somewhat surprising (to us) nor did it particularly fit our expectations of what might be thought of as seminal papers. We investigated self-citations which were generally too low (0-10 with a median of zero) to explain the variability.

We next normalised the citations by number of years, since publication (see Fig. 4); there remains high variability ranging from 0 to 30. The median for all years is only 1.2 citations per year and indeed 25/101 papers have no citations.

Next, we ask the question: does publication type influence citations? Given the highly skewed distributions, we compare medians and the 95% confidence intervals (see Table 3). There is some evidence of an effect since the median citation rate per year for a journal paper is 2.2 compared with 0.78 for a conference paper; however, the 95% confidence intervals overlap in part due to the high variance of paper citation counts, even when normalised.



The horizontal blue line along the x-axis shows the interquartile range, with Q1 and Q3 values given in white and the median represented as a blue circle. Thus, in this distribution, 50% of the papers are between 7 and 18 pages in length.

Fig. 2 Distribution of paper length by page count. The horizontal blue line along the x-axis shows the interquartile range, with Q1 and Q3 values given in white and the median represented as a blue circle. Thus, in this distribution, 50% of the papers are between 7 and 18 pages in length

Table 2 Counts of paper access types

Paper Access Type	Count	%
Gold/diamond	29	28.7
Green	23	22.8
Green (Not available)	41	40.6
No	8	7.9
Total	101	

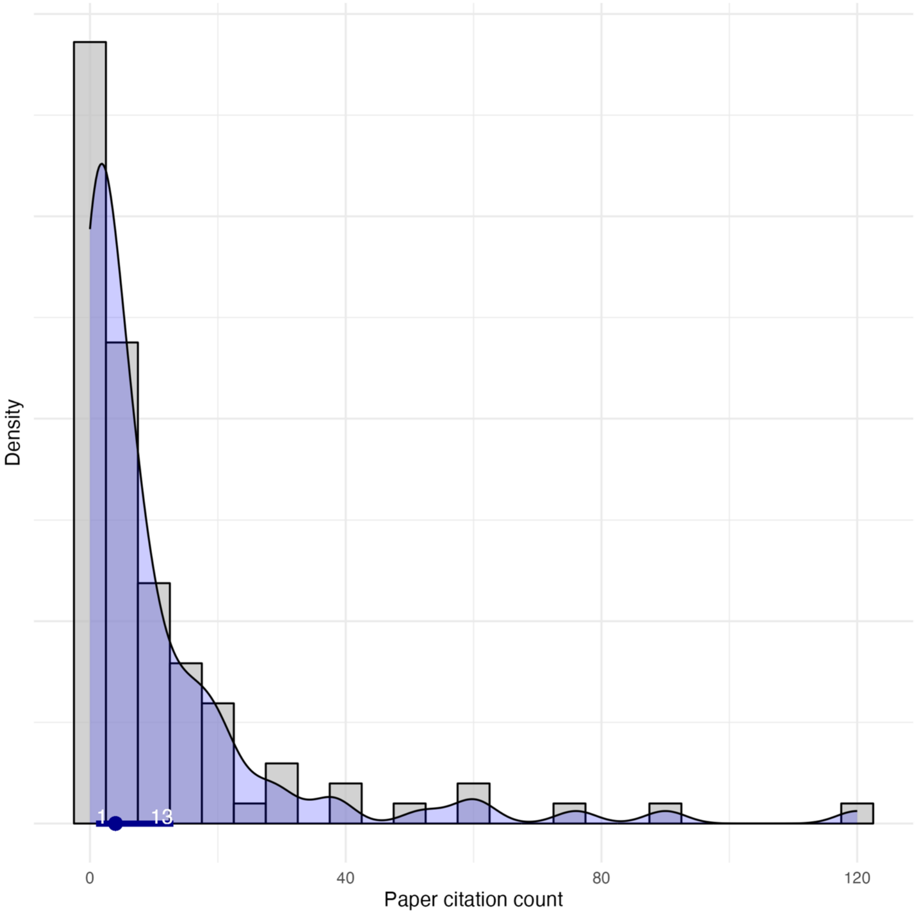


Fig. 3 Distribution of paper citation counts (unnormalised)

We also observed two papers containing multiple examples of tortured phrases i.e., “unexpected, weird phrases in lieu of established ones” (Cabanac et al. 2021) such as, in our audit, “novel insects” instead of “new bugs” and “arranging of deformities” instead of analysis of defects. Such phrases are usually indicative of problematic papers arising from the desire to evade plagiarism checkers by translating text from English to a second arbitrary language, sometimes multiple times and then finally translating back to English. Typically, these tactics are deployed by “paper mills” or other “bad actors” to create fake scientific papers (Richardson et al. 2025). Apart from the concern that the scientific body of work is being contaminated by these meaningless papers, it suggests a lack of oversight by the community and indeed both papers have been cited by others.

Since the tortured phrases occur from the very start of each paper, it does cause us to wonder how carefully they have been read. For this reason we unsuccessfully approached the authors, editors and publishers recommending they be retracted. The papers concerned are (i) “Machine Learning-Based Defect Prediction for Software Efficiency” published in 2023 in *The International Journal of Intelligent Systems and Applications in Engineering*

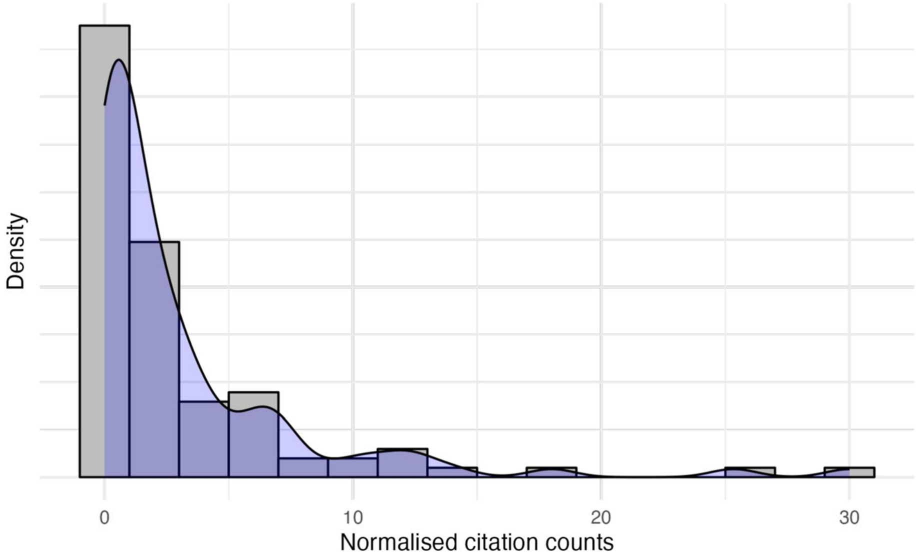


Fig. 4 Distribution of paper citation counts normalised by year

Table 3 Median citations per year by paper type

Paper Type	Median	Lower Bound	Upper Bound
Conference	0.775	0.60	1.33
Journal	2.200	1.00	3.40

and (ii) "Software Defect Prediction Framework Using Hybrid Software Metric" published in 2022 in *The International Journal on Informatics Visualization*. We also informed Scopus who responded that they no longer index *The Intl. J. of Intelligent Systems and Applications in Engineering* and are investigating (2/10/2024) the International Journal on Informatics Visualization.

It is hard to estimate the overall prevalence of papers containing tortured phrases. Apart from the two papers we located on our sample of 101 papers, by using the Problematic Paper Screener curated by Cabanac et al. (2022), we located a further 9 contained in our initial Scopus search of 7,375 papers. This is likely to be an underestimate because it is hard to anticipate a tortured phrase until it is encountered and Cabanac's background is the scientific literature more generally. However, using the two positive observations from a sample of 101, the exact Clopper-Pearson confidence interval, which is generally more reliable for small sample sizes or proportions close to 0 or 1 gives a 95% confidence interval of [0.002, 0.070]. In other words, a prevalence of between 0.2% and 7%.

So, to summarise, there seem disappointing levels of open access, much variability in citation patterns and in paper length and weak evidence that journal articles tend to be more highly cited. Worryingly, even in a carefully curated research paper collection such as Scopus, we see clear evidence of domain-incoherent terminology and suspect papers.

Table 4 Experimental design variables

Count	Min	Q1	Median	Mean	Q3	Max
Datasets	1	6	10	22	19	365
Learners	1	3	5	6	7	34
Metrics	1	2	3	4	5	9

4.2 RQ2: What Experimental Design Approaches are used in SDP Research?

First, we categorised papers by whether the prediction system under experimental investigation was a classifier — in practice a dichotomous classifier — or a regression system predicting fault count or severity. As it transpired, almost all studies focused on classifiers, although sometimes as part of a larger objective such as the effort-aware approaches. Only 4 out of 101 studies directly used regression systems to predict continuous valued outputs. It should be noted that regression systems naturally lead to different types of prediction accuracy metric; however, for the purposes of this audit, we do not differentiate between them and classifiers and use the more generic term "learner".

We observed considerable variability in experimental design. Table 4 shows summary statistics for the number of datasets (this includes new releases or versions of a system), distinct learning algorithms and performance evaluation metrics. If there is such a thing as a typical study, then it evaluated 5 learners over 10 datasets or system versions and used 3 performance metrics to make comparisons. Typically, the results were presented in tables, so for our hypothetical study, this would result in $5 \times 10 \times 3 = 150$ cells or results from which preference relations can be established.

Almost all studies used secondary datasets. The number ranged from 1 to a remarkable 365 with a median of 10. The Promise repository datasets dominated, with it seeming to serve a similar role to the UCI datasets used in machine learning experiments more generally.

Data pre-processing strategies also varied greatly. One important area is how to address the challenge of imbalanced training data in that the positive cases (i.e., defective components) are almost always in the minority. Typical strategies include SMOTE (Fernández et al. 2018). Overall, 42/97 studies explicitly used some imbalanced learning mechanism. We exclude regression studies since the notion of imbalance is not relevant to continuous target variables. Given the almost universal use of imbalanced datasets, this proportion appears low. However, we decided not to treat it as a quality issue since it depends on the specific purpose of the SDP experiment and optimising the prediction is not always the goal.

Similarly, there is significant variation in the number of learning algorithms[6] investigated. The median is 5 though the maximum was 34. There is a certain degree of subjectivity regarding whether some pre-processing, e.g., feature subset selection, constituted an additional algorithm or was essentially cleaning the data prior to the computational experiment. To determine this, we tried to follow the structure of a paper and, in particular, how the results were presented and interpreted.

We also see some variability in the number and choice of prediction accuracy metrics e.g., accuracy and AUC. These form the outcome variables for the experiments. There is a strong preference for employing multiple metrics, with a median of 3 and a maximum of

[6] In practice, almost all papers looked at classifiers; however, there are also four regression-based predictors aimed at continuous-valued outcomes so we use the more general term.

9. Of course, a challenge is the situation when the multiple metrics are discordant (Yao and Shepperd 2021) and how the paper's results should be interpreted.

Next, we consider the specific choices of accuracy performance metrics. These are given in Table 5. Note that the percentages are of 97 papers since the remaining four deal with regression systems. We see that the F1 metric is still the most widely used metric despite considerable adverse criticism, e.g., Hand and Christen (2018); Yao and Shepperd (2021). Unsurprisingly, since they are constituent parts of F1, Precision and Recall are also widely used. Area under the [ROC] Curve (AUC) (Provost and Fawcett 2001) is also used in half of the experiments we audited, but this metric conceptually differs from other metrics like F1, as it evaluates the performance of a classifier across a spectrum of decision thresholds, rather than at a single fixed threshold.

Given the well known and widespread criticisms of Accuracy as a performance metric for imbalanced datasets, which is almost invariably the case in the domain of software defect prediction, it is surprising that more than a third (38/97) of experiments still use this metric. Finally, the uptake of the more recently advocated Matthews Correlation Coefficient (MCC) (Baldi et al. 2000; Chicco and Jurman 2020) remains quite low at under 20%.

Of these metrics only MCC and AUC are chance-anchored, in that guessing will give predictive accuracy scores of 0 and 0.5 respectively (irrespective of the prevalence of the positive case). Consequently, it is possible to see how a classifier is performing with respect to a random strategy. In our audit, just under 60% (58/97) of experiments use metrics that allow us to compare the results with guessing. With other metrics such as the popular F1-measure, it is not possible to know how a result compares with simply guessing (Yao and Shepperd 2021). We return to this in the Discussion Section.

Next, we turn to the deployment of Out of Sample validation strategies. Table 6 summarises the range and frequency of approach. The NAs arise when essentially the style of algorithm or experiment means the test and training divide is fixed e.g., time order as in Just-in-time (JIT) prediction. The Unclear category arises when the checkers are unable to

Table 5 Frequency of accuracy metric usage

Metric	Chance	Uses	Percentage (n=97)
	Anchored		
F1	No	61	62.9
AUC	Yes	53	54.6
Recall	No	52	53.6
Precision	No	43	44.3
Accuracy	No	38	39.2
MCC	Yes	17	16.8
Specificity	No	10	10.3
Others	n.a.	86	n.a.

Table 6 Out of Sample validation strategies deployed

OOS validation Employed	Count	Percentage
Yes	66	65.3
No	24	23.8
Unclear	7	6.9
Not relevant (NA)	4	4.0

determine whether an OOS-validation strategy has been used and indicates problems with how the experiment is reported.

Where relevant, we also collected information on the number of folds and number of repetitions. The modal fold count is 10 (which is quite widely advocated in machine learning studies) and the number of replications ranged from one (in 26 studies thus the most common) to 100 (in 6 studies).

Where there was some element of repetition, either through multiple folds or replication (e.g., from repeating a bootstrap) we noted how the results were presented. In almost all cases, a measure of location, usually a mean, but sometimes a median, was used. Less typically, in 32/83 of relevant cases a measure of the dispersion of the results was provided, e.g., graphically using boxplots or a summary statistic such as standard deviation. We view this as necessary reporting since it is important to know how much predictive performance can vary as well as the most typical value.

In terms of inferential mechanisms, $\sim 45\%$ of studies (46/101) made use of statistical tests based around the idea of comparing a p-value with a pre-determined alpha value or acceptance threshold. When the p-value is less than alpha it is declared as statistically significant. Of these 46 studies, only 60% (28/46) made adjustments to alpha when conducting multiple tests (Bender and Lange 2001). There are philosophical and methodological ramifications, e.g., Colquhoun (2014); Greenland et al. (2016), but, suffice to say, not adjusting alpha in the context of multiple tests is problematic. This is because there are typically many inferential tests (the median is 24 and the maximum is 704).

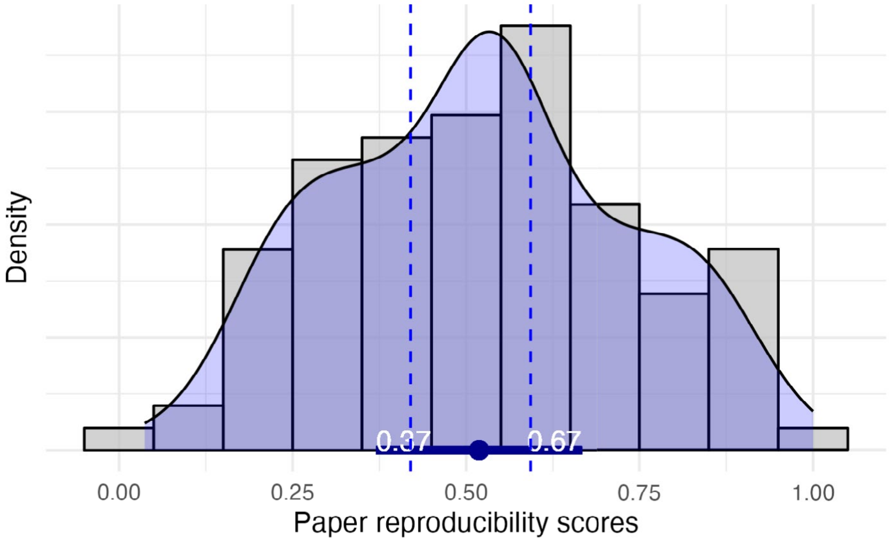
4.3 RQ3: How Reproducible are SDP Studies?

As described in Section 3, we used a variant of the González-Barahona and Robles (2012) Reproducibility Instrument which generated a score from 0-27 which we then normalised between zero and one. We found scores taking almost the entire range of possible values from 0.04 to a perfect 1.0, with a median of 0.52. Figure 5 shows the distribution of reproducibility scores.

We used the odds-ratio derived from the contingency table (see Table 7) to compare the lower and upper tertiles of conference and journal papers following a procedure recommended by Gelman and Park (2009). We therefore have the odds of a journal paper falling into the top tertile for reproducibility, as opposed to the bottom as being 20/7 and compare this with the same odds for conference papers of 14/23. From this, we can construct the odds ratio which is 4.694 with 95% CI [1.582, 13.923]. Although the interval is broad, it does not straddle unity and hence is supportive of journal papers being more reproducible than conference papers.

A common and useful way to assist with reproducibility is to link to the data or code. From the audit sample we found $67/101 = 66.3\%$ of studies contained a link or specific reference to the data used. Note that we excluded vague references to a general repository or where the version was unclear. Only $19/101 = 18.8\%$ of studies included links to the analysis code or scripts. Of the 67 papers that contained one or more links, in 24 cases (35.8%) at least one link was broken.

Next, we explore some associations with, and *possible* contributors to, reproducibility. First, consider the relationship between paper length (in pages) and reproducibility. Here we might expect shorter papers to be less reproducible since there is less space to convey



Additionally to the IQR, the vertical blue dashed lines represent the tertile boundaries.

Fig. 5 Histogram and density plot of paper reproducibility scores. Additionally to the IQR, the vertical blue dashed lines represent the tertile boundaries

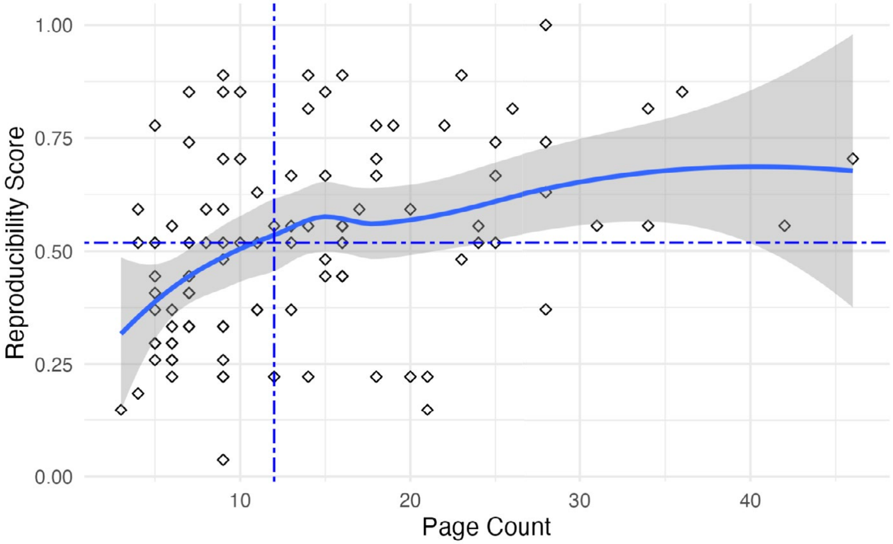
Table 7 Contingency table of Paper type by reproducibility score tertile

	Reproducibility		
Paper type	Tertile 1	Tertile 2	Tertile 3
Conference	20	13	7
Journal	14	24	23

relevant information. As expected, there is a positive relationship (the Spearman correlation coefficient is 0.42) but there is a lot of scatter and an inflexion point around the median of 12 pages (see Fig. 6). However, there are clearly many other factors and inspection of the scatterplot suggests many shorter papers score highly on reproducibility and vice versa. Potential reasons include different journal and conferencing formatting styles. This means that a page does not contain a uniform number of words and author preferences in what information to communicate.

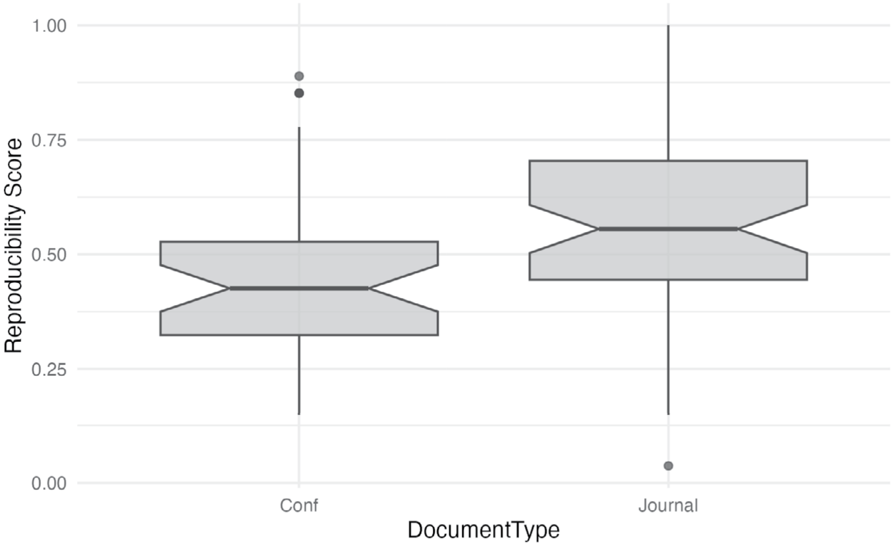
The side by side boxplots in Fig. 7 allow us to compare reproducibility between conference and journal papers. The medians are 0.426 and 0.556, respectively. In terms of effect size, this approximates to 3.5 additional positive answers out of 27 reproducibility questions. Using a robust method (5,000 bootstrap samples) to estimate the 95% confidence intervals for the medians, we find they touch (Conference = [0.352, 0.519] and Journal = [0.519, 0.593]. The method used by ggplot2 in the boxplots makes more assumptions about normality and has tighter bounds. Either way, there is some evidence that journal papers may be more replicable than conference ones.[7] We conjecture that the combination of more

[7] This small distinction between methods of estimating confidence intervals is another reason we chose to avoid traditional significance tests since; in this instance, the choice of method determines whether the relationship is ‘significant’ or not. We believe this to be a false dichotomy and prefer to focus on strength of



The blue fit line is a loess smoother with associated 95% confidence interval in grey. The blue dashed lines represent median values.

Fig. 6 Scatterplot of Reproducibility vs Paper Length. The blue fit line is a loess smoother with associated 95% confidence interval in grey. The blue dashed lines represent median values



The notches show a parametric estimate of the median 95% confidence intervals that assumes near normality.

Fig. 7 Boxplots of Reproducibility vs Document Type. The notches show a parametric estimate of the median 95% confidence intervals that assumes near normality

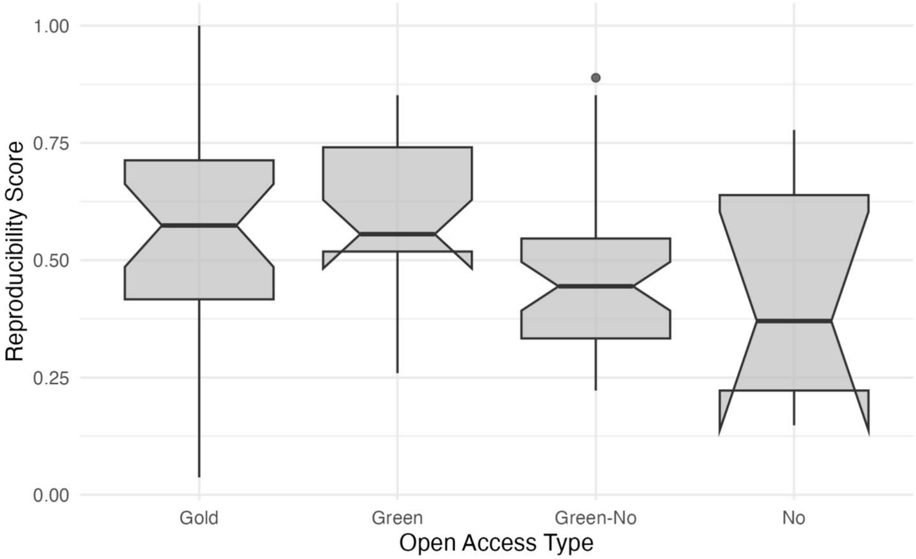


Fig. 8 Boxplots of Reproducibility vs Open Access Type

permissive page lengths and an iterative peer review process can lead to published experiments that are easier to reproduce. Of course, there are some conference papers that are clearly superior to the weaker journal papers; nevertheless, there is some pattern, which in terms of our conjectured causality, makes sense.[8]

Third, we examine the relationship between the type of Open Access for the paper (see Table 2) and Reproducibility. From the boxplots in Fig. 8 it would suggest that despite considerable variation within each category, there is some evidence that Gold and Green Access (where authors consciously choose to make their paper available) tend to be more reproducible than studies where the paper is protected by a paywall. Interestingly, the median score for Gold is lower than that for Green Access.

The relationship between reproducibility and citations per year is weak (Spearman's correlation coefficient = 0.24) indicating limited evidence that more reproducible work is more frequently cited. Nor is there any obvious relationship with experimental design such as the number of datasets (see the correlation matrix in Fig. 9. Note also, the stronger correlations are the consequence of functional relationships, so Total Problems = Experimental Probs + Reporting Problems. Finally, as one might expect, reporting problems are negatively correlated (-0.7) with Total Reproducibility.

Overall, we see a wide range of reproducibility scores with an unimpressive median score of 0.52 (or 14/27 questions obtaining positive scores). There is some evidence to suggest that journal papers tend to be more reproducible than conference papers. Other patterns such as those associated with page length and citations are less clear cut.

evidence and estimating effect size, as opposed to declaring whether an association is true or not (Greenland et al. 2016).

[8] Note that although our modelling approach is not directly causal since we explore association, nevertheless, awareness of potential causal mechanisms helps us interpret the results.

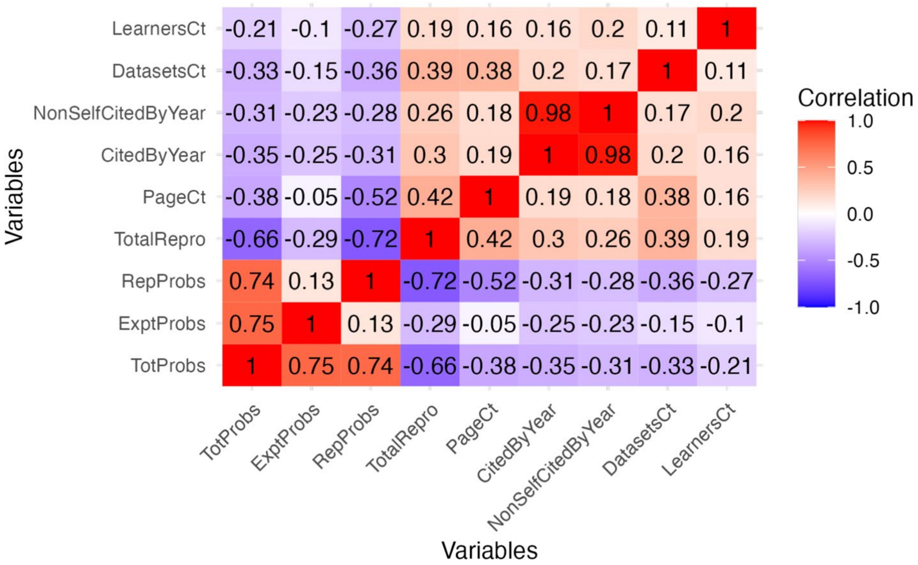


Fig. 9 Heat map of Spearman correlation coefficients

Table 8 Frequencies of issues in studies

Issue	Count
Using problematic metrics	68
Not addressing spread as well as location	59
Benchmarks and better than random	43
Lack of OOS-validation	31
Multiple statistical tests without correction	18
Total study design / implementation issues	219
No link to code	82
Behind a ‘paywall’	50
Low reproducibility	42
No link to data	34
Total issues in reporting	208
Grand Total	427

4.4 RQ4: What Kinds of Quality Issues are Found in SDP Studies?

Here, we summarise the prevalence of the issues outlined in Section 3 that are demonstrably weaknesses or problems in (i) the design and execution and (ii) the reporting of a computational experiment. In undertaking this audit we have sought to consider the context of an experiment; so, for example, if a study does not make use of statistical significance testing then it is not relevant to consider whether significance (alpha) thresholds need adjusting. For some studies, time ordering imposes a single analysis framework and hence there is only a single set of results and reporting the spread or dispersion is not relevant. Taking this into account, we tabulate our overall findings by problem (in decreasing order of prevalence) in Table 8.

Next, we address how issues distribute by individual paper. In total, we found 427 issues distributed across 101 papers (see Fig. 10). These ranged from 0 to 8 with a median of 4 and an IQR of 3-5. While not all the issues will invalidate the findings of a paper, e.g., placing a paper behind a ‘paywall’ does not mean the analysis is incorrect. However, even this hinders independent scrutiny and neither serves the research community nor Open Science well. Notably, only a single paper had zero issues.

The scatterplot (see Fig. 11) shows the distribution of types of issue (experimental and reporting) by paper. It is clear that they are essentially independent. The blue dashed lines represent the medians.

Second, we explore whether there is a difference between conference and journal papers for issues. Examining the boxplots in Fig. 12 would suggest that there is some tendency for journal papers to contain fewer issues than conference papers with medians of 4 and 5 respectively. The notches, representing the 95% confidence intervals, do not overlap, suggesting the difference is non-trivial.

A different approach is to think of the effect size as the difference between the medians for conference and journal papers. Using a bootstrap to establish the 95% confidence intervals for the effect, we see the difference is: 95% CI: 0.037, 0.222 which does not cover a zero or negative effect. Hence, again this gives some support for a *small* difference in issue counts between journal and conference papers. Note that this effect disappears when we *only* consider experimental issues (see Fig. 11) where the scatter of journal and conference data points shows little pattern on the x-axis for experimental issues. We also noted that the open papers tended to have fewer issues (see the side by side boxplots in Fig. 13 where the CI notches of the open papers do not overlap the paywalled papers). It is not clear whether this is a causal relationship, but making a research paper more widely available does not do harm and might possibly encourage authors to be more methodical.

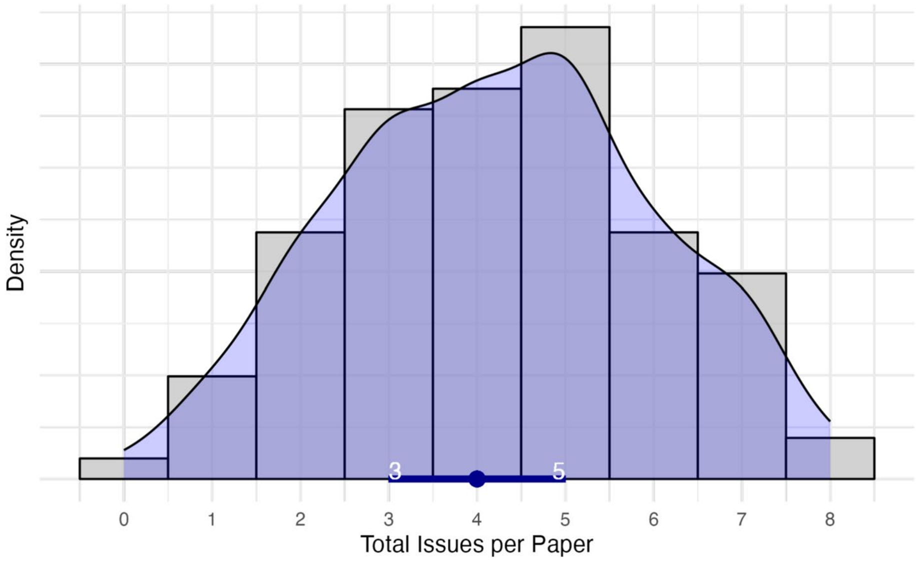


Fig. 10 The distribution of total issues per paper

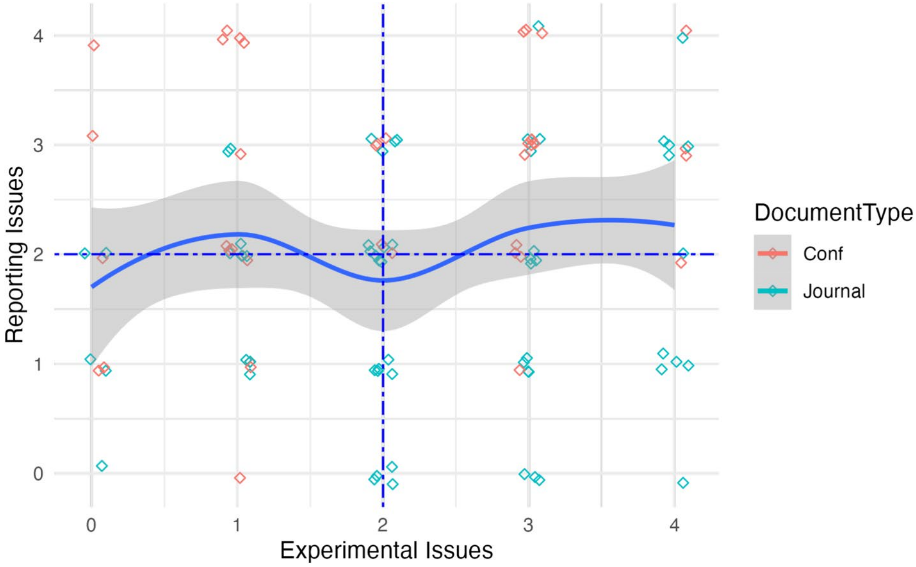


Fig. 11 Experimental versus reporting issues per paper

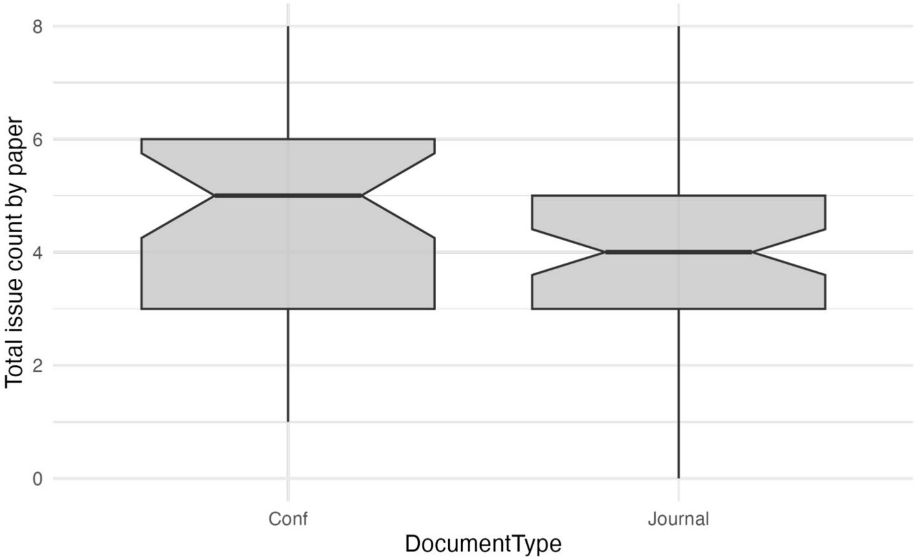


Fig. 12 Boxplots of issue counts of journal and conference papers

Again, we checked for any association with normalised citation counts (see Fig. 14). The Spearman correlation is -0.21 indicating a weak negative association; however, inspection of the scatterplot suggests much deviation and the fitted loess smoother is flat for much of the range of issue counts.

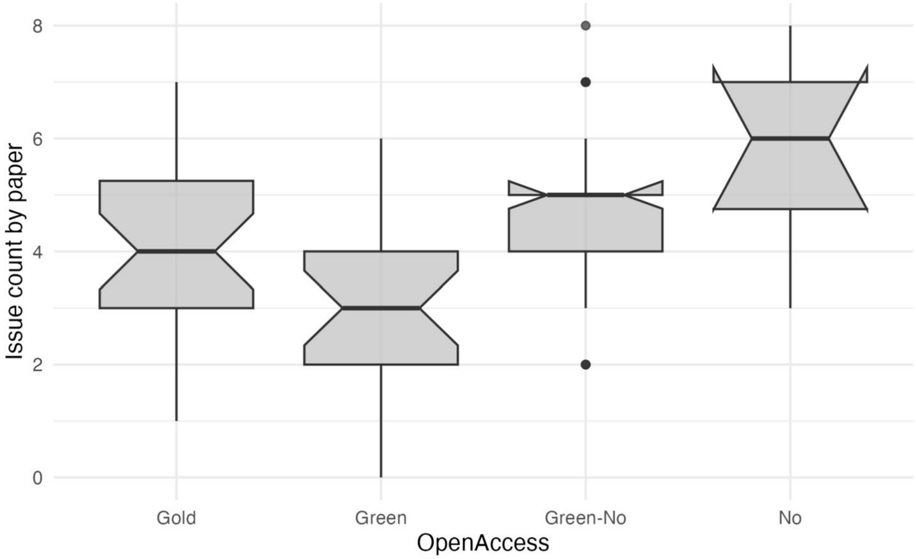


Fig. 13 Issues per paper by open access type

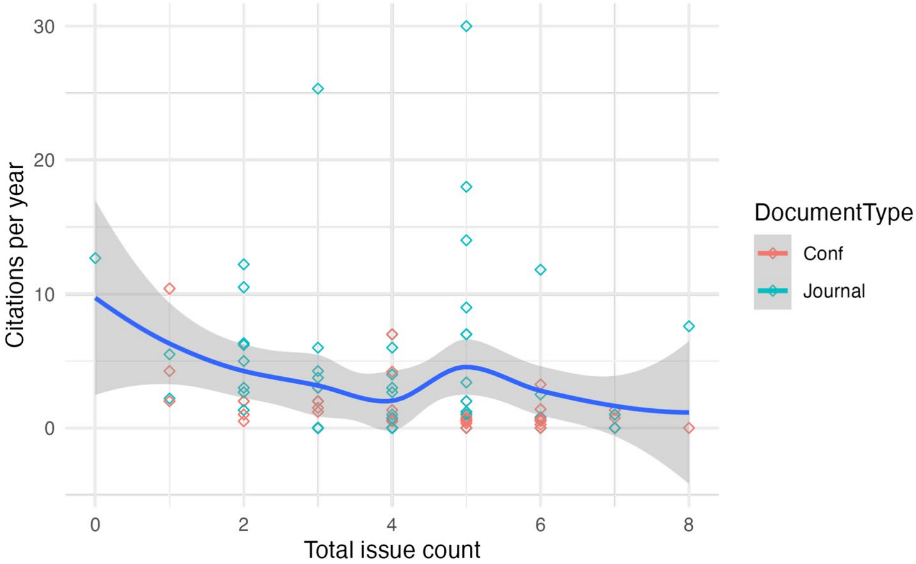


Fig. 14 Issues and citation counts

Finally, we consider whether there are any trends over time in Fig. 15 which reveals that, despite the year by year variability, there is no evidence to suggest that there is an overall effect over time. Unfortunately, we do not seem to be getting better, at least over the the five year window of this audit.

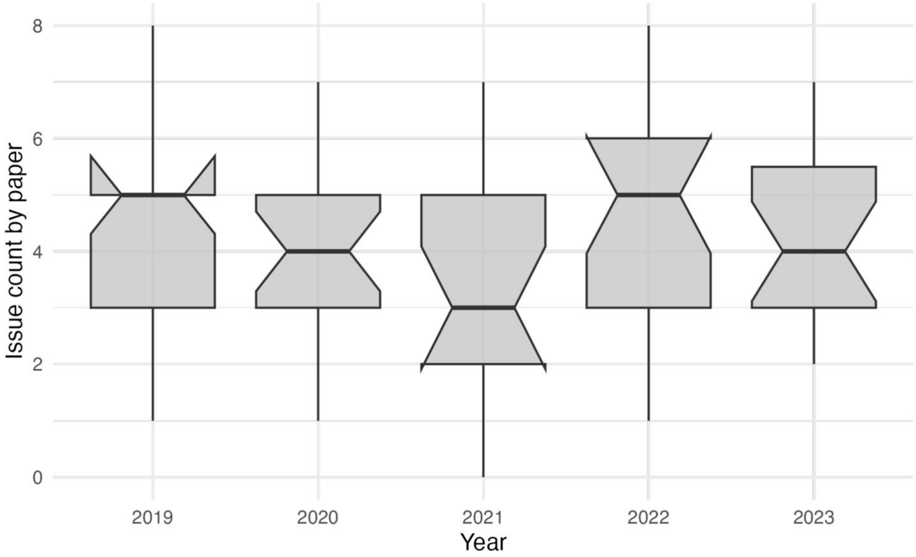


Fig. 15 Issues per paper by year

To summarise, the prevalence of issues with papers are widely distributed (almost all papers were impacted to some extent) and do not seem to be improving over time. Typically, a paper has four detected issues. There is weak evidence to suggest a relationship with Open Access publishing. Citation behaviour seems unrelated to either reproducibility or the count of issues.

5 Discussion

Our findings can be summarised as follows.

1. There is great diversity in the design of experiments including the number of learners, datasets and performance metrics. Scientific heterodoxy can be a powerful means to advance science but it also complicates reporting, understanding and making fair comparisons.
2. There is far less diversity in the choice of datasets (if not the absolute number), so a few curated collections such as the Promise datasets predominate. This means there is less independence than might be assumed for secondary analyses and meta-analysis. This can lead to difficulties with over-fitting to a repository curated more for reasons of convenience than representativeness. Ironically, as Hand points out (Hand 2006), the more successful a repository becomes, the greater the threat of distortion. It also means that the standard assumptions regarding random samples from some defined population are somewhat challenging to maintain.

3. Approximately 40% of the experiments explicitly undertake some kind of strategy to deal with imbalanced training data (where the prevalence of the positive case i.e., containing defects is low). Given the ubiquity of imbalanced datasets, this seems surprisingly low and will leave the analysis of results vulnerable to the potential biases of some of the performance metrics.
4. Performance metrics that are widely accepted as being problematic — certainly in the context of two-class problems and imbalanced datasets — continue to be deployed, with more than 60% of studies using F1 and 35% of studies using Accuracy. This can substantially impact results and the preference ordering of the learners under evaluation (Powers 2011; Yao and Shepperd 2021).
5. Almost half of the experiments do not use any chance-anchored metric, so it is not possible to determine whether an algorithm is actually performing better than guessing. This is problematic given that other studies, e.g., Yao and Shepperd (2020) have found a non-trivial number of learners doing worse than random.
6. Again, there is much diversity in the choice of validation strategies to simulate out-of-sample prediction which is problematic because the choice of validation strategy is well known to impact results (Kohavi 1995; Stapor et al. 2021). Clearly, this is an important area. The most common choice is 10-fold cross-validation, but this may be less relevant to how the SDP would actually be deployed than those based on cross-project and JIT prediction. More worrying is that some studies *appear* not to use any method so presumably are model-fitting; other papers are simply unclear on the matter.
7. Statistical errors affect almost a quarter of experiments where the acceptance or significance threshold is not adjusted for undertaking many, often in excess of 100, inferential tests. While we are of the view that estimating effect size is more valuable than significance testing, for this audit we adopt an agnostic stance. Self-evidently, acceptance thresholds, where used, must be adjusted in the face of multiple tests to prevent unacceptable false positive rates (Shaffer 1995). Approaches include (Benjamini and Hochberg 1995) and Nemenyi post hoc testing (Demšar 2006). To ignore this strikes us as quite problematic in that inferences and conclusions are directly impacted by the inflated possibility of false-positives.
8. Spread needs reporting as well centre. This may appear trivial, however, it is reasonable to communicate the variability of predictive results as well as typical values such as the mean. From a practitioner point of view, understanding the range of possible prediction outcomes can be important.
9. Informally, we found that for at least some of the papers, the descriptions of experiments were both hard to follow and omitted relevant information. More formally, our use of the González-Barahona and Robles instrument for assessing reproducibility revealed a good deal of scope for improvement. There may be occasions when an experiment is designed better than we judged, but without the relevant details we cannot know this. Replication and meta-analysis are also hindered.
10. Surprisingly few (50.5%), to us at least, of the studies were shared as open access. There is also some weak association between open access and reproducibility such that Gold and Green Open Access papers seemed to be more reproducible.

11. Journal papers tend to be more reproducible than conference papers although whether this due to typically being allowed more pages or a more constructive and iterative refereeing process is unclear.
12. Overall, we identified 427 issues with only one paper being entirely issue-free. The median number of issues is four. These split into 219 experimental issues and 208 reporting issues. The medians for both these are two.
13. Citation counts, even when taking into account the paper age do not seem to be a guide as to paper quality (either reproducibility or problem count). Given high citation counts are widely taken to connote high research quality or importance, this is both unexpected and a little concerning.

So what practical lessons can be derived? These can be organised into three groups: (i) relating to the design of the computational experiment, (ii) to the reporting of the experiment and (iii) for the wider research community.

Experimental design recommendations for the design and conduct of computational experiments.

R1 It is important to explicitly deploy some form of validation procedure to simulate prediction of out-of-sample instances and prevent over-fitting. We suggest that stochastic procedures such as k-fold cross-validation need $k \geq 10$ folds, or where large sample conditions do not hold then more repetitions, e.g., 2×5 folds (Kohavi 1995; Wong and Yeh 2019). However, the closer the procedure is to real-world usage the more useful the experimental results. Therefore, respecting time ordering, cross-project and JIT validation are important considerations.

R2 Use a chance-anchored performance metric, e.g., AUC or MCC or additionally report a simple metric such as Bookmaker's Odds (Powers 2011). Without this, it can be unclear whether a learning algorithm is doing better than guessing. With unbalanced datasets, values from metrics such as F1 can be quite unintuitive to interpret.

R3 Do not use problematic metrics: this seems self-evident and is widely discussed (e.g., Powers 2011; Chicco and Jurman 2020; Yao and Shepperd 2021) and easy to implement.

R4 Do not focus solely upon the prediction results central tendency but report dispersion. This may appear trivial, however, it is important to communicate the variability of predictive results as well as typical values such as the mean. A five-number summary or boxplots are examples of appropriate solutions. Many learning algorithms and OOS-validation strategies are stochastic, so understanding the variability of results is important especially to practitioners.

R5 If using multiple statistical inference tests, control for the false discovery rate. We recommend a modern approach such as proposed by Benjamini and Hochberg (1995) which is less conservative than traditional approaches such as the Bonferroni correction.

R6 In general, unless there are stronger methodological reasons to the contrary, restrict freedom to produce ever more complex and individual experimental designs, since this makes experiments harder to analyse, understand and comparisons between studies more awkward.

Reporting recommendations that relate to the reporting of both the experimental method and results.

R7 Fully report the experiment to include data (this is usually done), details of data pre-processing and analysis code (less commonplace). This should include all confusion matrices to allow alternative metrics to be computed, if desired.

R8 Where relevant, be clear what version of any dataset is used.

R9 Carefully describe all data cleaning or pre-processing to enable work to be reproduced.

R10 Use and report seeds for any stochastic analysis, e.g., the random allocation of instances to folds.

R11 Consider the durability of web links. We found that approximately 35% of links were broken in just the five year period covered by the audit.

R12 Behind a 'paywall' is related to reproducibility, since if many scientists are prevented from accessing a paper then they cannot build upon it. Therefore it is important to ensure an open-access version of a paper is available. For most publishers, this simply means posting the final postprint on a not-for-profit site such as arXiv or the author's own institution.

Community recommendations here we consider some recommendations for researchers and the wider community.

R13 We need careful reading of papers by reviewers, editors, readers and citers of published papers. Some of the issues we have detected, in particular the absence of explanation and use of tortured phrases, should be visible even upon a cursory reading. It would also be valuable to make greater use of public reviewing platforms such as PubPeer.

R14 Expect sharing of code and results, as well as data. The idea of data sharing is widely accepted, however sharing of code and raw results far less so. Improving upon this would help both better detection of issues and easier reproducibility.

R15 There may be value in the community providing more guidance and even prescription for the design of experiments. It is arguable if researchers have too many degrees of freedom this makes errors easier to commit and the research harder to understand.

R16 Journals seem to offer advantages for level of detail required for reproducibility. Here, we venture a little beyond our audit but there do seem benefits to the journal publication model as compared to conferences. Of course, conferences serve many purposes over and above publication vehicles. However, if software engineering were to place higher value upon journal papers we see it as a good thing.

6 Threats to Validity

6.1 Scope of Audit Checks

One potential limitation is the limited scope of checks we have performed during our audit. While our analysis is wide-ranging, it does not cover all possible types of errors or inconsistencies that could be present in the studies. For example, we did not scrutinise the internal consistency of the reported confusion matrices, an aspect highlighted in prior research as a source of error (Shepperd et al. 2019).

6.2 Measurement Error

Our approach to assessing reproducibility could itself be subject to measurement error. The metrics we used to evaluate reproducibility may not fully capture the nuances and complexities involved in replicating a study. In following the González-Barahona and Robles (2012) instrument, this does mean that each reproducibility item is equally weighted which is a simplifying assumption. However, we are of the view that gross differences between reproducibility levels will be highlighted even if small differences need to be treated with more circumspection.

6.3 Rater Bias and Subjectivity

The audit involved subjective judgments, such as the categorisation of different research designs. Despite efforts to standardize these evaluations, the possibility of rater bias cannot be entirely ruled out. To mitigate this, we used one paper for initial training and also independently double-checked and agreed all remaining studies.

6.4 Sample Size and Representativeness

Our study included 101 papers, which, while substantial, is not exhaustive and we estimate represents of the order of 6.5% of all the primary studies published 2019 to 2023. The papers were also selected based on specific inclusion criteria, possibly limiting the generalisability of our findings. The sample may not be fully representative of all work in the field, particularly studies published in venues not indexed by Scopus. However, if we restrict our population to studies that have been published in well-regarded venues and that have undergone meaningful peer review, then we have a reliable sample frame (the Scopus search results) and a demonstrably random sampling mechanism. Thus we can compute the standard error of sample statistics, e.g., the mean reproducibility score where the distribution is approximately symmetric. The sample Mean is 0.520, the adjusted (for the finite population correction) standard error of the mean 0.020 and the 95% Confidence Interval is [0.480, 0.560] which is reasonably precise i.e., $\pm 4\%$.

We also considered the sample representativeness in terms of the *perceived* quality of a venue. Our sample includes 4 papers from *IEEE TSE* and *Information & Software Technology* and 2 from *Empirical Software Engineering*. By its nature a sample will not cover every venue but we have good reasons to believe both ‘prestigious’ and ‘less prestigious’ venues are adequately covered.

6.5 Recommendations for Further Investigation

To mitigate the threats to validity identified in this study, we suggest researchers should aim for a more exhaustive range of quality checks, possibly incorporating automated tools to scrutinise aspects like confusion matrices (Bowes et al. 2014). Future studies might also consider employing multiple metrics for assessing reproducibility to capture its various dimensions. Finally, to enhance representativeness of the results, future audits could aim for a larger sample sizes and consider including papers from other sources, not just those indexed by Scopus so that it might be possible to compare the grey literature with the more regular refereed literature covered by Scopus.

7 Conclusion

To recap, in response to concerns about the quality and reproducibility of software defect prediction studies, we undertook an audit of experiments published from 2019-2023. We sampled 101 experiments and reviewed each one for the choice of training data, cleaning strategies relating to imbalance, the validation strategy, performance metrics, statistical inference and reproducibility. Unfortunately, this identified quite widespread issues consistent with other previous studies such as Liem and Panichella (2020); Liu et al. (2021); Shepperd et al. (2019).

Our findings can be summarised as follows.

- There is great diversity in the design of experiments but less so in the choice of datasets, where the Promise datasets predominate.
- Overall, we identified 427 issues (not all fatal, but minimally, unnecessary and unhelpful) with only one paper being entirely issue-free. The median number of issues was four per paper and cover both experimental design and reporting. Some of the more prestigious outlets, for example Transaction type papers are not immune. It should also be stressed our search for research quality issues is by no means exhaustive since we focused on issues that are easy to detect and cannot be considered a matter of opinion.
- Citation counts, even when normalised, do not seem to be a guide as to paper quality (for either reproducibility or issues).
- Journal papers tend to be more reproducible and contain fewer issues than conference papers although whether this is due to typically being allowed more pages or a more constructive and iterative refereeing process is unclear.
- We found 2 out of 101 papers to contain clear examples of tortured phrases, e.g., “software insects”, which could be indicative of paper mill activity.

Second, we list a total of 16 lessons that emerge from our audit to those conducting computational experiments to evaluate software defect prediction. These are relevant to researchers, reviewers and editors and to the wider software engineering community. One source of encouragement is that none of them are controversial or difficult to implement.

Finally, we wish to be clear that scientists are human. Furthermore, computational experiments, by their very nature tend to be complex. We all — the authors of this paper included — make mistakes, thus we do not wish this audit to be seen as some kind of schadenfreude-piece. Nevertheless, if we consider the astonishing amount of effort deployed in researching software defect prediction (considerably more than 1,500 primary studies in the past five years alone) then the lack of reliability, insight or impact should make us wish to do better.

Appendix: Raw Data Description

Variable Name	Type	Comment
PaperID	Character	Unique paper identifier that we can map back to the actual paper.
Checker	Factor	Person extracting data.
Checker2	Character	Person extracting data.
PredictionType	Factor	Either classification or regression.
Year	Factor	Year of publication (taken from Scopus).
SourceTitle	Character	Venue published e.g., journal or conference name.
Core_A	Factor	Publication venue is rated A*/A by CORE (Y or N).
PageCt	Integer	Length of article (taken from Scopus).
CitedBy	Integer	Citation count (taken from Scopus).
SelfCited	Integer	Self citations by any co-author.
OpenAccess	Factor	Is the paper publicly available (Y or N)?
DocumentType	Factor	Conf(erence) or Journal.
Tortured	Factor	Does the paper contain clear examples of "tortured phrases" (Y or N)?
Summary	Character	Free format narrative description by Checkers
Datasets	Character	Free format description of the datasets or repositories used.
DatasetsCt	Integer	How many distinct datasets used? Different releases are separate data sets eg Camel 1.0, 1.2 = 2 data sets.
LearnersCt	Integer	The number of learners/algorithms/treatments being evaluated. This might include pre-processing such as transformation or feature subset selection. Usually obvious from the results table(s).
MetricsText	Character	What classification performance metrics are collected? Items separated by comma.
F1	Integer	F1 is often referred to as the F-measure or even F-score (1 if used).
MCC	Integer	Matthews correlation coefficient (1 if used).
AUC	Integer	Area under the curve (AUC). Sometimes called ROC (1 if used).
Accuracy	Integer	(1 if used).
Precision	Integer	Is TP/(TP+FP) (1 if used).
Recall	Integer	Also known as sensitivity or TPR ie TP/(TP+FN) (1 if used).
Specificity	Integer	Also known as TNR ie TN/TN+FP (1 if used).
Other	Integer	The count of other metrics so if non-zero the total in the next column is correct.
MetricsCt	Integer	Total count of different performance metrics used in the analysis.
SummaryText	Character	Free format text on how the metrics summarised e.g., from multiple folds or multiple data sets? (Mean or boxplots are common).
Location	Factor	Is location reported e.g., mean or median (Y or N)?
Spread	Factor	Is spread or dispersion reported, e.g., IQR or variance (Y or N)?
RandomBenchmark	Factor	Do the authors compare prediction performance with chance e.g., Bookmakers odds or a chance-anchored metric such as AUC? (Y or N)?
StatInference	Factor	Do the authors make use of statistical inference, most likely via p-values and some significance threshold? (Y or N)?
TestsCt	Integer	How many statistical tests are reported (typically in tables of dataset v learning algorithm and then for each metric). Report NA if no tests (Y or N or NA)?
ImbalancedMethod	Factor	E.g., under/over sampling, SMOTE etc., but NA where irrelevant e.g., regression. (Y or N or NA)?

AdjustSig	Factor	Do the authors adjust significance threshold for multiple tests, otherwise NA if 0 or 1 test (Y or N or NA)?
OutOfSample	Factor	Is CV or other out of sample validation method e.g., bootstrap used or not and NA if CV isn't applicable to the study. (Y or N or NA)?
TypeCV	Character	Free format description of CV method.
m	Integer	Number of replications of the validation process or NA.
n	Integer	Number of folds for CV or NA.
		The next 27 columns address the **reproducibility** of a study as per González-Barahona & Robles 2012. 0 = not present/not ok; 1 = present / ok.
RawIdentification	Integer	Is it clear where the (original) raw data be obtained from?
RawDescription	Integer	Is the published information about the raw data, including its internal organization and structure and its semantics sufficiently detailed?
RawAvailability	Integer	Is it easy for a researcher to obtain the raw data, or have access to it?
RawPersistence	Integer	Is it likely the raw data will be available in the future?
RawFlexibility	Integer	Is the raw data flexible, how easily can it be adapted to new environments?
ExtractionIdentification	Integer	Is it clear where the (original) extraction method be obtained from?
ExtractionDescription	Integer	Is the published information about the extraction method sufficiently detailed?
ExtractionAvailability	Integer	Is it easy for a researcher to obtain the extraction method, or have access to it?
ExtractionPersistence	Integer	Is it likely the extraction method will be available in the future?
ExtractionFlexibility	Integer	Is the extraction method flexible, how easily can it be adapted to new environments?
ProcessedIdentification	Integer	Is it clear where the processed dataset be obtained from?
ProcessedDescription	Integer	Is the published information about the processed dataset sufficiently detailed?
ProcessedAvailability	Integer	Is it easy for a researcher to obtain the processed dataset, or have access to it?
ProcessedPersistence	Integer	Is it likely the processed dataset will be available in the future?
ProcessedFlexibility	Integer	Is the processed dataset flexible, how easily can it be adapted to new environments?
AnalysisIdentification	Integer	Is it clear where the analysis methodology/tools can be obtained?
AnalysisDescription	Integer	Is the published information about the analysis methodology/tools sufficiently detailed?
AnalysisAvailability	Integer	Is it easy for a researcher to obtain the analysis methodology/tools, or have access to it?
AnalysisPersistence	Integer	Is it likely the analysis methodology/tools will be available in the future?
AnalysisFlexibility	Integer	Are the analysis methodology/tools flexible, how easily can they be adapted to new environments?
ResultsIdentification	Integer	Is it clear where the results dataset be obtained from?
ResultsDescription	Integer	Is the published information about the results dataset sufficiently detailed?
ResultsAvailability	Integer	Is it easy for a researcher to obtain the results dataset, or have access to it?
ResultsPersistence	Integer	Is it likely the results dataset will be available in the future?

ResultsFlexibility	Integer	Is the results dataset flexible, how easily can it be adapted to new environments?
Identification	Integer	Are all the study parameters clearly identified?
Description	Integer	Are all the study parameters adequately described including their setting(s)?
TotalRepro	Numeric	The overall reproducibility score out of 27 and then normalised 0-1.
DataLink	Factor	Is there a link to the data used (Y or N)?
CodeLink	Factor	Is there a link to the code to generate the analysis (Y or N)?
BrokenLink	Factor	NA if no links, Y if one or more links are broken otherwise N.
CitedByYear	Numeric	Total citations divided by years available.
NonSelfCitedByYear	Numeric	(Total citations minus self-citations) divided by years available.
CVprob	Numeric	Are there issues relating to the absence of cross-validation where expected (or any other appropriate out-of-sample testing)?
Randprob	Numeric	Is there an issue due to the absence of random prediction benchmarks?
BadMprob	Numeric	Problematic use of performance metrics i.e., F1 or Accuracy?
AdjustProb	Numeric	Issue arising from failure to adjust significance threshold (alpha) when undertaking multiple significance tests?
NoVarProb	Numeric	Issue arising from the failure to report the spread or variability of results.
LowReproProb	Numeric	Issue arising from low reproducibility of the study, defined as the lowest tertile.
NotOpenPubProb	Numeric	Issue arising from unavailability of the paper due to “pay walls”.
NotOpenDataProb	Numeric	Issue arising from the unavailability of the data.
NotOpenCodeProb	Numeric	Issue arising from not sharing analysis methods.
TotProbs	Numeric	Total count of issues (ranges from 0 to 9).
ExptProbs	Numeric	Total count of experimental design issues (ranges from 0 to 5).
RepProbs	Numeric	Total count of reporting issues (ranges from 0 to 4).

Author Contributions All authors contributed to the study conception, design. Material preparation was performed by Martin Shepperd. Data collection and validation were performed by all authors. The data analysis was performed by Martin Shepperd and Leila Yousefi. The first draft of the manuscript was written by Giuseppe Destefanis, Martin Shepperd and Steve Counsell. All authors commented on previous versions of the manuscript and all authors read and approved the final manuscript. The order of authors was determined randomly.

Funding No specific funding was received for conducting this study.

Data and Code Availability The dataset used in this study and the code to analyse it are available from here https://zenodo.org/records/17696089 and DOI https://doi.org/10.5281/zenodo.13927601.

Declarations

Conflicts of Interest The authors declared that they have no conflict of interest.

References

ACM (2020) Artifact review and badging version 1.1

Baas J, Schotten M, Plume A, Côté G, Karimi R (2020) Scopus as a curated, high-quality bibliometric data source for academic research in quantitative science studies. Quant Sci Stud 1(1):377–386

Baldi P, Brunak S, Chauvin Y, Andersen C, Nielsen H (2000) Assessing the accuracy of prediction algorithms for classification: an overview. Bioinformatics 16(5):412–424

Baltes S, Ralph P (2022) Sampling in software engineering research: a critical review and guidelines. Empir Softw Eng 27(4):94

Benavoli A, Corani G, Demšar J, Zaffalon M (2017) Time for a change: a tutorial for comparing multiple classifiers through Bayesian analysis. J Mach Learn Res 18(1):2653–2688

Bender R, Lange S (2001) Adjusting for multiple testing—when and how? J Clin Epidemiol 54(4):343–349

Benjamini Y, Hochberg Y (1995) Controlling the false discovery rate: a practical and powerful approach to multiple testing. J Roy Stat Soc: Ser B (Methodol) 57(1):289–300

Bowes D, Hall T, Gray D (2014) DConfusion: a technique to allow cross study performance evaluation of fault prediction studies. Autom Softw Eng 21(2):287–313

Brewin C (2023) Inaccuracy in the scientific record and open postpublication critique. Perspectives on Psychological Science, pp 17456916221141357

Cabanac G, Labbé C, Magazinov A (2021) Tortured phrases: a dubious writing style emerging in science. Evidence of critical issues affecting established journals. arXiv:2107.06751

Cabanac G, Labbé C, Magazinov A (2022) Flagging suspect publications and crowdsourcing post-publication reassessments: the ‘problematic paper screener’. HAL UT3 and Toulouse INP portal: hal-03603538

Candal-Pedreira C, Ross J, Ruano-Ravina A, Egilman D, Fernández E, Pérez-Ríos, M (2022) Retracted papers originating from paper mills: cross sectional study. BMJ, 379

Chicco D, Jurman G (2020) The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. BMC Genomics 21(1):1–13

Chicco D, Warrens M, Jurman G (2021) The Matthews correlation coefficient (MCC) is more informative than Cohen’s Kappa and Brier score in binary classification assessment. IEEE Access 9:78368–78381

Colquhoun D (2014) An investigation of the false discovery rate and the misinterpretation of p-values. Royal Soc Open Sci 1(3):140216

Demšar J (2006) Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 7:1–30

Destefanis G, Yousefi L, Shepperd M, Tucker A, Swift S, Counsell S, Arzoky M (2024) An audit of machine learning experiments on software defect prediction - dataset

Diong J, Butler A, Gandevia S, Héroux M (2018) Poor statistical reporting, inadequate data presentation and spin persist despite editorial advice. PLoS ONE 13(8):e0202121

Earp B, Trafimow D (2015) Replication, falsification, and the crisis of confidence in social psychology. Front Psychol 6:621

Fazekas A, Kovács G (2024) Testing the consistency of performance scores reported for binary classification problems. Appl Soft Comput 164:111993

Fernández A, Garcia S, Herrera F, Chawla N (2018) SMOTE for learning from imbalanced data: progress and challenges, marking the 15-year anniversary. J Artif Intell Res 61:863–905

Gelman A, Park D (2009) Splitting a predictor at the upper quarter or third and the lower quarter or third. Am Stat 63(1):1–8

González-Barahona J, Robles G (2012) On the reproducibility of empirical software engineering studies based on data retrieved from development repositories. Empir Softw Eng 17:75–89

González-Barahona J, Robles G (2023) Revisiting the reproducibility of empirical software engineering studies based on data retrieved from development repositories. Information and Software Technology, online:107318

Greenland S, Senn S, Rothman K, Carlin J, Poole C, Goodman S, Altman D (2016) Statistical tests, p values, confidence intervals, and power: a guide to misinterpretations. Eur J Epidemiol 31:337–350

Hand D (2006) Classifier technology and the illusion of progress. Stat Sci 21(1):1–14

Hand D, Christen P (2018) A note on using the F-measure for evaluating record linkage algorithms. Stat Comput 28(3):539–547

Harnad S, Brody T, Vallia F, Carr L, Hitchcock S, Gingras Y, Oppenheim C, Stamerjohanns H, Hilf E (2004) The access/impact problem and the green and gold roads to open access. Ser Rev 30(4):310–314

Héroux M, Diong J, Bye E, Fisher G, Robertson L, Butler A, Gandevia S (2023) Poor statistical reporting inadequate data presentation and spin persist despite journal awareness and updated information for authors. F1000Research 12

Heumüller R, Nielebock S, Krüger J, Ortmeier F (2020) Publish or perish, but do not forget your software artifacts. Empir Softw Eng 25(6):4585–4616

Huang C, Neylon C, Montgomery L, Hosking R, Diprose J, Handcock R, Wilson K (2024) Open access research outputs receive more diverse citations. Scientometrics 129:825–845

Ioannidis J (2005) Why most published research findings are false. PLoS Med 2(8):e124

Kamei F, Wiese I, Lima C, Polato I, Nepomuceno V, Ferreira W, Ribeiro M, Pena C, Cartaxo B, Pinto G, Soares S (2021) Grey literature in software engineering: a critical review. Inf Softw Technol 138:106609

Kapoor S, Narayanan A (2023) Leakage and the reproducibility crisis in machine-learning-based science. Patterns 4(9)

Kitchenham B, Pfleeger S, Pickard L, Jones P, Hoaglin D, El Emam K, Rosenberg J (2002) Preliminary guidelines for empirical research in software engineering. IEEE Trans Software Eng 28(8):721–734

Kohavi R (1995) A study of cross-validation and bootstrap for accuracy estimation and model selection. In: 14th International Joint Conference on Artificial Intelligence (IJCAI), vol 14. pp 1137–1145

Lavazza L, Morasca S (2022) Comparing ϕ and the F-measure as performance metrics for software-related classifications. Empir Softw Eng 27(7):185

Li L, Lessmann S, Baesens B (2019) Evaluating software defect prediction performance: an updated benchmarking study. arXiv:1901.01726

Liem C, Panichella, A (2020) Run, forest, run? On randomization and reproducibility in predictive software engineering. arXiv:2012.08387

Liu C, Gao C, Xia X, Lo D, Grundy J, Yang X (2021) On the reproducibility and replicability of deep learning in software engineering. ACM Trans Softw Eng Methodol 31(1)

Madeyski L, Kitchenham B (2017) Would wider adoption of reproducible research be beneficial for empirical software engineering research? J Intell Fuzzy Syst 32(2):1509–1521

Meagher K (2021) Introduction: The politics of open access—decolonizing research or corporate capture? Dev Chang 52(2):340–358

Menzies T, Shepperd M (2012) Editorial: special issue on repeatable results in software engineering prediction. Empir Softw Eng 17(1–2):1–17

Midway S, Robertson M, Flinn S, Kaller M (2020) Comparing multiple comparisons: practical guidance for choosing the best multiple comparisons test. PeerJ 8:e10387

Mohammadi M, Di Nucci D, Tamburri D (2023) Bayesian meta-analysis of software defect prediction with machine learning. IEEE Trans Ind Cyber-Phys Syst 1:147–156

Munafò M, Nosek B, Bishop D, Button K, Chambers C, Percie du Sert N, Simonsohn U, Wagenmakers E, Ware JJ, Ioannidis J (2017) A manifesto for reproducible science. Nat Hum Behav 1(1):1–9

Nuijten M, Hartgerink C, van Assen M, Epskamp S, Wicherts J (2016) The prevalence of statistical reporting errors in psychology (1985–2013). Behav Res Methods 48(4):1205–1226

Open Science Collaboration (2015) Estimating the reproducibility of psychological science. Science 349(6251):aac4716

Powers D (2011) Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. J Mach Learn Technol 2(1):37–63

Provost F, Fawcett T (2001) Robust classification for imprecise environments. Mach Learn 42:203–231

Ralph P, Baltes S (2022) Paving the way for mature secondary research: the seven types of literature review. In: 30th ACM joint european software engineering conference and symposium on the foundations of software engineering. pp 1632–1636

Rana R, Staron M, Hansson J, Nilsson M, Meding W (2014) A framework for adoption of machine learning in industry for software defect prediction. In: 9th IEEE International Conference on Software Engineering and Applications (ICSOFT-EA). pp 383–392

Richardson R, Hong S, Byrne J, Stoeger T, Amaral L (2025) The entities enabling scientific fraud at scale are large, resilient, and growing rapidly. Proc Natl Acad Sci 122(32):e2420092122

Shaffer J (1995) Multiple hypothesis testing. Annu Rev Psychol 46(1):561–584

Shepperd M, Bowes D, Hall T (2014) Researcher bias: The use of machine learning in software defect prediction. IEEE Trans Software Eng 40(6):603–616

Shepperd M, Guo Y, Li N, Arzoky M, Capiluppi A, Counsell S, Destefanis G, Swift S, Tucker A, Yousefi L (2019). The prevalence of errors in machine learning experiments. In: Intelligent data engineering and automated learning-IDEAL 2019: 20th international conference, Manchester, UK, November 14–16, 2019, Proceedings, Part I 20. Springer, pp 102–109

Stapor K, Ksieniewicz P, García S, Woźniak M (2021) How to design the fair experimental classifier evaluation. Appl Soft Comput 104:107219

Stradowski S, Madeyski, L (2022) Machine learning in software defect prediction: a business-driven systematic mapping study. Information and Software Technology, pp 107128

van Rijsbergen C (1979) Information retrieval. Butterworths, 2nd edn

Varoquaux G, Cheplygina V (2022) Machine learning for medical imaging: methodological failures and recommendations for the future. NPJ Digit Med 5(1):48

Wong T, Yeh P (2019) Reliable accuracy estimates from k-fold cross validation. IEEE Trans Knowl Data Eng 32(8):1586–1594

Wong T, Yeh P (2020) Reliable accuracy estimates from k-fold cross validation. IEEE Trans Knowl Data Eng 32(8):1586–1594. First published online 2019–04-25

Yao J, Shepperd M (2020) Assessing software defection prediction performance: Why using the Matthews correlation coefficient matters. In: Proceedings of the ACM Evaluation and Assessment in Software Engineering (EASE) Conference. pp 120–129

Yao J, Shepperd M (2021) The impact of using biased performance metrics on software defect prediction research. Inf Softw Technol 139:106664

Zhu Q (2020) On the performance of Matthews correlation coefficient (MCC) for imbalanced dataset. Pattern Recogn Lett 136:71–80

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Giuseppe Destefanis[1] · Leila Yousefi[2] · Martin Shepperd[2] · Allan Tucker[2] · Stephen Swift[2] · Steve Counsell[2] · Mahir Arzoky[2]

✉ Martin Shepperd
martin.shepperd@brunel.ac.uk

[1] Department of Computer Science, University College London, London, UK

[2] Department of Computer Science, Brunel University London, West London UB8 3PH, UK