Reflecting on Empirical and Sustainability Aspects of Software Engineering Research in the Era of Large Language Models

David Williams
University College London
United Kingdom
david.williams.22@ucl.ac.uk

Max Hort
Simula Research Laboratory
Norway
maxh@simula.no

Maria Kechagia
National and Kapodistrian University
of Athens
Greece
makechag@ba.uoa.gr

Aldeida Aleti
Monash University
Australia
aldeida.aleti@monash.edu

Justyna Petke
University College London
United Kingdom
j.petke@ucl.ac.uk

Federica Sarro
University College London
United Kingdom
f.sarro@ucl.ac.uk

Abstract

Software Engineering (SE) research involving the use of Large Language Models (LLMs) has introduced several new challenges related to rigour in benchmarking, contamination, replicability, and sustainability. In this paper, we invite the research community to reflect on how these challenges are addressed in SE. Our results provide a structured overview of current LLM-based SE research at ICSE, highlighting both encouraging practices and persistent shortcomings. We conclude with recommendations to strengthen benchmarking rigour, improve replicability, and address the financial and environmental costs of LLM-based SE.

1 Introduction

Large Language Models (LLMs) are now being integrated into software development practices, influencing safety-critical applications, projects, and codebases. As these systems become more prevalent, Software Engineering (SE) researchers are uniquely positioned to address challenges and opportunities across the entire software lifecycle. In fact, despite the potential to augment human capabilities and streamline software development workflows, the integration of LLMs into software systems raises critical concerns. For instance, benchmarking should be performed systematically to ensure fair evaluation, particularly in light of potential data leakage from training corpora that can skew performance metrics [20]. Reproducibility and replicability should become mandatory to address the stochastic nature of LLMs and the lack of standardised evaluation protocols [14]. Further efforts should be made to reduce the computational costs associated with training and inference – both in terms of energy and infrastructure – which pose significant barriers to accessibility and sustainability [9]. As LLM-based SE becomes increasingly prevalent, addressing these challenges is crucial to harnessing the full potential of LLMs while maintaining scientific rigour, ethical standards, and equitable access.

To understand how our community is tackling the aforementioned challenges, we conduct an initial analysis of the empirical research literature published in the research track of the top SE conference, the International Conference on Software Engineering (ICSE), between 2023 and 2025. We identified 177 relevant papers and manually extracted several key pieces of information, including the SE tasks addressed, the types of models used, the experimental protocols, the benchmarks, and other details relevant to empirical studies. As expected, the proportion of published papers at ICSE that include an empirical evaluation using LLMs has grown dramatically, from 15.2% of accepted papers in 2023 to 36.6% in 2025. We invite the community to reflect on crucial empirical and ethical aspects of current SE research involving LLMs.

2 Research Questions

We analyse relevant ICSE papers to establish:

RQ1. Models & Benchmarking *Which LLMs are used in SE research and how are they benchmarked?* We want to understand whether current studies provide reliable evidence of improvement over non-LLM techniques, and to highlight gaps where more rigorous evaluation protocols are needed. We check (i) whether open or closed models are used, (ii) which model families are chosen, (iii) whether comparisons include non-LLM baselines, and (iv) which programming languages are targeted.

RQ2. Contamination. *How well do authors tackle the problem of data leakage/contamination?* Data leakage and contamination directly threaten the validity of empirical findings. If evaluation datasets overlap with model pre-training data, performance may reflect memorisation rather than genuine reasoning capabilities. We check i) whether authors mention data contamination as a threat to the validity of the results and ii) the strategy proposed to mitigate contamination.

RQ3. Replicability. *How replicable are LLM-based studies?* Replicability underpins scientific credibility. LLM-based SE research is challenged by the stochastic nature of model outputs, opaque APIs, and frequent model updates. We check (i) whether LLM configuration details are reported and (ii) whether artefacts are shared and recognised with a badge.

RQ4. Sustainability. *What are the costs of LLM-based SE research?* Training and deploying LLMs is resource-intensive, with high financial and environmental costs. By examining how sustainability is addressed, we aim to raise awareness of these hidden costs and encourage the community to adopt more transparent, responsible practices that balance innovation, accessibility, and environmental responsibility. We check (i) which experimental costs are reported, (ii) the costs of LLM-based SE research, and (iii) how they are sustained.

3 Methodology

To answer our RQs, we used a mixed-methods approach, integrating quantitative and qualitative data gathered through a literature review and user survey. We incorporate temporal analyses, where possible, to analyse how trends have shifted.

Article Search. The study is conducted on the 692 articles published at the technical research track of ICSE 2023, 2024, and 2025. As the premier venue in SE, ICSE sets the benchmark for methodological rigour and research quality in the field. Consequently, the insights we draw from this body of work can be regarded as a reliable reflection of the community's current priorities and practices. Moreover, these years coincide with the widespread advent of LLMs, following the release of ChatGPT in late 2022.

The initial list of 692 papers was filtered based on the following list of keywords: "*LLM*", "*language model*", "*conversational AI*", "*chatbot*", "*genAI*", "*generative*", "*AI-assisted*", "*foundation model*", "*BERT*", "*encoder*", "*decoder*", "*autoencoder*", "*transformer*", and "*agent*". A title and abstract search using these terms returned 304 articles from the initial 692. Among these 304, we manually identified relevant publications matching the following inclusion criteria: *articles that present an empirical experiment involving at least one LLM, either as the main approach or as a baseline.* For example, we exclude papers focusing solely on qualitative insights derived from LLM user studies. When referring to LLMs, we use the classification proposed by Fan et al. [6], namely encoder-only (e.g., BERT, DeBERTa, RoBERTa), encoder-decoder (e.g., T5, BART), and decoder-only (e.g., GPT, LLaMA, Claude). We followed a three-step process, widely adopted in previous surveys (e.g. [10, 15, 16]): i) We first excluded all publications whose titles clearly did not meet our inclusion criteria; ii) next, we judged inclusion based on the abstract; iii) finally, we reviewed the content of all remaining articles from the previous two steps and excluded those that did not meet our inclusion criteria. This process resulted in a total of 174 articles meeting the inclusion criteria and thus selected for data extraction. To mitigate the risks of missing relevant articles, we (a) checked whether we missed any keywords in our search by computing word clouds from the abstracts of these 174 articles (see [24] for full results), finding no additional relevant keywords were missing; (b) randomly sampled 20% of the 338 papers not selected by our keyword search for each year. This comprised a total of 80 papers (32 for 2023, 28 for 2024, and 20 for 2025), which we manually inspected using the process described above. We found that only three of these articles were false negatives and subsequently included them in the survey. The resulting number of relevant papers we analyse is thus 177. We provide the full list of papers in our online repository [24].

Data Extraction. Next, we performed data extraction guided by the RQs. We minimise the threat arising from data extraction and interpretation bias by standardising the processes for extracting and consolidating data from the papers. We minimised the threat of reviewer bias by involving at least two people at each step of this study. For each RQ, we identified the specific information that had to be collected from the selected papers. To operationalise this, each author was assigned a subset of papers and independently extracted the required information. To ensure reliability, we began with a pilot phase in which each author analysed four papers, then met to discuss the extraction process, clarify ambiguities, and refine

Table 1: Num. papers employing open & commercial models.

Year	Only Open	Only Comm.	Open & Comm.
2023	22 / 32 (68.8%)	4 / 32 (12.5%)	6 / 32 (18.8%)
2024	25 / 55 (45.5%)	13 / 55 (23.6%)	17 / 55 (30.9%)
2025	24 / 90 (26.7%)	24 / 90 (26.7%)	42 / 90 (46.7%)
Total	71 / 177 (40.1%)	41 / 177 (23.2%)	65 / 177 (36.7%)

the extraction schema. Next, the process was repeated for a new set of four papers each to ensure our ambiguities had been resolved before analysing the full set.

Once the extraction from all 177 papers was completed, the information was consolidated into a unified taxonomy. This consolidation was performed iteratively: one of the authors compared entries across papers, standardised terminology, and grouped similar items under common categories. Any uncertainties or disagreements were resolved collaboratively among all authors to ensure consensus. Finally, the resulting taxonomy provided a structured view of how the ICSE community addresses benchmarking, contamination, replicability, and sustainability, which served as the basis for our subsequent analysis.

User Study. To better understand the costs incurred by the SE research community for using LLMs in empirical studies, we developed a brief questionnaire and distributed it to the authors of relevant papers included in our survey. This consisted of 10 questions aimed at gaining insights into the costs incurred by researchers using LLMs over the past year, how these costs were covered, and their expectations regarding both aspects for the next 12 months. Due to space constraints, the full questionnaire is available online [24]. This study received approval from the ethics committee at the Computer Science Department at UCL (Project ID: 1819).

4 Results & Reflections

We analysed 177 papers reporting empirical SE studies using LLMs: 32 from ICSE 2023, 55 from ICSE 2024, and 90 from ICSE 2025. Although it is expected that the number of such studies would have grown over the years, we observe that these studies accounted for 2.4 times as many articles published in the ICSE research track in 2025 compared to 2023 (36.6% vs. 15.2%).

RQ1. Models and Benchmarking

Open vs. commercial models. Our analysis reveals that 136 out of the 177 relevant papers (76.8%) use open models (e.g., DeepSeek, CodeT5), whereas 106 out of 177 papers (59.9%) consider closed models (e.g., OpenAI's GPT models, Google's Gemini, Anthropic's Claude). Table 1 describes the prevalence of open and commercial LLMs used in empirical LLM-based studies.

☛ Overall, there is an increasing use of closed models over time, particularly in the number of studies that feature only closed or commercial models. We raise this as a concern, as commercial models are typically less accessible to researchers due to financial constraints, and replicability is hindered as said models are often deprecated when newer versions are released.

Model Family & Benchmarking. We examined the number of papers that consider a particular model family. The top 5 families of models used across all three years are: **1.** GPT-4 (47 papers), **2.** GPT-3.5 (44 papers), **3.** CodeBERT (34 papers), **4.** CodeLlama

Table 2: Prevalence of the top 5 most popular targeted programming languages for LLM-based empirical studies.

Language	2023 - 32	2024 - 55	2025 - 90	Total - 177
Java	14 (43.8%)	22 (40.0%)	31 (34.4%)	67 (37.9%)
Python	9 (28.1%)	18 (32.7%)	29 (32.2%)	56 (31.6%)
C	7 (21.9%)	11 (20.0%)	12 (13.3%)	30 (16.9%)
C++	2 (6.3%)	9 (16.3%)	12 (13.3%)	23 (13.0%)
JavaScript	4 (12.5%)	3 (5.5%)	6 (6.7%)	13 (7.3%)

(26 papers), and **5.** CodeT5 (22 papers). A full table of the most popular models per year (and additional analysis on how many models were used in various papers) can be found in our online appendix [24]. Most notably, we observe a significant shift toward OpenAI's commercial models from 2023 to 2025: while CodeBERT was the most popular model in 2023 (featured in 11 out of 32 papers, 34%), by 2025 it appears in only 12 out of 90 papers (13%), ranking fifth, whereas GPT-3.5 and GPT-4 feature in 39 (43%) and 45 (50%) papers, respectively.

For each study utilising an LLM to solve a given task, we checked whether the LLM is benchmarked with at least one non-LLM approach. While the number of studies benchmarking their proposal against non-LLM techniques has increased over time, the proportion of those doing so has decreased. In particular, 27 out of 32 papers (84.4%) did so in 2023, 36 out of 55 (65.5%) in 2024, and 51 out of 90 (56.7%) in 2025.

☛ We observe a shift in the technologies adopted, from BERT-based models in 2023 to GPT-based models in 2025. While the adoption of more complex and costly technologies is increasing, there does not seem to be a corresponding increase in benchmarking them against non-LLM counterparts. This should be prioritised to ensure the adoption of these technologies remains cost-effective.

Targeted Programming Languages. We examined the prevalence of programming languages (PLs) used in LLM-based empirical studies and found that Java was the most popular language overall, followed by Python, C, C++, and JavaScript. Interestingly, Table 2 shows that Java is becoming less common in studies over the years (from 43.8% of studies in 2023 to 37.9% in 2025), while Python has grown more popular (28.1% to 31.6%). In terms of papers evaluating multiple PLs, we see more concerning trends. Overall, only 27.1% of papers evaluate with more than one PL. Notably, only 22.2% of 2025 papers evaluate with more than one PL, compared to the more positive results of 2023 (31.2%) and 2024 (32.7%). This is problematic when considering replicability, as LLMs have been shown to perform better on some PLs than others [21], meaning it is not a given that reported performance on a single PL would generalise to other PLs or contexts.

☛ To mitigate the risk of reporting potentially inflated performance estimates, we urge researchers to, where possible, evaluate novel LLM SE approaches on multiple programming languages.

RQ2: Contamination

Data contamination or leakage refers to the scenario where a model is evaluated using data present in its training set, which can lead to misleading results and inflated performance estimates [7]. As LLMs are trained on vast, often closed-source corpora, it can

be challenging to determine whether an evaluation dataset is contaminated. Regardless, papers should mention such cases as threats to validity and discuss any strategies considered to address and mitigate these issues.

We found that only three articles (two published in 2024 [2, 27] and one in 2025 [17]) specifically tackle the problem of contamination/memorisation. Moreover, only 58 out of 177 papers (32.8%) discuss data contamination as a threat to validity. We observed an upward trend in the number of papers reporting data contamination over the examined period from 2023 to 2025. Specifically, for 2023, contamination is reported in 6 out of 32 papers (18.8%), 14 out of 55 (25.5%) for 2024, and 38 out of 90 (42.2%) for 2025. However, despite the rising trend, the percentage of papers that mention such threats remains less than 50%.

Some papers acknowledged the issue of data contamination without providing strategies to minimise the risk to the validity of the results [1]. In other papers, a common approach to addressing contamination is temporal filtering, which involves collecting data created after the training cutoff date of the evaluated models [11, 23, 25, 30]. For example, papers use datasets like GitBug-Java [4], JLeaks [22], DebugBench [26], and ClassEval [30] that were constructed specifically to minimise overlap. Another strategy is benchmark design through obfuscation or perturbation, such as OBFUSEVAL [31] or DS-1000 [19], which alters code or phrasing while preserving functionality so that models cannot rely on memorised examples. Data cleaning [18], data splitting [5], and deduplication are widely applied, with clone detection [28] used to eliminate duplicates across training, validation, and test sets. In many cases, experiments are replicated across multiple datasets or with synthetic benchmarks specifically designed to minimise leakage risk. Data contamination is further mitigated by cross-dataset evaluation [3] and ablation studies [8, 12, 22]. By evaluating on multiple benchmarks, including both classic and recently introduced datasets, researchers can assess whether improvements hold beyond data that may be contaminated. Ablation studies, meanwhile, demonstrate that performance gains stem from specific methods or frameworks rather than from data memorisation, since results drop when key components are removed.

Finally, reasoning-based safeguards, such as complex prompting [13] or zero-shot setups [29], help reduce the likelihood that models succeed by recalling exact matches.

☛ Overall, while no strategy can fully eliminate the risk of contamination when the training dataset is unavailable, the combination of temporal filtering, benchmark design, deduplication, cross-dataset validation, and transparent reporting significantly reduces its impact. These measures ensure results are more reflective of model capability rather than memorisation, and provide the community with practical standards for trustworthy LLM evaluation.

RQ3: Replicability and Reproducibility

Model Configuration. While the total number of studies reporting on inference (generation) configuration remains relatively low – 89 out of 177 papers (50.3%) – we observe a positive trend over time: in 2023 10 out of 32 papers (31.2%) included details about inference (generation) configuration; in 2024 this increased to 27 out of 55 papers (49.1%), and by 2025 52 out of 90 papers (57.8%). This upward trajectory suggests growing awareness of the importance of documenting inference settings, even though half of the studies

still omit them. Overall, we observe that model configurations are reported inconsistently across different studies: some report only on training parameters, some only on generation parameters, and others report neither. While we acknowledge that page limits in conference venues make it difficult to document all details, the absence of such information significantly undermines replicability.

☞ To mitigate these shortcomings, we recommend that empirical studies include, at a minimum, a permanent link to an external appendix or artefact repository where prompts (along with their rationale) and model configurations are described in detail. This practice would not only enhance transparency but also ensure that results can be meaningfully reproduced and built upon.

Artefact Availability. In our analysis, we find that 33 out of 177 (18.6%) of empirical studies involving LLMs received an "Artifact Available" badge. This is in stark contrast to the remaining 515 articles published at ICSE, for which 213 badges were awarded (41.4%), thus suggesting that authors conducting research involving LLMs have been less proactive in submitting their contributions to the artefact evaluation tracks. We observe similar trends for "Artifact Reusable badges" (11.9% LLM-based vs. 29.1% other) and "Artifact Functional" badges (7.3% LLM-based vs. 13.6% other). Over the three years examined, we observe a minor increase in the number of studies awarded availability badges. Per year, 6 out of 32 papers (18.75%) from 2023 have these badges, 9 out of 55 papers (16.36%) in 2024, and 18 out of 90 papers (20.00%) in 2025.

We manually analysed the links to the artefacts provided in the examined papers and found that nine papers contain expired links. Of these nine, two had "Artifact Available" badges, and one even had an "Artifact Reusable" badge.

☞ The community should actively promote the permanent availability of artefacts from LLM-based SE research and incentivise researchers to submit their work to artefact evaluation tracks. Such practices not only strengthen replicability and reproducibility but also foster transparency, enable fair comparisons, and accelerate cumulative progress in the field.

RQ4: Sustainability

Reported Costs. The most commonly reported cost among the 177 publications is the computational environment used to conduct experiments, as done by 89 papers (50.3%). This entails the use of dedicated hardware, such as GPUs (77 times) or, in 2 cases, TPUs.

Time is the next most frequently discussed cost, being mentioned 68 times. This includes any notion of time reported (e.g., total execution time or time required for a single LLM invocation). Thirty-six publications (20%) reported both the computational environment and time costs, which are necessary to estimate the energy requirements for conducting experiments. This is an improvement over the 10.2% (30 out of 293 publications) reported by Hort et al. [9], which considered LLM studies on code-related tasks published up to 2022. However, we found no study that directly reported the energy consumed or CO_2 emissions.

Monetary costs have been reported 18 times, with budgets up to 3000$. Twelve publications reported the sizes of input or output content (e.g., number of tokens or files) processed by LLMs, which dictate the incurred costs. Less frequent cost categories include memory requirements (6 times), number of LLM invocations (3 times), and number of floating point operations (1 time).

☞ We encourage authors to clearly report the actual costs incurred to perform experiments, as well as an estimate of the costs associated with using the proposed approach in practice over time. To increase attention to sustainability matters, we believe this should become a standard practice required by conferences and journals, as is already the case for data availability.

User Study. We received 57 responses to our questionnaire. 95% of participants used commercial models in the past 12 months, and 89% said they are very likely (65%) or likely (24%) to continue using them in the next 12 months. However, the majority (65%) do not know whether they will be able to sustain the costs over time, and 9% believe that they will not be able to do so at all.

In contrast, 82% of participants used open models in the last 12 months, and 95% said they are very likely (89%) or likely (6%) to continue using them in the next 12 months. Most participants (65%) believe they will be able to sustain the cost of using specialised hardware required to run open models over time, while 38% are uncertain, and 2% say they will not be able to.

Participants reported spending as little as 200$ to as much as 50,000$ on commercial models in the last 12 months. As for open-source models, generally, researchers do not incur any direct costs; however, the hardware costs to run these models can be high, depending on usage goals (e.g., inference, training, fine-tuning), and our respondents reported spending from 5,000$ to 40,000$ to buy such hardware in the last year. Many mentioned electricity costs, yet no one could provide a monetary value because their institutions cover them or because they are difficult to measure.

When asked how they plan to sustain these costs in future, the participants gave very similar responses for commercial and open models: They plan to rely on a combination of academic funding like research grants (36% for commercial, 37% for open) and academic institutional support (26% and 28%), industrial support in the form of free credits and collaboration (24% and 19%), and some are willing to use their own personal funds (12% and 9%).

☞ While researchers are adopting both commercial and open models, costs are a pressing concern. Nearly all participants have used commercial models in the past year and intend to continue, yet most are unsure whether they can afford long-term usage due to API fees. Open models are also widely used, and researchers are optimistic about sustaining their costs. Still, the bottleneck lies in the steep hardware and energy costs to run and fine-tune them.

5 Future Plans

In future, we plan to expand our analysis to other important empirical aspects, such as the clarity of reporting on prompts and datasets used. Further, we aim to tackle additional ethical and social factors, such as whether the cost and infrastructure demands of LLMs are contributing to a digital divide in SE research, particularly between well-resourced and low-resourced institutions.

6 Conclusions

Our study provides an initial reflection on LLM-based empirical SE studies with respect to crucial empirical and ethical aspects. We hope that our results encourage the community to take up the open challenges and onboard the suggestions highlighted in our work.

Acknowledgments

References

[1] Toufique Ahmed, Kunal Suresh Pai, Premkumar Devanbu, and Earl Barr. 2024. Automatic Semantic Augmentation of Language Model Prompts (for Code Summarization). In *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering* (Lisbon, Portugal) *(ICSE '24)*. Association for Computing Machinery, New York, NY, USA, Article 220, 13 pages. doi:10.1145/3597503.3639183

[2] Ali Al-Kaswan, Maliheh Izadi, and Arie van Deursen. 2024. Traces of Memorisation in Large Language Models for Code. In *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering* (Lisbon, Portugal) *(ICSE '24)*. Association for Computing Machinery, New York, NY, USA, Article 78, 12 pages. doi:10.1145/3597503.3639133

[3] Fraol Batole, David OBrien, Tien N. Nguyen, Robert Dyer, and Hridesh Rajan. 2025. An LLM-Based Agent-Oriented Approach for Automated Code Design Issue Localization. In *Proceedings of the IEEE/ACM 47th International Conference on Software Engineering* (Ottawa, Ontario, Canada) *(ICSE '25)*. IEEE Press, 1320–1332. doi:10.1109/ICSE55347.2025.00100

[4] Islem Bouzenia, Premkumar Devanbu, and Michael Pradel. 2025. RepairAgent: An Autonomous, LLM-Based Agent for Program Repair. In *2025 IEEE/ACM 47th International Conference on Software Engineering (ICSE)*. IEEE Computer Society, 694–694.

[5] Yangruibo Ding, Yanjun Fu, Omniyyah Ibrahim, Chawin Sitawarin, Xinyun Chen, Basel Alomair, David Wagner, Baishakhi Ray, and Yizheng Chen. 2024. Vulnerability Detection with Code Language Models: How Far Are We?. In *2025 IEEE/ACM 47th International Conference on Software Engineering (ICSE)*. IEEE Computer Society, 469–481.

[6] Angela Fan, Beliz Gokkaya, Mark Harman, Mitya Lyubarskiy, Shubho Sengupta, Shin Yoo, and Jie M. Zhang. 2023. Large Language Models for Software Engineering: Survey and Open Problems. In *2023 IEEE/ACM International Conference on Software Engineering: Future of Software Engineering (ICSE-FoSE)*. 31–53. doi:10.1109/ICSE-FoSE59343.2023.00008

[7] Shahriar Golchin and Mihai Surdeanu. 2025. Data Contamination Quiz: A Tool to Detect and Estimate Contamination in Large Language Models. *Transactions of the Association for Computational Linguistics* 13 (2025), 809–830. doi:10.1162/TACL.a.20

[8] Qi Guo, Xiaofei Xie, Shangqing Liu, Ming Hu, Xiaohong Li, and Lei Bu. 2025. Intention is All You Need: Refining Your Code from Your Intention. In *2025 IEEE/ACM 47th International Conference on Software Engineering (ICSE)*. IEEE Computer Society, 728–728.

[9] Max Hort, Anastasiia Grishina, and Leon Moonen. 2023. An Exploratory Literature Study on Sharing and Energy Use of Language Models for Source Code. In *2023 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*. IEEE Computer Society, Los Alamitos, CA, USA, 1–12. doi:10.1109/ESEM56168.2023.10304803

[10] Max Hort, Maria Kechagia, Federica Sarro, and Mark Harman. 2021. A Survey of Performance Optimization for Mobile Applications. *IEEE TSE* (2021).

[11] Xue Jiang, Yihong Dong, Yongding Tao, Huanyu Liu, Zhi Jin, and Ge Li. 2025. ROCODE: Integrating Backtracking Mechanism and Program Analysis in Large Language Models for Code Generation. In *2025 IEEE/ACM 47th International Conference on Software Engineering (ICSE)*. IEEE Computer Society, 670–670.

[12] Kaiyao Ke. 2025. NIODebugger: A novel approach to repair non-idempotent-outcome tests with LLM-based agent. In *2025 IEEE/ACM 47th International Conference on Software Engineering (ICSE)*. IEEE Computer Society, 762–762.

[13] Yichen Li, Yulun Wu, Jinyang Liu, Zhihan Jiang, Zhuangbin Chen, Guangba Yu, and Michael R Lyu. 2025. COCA: Generative Root Cause Analysis for Distributed Systems with Code Knowledge. In *2025 IEEE/ACM 47th International Conference on Software Engineering (ICSE)*. IEEE, 1346–1358.

[14] Jenny T. Liang, Carmen Badea, Christian Bird, Robert DeLine, Denae Ford, Nicole Forsgren, and Thomas Zimmermann. 2024. Can GPT-4 Replicate Empirical Software Engineering Research? *Proc. ACM Softw. Eng.* 1, FSE, Article 60 (2024), 24 pages. doi:10.1145/3660767

[15] William Martin, Federica Sarro, Yue Jia, Yuanyuan Zhang, and Mark Harman. 2016. A survey of app store analysis for software engineering. *IEEE TSE* 43, 9 (2016), 817–847.

[16] Rebecca Moussa and Federica Sarro. 2022. On the use of evaluation measures for defect prediction studies. In *Proceedings of the 31st ACM SIGSOFT International Symposium on Software Testing and Analysis* (Virtual, South Korea) *(ISSTA 2022)*. Association for Computing Machinery, New York, NY, USA, 101–113. doi:10.1145/3533767.3534405

[17] Yuqing Nie, Chong Wang, Kailong Wang, Guoai Xu, Guosheng Xu, and Haoyu Wang. 2025. Decoding Secret Memorization in Code LLMs Through Token-Level Characterization. In *2025 IEEE/ACM 47th International Conference on Software Engineering (ICSE)*. 2880–2892. doi:10.1109/ICSE55347.2025.00229

[18] Yuqing Nie, Chong Wang, Kailong Wang, Guoai Xu, Guosheng Xu, and Haoyu Wang. 2025. Decoding Secret Memorization in Code LLMs Through Token-Level Characterization. In *2025 IEEE/ACM 47th International Conference on Software Engineering (ICSE)*. IEEE, 2880–2892.

[19] Shuyin Ouyang, Jie Zhang, Zeyu Sun, and Albert Merono Penuela. 2025. Knowledge-Enhanced Program Repair for Data Science Code. In *2025 IEEE/ACM 47th International Conference on Software Engineering (ICSE)*. IEEE Computer Society, 782–782.

[20] June Sallou, Thomas Durieux, and Annibale Panichella. 2024. Breaking the Silence: the Threats of Using LLMs in Software Engineering. In *Proceedings of the 2024 ACM/IEEE 44th International Conference on Software Engineering: New Ideas and Emerging Results (ICSE-NIER'24)*. Association for Computing Machinery, 102–106. doi:10.1145/3639476.3639764

[21] Lukas Twist, Jie M. Zhang, Mark Harman, Don Syme, Joost Noppen, Helen Yannakoudakis, and Detlef Nauck. 2025. A Study of LLMs' Preferences for Libraries and Programming Languages. arXiv:2503.17181 [cs.SE] https://arxiv.org/abs/2503.17181

[22] Chong Wang, Jianan Liu, Xin Peng, Yang Liu, and Yiling Lou. 2025. Boosting static resource leak detection via llm-based resource-oriented intention inference. In *2025 IEEE/ACM 47th International Conference on Software Engineering (ICSE)*. IEEE Computer Society, 668–668.

[23] Moshi Wei, Nima Shiri Harzevili, YueKai Huang, Jinqiu Yang, Junjie Wang, and Song Wang. 2024. Demystifying and detecting misuses of deep learning apis. In *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*. 1–12.

[24] Dave Williams, Max Hort, Maria Kechagia, Aldeida Aleti, Justyna Petke, and Federica Sarro. 2025. https://figshare.com/s/903589cadfd84c50613f.

[25] Yifan Wu, Yunpeng Wang, Ying Li, Wei Tao, Siyu Yu, Haowen Yang, Wei Jiang, and Jianguo Li. 2025. An Empirical Study on Commit Message Generation Using LLMs via In-Context Learning. In *2025 IEEE/ACM 47th International Conference on Software Engineering (ICSE)*. IEEE, 553–565.

[26] Junjielong Xu, Ying Fu, Shin Hwei Tan, and Pinjia He. 2025. Aligning the Objective of LLM-Based Program Repair. In *2025 IEEE/ACM 47th International Conference on Software Engineering (ICSE)*. IEEE, 2548–2560.

[27] Zhou Yang, Zhipeng Zhao, Chenyu Wang, Jieke Shi, Dongsun Kim, Donggyun Han, and David Lo. 2024. Unveiling Memorization in Code Models. In *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering* (Lisbon, Portugal) *(ICSE '24)*. Association for Computing Machinery, New York, NY, USA, Article 72, 13 pages. doi:10.1145/3597503.3639074

[28] Zhou Yang, Zhipeng Zhao, Chenyu Wang, Jieke Shi, Dongsun Kim, Donggyun Han, and David Lo. 2024. Unveiling memorization in code models. In *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*. 1–13.

[29] Nusrat Zahan, Philipp Burckhardt, Mikola Lysenko, Feross Aboukhadijeh, and Laurie Williams. 2025. Leveraging Large Language Models to Detect npm Malicious Packages. In *2025 IEEE/ACM 47th International Conference on Software Engineering (ICSE)*. IEEE Computer Society, 683–683.

[30] Tanghaoran Zhang, Yue Yu, Xinjun Mao, Shangwen Wang, Kang Yang, Yao Lu, Zhang Zhang, and Yuxin Zhao. 2025. Instruct or Interact? Exploring and Eliciting LLMs' Capability in Code Snippet Adaptation Through Prompt Engineering. In *2025 IEEE/ACM 47th International Conference on Software Engineering (ICSE)*. IEEE, 566–577.

[31] Yuanliang Zhang, Yifan Xie, Shanshan Li, Ke Liu, Chong Wang, Zhouyang Jia, Xiangbing Huang, Jie Song, Chaopeng Luo, Zhizheng Zheng, et al. 2025. Unseen Horizons: Unveiling the Real Capability of LLM Code Generation Beyond the Familiar. In *2025 IEEE/ACM 47th International Conference on Software Engineering (ICSE)*. IEEE Computer Society, 619–619.