

How Computational Notebooks Are Implemented in the Classroom: Challenges and Impacts—A Systematic Review

Joko Saefan

Faculty of Mathematics, Natural Sciences, and Information Technologies Education, Universitas PGRI Semarang
<https://orcid.org/0000-0002-2810-9628>

Siti Wahyuni

Faculty of Mathematics and Natural Sciences, Universitas Negeri Semarang <https://orcid.org/0000-0001-7237-8115>

Wahyu Hardyanto

hardy@mail.unnes.ac.id

Faculty of Mathematics and Natural Sciences, Universitas Negeri Semarang <https://orcid.org/0000-0001-7556-2057>

Wiyanto

Faculty of Mathematics and Natural Sciences, Universitas Negeri Semarang <https://orcid.org/0000-0002-3766-1684>

Systematic Review

Keywords: computational notebook, Jupyter, Google Colab, classroom implementation, systematic literature review

Posted Date: March 17th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-9124168/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License. [Read Full License](#)

Additional Declarations: The authors declare no competing interests.

Abstract

Literature on the use of Computational Notebook (Notebook) in the classroom remains fragmented, often limiting the relationship between implementation challenges and impacts to narrative descriptions. Consequently, a systematic literature review (SLR) is required to systematically extract the implementation–challenges–impact triad. This SLR aims to synthesize evidence on how notebooks are implemented in classrooms, including the challenges and impacts associated with their use. Following the PRISMA 2020 guidelines, this study focuses on three synthesis constructs: implementation, challenges, and effects. The search was conducted across Scopus and Web of Science Core for publications from 2021 to 2025, resulting in 71 included studies, with the highest concentration in 2023. The dominant platforms identified were the Jupyter ecosystem and Google Colab, with implementation contexts spanning schooling, higher education, and teacher training. In general, implementation patterns indicate that notebooks are more frequently positioned as a core component, characterized by moderate scaffolding and relatively high support layering or workflow. This suggests that the adoption of the Notebook in classroom practice is supported more by operational and workflow regularity than by the intensification of conceptual assistance. While technical and pedagogical-cognitive challenges recur, they are often reported narratively. In contrast, challenges regarding assessment and integrity are more explicit because they directly affect the legitimacy of grading. As a result, the correlation between challenges and impacts remains less accessible across various studies. The practical implications point to a need to balance workflow strengths with reinforced conceptual scaffolding. In contrast, the research implications emphasize improving documentation quality and challenge detection to ensure that 21st-century skill outcomes are more grounded in structured evidence. Ultimately, this study offers an operational perspective on what makes technology “work” in the classroom and provides a shared language for mapping technology adoption across core activity integration, scaffolding levels, and operational support.

1. Introduction

Computational Notebooks, commonly referred to simply as Notebooks, are widely used in education for their ability to integrate explanatory narratives, code, and computational output into a single document. This ecosystem evolved from interactive computing practices that emphasize exploration and human-computer dialogue (Perez & Granger, 2007). In an educational context, this narrative and execution format makes thought processes and problem-solving steps easy to trace. Such teaching practices facilitate a seamless blend of conceptual context and coding practice, particularly when learning requires step-by-step data visualization and analysis (Reades, 2020).

Regarding the quality of computational practices, Notebooks bring forward critical issues of reproducibility and process traceability. On the one hand, they simplify the sharing of analyses through explicit procedural traces; on the other hand, the quality of a Notebook determines whether others can replicate its results. Specific Notebook characteristics influence reproducibility and inform best practices when used for instruction or assessment (Pimentel et al., 2021). Further research reveals that reproducibility challenges lie within the code, execution environment, and dependencies. In a study of millions of notebooks on GitHub, Pimentel et al. (2021) found that poor documentation practices, reliance on libraries with ambiguous versions, and non-sequential cell execution are common causes of reproduction failure. This is exacerbated in educational contexts, where students tend to explore code at random, thereby deviating from the instructor's intended workflow (Samuel & Mietchen, 2024). To address these issues, various tools are being developed to identify potential problems in notebooks and suggest improvements.

Furthermore, the effectiveness of Notebooks is inseparable from the pedagogical design in which they are utilized. The existence of Notebooks as digital content opens significant opportunities for the application of active learning and blended learning models. A blended learning approach is particularly relevant, as Notebooks can serve as a bridge between classroom theory and independent practice at home (De Santo et al., 2022). Through Notebooks, students can actively engage with the material as experimenters, receiving instant feedback from code execution. This supports a shift from traditional teacher-centered instruction toward a more student-centered learning environment, where exploration and discovery are key.

Within a broader framework, Notebooks are highly relevant to 21st-century learning, which demands cross-disciplinary competencies. Comparative analyses show a consistent spectrum of skills associated with Notebook use, despite differences in terminology and emphasis (Voogt & Roblin, 2012). A foundational concept in this regard is computational thinking, in which Notebooks serve as a practical medium for orchestrating learning activities that combine modeling, exploration, and communication through computational artifacts (Wing, 2006). Computational Thinking (CT), with its core pillars of decomposition, abstraction, pattern recognition, and algorithmic design, has become increasingly crucial across all disciplines in the digital era. Notebooks provide a rich environment for teaching and practicing these CT skills. They facilitate the learning of both programming syntax and the computational mindset essential for solving real-world problems (De Santo et al., 2022).

Despite their growing adoption, empirical evidence suggests that effectiveness depends on implementation design, student readiness, and infrastructural support. Classroom studies indicate that Notebooks can enhance engagement, learning experiences, and outcomes (Amoudi & Tbaishat, 2023). At the instructional level, their use entails varying operational consequences across courses and institutions (Reades, 2020), including the need for computational scaffolding, material management, and support for self-directed learning. A significant gap exists concerning the quality of notebook artifacts and reproducibility. Research shows that issues with documentation, execution order, and re-execution failures are common; a Notebook may "appear successful" when created but remain difficult for others to verify or replicate (Pimentel et al., 2021; Samuel & Mietchen, 2024). In practice, best-practice recommendations emphasize that the quality of process documentation, narrative structure, and execution habits determines the Notebook's readability, repeatability, and utility (Rule et al., 2019). Furthermore, while Notebooks are often used as assessment media, this adds a layer of complexity. Studies show that using Notebooks for formative assessment can improve attitudes and self-efficacy but can also introduce hurdles, such as programming anxiety, resistance to open-source technology, and specific technical requirements (Temel et al., 2025). On the tooling side, ecosystems like nbgrader demonstrate that automating the distribution, collection, and grading of Notebooks requires standardized task structures and consistent management practices (Jupyter et al., 2019).

The current literature remains fragmented: some studies focus on implementation patterns and learning experiences, others on quality, and others on assessment and grading automation. Consequently, evidence regarding the relationship between implementation, challenges, and impacts is largely limited to narrative descriptions. Therefore, a Systematic Literature Review (SLR) is needed to structurally extract: (a) implementation characteristics, (b) types of challenges and their detection methods, and (c) types of impacts. This extraction enables a traceable synthesis and opens opportunities for quantitative analysis.

This study is designed as an SLR reported following the PRISMA 2020 guidelines through a process of identification, selection, and synthesis (Page, McKenzie, et al., 2021). The review focuses on three synthesis

constructs: implementation, challenges, and impacts. Based on this foundation, the study synthesizes evidence to answer the following research questions:

RQ 1: How are Notebooks implemented in learning environments?

RQ 2: What are the challenges of using Notebooks in learning?

RQ 3: What is the impact of using Notebooks in learning on 21st-century skills?

The primary contribution of this study is providing an evidence map that links the role of Notebooks in learning across three areas: implementation practices, quality issues as computational artifacts, and classroom assessment issues. By combining these perspectives, this study positions the Notebook as a learning artifact whose quality influences the learning experience, challenges, and outcomes. Thus, this SLR provides a basis for more evidence-informed Notebook implementation design and paves the way for further quantitative analysis.

2. Literature Review

Notebooks are presented as tools and formats for executable computational artifacts. Consequently, their existence carries pedagogical implications that support exploration, as well as methodological consequences involving risks of statefulness and iterative processes. This concept originates in literate programming, which treats a program as a narrative that explains computational logic and procedures (Knuth, 1984). This pioneering work evolved through interactive computing, such as IPython, into Jupyter Notebooks (Perez & Granger, 2007). The resulting document is an integrated fusion of text, code, and outputs (Kluyver et al., 2016).

2.1 The Evolution and Characteristics of Computational Notebooks

Notebooks treat programs as narratives that explain computational logic, allowing readers to follow the reasoning, steps, and structure of problem-solving. This narrative and code format facilitates procedural reasoning and step-by-step reflection during the production process (Knuth, 1984). Developed through interactive computing, notebooks reinforce iterative exploration in scientific computing. Users can test code snippets, modify them, rerun them, and immediately interpret the output. This characteristic aligns with exploration-based learning patterns, where understanding is formed through a "test–interpret–revise" cycle (Perez & Granger, 2007). Building on this, Jupyter Notebook formalized the notebook as a document that integrates text, code, and output into a single artifact. Kluyver et al., (2016) emphasize the notebook as a publication format for reproducible computational workflows. This is documentation of an analytical process that readers can follow and re-execute.

The next development is the shift toward cloud-based platforms, with Google Colab serving as a prominent representation of this phase. Colab is a hosted Jupyter Notebook service that runs in a browser and provides computational access, making it easier to adopt for network-based learning and computational projects (Google, 2025). In an educational context, Colab's primary value lies in lowering initial technical barriers—such as installation, OS compatibility, and dependencies—so the focus can shift to learning activities. Google has even positioned Colab as the easiest way to start Python programming since 2017 (Google, 2025). Evidence of its utility can be seen in studies integrating GitHub and Colab to ensure equitable access across devices and eliminate installation friction for students (Osório & Garma, 2025). Based on this evolution, the key characteristics of Notebooks can be summarized as:

- Narrative–Executable: Explanation and computation within a single document.
- Interactive–Iterative: Readily supporting rapid exploration and revision.
- Stateful: Results are influenced by the execution order of cells and the environment.
- Service-based Accessibility: Minimal setup, available computing power, and streamlined sharing/collaboration.

2.2 Implementing Notebooks in Education

The implementation of Notebooks is understood as the design of a learning ecology that orchestrates (i) the extent to which Notebooks are integrated into the course structure, (ii) the types of computational activities at the core of learning, and (iii) the sustained practical support provided to students. Notebooks function as a learning environment that binds together content, exercises, and computational artifacts. Successful implementation requires a well-contained approach toward both the depth of curricular integration and the specific forms of activity (Reades, 2020; Rowe et al., 2021). For Notebooks to serve effectively as a learning medium, the role of scaffolding—through templates, step-by-step guidance, and interpretation prompts—must be carefully managed (Vallejo et al., 2022). Furthermore, implementation almost always depends on the chosen deployment ecosystem and distribution workflow.

Notebooks appear across a broad spectrum of integration: from structured worksheets interspersed within a course to serving as the pedagogical backbone, where they are the primary medium. Reads (2020) describes Notebooks as a teaching infrastructure that accelerates curriculum development and data-driven learning activities. Physics laboratory studies show that Notebooks can be integrated without drastically altering the course structure while remaining the primary media for data analysis exercises (Tufino et al., 2024). At the assignment level, Notebooks can be positioned as the central artifact for narrative-computational tasks, such as computational essays that combine scientific modeling with communication (Odden & Malthe-Sørensen, 2020).

Notebook implementations can be categorized into several activity patterns: (a) data analysis and visualization, (b) modeling, and (c) narrative-computational assignments. Examples of data analysis include integrating Python for physics lab analysis (Tufino et al., 2024) and using notebooks for problem-based learning in geosciences (Campbell et al., 2025; Rowe et al., 2021). Modeling examples are seen in thermodynamics notebooks paired with virtual simulations in Google Colab (Vallejo et al., 2022) and the redesign of Computational Fluid Dynamics (CFD) learning that combines Notebooks with industrial packages (Seddighi et al., 2020).

Implementation is generally supported by scaffolding via templates, sequenced steps, worked examples, interpretation cues, and tiered exercises. In Colab-based thermodynamics learning, Notebooks are structured as learning objects containing both exercises and solutions. In laboratory data analysis, Notebooks are designed with exercises and physics application examples to guide mastery of programming. In data-driven machine learning, modules are equipped with worked examples and modeling process structures so students can emulate generalized modeling practices (Fleischer et al., 2022).

Implementation choices are often determined by how the Notebook is brought to life in the classroom: via local installations (e.g., Jupyter) or cloud platforms (e.g., Colab) to minimize installation and compatibility barriers. Studies on Colab for thermodynamics highlight browser-based Notebooks as e-learning resources and a gateway to coding for students without prior programming experience. Meanwhile, implementations in computing-heavy courses may be paired with other tools that require managing computational environments and technical support readiness (Seddighi et al., 2020). In practice, literature also indicates that implementation shifts the friction from

installation to activity design and support, such as through online resources, problem-solving sessions, or accessible material repositories (Campbell et al., 2025; Vladis & Coleman, 2021).

A decisive dimension for implementation consistency is how Notebooks are distributed, completed, and collected. In this context, tooling such as nbgrader marks a more "systematic" implementation approach, as it supports structured assignment creation, distribution, and grading. Examples include integration with Learning Management Systems (LMS) within the Jupyter ecosystem (Kluyver et al., 2016). Conversely, some computing education studies utilize code repositories to support material replication, version tracking, and student access. When Notebooks are positioned as a medium for formative assessment, implementation can also take the form of assessment activities designed directly within the notebook (Temel et al., 2025).

2.3 Challenges in Notebook Implementation

Implementation challenges arise because Notebooks play a dual role: as a computational environment (runtime, dependencies, execution, reproducibility) and as a pedagogical artifact (narrative–code–output serving as a medium for learning and assessment). Their interactive and stateful nature means the execution order of cells, environment configurations, and computational work habits heavily influences the learning experience. Consequently, challenges are defined by activity design, heterogeneity in programming proficiency, and assessment mechanisms (Johnson, 2020; Pimentel et al., 2021; Reades, 2020; Rule et al., 2019). Across various educational contexts, reports consistently highlight dependency friction, debugging burdens, and the complexities of evaluating notebooks as both products and processes. Therefore, literature emphasizes the importance of identifying challenges and their detection methods—whether based on perception (surveys), artifacts (Notebooks), process traces (logs), or assessment mechanisms (workflows) (González-Carrillo et al., 2021; Kluyver et al., 2016; Nwulu et al., 2021; Vladis & Coleman, 2021).

Technical challenges center on reproducibility and environment stability: package and version dependencies, configuration errors, and inconsistent results due to cell execution order or runtime states. Literature indicates that execution order issues, incomplete artifacts, and poor documentation practices can diminish the reproducibility of an analysis. These issues become critical when Notebooks are used as instructional materials that must be executed by many students across different devices (Pimentel et al., 2021; Rule et al., 2019). In teaching practice, deployment choices are often presented as strategies to reduce installation friction, yet they still leave unresolved issues of compatibility, environmental management, and packaging requirements (Campbell et al., 2025; Reades, 2020; Seddighi et al., 2020). Cloud-based studies confirm the benefits of accessibility, but the technical consequences shift toward file management, connectivity, resource limits, or service dependencies (Osório & Garma, 2025; Vallejo et al., 2022). Within a course context, environmental stability and device readiness are often prerequisites to ensure that the learning focus is not consumed by troubleshooting (Domínguez et al., 2021; Ruiz-Sarmiento et al., 2021).

In the cognitive domain, the key challenge is the dual learning load: students must grasp domain concepts while simultaneously building computational competence. Many reports show that basic programming hurdles can submerge conceptual goals if scaffolding is inadequate (Fleischer et al., 2022; Johnson, 2020; Reades, 2020). Heterogeneity in prior programming skills is frequently cited as a source of gaps in participation and learning tempo. Evidence for this emerges from both student-perception studies and classroom-implementation narratives (Campbell et al., 2025; Nwulu et al., 2021; Vladis & Coleman, 2021). In some studies, highly structured Notebook designs are used to lower cognitive load and help students link computational output with conceptual meaning (Domínguez et al., 2021; Fleischer et al., 2022; Vallejo et al., 2022). In team-based learning or guided inquiry, social

support and collaborative structures are employed to reduce debugging friction and facilitate more productive problem-solving (Kumwihar, 2023; Osório & Garma, 2025; Rowe et al., 2021).

Assessment challenges arise when the Notebook is used as an assessment format that combines narrative, code, and output: what should be graded, and how can fairness and consistency be maintained? A Notebook may appear "correct" in one runtime but fail to execute cleanly upon a fresh re-run, making assessment based on re-execution or test cases complicated (Pimentel et al., 2021; Rule et al., 2019). Tooling such as nbgrader institutionalizes workflows to reduce operational burdens and improve consistency. Autograding studies highlight the challenges of maintaining validity and fairness when student solutions vary, as well as the risk of teaching to the tests if feedback is not designed to be educational (González-Carrillo et al., 2021). In studies where Notebooks serve as a digital assessment medium, issues of readiness and perception toward the assessment itself become part of the implementation challenges (Amoudi & Tbaishat, 2023; Temel et al., 2025).

The three categories of challenges mentioned above serve as a framework for detection. Challenges reported through surveys provide an overview of the learning experience, but the weight of their evidence differs from challenges demonstrated by artifacts, process traces, or assessment mechanisms that yield operational evidence.

2.4 The Impact of Notebooks on 21st-Century Skills

Within the framework of 21st-century competencies, the impact of Notebook implementation is understood as a multi-domain output: cognitive-conceptual, computational, and socio-affective. 21st-century competencies emphasize higher-order thinking skills and collaborative practices embedded in authentic activities. Conceptual and professional outcomes emerge when Notebooks are positioned as artifacts where students link narratives, models, and evidence within a single document. This pattern frequently appears in the tradition of computational essays in physics, where Notebooks serve as a medium to demonstrate scientific reasoning and professional practice (Odden, 2019; Odden & Malthe-Sørenssen, 2020). Furthermore, integrating Notebooks into laboratory settings emphasizes data interpretation and evidence-based argumentation (Casebeer & Frano, 2025; Tufino et al., 2024). In several studies, Notebooks are linked to enhanced learning processes and module comprehension through problem-based tasks and automated question generators (Bascuñana et al., 2023; Castilla & Peña, 2023; Domínguez et al., 2021; Seddighi et al., 2020).

The Computational Thinking (CT) framework positions Notebooks as a means to develop computational ways of thinking (Wing, 2006). This outcome typically occurs when Notebooks are used for data analysis, modeling, and programming exercises integrated with domain-specific goals. For instance, learning Python or R through Notebooks in biomedical and health contexts emphasizes computational analytical skills and data-driven problem solving (Gupta et al., 2023; Kumwihar, 2023; Vladis & Coleman, 2021). Robotics contexts demonstrate CT as the ability to apply computation to real-world systems and applied tasks (Castilla & Peña, 2023; Ruiz-Sarmiento et al., 2021; Seddighi et al., 2020). In implementations where Notebooks are the core of activity design, CT outcomes also intersect with computational literacy: writing, executing, debugging, and communicating code (Campbell et al., 2025; Osório & Garma, 2025; Rowe et al., 2021). Even when the research focus is on improving module learning, Notebooks are positioned as a medium that "compels" students to explicitly practice CT through computational tasks.

Notebooks can be understood as a learning-as-assessment medium that influences engagement and agency. Students observe a direct correlation between their actions and the resulting consequences, making the learning experience more meaningful. Impact is often reported through indicators of attitude, perception, and learning

experience—such as engagement, perceived usefulness, or readiness to use Notebooks (Amoudi & Tbaishat, 2023; Temel et al., 2025). Implementing cloud-based Notebook designs that blend simulations with interactive activities is positioned as a factor that enhances accessibility and learning engagement. Additionally, when Notebooks are used as an assessment format, aspects of self-confidence and digital assessment readiness become vital outcomes. This outcome framework confirms that the impact of Notebooks on 21st-century skills can manifest as reinforced conceptual understanding and reasoning, enhanced CT and computational practices, or affective-agency shifts and technological readiness.

3. Methods

3.1 Research Design and Protocol

This study employs an SLR design to synthesize empirical evidence concerning the implementation of Notebooks in learning (IMP), the challenges arising from such implementation (CHA), and the reported outcomes regarding 21st-century skills (OUT). The review adopts the principles of evidence-informed and transparent reviewing, which include the formulation of explicit research questions, replicable search and screening procedures, and auditable data extraction rules (Tranfield et al., 2003; Xiao et al., 2021). Reporting is structured according to the PRISMA 2020 guidelines, providing a traceable account of the identification, screening, eligibility, and inclusion processes (Page, McKenzie, et al., 2021; Page, Moher, et al., 2021). In line with documentation recommendations, this protocol has been archived in the Zenodo repository to provide a permanent methodological record.

3.2 Search Strategy

Two multidisciplinary bibliographic databases, Scopus and Web of Science (WoS), were utilized for the literature search. The search was conducted via the advanced search interface of each database to enable field-specific retrieval. The search strategy and reporting methods were aligned with established guidelines for reporting literature searches in SLR (Rethlefsen et al., 2021).

The search query combined two conceptual blocks using Boolean logic:

1. Computational notebook terms: "computational notebook" OR Jupyter* OR "Google Colab*" OR "Kaggle"
2. Education/Learning context terms: classroom OR educat* OR teach* OR student* OR course* OR curricul* (with the addition of "pedagogy" in WoS).

Wildcard symbols (*) were employed to capture spelling variations and morphological forms; for instance, educat* retrieves "educate," "education," and "educational." The timeframe targeted publications from 2021 to 2025 to capture the post-2020 growth in notebook-aided learning practices as a window into current state-of-the-art developments.

In Scopus, the query was executed across titles, abstracts, and keywords, with filters for publication years 2021–2025, document types (article and conference paper), English language, and Open Access (all):

(TITLE-ABS-KEY ("computational notebook" OR Jupyter* OR "Google Colab*" OR "Kaggle")

AND TITLE-ABS-KEY (classroom OR educate* OR teach* OR student* OR course* OR curricul*))

AND PUBYEAR > 2020 AND PUBYEAR < 2026

AND (LIMIT-TO (DOCTYPE, "cp") OR LIMIT-TO (DOCTYPE, "ar"))

AND (LIMIT-TO (LANGUAGE, "English"))

AND (LIMIT-TO (OA, "all"))

AND (LIMIT-TO (SRCTYPE, "j") OR LIMIT-TO (SRCTYPE, "p"))

This search yielded n = 262 records.

In WoS, the query utilized the same combination of notebook and education concept blocks, limiting results to English, Open Access, and document types (Article or Proceedings Paper):

("computational notebook" OR "Jupyter*" OR "Google Colab*" OR "Kaggle")

AND (classroom OR teach* OR educat* OR course* OR pedagogy OR curricul*)

AND (Publication Years: 2021 OR 2022 OR 2023 OR 2024 OR 2025 OR 2026)

AND (Document Types: Article OR Proceedings Paper)

AND (Language: English)

AND (Open Access)

This search yielded n = 190 records.

All retrieved records were exported from their respective databases and merged into a Zotero library for duplicate checking. The de-duplication procedure involved matching combinations of title, author, year, and DOI. In total, the combined search from Scopus and WoS gathered 452 records (262 + 190). After removing 132 duplicates, n = 320 unique records remained for the abstract screening phase.

3.3 Eligibility Criteria

Eligibility criteria were established to ensure a transparent and replicable study selection process. Studies were included if they met all of the following conditions: (1) involved learning activities within a classroom-like setting, such as lectures, laboratory sessions, tutorials, or workshops; (2) utilized Notebooks, such as Jupyter Notebook/JupyterLab, Google Colab, or other notebook environments; and (3) reported assessments, such as pre/post-tests, concept inventories, assignment scores, artifact analysis, learning gains, or evidence of conceptual understanding. This structural framework is compatible with educational technology research to avoid overly broad specifications (Cooke et al., 2012; Xiao et al., 2021).

Several exclusion rules served as primary filters, including:

1. Non-classroom: Excluded when the abstract or full text indicated that the setting was not a formal learning activity or the participants were not learners or instructors. Common indicators included an emphasis on "research workflow," "pipeline," "reproducible research," or "scientific computing" without a pedagogical context.
2. Non-computational notebook: Excluded when the technology used was not notebook-based. Required indicators for inclusion included explicit mentions of "Jupyter Notebook/JupyterLab," "Google Colab," "computational notebook," "notebook-based," or R Markdown notebooks.

3. Non-learning assessment: Excluded when the publication did not evaluate learning. Studies reporting only satisfaction surveys were excluded unless accompanied by evidence relevant to learning, such as pre-/post-measures, artifacts, performance data, or interviews.

3.4 Screening and Selection

The screening and selection of studies followed a two-stage process, as reported in the PRISMA 2020 flow diagram (Fig. 1) (Page, McKenzie, et al., 2021; Page, Moher, et al., 2021).

First, an abstract screening was conducted in accordance with the eligibility criteria. Out of 320 records, 197 were excluded at this stage for failing to meet core requirements: non-classroom context ($n = 134$), non-notebook technology ($n = 43$), and lack of learning assessment ($n = 20$). The remaining 123 records underwent full-text retrieval; however, 14 full-text articles were unavailable and excluded at the eligibility stage.

Second, full-text screening was performed on 109 articles to confirm eligibility and ensure that the notebook intervention, classroom-based learning context, and learning assessments were explicitly documented in the full manuscripts. At this stage, 38 studies were excluded: non-classroom context ($n = 20$), lack of learning assessment ($n = 15$), and non-notebook technology ($n = 3$). Following these exclusions, 71 studies met all inclusion criteria and were included in the final synthesis.

3.5 Data Synthesis and Analysis Plan

The data synthesis is designed to maintain auditability while enabling a scoring process for IMP, CHA, and OUT. Coding is guided by a codebook that treats each field as a structured evidence container: coders record brief descriptors and concise paraphrases directly linked to the article's content. The codebook emphasizes: (i) descriptive coding for implementation features regarding what was done, how often, and with what infrastructure; (ii) analytical categorization for types of challenges and outcome domains; and (iii) explicit statements regarding the strength of evidence. This approach requires transparent coding rules and traceable evidence, in the form of extracts or citations.

The synthesis is conducted in two steps. First, all included studies are summarized through a structured extraction table that preserves the evidence for IMP, CHA, and OUT. Second, a deterministic scoring scheme is applied to the extracted fields to generate numerical indicators. This two-step approach—preserving raw evidence before applying transparent, rule-based quantification—supports traceability and mitigates subjective bias when synthesizing heterogeneous studies.

Table 1
Scoring rules for the implementation, challenges, and impacts of Notebook use.

Indikator	1	2	3	4	5
Depth/Role & intensity (IMP1)	NR	Demo role or single/once use	Repeated use (≥ 2 activities/tasks)	Core role or modules ≥ 2 or replaces part of core instruction	Core + high-intensity implementation,
Scaffold richness (IMP2)	0 token	1–2 tokens	3–4 tokens	5–7 tokens	≥ 8 tokens or ≥ 5 tokens + explicit advanced scaffolds
Support layering (IMP3)	NR	1 support element	2 support elements	≥ 3 elements or structured support	Layered support + at least one “strong package”:
Technical challenges (CHA1)	NR	Mentioned generally (no concrete evidence)	Reported narratively (reported = Y, sources/freq NR)	Explicit source	Strong evidence: ≥ 2 source types and quantified frequency
Pedagogical/ cognitive challenges (CHA2)	NR	Implied/ mentioned generally	Reported narratively	Explicit source but no frequency/triangulation	Strong evidence + quantified/triangulated
Assessment/ integrity challenges (CHA3)	NR	Implied/ mentioned generally	Reported narratively	Explicit source but no frequency/triangulation	Strong evidence + quantified/triangulated
Conceptual/ professional outcomes (OUT1)	NR	Intended benefit claim only	Reported narratively	Clear measure/instrument	Statistical evidence
Computational outcomes (OUT2)	NR	Intended claim only	Narrative qualitative evidence	Clear measure/instrument	Statistical/strong design evidence
Affective/ agency outcomes (OUT3)	NR	Intended claim only	Narrative qualitative evidence	Clear measure/instrument,	Statistical/strong design evidence

The scoring rules in this article follow the rubric presented in Table 1. A score of 1 represents the lowest, while 5 represents the highest. Certain indicators include additional explanations; for instance, IMP1 with a score of 5, e.g., any ≥ 2 strong signals: multiweek; managed environment; gradebook; containerization; scaled repeated events. IMP 2 strong packages, for example: tiered tasks, worked examples, reflection, and step-by-step transparency. IMP3 support element: documentation, link, instructor support; structured support: (training, TA, clinic, or formal troubleshooting); strong packages: autograder, versioning, or formal troubleshooting pipeline. CHA1 Explicit source: survey, interview, artefact, assessment, or material, but no frequency or triangulation. OUT1 Reported narratively with qualitative evidence reflections, observations, or quotes; Clear measure: (test, rubric, survey, or artefact) but no robust stats; Statistical evidence: p effect size, pre–post, or comparator.

4. Result

4.1 Study characteristics

A total of 71 studies were included in the quantitative synthesis (Table 2 in the Appendix), published from 2021 to 2025, with the highest concentration in 2023, as illustrated in Fig. 2. The most dominant platforms identified belong to the Jupyter ecosystem and Google Colab, as shown in Fig. 3. Based on the contextual descriptions in the evidence summary, Notebook implementation is distributed across K-12 schooling, higher education, and teacher training, as depicted in Fig. 4. The subject areas represented are quite diverse; while several studies mention specific fields, others are not sufficiently explicit to be classified.

Table 2
Sinteszid data

ID	Article	Platform
S017	(Allen et al., 2025)	Jupyter Notebook on HPC via web portal (TAP) + CLI + DCV; containers (Apptainer/Docker)
S036	(Sytnykova et al., 2025)	Google Colab (cloud Jupyter notebook environment; Python); interactive widgets (ipywidgets) + plots/visualization
S039	(Ho et al., 2025)	Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence (AAAI-25); copyright AAAI; DOI = NR; indexing = NR
S079	(Sugiarto et al., 2024)	Jupyter Lab + Python + SymPy
S093	(Conroy et al., 2024)	Python Jupyter notebooks (interactive training repository) + ATLAS Open Data; tools: ROOT/uproot, python stack
S114	(Balovsyak et al., 2024)	Google Colab (Jupyter Notebook) + Python; hardware: PC atau Raspberry Pi 3B + + USB cameras
S144	(Seebut et al., 2024)	Google Colab (Jupyter) + GPT (GPT-3.5) + Python
S151	(Seth et al., 2023)	Pluto notebooks (Julia/Pluto.jl) + AeroFuse (MADE software; online, open-source)
S162	(W. B. Lane et al., 2023)	Jupyter Notebooks (Python) + Zoom (online synchronous PD)
S175	(Lo et al., 2023)	Google Colab (Jupyter Notebook)
S178	(Heredia-Negron et al., 2023)	TalentLMS asynchronous course + Jupyter Notebook framework; coding with Python & R; ML libraries
S211	(Lyu et al., 2022)	Google Colab (Jupyter Notebook) + Web-based game (Unity/WebGL) + Kahoot + Zoom
S212	(Lee & Perret, 2022)	Google Colab (Jupyter Notebook) + interactive web tools/games + online PD (sync)
S220	(Kozakai et al., 2022)	Google Colaboratory (Jupyter) + Google Drive (sharing); senior online via Zoom; junior face-to-face
S227	(Podworny et al., 2022)	Jupyter Notebook (browser-based) + Python; prepared notebooks (markdown+code cells); sensor boxes (senseBox)
S226	(Fleischer et al., 2022)	Jupyter Notebook (Python) as computational essay; decision-tree ML; project-course context
S248	(Tang, 2021)	Jupyter-Python Notebook; materials via GitHub; plotting libs
S251	(Foster & Wagner, 2021)	Google Colab (Jupyter Notebook); scikit-learn; HuggingFace Transformers; PyTorch Lightning;
W001	(Truong et al., 2024)	Google Colab (Jupyter Notebook); SLEAP (open-source ML); resources via GitHub/Linktree
W004	(De Santo et al., 2022)	Graasp digital notebook; built-in Python code app + Answer app + Point-counter gamification app
W006	(Llerena-Izquierdo et al., 2024)	Google Colab (Jupyter Notebook) + Gemini (GenAI); also Moodle + PSeInt (pseudocode/flowchart)

ID	Article	Platform
W015	(Vidal-Silva et al., 2022)	Google Colab (Jupyter Notebook) + Python (in remote teaching); comparison baselines: Java/Eclipse and C/Linux (2020)
W016	(Tufino et al., 2025)	Jupyter Notebooks (Python) via Anaconda on lab PCs; GitHub materials; (Colab discussed but not adopted due to privacy)
W017	(Laky et al., 2023)	Jupyter Notebook + PharmaPy (open-source Python pharmaceutical manufacturing process simulator); Anaconda for package/env management
W018	(Temel et al., 2025)	Jupyter Notebooks; Python + R; (nbgrader, jupyterquiz, jupytercards, Graphviz/pydot, Markdown)
W020	(Kumwichar, 2023)	Jupyter Notebook for R via self-hosted JupyterHub server (online); PDF instruction + GitHub materials
W022	(Betlem et al., 2025)	Jupyter Book integrated with GitHub backend + multimedia (GIFs/videos) + code snippets/templates
W023	(Lonsky et al., 2024)	Jupyter Notebook controlling Ubermag (interfaces OOMMF/mumax3); cloud via Binder/MyBinder;
W024	(Wang et al., 2023)	Jupyter Notebook + Google Colaboratory (runs in browser via link); optional local Jupyter via Anaconda
W025	(Tufino et al., 2024)	Jupyter Notebooks (Python) in Anaconda environment (in-class); GitHub repository (EN/DE versions); cloud (Google Colab)
W027	(Spencer-Tyree et al., 2024)	Jupyter Notebook (Python) used in-class; students used computers during labs
W028	(Liebal et al., 2023)	Jupyter Notebook + silvio virtual-organism simulator; via JupyterHub (RWTH) and/or Binder; Moodle as course hub;
W029	(Wagemann et al., 2022)	Jupyter Notebooks + hosted JupyterHub training platform (LTPy) + JupyterBook + GitLab
W031	(Nwulu et al., 2021)	Jupyter Notebook (Python) + GEKKO optimizer (FOSS)
W033	(Elshall & Badir, 2025)	Environmental Data Science course ; materials via Jupyter Book; AI coding assistance via Jupyter AI + ChatGPT (3.5 Turbo)
W037	(King & Sharifi Far, 2024)	Jupyter notebooks; (Noteable/ EDINA) integrated with institutional VL; uses Microsoft Teams + Miro; assessment tools nbgrader + CodeRunner
W043	(Goswami et al., 2023)	Jupyter Notebook (Python) on JupyterHub; collaborative extension “Thyone” (Flowchart + Discuss + Share Cell)
W044	(Xiao et al., 2021)	Jupyter Notebook cell-based workflow used consistently across 3 languages (Python, R, MATLAB); course materials publicly available (Google Sites)
W045	(Zhao et al., 2025)	CyberFaCES platform: Halcyon-based CMS front end + JupyterHub back end; Jupyter Notebook environment; backend connects to HPC resources
W046	(W. B. Lane et al., 2023)	Python workshop for high school physics teachers using Jupyter Notebooks (online synchronous; breakout rooms)
W054	(Cai et al., 2025)	Jupyter Analytics: two JupyterLab extensions (Telemetry + Dashboard) + cloud backend server; embedded real-time dashboards inside JupyterLab
W058	(Bascuñana et al., 2023)	Jupyter Notebook (Python) (interactive notebook untuk pembelajaran konsep chemical engineering)

ID	Article	Platform
W059	(Castilla & Peña, 2023)	Jupyter Notebooks (Python; Jupyter/JupyterLab) + Moodle forum; public GitHub repo for course notebooks
W061	(Chen & Asta, 2022)	Jupyter Book + cloud execution via JupyterHub (UC Berkeley DataHub) and optional Google Colab; hosted on GitHub Pages; files on GitHub (CC BY-SA 4.0)
W064	(Biehler & Fleischer, 2021)	CODAP (Arbor decision-tree plug-in; web-based) + ProDaBi Decision Tree Jupyter Notebook
W065	(González-Carrillo et al., 2021)	UNCode (built on INGINious) + Jupyter Notebooks (Python) + OK CLI + Docker sandbox; web platform https://uncode.unal.edu.co
W066	(Ruiz-Sarmiento et al., 2021)	Jupyter Notebooks (Python) educational notebook suite for mobile robotics; public student notebooks on GitHub; executable in class via web browser
W075	(Alzahrani, 2025)	Web-based Jupyter notebooks hosted on GitHub; runnable via Google Colab and/or JupyterHub (incl. HPC via ACCESS); accessible on any device incl. smartphones
W091	(Zabasta et al., 2024)	SMSE (Shared Modeling and Simulation Environment) integrates Jupyter (Notebooks) + Moodle LMS; virtual labs via Jupyter capabilities
W097	(Rainey et al., 2024)	Google Colab Notebook (Python) for postlab data analysis; queries NIST Atomic Lines DB
W101	(Santos & Collaboration, 2025)	Auger Open Data portal + Python Jupyter notebooks; Auger 3-D Event Display (Unity);
W106	(Fransson et al., 2023)	eChem e-book (web) built with Jupyter Notebook + Jupyter Book; workflows run with Python-driven QC packages
W119	(Kayhan & Berndt, 2023)	Jupyter Notebook + MongoDB Atlas + MongoDB Compass; notebooks/materials hosted on GitHub
W120	(Kim & Henke, 2021)	Jupyter Notebook + GitHub + Binder; SQL via BeakerX;
W132	(Lapeña-Mañero et al., 2022)	Python-based open-source assignment generator + auto-grader using Jupyter Notebook as GUI
W144	(Elhayany & Meinel, 2023)	Integrates JupyterLab with MOOCs/openHPI)
W148	(Roundy et al., 2022)	HydroLearn (edx.hydrolearn.org) module + Google Colab with Python ipywidgets-based interactive UI
W149	(Krugh & Mears, 2021)	Python JupyterLab notebooks + Microsoft PowerBI dashboard; IoT sensor network
W150	(Werth et al., 2022)	Google Colaboratory (Colab) used throughout an online large-enrollment physics CURE
W153	(Resendez et al., 2025)	JupyterHub + virtual delivery (live lectures + recorded asynchronous viewing)
W157	(Osório & Garma, 2025)	Cloud-hosted interactive Python notebooks: Jupyter notebooks hosted on GitHub and run via Colab; post-session survey via Microsoft Forms
W167	(Hall & Cantrell, 2024)	Google Colaboratory (hosted Jupyter Notebook) + GitHub repository with student notebooks and stand-alone tutorial notebooks

ID	Article	Platform
W169	(Resendez et al., 2025)	JupyterHub + virtual recorded lectures (asynchronous viewing)
W173	(Angara et al., 2022)	IBM Quantum Experience with native Qiskit + Jupyter Notebooks + Circuit Composer;
W175	(Zhang et al., 2023)	JupyterLab “fillable worksheets” + supporting Python library functions; code/notebooks available on GitHub
W176	(Grazioli et al., 2023)	Python + Jupyter Notebook; open-source Lennard-Jones (LJ) fluid simulation code + notebooks on GitHub
W177	(Vanegas-Guillén et al., 2023)	RemoteLabo RLMS; JupyterLite + student sandbox + lab interface; MQTT pub-sub (AWS IoT) + WebRTC video
W178	(Sánchez-Peña et al., 2023)	R Markdown (LearnR package) for an interactive tutorial; R programming environment used in course
W180	(Wen et al., 2022)	JupyterHub + Kubernetes cluster; integrated Android Emulator + OpenAirInterface (gNB/nrUE) + Rust-based 5G core + P4 (BMv2)
W184	(B. Lane et al., 2021)	HydroLearn open + integrated CUAHSI HydroShare + CUAHSI JupyterHub + ESRI Story Maps; also uses open data services
W190	(Callupe et al., 2021)	Google Colab (Jupyter notebooks, Python) + open-source data-science stack

The surge in publications in 2023, shown in Fig. 2, can be explained by the combined effect of a post-COVID-19 "research maturation" following the accelerated digital transformation in education and a "new topical wave" that prompted many researchers to revisit digital learning practices. Conceptually, the accelerated digitalization during the pandemic forced institutions to establish more permanent digital learning ecosystems; consequently, research data collected between 2020 and 2022 entered the writing phase and was published in 2022–2023 (Bygstad et al., 2022). Furthermore, the emergence of chatbots and Large Language Models (LLMs) in education, which peaked in 2023, contributed to the increased publication volume, as many early articles focused on mapping opportunities and limitations, academic integrity, and pedagogical implications (Memarian & Doleck, 2023).

Meanwhile, the decline observed after 2023 does not necessarily reflect a decrease in research activity; rather, it is influenced by bibliometric artifacts: (1) publication lags that can span several months, preventing articles from being published or counted in time (Björk & Solomon, 2013), and (2) indexing delays in databases after articles become available online, with indexing speeds ranging from weeks to months (Moed et al., 2016). Consequently, the post-2023 decline indicates that data for the most recent years have not yet stabilized, suggesting a field moving from initial euphoria toward a slower phase of empirical consolidation.

The dominance of Jupyter Notebook and Google Colab, as shown in Fig. 3, stems from their "fitness for purpose" in educational contexts. Both offer a mature computational-narrative format (text, code, and output in a single document) with low barriers to adoption. From its inception, Jupyter Notebook has been designed to support explainable, shareable computational analysis, making it ideal for learning activities that require step-by-step demonstrations (Pimentel et al., 2021; Rule et al., 2019). Meanwhile, Google Colab reinforces the Jupyter ecosystem's dominance by providing a similar notebook experience hosted in the cloud, allowing many classroom contexts to operate without the burden of software installation or environment conflicts (Vallejo et al., 2022).

The dominance of K–12/secondary and higher education contexts in Fig. 4 is driven by (a) curricular agendas and (b) the readiness of implementation ecosystems. At the K–12 level, many countries have begun positioning coding and computational thinking as core competencies across various subjects (Mills et al., 2025). In higher education, this dominance arises because Notebooks are relatively easy to integrate into STEM courses as interactive worksheets, computational labs, or assessment tools. Furthermore, university instructors and researchers typically have more stable access to infrastructure and devices, and greater freedom to design their own evaluation methods (Bascuñana et al., 2023).

Notebooks are designed for data-centric work, combining code, narrative, and output into a single interactive, easily shareable document, as illustrated in Fig. 5. This characteristic aligns perfectly with typical data science activities that require rapid iteration and analytical transparency. Consequently, studies on notebook-based learning appear most frequently in data science and statistics contexts (Pimentel et al., 2021; Samuel & Mietchen, 2024). Conversely, the lower proportion in Mathematics/Modeling suggests a long-standing tradition of established tools in mathematics education, such as Computer Algebra Systems (CAS), whose research and practice ecosystems flourished well before the advent of modern Notebooks (Marshall et al., 2012). For Chemistry, several factors contribute to the scarcity of notebook studies. Chemistry curricula rely heavily on "wet lab" practical components. Additionally, programming is not yet a conventional part of the undergraduate chemistry curriculum in many contexts, leading to slower adoption of Notebooks as a learning medium compared to data science (Vallejo et al., 2022).

4.2. Notebook Implementation in the Classroom

Notebook implementation in learning appears predominantly as a core instructional component, characterized by moderate scaffolding and robust workflows, as illustrated in Fig. 6.

The dominance of Score 4 in Role Depth and Intensity of Use (IMP1) indicates that Notebooks are a core part of the learning process, used across multiple activities or as a replacement for significant portions of core instruction. In practice, when instructors decide to adopt Notebooks, they rarely use them as a one-off tool. Instead, Notebooks are bundled into a series of activities or modules repeated over several sessions to justify their evaluation as a learning intervention. This pattern is evident in studies that design Jupyter-based modules for curriculum development (Reades, 2020) and in those that develop activity sets to support course modules and track learning progress (Bascuñana et al., 2023). Furthermore, the Notebook ecosystem "encourages" instructional designs based on recurring activity packages because they are easily structured as lesson-by-lesson units, shareable, and adaptable across topics. The prevalence of Score 4 in IMP1 signals that the majority of studies describe Notebooks as a well-established, recurring intervention within core learning.

The high frequency of Score 2 in Scaffolding Richness (IMP2) reveals a design pattern favoring minimal-to-moderate scaffolding. From the perspectives of Cognitive Load Theory and worked example research, learning computational skills is most effective when novices receive clear guidance, yet such guidance is often not "maximized" at every stage. Evidence suggests that increasing the proportion of worked-solution steps can reduce extraneous load; however, "highly guided" designs must be managed to facilitate a transition toward independent problem-solving. Thus, many educators opt for "moderate" scaffolding as a practical compromise: providing enough help for beginners while maintaining space for exploration and incremental problem-solving (Kirschner et al., 2006; Renkl & Atkinson, 2003; Schwonke et al., 2011). Educational reports often lack detailed descriptions of scaffolding; consequently, in feature-based quantization, articles may be scored as 2 even when additional teacher-led support is present in the actual classroom. Literature reviews on scaffolding highlight that contingent

scaffolding processes are frequently underdocumented in research reports, which explains why "highly rich scaffolding" categories appear rare in data extracted from published articles (Dominguez & Svihla, 2023).

Conversely, the relative scarcity of Score 5 suggests that instructional designs involving numerous integrated components are often avoided or not reported in granular detail. "Rich" scaffolding requires significant design time, pedagogical expertise, and high-level technological support. Even in programming research, studies testing fade-in or fade-out scaffolding paradigms emphasize the complexity of design, timing, and support adjustment within instructional materials (Zheng et al., 2022).

The dominance of Score 5 in Support Layering and Workflow (IMP3) demonstrates that operational support and workflow management are "mandatory" requirements for successful notebook-based learning. Since Notebooks serve as both a reading medium and an execution environment, the risk of environment errors is high without clear guidance. Therefore, authors often provide layered support packages that include execution instructions, step-by-step instructions, file links, templates, output examples, troubleshooting guides, and streamlined workflows. Good Notebook writing practices also emphasize the need to "narrate the analytical flow," ensuring the content is understandable and shareable (Rule et al., 2019).

Based on the combination of high "core use" (IMP1), high "strong workflow support" (IMP3), and moderate "scaffolding" (IMP2), implementation practices can be summarized into four major types:

1. Course-integrated Modules: Notebooks are used across multiple sessions to build computational competence incrementally. Layered support for instructions, data, workflow, and troubleshooting is provided to ensure classroom stability.
2. Interactive Lab Worksheets: Notebooks function as "computational laboratories" for running models, adjusting parameters, and interpreting output. Workflow support is explicitly defined, while cognitive scaffolding tends to vary.
3. Product-based Analysis Projects: Notebooks are used for data exploration, modeling, or mini-research. This type aligns with the Notebook's character as a shareable and reproducible analytical narrative medium.
4. Textbook-like Delivery: Notebooks are packaged as self-paced learning materials. These packages emphasize flow structure and operational support, while the depth of conceptual scaffolding varies according to the author's design.

4.3. Challenges in Notebook Usage

The dominance of narrative reporting in CHA1 indicates that many articles present technical hurdles as a "classroom reality" without necessarily documenting them as measurable data, as shown in Fig. 7. These technical obstacles include library errors, internet connectivity issues, hardware limitations, file management difficulties, or non-linear cell execution behavior. These findings align with the literature, which emphasizes that Notebooks possess a unique complexity that can disrupt both learning flow and the replication of activities (Pimentel et al., 2021; Rule et al., 2019).

The prevalence of Score 3 in CHA2 suggests that most studies report learning difficulties primarily as a narrative: heterogeneous participants (ranging from beginners to advanced) often experience cognitive load when simultaneously grasping domain concepts and syntax, compounded by time-consuming "debugging" friction. Theoretically, this condition is consistent with arguments that minimally guided instruction tends to be inefficient for novices due to limitations in working memory (Kirschner et al., 2006; Renkl & Atkinson, 2003; Schwonke et al., 2011). Consequently, this dominance signals that studies more frequently "recount" cognitive challenges rather

than measuring them in detail. In contrast, the instructional design literature emphasizes the importance of evidence on assistance levels and cognitive load when working with novice learners.

The dominance of Score 4 in CHA3 indicates that many articles explicitly address threats to validity or assessment integrity. In notebook-based assignments, plagiarism or solution similarity represents a classic risk; thus, it is common for studies to include notes on integrity, policies, or mitigation strategies (Joy & Luck, 1999; Karnalim, 2023). Furthermore, several notebook learning studies still rely on self-report measures (perceptions, satisfaction, "perceived improvement"), which are susceptible to reporting bias (Tourangeau & Yan, 2007). When studies utilize log/trace data or learning analytics, validity issues may shift toward data interpretation, ethics, and privacy, as learning data collection carries policy and moral consequences that must be explicitly stated (Rubel & Jones, 2016; Slade & Prinsloo, 2013). The more "explicit" pattern in CHA3 compared to CHA1–CHA2 arises because threats to validity/integrity are typically standard components of research reporting, while technical and cognitive problems are often relegated to the level of "implementation experience" and are not always quantified.

3.4. Impact of Notebooks on 21st-Century Skills

Figure 8 shows that the majority of studies report that outcomes were measured (Measure = Y), particularly for learning outcomes (OUT1) and computational/digital skills (OUT2). However, markers of strong statistical evidence (Stats strong = Y) appear more frequently in OUT1 and OUT2 than in Affective/Agency Outcomes (OUT3). This is consistent with the score distribution: OUT1 and OUT2 have a higher proportion of Score 5, while OUT3 more frequently plateaus at Score 4. Overall, these findings indicate that in the included literature, the impact of Notebooks on 21st-century skills is most often reported as strong to very strong for OUT1–OUT2. In contrast, for OUT3, the impact remains dominantly strong but less frequently reaches the "very strong" category according to the quantization criteria used.

The dominance of OUT1 in the Score 4–5 range can be attributed to two factors: (a) edtech evaluation patterns that prioritize "easily measurable outcomes," and (b) reporting structures that provide OUT1 with robust measures and statistical evidence. Generally, educational technology evaluation research shows that the most dominant focus is on learning outcomes, as studies are better equipped to provide quantitative instruments and reporting for these metrics (Lai & Bower, 2019). Furthermore, OUT1 tends to be more easily driven toward "strong" evidence because many studies utilize designs that yield direct numerical data, such as pre-post tests, assignment grades, or performance indicators. In contrast, other dimensions of 21st-century skills often require more complex operationalization (performance rubrics, process observation, triangulation, or longitudinal tracking); thus, while impacts are reported, "Level 5" evidence is harder to achieve consistently. When studies rely on self-reports for certain outcomes (e.g., perceived skills or confidence), the evidence is often weaker due to inherent reporting bias, leading to rarer top scores for perception-based indicators than for performance-based ones (Tourangeau & Yan, 2007).

The dominance of Scores 4–5 in OUT2 is logical, as OUT2 typically reflects impacts proximal to the Notebook's use—skills directly practiced. At the same time, students work within the environment, such as computational practices, problem-solving, and digital literacy. Consequently, many studies provide strong evidence, at least at the level of task performance, work products, or skill indicators directly tied to computational tasks. From a platform perspective, Notebooks support the "verification" of OUT2 by facilitating an explicit workflow (code + output + narrative), allowing learning activities and achievements to be documented as assessable artifacts. Evidence from best practices indicates that Notebooks are ideal for constructing readable, shareable analyses, making student work processes easily accessible for assessment and reporting (Rule et al., 2019).

The prevalence of Score 4 in OUT3 reflects affective-agency outcomes, including engagement, motivation, attitude, self-efficacy, and collaborative experience. While many studies "measure and report" these impacts, they rarely reach the "strongest" evidence level for two main reasons. First, OUT3 measurements are often dominated by self-reports or post-course feedback (questionnaires, reflections), which are practical for real-world classrooms but susceptible to context bias and social desirability. Thus, even with positive results, the evidentiary strength often stays at "moderately strong" (Score 4) rather than "very strong" (Score 5). Methodological studies suggest that social desirability can affect students' reports of motivation; therefore, claims based on self-reports should be interpreted cautiously and ideally supported by data triangulation (Lavidas et al., 2022). Second, OUT3 is a multi-dimensional "layered" construct—covering behavioral, cognitive, and affective aspects—leading to inconsistencies in operational definitions, instruments, and measurement timing (Buntins et al., 2021).

Based on the score distributions of OUT1–OUT3, three dominant impact typologies emerge:

1. **Uniformly Strong Impact (4-4-4):** This is the most frequent pattern. In this group, all three outcomes are measured, but they are rarely supported by evidence deemed "strong." This aligns with trends in edtech research that demonstrate "strong" results across dimensions in classroom implementation, but with reporting quality and instrument consistency varying significantly (Lai, 2019).
2. **Very Strong Core Outcomes (5-5-4):** Many studies achieve the highest evidence levels in OUT1 and OUT2, while OUT3 remains at the "strong" level. Substantively, this occurs because 21st-century skill frameworks place competencies on a broad spectrum. "Technical-cognitive" dimensions are easier to operationalize into performance-based assessments, whereas affective/agency dimensions often require more complex or multi-source instruments (Voogt & Pareja Roblin, 2012).
3. **Comprehensive Very Strong Impact (5-5-5):** This group represents studies that not only report high impacts across all three dimensions but also include very strong statistical evidence and measurement. Methodologically, this "5-5-5" pattern typically emerges when assessments are authentic/performance-based, utilize multiple indicators, and feature transparent evaluation reporting (Vlachopoulos & Makri, 2024).

4.5 Best Practices and Opportunities for Notebook Implementation

Based on the synthesis of the results, the most consistent best practices for implementing Notebooks in educational settings can be formulated as follows:

1. **Position Notebooks as "Core Tools"** rather than accessories. Implementation should involve recurring designs across activities and be deeply integrated into course modules. In practice, Notebooks should be structured as a clear learning path—starting with orientation and guided exercises, then progressing to performance tasks that demand student-led modification and exploration.
2. **Employ "Precise and Economical" Scaffolding.** Since scaffolding tends to be moderate in effective implementations, the focus should be on targeted support. This involves utilizing worked examples at the beginning and gradually reducing support (fading) to prevent cognitive overload for novices while still providing space for problem-solving as their competency stabilizes.
3. **Provide Layered Operational Support.** A prominent finding is that successful Notebooks are rarely just standalone code files; they are wrapped in a comprehensive support workflow. This includes step-by-step instructions, starter templates, checklists, assessment rubrics, "correct output" examples, debugging FAQs, links to supplementary resources, and clear help-seeking channels.

4. **Maintain Pedagogical Coherence to Address Cognitive Challenges.** To mitigate pedagogical-cognitive hurdles, the best practice is not necessarily to add more features, but rather to ensure internal pedagogical coherence. Every code block should have an explicit conceptual objective, accompanied by brief reflective questions that force students to link representations and explain their modeling decisions.
5. **Shift Assessment toward Process and Authenticity.** To address academic integrity and validity concerns, the most robust practice is to shift from "final answer" evaluation to process-oriented, authentic performance assessment. This includes context-based assignments, Notebook artifacts that display "thinking traces," and short reflective components justifying the choice of models.
6. **Facilitate Measurable Performance for 21st-Century Skills.** To ensure a strong impact on 21st-century skills, Notebooks must simultaneously facilitate measurable cognitive-technical performance and a "visible" learning experience. Authentic modeling or data-based tasks and performance assessments are typically most effective because they align with the broad spectrum of 21st-century competencies outlined in international frameworks.

5. Discussion

The research conducted is an effort to interpret an SLR regarding the use of Notebooks in education, focusing on three domains: implementation (IMP), challenges (CHA), and outcomes (OUT). This discussion emphasizes the significance of emerging patterns, explains their underlying mechanisms, and situates Notebooks within the broader context of learning.

5.1 Interpretation of Finding

5.1.1 Implementation (IMP): From Auxiliary Media to Learning Architecture

Notebooks are frequently a core component, consistent with the argument that they integrate narrative, code, and output into a single learning artifact. Consequently, the Notebook becomes the locus of learning activities—ranging from conceptual exploration and practice to assessment (Amoudi & Tbaishat, 2023; Temel et al., 2025). In this context, the didactic consequence is a shift from teaching with code to teaching through an executable artifact. Material is not merely read but is executed and modified by students. Studies in chemistry education, for instance, demonstrate that Notebooks can be used to strengthen conceptual understanding through interactive activities and self-assessment (Bascuñana et al., 2023).

However, the literature also cautions that establishing the Notebook as a core component demands high standards for artifact quality to ensure that learning is not derailed by technical issues or procedural confusion. Educational Notebook design principles emphasize the importance of instructions that are both machine- and human-readable, dependency management, and the reinforcement of reproducible practices (Wagemann et al., 2022). Critically, a "counter-intuitive" potential arises when Notebooks become central: while some studies report pedagogical benefits, they also reveal initial resistance, such as prejudices against programming, anxiety, and the need for intensive support during the early stages (Temel et al., 2025). This reinforces the implication that the decision to make Notebooks a core component must be accompanied by transition strategies, such as technical orientation, foundational exercises, and measures to mitigate initial barriers.

The "moderate" pattern in scaffolding richness aligns with Cognitive Load Theory: for novice learners, cognitive load can surge if tasks require exploration that is too open-ended or lacks direction. Thus, support must be provided. However, excessive support can also increase extraneous load (Sweller, 1988). Critics of minimal-guidance approaches argue that novices require sufficient guidance for efficient learning (Kirschner et al., 2006). Therefore, moderate scaffolding often serves as a pragmatic choice—providing enough structure so that beginners do not become lost, while leaving space for realistic exploration. Many modern classroom Notebook practices rely on feedback mechanisms that indirectly function as scaffolding. Nevertheless, their development requires mature test designs, rubrics, and mechanisms (González-Carrillo et al., 2021).

From a didactic perspective, the tendency toward a high "workflow as pedagogy" suggests that the learning structure is derived not only from conceptual sequencing but also from predictable, repetitive action sequences. This process helps students reduce "procedural uncertainty," allowing them to focus on problem-solving. Empirical evidence shows that Notebooks can be utilized for formative assessment by integrating various tools and packages. However, the success of these practices is heavily determined by workflow design, user readiness, and process support (Temel et al., 2025).

Interestingly, the high level of support layering/workflow appears to be a solution to the "moderate" level of scaffolding richness. When conceptual scaffolding is not exhaustive, instructors and systems often "compensate" by enriching the workflow scaffolding. Educational Notebook design literature also emphasizes that reproducibility is an inherent part of the workflow; thus, layered support (covering environment, instructions, and artifacts) is a prerequisite for the Notebook to function as a reliable learning medium (Wagemann et al., 2022). This situation confirms that the "superiority" of Notebooks in the classroom is not merely due to their interactivity, but because they enable practice-based learning through infrastructure-enabled routines.

5.1.2 Challenges (CHA): From Operational Hurdles to Learning Validity Issues

Technical challenges frequently emerge in narrative reports, aligning with the literature, which identifies environmental management as a primary source of friction in Notebook usage. Research on reproducibility indicates that Notebooks are not "automatically reproducible"; re-execution failures are often triggered by ambiguous dependencies, library versions, and environment configurations (Samuel & Mietchen, 2024). Conversely, educational Notebook design literature asserts that most technical issues can be mitigated through workflow hygiene: documenting environments, managing data paths, and maintaining habits such as "run-all" and output verification (Wagemann et al., 2022). Furthermore, Notebooks possess unique challenges that may seem minor but have significant impacts, such as non-linear cell execution and hidden states. The reproducibility literature highlights that practices such as out-of-order execution, hard-coded paths, or residual kernel states can compromise result repeatability (Samuel & Mietchen, 2024). Consequently, technical challenges encompass not only internet access or installation issues but also the computational execution mental models that students must grasp in the early stages.

Pedagogical challenges are frequently reported but not always rigorously measured, reflecting the consequences of heterogeneous prior knowledge and the Notebook's nature of combining conceptual and procedural demands within a single activity. Theoretically, this is consistent with Cognitive Load Theory (Sweller, 1988). For novices, Notebook tasks can increase extraneous load (e.g., managing errors, understanding syntax, interpreting output, and following technical instructions), thereby disrupting the focus on germane processing for the target scientific concepts. Novice learners generally require sufficient guidance to avoid becoming mired in unproductive trial-and-

error (Kirschner et al., 2006). In the context of Notebooks, "guidance" does not necessarily mean lengthy theoretical explanations; it can take the form of worked examples, code templates, step-by-step instructions, checkpoints, and formative feedback.

In notebook-based learning, assessment issues often revolve around: (i) whether the evaluation measures true understanding or merely "working code," (ii) how to ensure clean re-runs for grading, and (iii) how to minimize copy-pasting. Literature on Notebook autograding emphasizes a dual perspective: autograding accelerates feedback and scalability, but its quality depends on the evaluation design. It can foster perceptions of "unfairness" if the feedback is uninformative (González-Carrillo et al., 2021). Regarding integrity, common mitigation strategies include task variation and automated problem generation, ensuring each student receives a different version without drastically increasing the grading workload. An example of this approach is found in generative grading systems designed to reduce opportunities for cheating while maintaining efficiency (Lapeña-Mañero et al., 2022).

Technical and pedagogical challenges often conclude as "lessons learned" without supporting frequency data, activity logs, or triangulation. In the literature, this is evidenced by the dominance of perception data or implementation reflections, which—while useful for early-stage adoption—limit the precision of causal claims (Amoudi & Tbaishat, 2023; Temel et al., 2025). Large-scale reproducibility evidence provides a strong argument that technical problems are systemic patterns. When a majority of Notebooks fail to re-execute due to dependency and environment documentation issues, the need for standardized workflows, documentation, and verification becomes a methodological necessity—for both research and education (Samuel & Mietchen, 2024; Wagemann et al., 2022).

5.1.3 Outcomes (OUT): Strong Proximal Achievements, Varied Affective-Agency Impact

The pattern of strong conceptual outcomes aligns with the literature viewing the Notebook as an executable narrative. By integrating text, code, and output within a single space, students can test concepts directly and revise their understanding based on empirical evidence. In higher education contexts, replacing portions of traditional lectures with Jupyter has been reported to facilitate deeper conceptual understanding and the achievement of learning outcomes while simultaneously increasing student engagement (Amoudi & Tbaishat, 2023). On a more applied level, Notebooks are used to reinforce key concepts through interactive activities and self-assessment, which are subsequently evaluated via learning achievement indicators and student feedback. Studies indicate that such practices can improve learning and provide a positive experience (Bascuñana et al., 2023). However, the literature also signals that high conceptual outcomes typically emerge when Notebooks do not merely "present code" but actively guide scientific practice.

Computational outcomes tend to be very strong, consistent with the argument that computational thinking flourishes through the habits of formulating problems, executing procedures, verifying results, and iterating (Wing, 2008). Compelling empirical support comes from "interactive computing textbook" studies based on Jupyter: active interaction is a stronger predictor of performance than traditional "reading" metrics. This reinforces the interpretation that computational outcomes are forged through coding activities and traceable engagement—computational activities that are actually performed (Smith et al., 2021). In practice, computational outcomes are often higher when Notebooks are supported by an assessment ecosystem that enables rapid iteration and execution verification. Tools like nbgrader are built for the release–work–collect–execute–grade cycle, effectively making the workflow an inherent part of computational learning itself (Jupyter et al., 2019).

Affective outcomes tend to be strong but are less likely to reach the "strongest" level; this pattern is consistent with the nature of affective constructs. Motivation, self-efficacy, attitudes, and agency are predominantly measured through self-reporting and are highly susceptible to contextual influences (Temel et al., 2025). Here, there is a productive "counter-point" to discuss. While literature often reports more enjoyable or engaged learning experiences when Notebooks are interactive, affective effects can fluctuate depending on whether technical hurdles and debugging burdens are successfully mitigated. This implies that affective outcomes are likely mediated by the quality of implementation and the intensity of technical and pedagogical challenges.

5.2 Implications

This study contributes to the broader field of educational technology by offering a more operational framework for understanding "what makes technology work in the classroom." By treating implementation, challenges, and impacts as a cohesive object of study, this SLR shifts the focus from mere platform selection or feature sets toward a more fundamental question: whether a technology truly shapes a learning workspace that remains stable and executable for diverse learners within real-world classroom conditions. This benefit is cross-contextual, providing a lens to analyze both the successes and failures of various instructional technologies.

A further implication is the provision of a shared language—simple yet robust—to map technology adoption through three lenses: technology as the core of learning activities, the level of instructional assistance provided, and the strength of the supporting operational workflow. In practice, this language facilitates coordination among stakeholders who often speak in different "dialects," including educators, technical teams, curriculum developers, and policymakers. When an innovation falters, this framework allows for a fairer and more precise diagnosis. Often, the issue lies not in the pedagogical concept itself, but in a fragile workflow that leads to inconsistent learning experiences.

Furthermore, this study offers a realistic outlook on educational technology. The finding that impacts are most consistent in outcomes proximal to the technological activity itself provides a roadmap for program planning. Technology typically yields "quick wins" in skills directly practiced through digital routines, whereas changes in affective agency tend to require longer time horizons and a broader ecology of support. Armed with this understanding, institutions can design incremental strategies: securing proximal achievements as a foundation, then building toward affective-agency goals through richer and more sustainable learning experience designs.

Finally, this study serves as a tool for refining the evaluation of learning technologies. It encourages assessing the infrastructure of the learning experience: whether the process is repeatable, whether errors serve as productive feedback, whether operational support mitigates classroom heterogeneity, and whether impact claims are supported by sufficiently robust evidence.

5.3 Limitations

The primary limitation of this study stems from its reliance on the quality of reporting in the primary studies. Many articles describe implementation and challenges narratively but do not always provide sufficient detail to consistently assess the strength of evidence. Consequently, some emerging patterns in this synthesis may reflect "reporting richness" rather than the actual intensity of classroom phenomena.

Furthermore, while the IMP–CHA–OUT rubric enhances comparability across studies, the quantization process still involves interpretive decisions when information in the articles is ambiguous or incomplete. Simultaneously, the heterogeneity of contexts limits generalizability; findings should be read as ecological tendencies across contexts rather than uniform causal claims. Additionally, the search strategy and inclusion criteria may introduce coverage

bias—for instance, reducing the representation of field practices that are technically rich but not indexed in the targeted publication channels. As many studies do not explicitly test the implementation–challenge–impact relationship, the discussion and implications presented here should be understood as a mapping of patterns and hypothesized mechanisms that require more rigorous empirical testing in future research.

6. Conclusion

This SLR concludes that the use of Computational Notebooks in education is best understood as a reconfiguration of the learning work structure. Three emerging implementation patterns: (i) the positioning of Notebooks as a core component, (ii) a tendency toward moderate scaffolding richness, and (iii) relatively high support workflow. These patterns indicate that Notebook adoption is sustained more by operational regularity than by the intensification of conceptual assistance. In other words, Notebooks are rapidly becoming the center of classroom activity, while cognitive scaffolding enrichment proceeds more cautiously, constrained by learner heterogeneity, design costs, and instructional time.

Simultaneously, the patterns of challenges confirm that Notebooks introduce two distinct types of issues. Technical and pedagogical-cognitive challenges tend to recur as friction points but are often reported narratively. In contrast, challenges regarding assessment and integrity are more frequently stated explicitly because they directly affect the legitimacy of grading. Consequently, the most significant hurdle to adoption is that learning continuity is often insufficiently measured. This explains why workflow has become dominant: when measuring challenges is not yet robust, the most viable and widely practiced solution is to strengthen operational routines that reduce friction.

Thus, current literature presents the Notebook more as a learning work system than as a standalone pedagogical strategy. The direct implication for future research and practice is the urgent need to improve the quality of documentation and the measurement of challenges. This shift is necessary to move the academic discourse beyond what is frequently recounted toward identifying the factors that most decisively determine successful implementation in real-world classrooms.

References

1. Allen WJ, Beavers KM, Ferlanti E, Concia L, Urrutia J, Lima EABF, Fonner JM, Zuo F, Seymour HED, Kahn AB, Stubbs J, Jamthe A, Baker SN, Khan T, Carson JP (2025) A Model for Teaching Machine Learning, Deep Learning, and Research Computing to Domain Scientists on HPC Resources. *Proc. Workshops Int. Conf. High Perform. Comput., Netw., Storage, Anal., SC Workshops*, 401–408. <https://doi.org/10.1145/3731599.3767380>
2. Alzahrani N (2025) Accessible AI and HPC Education for All. *Practice and Experience in Advanced Research Computing 2025: The Power of Collaboration, PEARC '25*:1–4. <https://doi.org/10.1145/3708035.3736048>
3. Amoudi G, Tbaishat D (2023) Interactive notebooks for achieving learning outcomes in a graduate course: A pedagogical approach. *Educ Inform Technol* 1–36. <https://doi.org/10.1007/s10639-023-11854-x>
4. Angara PP, Stege U, MacLean A, Müller HA, Markham T (2022) Teaching Quantum Computing to High-School-Aged Youth: A Hands-On Approach. *IEEE Trans Quantum Eng* 3:1–15. <https://doi.org/10.1109/TQE.2021.3127503>
5. Balovsyak S, Derevyanchuk O, Kovalchuk V, Kravchenko H, Ushenko Y, Hu Z (2024) STEM Project for Vehicle Image Segmentation Using Fuzzy Logic. *Int J Mod Educ Comput Sci* 16(2):45. <https://doi.org/10.5815/ijmecs.2024.02.04>

6. Bascuñana J, León S, González-Miquel M, González EJ, Ramírez J (2023) Impact of Jupyter Notebook as a tool to enhance the learning process in chemical engineering modules. *Educ Chem Eng* 44:155–163. <https://doi.org/10.1016/j.ece.2023.06.001>
7. Betlem P, Rodes N, Cohen SM, Vander Kloet MA (2025) Jupyter Book as an open online teaching environment in the geosciences: Lessons learned from Geo-SfM and Geo-UAV. *Geoscience Communication* 8(1):51–65. <https://doi.org/10.5194/gc-8-51-2025>
8. Biehler R, Fleischer Y (2021) Introducing students to machine learning with decision trees using CODAP and Jupyter Notebooks. *Teach Stat* 43(S1):S133–S142. <https://doi.org/10.1111/test.12279>
9. Björk B-C, Solomon D (2013) The publishing delay in scholarly peer-reviewed journals. *J Informetrics* 7(4):914–923. <https://doi.org/10.1016/j.joi.2013.09.001>
10. Buntins K, Kerres M, Heinemann A (2021) A scoping review of research instruments for measuring student engagement: In need for convergence. *Int J Educational Res Open* 2:100099. <https://doi.org/10.1016/j.ijedro.2021.100099>
11. Bygstad B, Øvrelid E, Ludvigsen S, Dæhlen M (2022) From dual digitalization to digital learning space: Exploring the digital transformation of higher education. *Comput Educ* 182:104463. <https://doi.org/10.1016/j.compedu.2022.104463>
12. Cai Z, Davis RL, Mariétan R, Tormey R, Dillenbourg P (2025) Jupyter Analytics: A Toolkit for Collecting, Analyzing, and Visualizing Distributed Student Activity in Jupyter Notebooks. *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 1, SIGCSETS 2025*, 172–178. <https://doi.org/10.1145/3641554.3701971>
13. Callupe M, Fumagalli L, Nucera DD (2021), May 14 Development of a learning pilot for the remote teaching of Smart Maintenance using open source tools. *Seventh International Conference on Higher Education Advances*. Seventh International Conference on Higher Education Advances. <https://ocs.editorial.upv.es/index.php/HEAD/HEAD21/paper/view/13140>
14. Campbell EC, Christensen KM, Nuwer M, Ahuja A, Boram O, Liu J, Miller R, Osuna I, Riser SC (2025) Cracking the code: An evidence-based approach to teaching Python in an undergraduate earth science setting. *J Geosci Educ* 73(3):239–258. <https://doi.org/10.1080/10899995.2024.2384338>
15. Casebeer MD, Frano A (2025) Incorporating a research project and coding exercises into existing undergraduate physics courses. *Am J Phys* 93(9):724–729. <https://doi.org/10.1119/5.0227376>
16. Castilla R, Peña M (2023) Jupyter Notebooks for the study of advanced topics in Fluid Mechanics. *Comput Appl Eng Educ* 31(4):1001–1013. <https://doi.org/10.1002/cae.22619>
17. Chen E, Asta M (2022) Using Jupyter Tools to Design an Interactive Textbook to Guide Undergraduate Research in Materials Informatics. *J Chem Educ* 99(10):3601–3606. <https://doi.org/10.1021/acs.jchemed.2c00640>
18. Conroy E, Barr A, Harris Y, Kirk J, Olaiya E, Phillips R (2024) Real particle physics analysis by UK secondary school students using the ATLAS Open Data: An illustration through a collection of original student research. *Eur Phys J Plus* 139(9). <https://doi.org/10.1140/epjp/s13360-024-05518-z>
19. Cooke A, Smith D, Booth A (2012) Beyond PICO: The SPIDER Tool for Qualitative Evidence Synthesis. *Qual Health Res* 22(10):1435–1443. <https://doi.org/10.1177/1049732312452938>
20. De Santo A, Farah JC, Martínez ML, Moro A, Bergram K, Purohit AK, Felber P, Gillet D, Holzer A (2022) Promoting Computational Thinking Skills in Non-Computer-Science Students: Gamifying Computational Notebooks to Increase Student Engagement. *IEEE Trans Learn Technol* 15(3):392–405. <https://doi.org/10.1109/TLT.2022.3180588>

21. Domínguez JC, Alonso MV, González EJ, Guijarro MI, Miranda R, Oliet M, Rigual V, Toledo JM, Villar-Chavero MM, Yustos P (2021) Teaching chemical engineering using Jupyter notebook: Problem generators and lecturing tools. *Educ Chem Eng* 37:1–10. <https://doi.org/10.1016/j.ece.2021.06.004>
22. Dominguez S, Svihla V (2023) A review of teacher implemented scaffolding in K-12. *Social Sci Humanit Open* 8(1):100613. <https://doi.org/10.1016/j.ssaho.2023.100613>
23. Elhayany M, Meinel C (2023) Towards Automated Code Assessment with OpenJupyter in MOOCs. *Proceedings of the Tenth ACM Conference on Learning @ Scale, L@S '23*, 321–325. <https://doi.org/10.1145/3573051.3596180>
24. Elshall AS, Badir A (2025) Balancing AI-assisted learning and traditional assessment: The FACT assessment in environmental data science education. *Frontiers in Education*, 10. <https://doi.org/10.3389/feduc.2025.1596462>
25. Fleischer Y, Biehler R, Schulte C, DATA-DRIVEN, MACHINE LEARNING WITH EDUCATIONALLY DESIGNED JUPYTER NOTEBOOKS (2022) *Stat Educ Res J*, 21(2). <https://doi.org/10.52041/serj.v21i2.61>
26. Foster J, Wagner J (2021) Naive Bayes versus BERT: Jupyter notebook assignments for an introductory NLP course. *Teach. NLP - Proc. Workshop Teach. Nat. Lang. Process.*, 112–114. <https://doi.org/10.18653/v1/2021.teachingnlp-1.20>
27. Fransson T, Delcey MG, Brumboiu IE, Hodecker M, Li X, Rinkevicius Z, Dreuw A, Rhee YM, Norman P (2023) eChem: A Notebook Exploration of Quantum Chemistry. *J Chem Educ* 100(4):1664–1671. <https://doi.org/10.1021/acs.jchemed.2c01103>
28. González-Carrillo CD, Restrepo-Calle F, Ramírez-Echeverry JJ, González FA (2021) Automatic Grading Tool for Jupyter Notebooks in Artificial Intelligence Courses. *Sustainability* 13(21):12050. <https://doi.org/10.3390/su132112050>
29. Google (2025) *Google Colab*. https://research.google.com/colaboratory/faq.html?utm_source=chatgpt.com
30. Goswami L, Senges A, Estier T, Cherubini M (2023) Supporting Co-Regulation and Motivation in Learning Programming in Online Classrooms. *Proc. ACM Hum.-Comput. Interact.*, 7(CSCW2), 298:1-298:29. <https://doi.org/10.1145/3610089>
31. Grazioli G, Ingwersen A, Santiago D Jr., Regan P, Cho H (2023) Foregrounding the Code: Computational Chemistry Instructional Activities Using a Highly Readable Fluid Simulation Code. *J Chem Educ* 100(3):1155–1163. <https://doi.org/10.1021/acs.jchemed.2c00838>
32. Gupta YM, Kirana SN, Homchan S, Tanasarnpaiboon S (2023) Teaching Python programming for bioinformatics with Jupyter notebook in the Post-COVID-19 era. *Biochem Mol Biol Educ* 51(5):537–539. <https://doi.org/10.1002/bmb.21746>
33. Hall WP, Cantrell K (2024) Exploring the Connection between Atmospheric Carbon Dioxide and Ocean Acidification through a Python Coding Exercise. *J Chem Educ* 101(9):3922–3927. <https://doi.org/10.1021/acs.jchemed.4c00462>
34. Heredia-Negron F, Alamo-Rodriguez N, Oyola-Velazquez L, Nieves B, Carrasquillo K, Hochheiser H, Fristensky B, Daluz-Santana I, Fernandez-Repollet E, Roche-Lima A (2023) Evaluation of AIML + HDR—A Course to Enhance Data Science Workforce Capacity for Hispanic Biomedical Researchers. *Int J Environ Res Public Health* 20(3). <https://doi.org/10.3390/ijerph20032726>
35. Ho L, McErlean M, You Z, Blank D, Meeden L (2025) AI Toolkit: Libraries and Essays for Exploring the Technology and Ethics of AI. *Proc. AAAI Conf. Artif. Intell.*, 39(28), 29013–29018. <https://doi.org/10.1609/aaai.v39i28.35171>

36. Johnson JW (2020) Benefits and Pitfalls of Jupyter Notebooks in the Classroom. *Proceedings of the 21st Annual Conference on Information Technology Education, SIGITE '20*, 32–37.
<https://doi.org/10.1145/3368308.3415397>
37. Joy M, Luck M (1999) Plagiarism in programming assignments. *IEEE Trans Educ* 42(2):129–133.
<https://doi.org/10.1109/13.762946>
38. Jupyter P, Blank D, Bourgin D, Brown A, Bussonnier M, Frederic J, Granger B, Griffiths TL, Hamrick J, Kelley K, Pacer M, Page L, Pérez F, Ragan-Kelley B, Suchow JW, Willing C (2019) nbgrader: A Tool for Creating and Grading Assignments in the Jupyter Notebook. *J Open Source Educ* 2(16):32.
<https://doi.org/10.21105/jose.00032>
39. Karnalim O (2023) Maintaining Academic Integrity in Programming: Locality-Sensitive Hashing and Recommendations. *Educ Sci* 13(1):54. <https://doi.org/10.3390/educsci13010054>
40. Kayhan V, Berndt D (2023) Navigating Workload Compatibility Between a Recommender System and a NoSQL Database: An Interactive Tutorial. *Commun Association Inform Syst* 53(1):667–681.
<https://doi.org/10.17705/1CAIS.05327>
41. Kim B, Henke G (2021) Easy-to-Use Cloud Computing for Teaching Data Science. *J Stat Data Sci Educ* 29(sup1):S103–S111. <https://doi.org/10.1080/10691898.2020.1860726>
42. King S, Sharifi Far S (2024) Teaching Data Science to Diverse Learners: A Hybrid and Interdisciplinary Approach. *Teaching Statistics*, *n/a(n/a)*. <https://doi.org/10.1111/test.70014>
43. Kirschner PA, Sweller J, Clark RE (2006) Why Minimal Guidance During Instruction Does Not Work: An Analysis of the Failure of Constructivist, Discovery, Problem-Based, Experiential, and Inquiry-Based Teaching. *Educational Psychol* 41(2):75–86. https://doi.org/10.1207/s15326985ep4102_1
44. Kluyver T, Ragan-Kelley B, Perez F, Granger B, Bussonnier M, Frederic J, Kelley K, Hamrick J, Grout J, Corlay S, Ivanov P, Avila D, Abdalla S, Willing C, Jupyter Development Team (2016) Jupyter Notebooks: A publishing format for reproducible computational workflows. *Positioning and Power in Academic Publishing: Players, Agents and Agendas*. IOS. <https://doi.org/10.3233/978-1-61499-649-1-87>
45. Knuth DE (1984) Literate Programming. *Comput J* 27(2):97–111. <https://doi.org/10.1093/comjnl/27.2.97>
46. Kozakai R, Kobayashi T, Wenxuan Z, Watanabe Y (2022) Tendency Analysis of Python Programming Classes for Junior and Senior High School Students. *Procedia Comput Sci* 207:4603–4612.
<https://doi.org/10.1016/j.procs.2022.09.524>
47. Krugh M, Mears L (2021) Pervasive environmental sensing for Industry 4.0 as an educational tool. *Procedia Manufacturing, 49th SME North American Manufacturing Research Conference (NAMRC 49, 2021)*, 53, 790–801. <https://doi.org/10.1016/j.promfg.2021.06.086>
48. Kumwiche P (2023) for Graduate Students in Medical Fields With Jupyter Notebook: Classroom Action Research. *JMIR Med Educ* 9(1):e47394. <https://doi.org/10.2196/47394>. Enhancing Learning About Epidemiological Data Analysis Using R
49. Lai JWM, Bower M (2019) How is the use of technology in education evaluated? A systematic review. *Comput Educ* 133:27–42. <https://doi.org/10.1016/j.compedu.2019.01.010>
50. Laky DJ, Casas-Orozco D, Abdi M, Feng X, Wood E, Reklaitis GV, Nagy ZK (2023) Using PharmaPy with Jupyter Notebook to teach digital design in pharmaceutical manufacturing. *Comput Appl Eng Educ* 31(6):1662–1677.
<https://doi.org/10.1002/cae.22660>
51. Lane B, Garousi-Nejad I, Gallagher MA, Tarboton DG, Habib E (2021) An open web-based module developed to advance data-driven hydrologic process learning. *Hydrol Process* 35(7):e14273.

<https://doi.org/10.1002/hyp.14273>

52. Lane WB, Galanti TM, Rozas XL (2023) Teacher Re-novicing on the Path to Integrating Computational Thinking in High School Physics Instruction. *J STEM Educ Res* 6(2):302–325. <https://doi.org/10.1007/s41979-023-00100-1>
53. Lapeña-Mañero P, García-Casuso C, Montenegro-Cooper JM, King RW, Behrens EM (2022) An Open-Source System for Generating and Computer Grading Traditional Non-Coding Assignments. *Electronics* 11(6). <https://doi.org/10.3390/electronics11060917>
54. Lavidas K, Papadakis S, Manesis D, Grigoriadou AS, Gialamas V (2022) The Effects of Social Desirability on Students' Self-Reports in Two Social Contexts: Lectures vs. Lectures Lab Classes *Information* 13(10):491. <https://doi.org/10.3390/info13100491>
55. Lee I, Perret B (2022) Preparing High School Teachers to Integrate AI Methods into STEM Classrooms. *Proc. AAAI Conf. Artif. Intell., AAAI*, 36, 12783–12791. <https://doi.org/10.1609/aaai.v36i11.21557>
56. Liebal UW, Schimassek R, Broderius I, Maaßen N, Vogelgesang A, Weyers P, Blank LM (2023) Biotechnology Data Analysis Training with Jupyter Notebooks. *J Microbiol Biology Educ* 24(1):e00113–e00122. <https://doi.org/10.1128/jmbe.00113-22>
57. Llerena-Izquierdo J, Mendez-Reyes J, Ayala-Carabajo R, Andrade-Martinez C (2024) Innovations in Introductory Programming Education: The Role of AI with Google Colab and Gemini. *Educ Sci* 14(12). <https://doi.org/10.3390/educsci14121330>
58. Lo D, Shahriar H, Qian K, Whitman M, Wu F, Thomas C (2023) Authentic Learning on Machine Learning for Cybersecurity. *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 2, SIGCSE 2023*, 1299. <https://doi.org/10.1145/3545947.3576245>
59. Lonsky M, Lang M, Holt S, Pathak SA, Klause R, Lo T-H, Beg M, Hoffmann A, Fangohr H (2024) Numerical simulation projects in micromagnetics with Jupyter. *Am J Phys* 92(10):794–800. <https://doi.org/10.1119/5.0149038>
60. Lyu Z, Ali S, Breazeal C (2022) Introducing Variational Autoencoders to High School Students. *Proc. AAAI Conf. Artif. Intell., AAAI*, 36, 12801–12809. <https://doi.org/10.1609/aaai.v36i11.21559>
61. Marshall N, Buteau C, Jarvis DH, Lavicza Z (2012) Do mathematicians integrate computer algebra systems in university teaching? Comparing a literature review to an international survey study. *Comput Educ* 58(1):423–434. <https://doi.org/10.1016/j.compedu.2011.08.020>
62. Memarian B, Doleck T (2023) ChatGPT in education: Methods, potentials, and limitations. *Computers Hum Behavior: Artif Hum* 1(2):100022. <https://doi.org/10.1016/j.chbah.2023.100022>
63. Mills KA, Cope J, Scholes L, Rowe L (2025) Coding and Computational Thinking Across the Curriculum: A Review of Educational Outcomes. *Rev Educ Res* 95(3):581–618. <https://doi.org/10.3102/00346543241241327>
64. Moed HF, Bar-Ilan J, Halevi G (2016) A new methodology for comparing Google Scholar and Scopus. *J Informetrics* 10(2):533–551. <https://doi.org/10.1016/j.joi.2016.04.017>
65. Nwulu NI, Damisa U, Gbadamosi SL (2021) Students Perception about the Use of Jupyter Notebook in Power Systems Education. *Int J Eng Pedagogy (iJEP)* 11(1):78–86. <https://doi.org/10.3991/ijep.v11i1.14769>
66. Odden TOB (2019) Physics computational literacy: An exploratory case study using computational essays. *Phys Rev Phys Educ Res* 15(2). <https://doi.org/10.1103/PhysRevPhysEducRes.15.020152>
67. Odden TOB, Malthé-Sørenssen A (2020) Using computational essays to scaffold professional physics practice. *Eur J Phys* 42(1):015701. <https://doi.org/10.1088/1361-6404/abb8b7>

68. Osório NS, Garma LD (2025) Teaching Python with team-based learning: Using cloud-based notebooks for interactive coding education. *FEBS Open Bio* 15(12):2054–2066. <https://doi.org/10.1002/2211-5463.70097>
69. Page MJ, McKenzie JE, Bossuyt PM, Boutron I, Hoffmann TC, Mulrow CD, Shamseer L, Tetzlaff JM, Akl EA, Brennan SE, Chou R, Glanville J, Grimshaw JM, Hróbjartsson A, Lalu MM, Li T, Loder EW, Mayo-Wilson E, McDonald S, Moher D (2021) *The PRISMA 2020 statement: An updated guideline for reporting systematic reviews*. <https://doi.org/10.1136/bmj.n71>
70. Page MJ, Moher D, Bossuyt PM, Boutron I, Hoffmann TC, Mulrow CD, Shamseer L, Tetzlaff JM, Akl EA, Brennan SE, Chou R, Glanville J, Grimshaw JM, Hróbjartsson A, Lalu MM, Li T, Loder EW, Mayo-Wilson E, McDonald S, McKenzie JE (2021) *PRISMA 2020 explanation and elaboration: Updated guidance and exemplars for reporting systematic reviews*. <https://doi.org/10.1136/bmj.n160>
71. Perez F, Granger BE (2007) IPython: A System for Interactive Scientific Computing. *Comput Sci Engg* 9(3):21–29. <https://doi.org/10.1109/MCSE.2007.53>
72. Pimentel JF, Murta L, Braganholo V, Freire J (2021) Understanding and improving the quality and reproducibility of Jupyter notebooks. *Empir Softw Eng* 26(4):65. <https://doi.org/10.1007/s10664-021-09961-9>
73. Podworny S, Hüsing S, Schulte C, DATA SCIENCE INTRODUCTION IN SCHOOL: BETWEEN STATISTICS AND PROGRAMMING (2022) *Stat Educ Res J* 21(2). <https://doi.org/10.52041/serj.v21i2.46>. A PLACE FOR A
74. Rainey MA, Benda MC, Mayberry KA, Smeekens JM, Braga RA, Bottomley LA, O'Mahony CM (2024) Data Science Meets Mineral Analysis: An Innovative Laser-Induced Breakdown Spectroscopy Experiment for Undergraduate Chemistry Students. *J Chem Educ* 101(7):2869–2879. <https://doi.org/10.1021/acs.jchemed.4c00421>
75. Reades J (2020) Teaching on Jupyter: Using notebooks to accelerate learning and curriculum development. *REGION* 7(3):21–34. <https://doi.org/10.18335/region.v7i1.282>
76. Renkl A, Atkinson RK (2003) Structuring the Transition From Example Study to Problem Solving in Cognitive Skill Acquisition: A Cognitive Load Perspective. *Educational Psychol* 38(1):15–22. https://doi.org/10.1207/S15326985EP3801_3
77. Resendez SD, Franklin G, Tomlin C, Stephens R, Maness H, Chamala S, Koppel R, Elkin PL (2025) Surveying the Efficacy of an Open Access Biomedical Informatics Boot Camp. *Appl Clin Inf* 16:583–588. <https://doi.org/10.1055/a-2547-5208>
78. Rethlefsen ML, Kirtley S, Waffenschmidt S, Ayala AP, Moher D, Page MJ, Koffel JB, Blunt H, Brigham T, Chang S, Clark J, Conway A, Couban R, de Kock S, Farrah K, Fehrmann P, Foster M, Fowler SA, Glanville J, PRISMA-S Group (2021) PRISMA-S: An extension to the PRISMA Statement for Reporting Literature Searches in Systematic Reviews. *Syst Reviews* 10(1):39. <https://doi.org/10.1186/s13643-020-01542-z>
79. Roundy JK, Gallagher MA, Byrd JL (2022) An innovative active learning module on snow and climate modeling. *Front Water*. <https://doi.org/10.3389/frwa.2022.912776>
80. Rowe PM, Fortmann L, Guasco TL, Wright A, Ryken A, Sevier E, Stokes G, Mifflin A, Wade R, Cheng H, Pfalzgraff W, Beaudoin J, Rajbhandari I, Fox-Dobbs K, Neshyba S (2021) Integrating polar research into undergraduate curricula using computational guided inquiry. *J Geosci Educ* 69(2):178–191. <https://doi.org/10.1080/10899995.2020.1768004>
81. Rubel A, Jones KML (2016) Student privacy in learning analytics: An information ethics perspective. *Inform Soc* 32(2):143–159. <https://doi.org/10.1080/01972243.2016.1130502>
82. Ruiz-Sarmiento J-R, Baltanas S-F, Gonzalez-Jimenez J (2021) Jupyter Notebooks in Undergraduate Mobile Robotics Courses: Educational Tool and Case Study. *Appl Sci* 11(3):917. <https://doi.org/10.3390/app11030917>

83. Rule A, Birmingham A, Zuniga C, Altintas I, Huang S-C, Knight R, Moshiri N, Nguyen MH, Rosenthal SB, Pérez F, Rose PW (2019) Ten simple rules for writing and sharing computational analyses in Jupyter Notebooks. *PLoS Comput Biol* 15(7):e1007007. <https://doi.org/10.1371/journal.pcbi.1007007>
84. Samuel S, Mietchen D (2024) Computational reproducibility of Jupyter notebooks from biomedical publications. *GigaScience*, 13, giad113. <https://doi.org/10.1093/gigascience/giad113>
85. Sánchez-Peña M, Vieira C, Magana AJ (2023) Data science knowledge integration: Affordances of a computational cognitive apprenticeship on student conceptual understanding. *Comput Appl Eng Educ* 31(2):239–259. <https://doi.org/10.1002/cae.22580>
86. Santos E, Collaboration PA (2025) Auger Open Data and the Pierre Auger Observatory International Masterclasses. *Journal of Physics: Conference Series*, 3053(1), 012040. <https://doi.org/10.1088/1742-6596/3053/1/012040>
87. Schwonke R, Renkl A, Salden R, Aleven V (2011) Effects of different ratios of worked solution steps and problem solving opportunities on cognitive load and learning outcomes. *Computers Hum Behav Curr Res Top Cogn Load Theory* 27(1):58–62. <https://doi.org/10.1016/j.chb.2010.03.037>
88. Seddighi M, Allanson D, Rothwell G, Takroui K (2020) Study on the use of a combination of IPython Notebook and an industry-standard package in educating a CFD course. *Comput Appl Eng Educ* 28(4):952–964. <https://doi.org/10.1002/cae.22273>
89. Seebut S, Wongsason P, Kim D (2024) Combining GPT and Colab as learning tools for students to explore the numerical solutions of difference equations. *Eurasia J Math Sci Technol Educ* 20(1). <https://doi.org/10.29333/ejmste/13905>
90. Seth A, Redonnet S, Liem RP (2023) MADE: A Multidisciplinary Computational Framework for Aerospace Engineering Education. *IEEE Trans Educ* 66(6):622–631. <https://doi.org/10.1109/TE.2023.3281825>
91. Slade S, Prinsloo P (2013) Learning Analytics: Ethical Issues and Dilemmas. *Am Behav Sci* 57(10):1510–1529. <https://doi.org/10.1177/0002764213479366>
92. Smith DH, Hao Q, Hundhausen CD, Jagodzinski F, Myers-Dean J, Jaeger K (2021) Towards Modeling Student Engagement with Interactive Computing Textbooks: An Empirical Study. *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education, SIGCSE '21*, 914–920. <https://doi.org/10.1145/3408877.3432361>
93. Spencer-Tyree B, Bowen BD, Olaguro M (2024) The Impact of Computational Labs on Conceptual and Contextual Understanding in a Business Calculus Course. *Int J Res Undergrad Math Educ*. <https://doi.org/10.1007/s40753-024-00255-1>
94. Sugiarto S, Lekitoo JN, Ma K, R (2024) PYTHON IN ORDINARY DIFFERENTIAL EQUATIONS LEARNING. *Barekeng* 18(4):2531–2542. <https://doi.org/10.30598/barekengvol18iss4pp2531-2542>
95. Sweller J (1988) Cognitive Load During Problem Solving: Effects on Learning. *Cogn Sci* 12(2):257–285. https://doi.org/10.1207/s15516709cog1202_4
96. Sytnykova Y, Kyrpenko V, Palevych S, Pochuieva O, Lamtiuhova S, Chaika O (2025) Implementation of Professionally Oriented Tasks with Interactive Cloud Environment Google Colab. *Int J Interact Mob Technol* 19(9):73–91. <https://doi.org/10.3991/ijim.v19i09.53459>
97. Tang C (2021) Computer-aided Linear Algebra Course on Jupyter-Python Notebook for Engineering Undergraduates. *J Phys Conf Ser* 1815(1). <https://doi.org/10.1088/1742-6596/1815/1/012004>
98. Temel GY, Barentien J, Padubrin T (2025) Using Jupyter Notebooks as digital assessment tools: An empirical examination of student teachers' attitudes and skills towards digital assessment. *Educ Inform Technol*

- 30(13):18621–18650. <https://doi.org/10.1007/s10639-025-13507-7>
99. Tourangeau R, Yan T (2007) Sensitive questions in surveys. *Psychol Bull* 133(5):859–883. <https://doi.org/10.1037/0033-2909.133.5.859>
100. Tranfield D, Denyer D, Smart P (2003) Towards a Methodology for Developing Evidence-Informed Management Knowledge by Means of Systematic Review. *Br J Manag* 14(3):207–222. <https://doi.org/10.1111/1467-8551.00375>
101. Truong V, Moore JE, Ricoy UM, Verpeut JL (2024) Low-Cost Approaches in Neuroscience to Teach Machine Learning Using a Cockroach Model. *eNeuro* 11(12). <https://doi.org/10.1523/ENEURO.0173-24.2024>
102. Tufino E, Oss S, Alemani M (2024) Integrating Python data analysis in an existing introductory laboratory course. *Eur J Phys* 45(4):045707. <https://doi.org/10.1088/1361-6404/ad4fcc>
103. Tufino E, Oss S, Alemani M (2025) Using Jupyter Notebooks to foster computational skills and professional practice in an introductory physics lab course. *Journal of Physics: Conference Series*, 2950(1), 012022. <https://doi.org/10.1088/1742-6596/2950/1/012022>
104. Vallejo W, Díaz-Urbe C, Fajardo C (2022) Google Colab and Virtual Simulations: Practical e-Learning Tools to Support the Teaching of Thermodynamics and to Introduce Coding to Students. *ACS Omega* 7(8):7421–7429. <https://doi.org/10.1021/acsomega.2c00362>
105. Vanegas-Guillén O, Parra-Rosero P, Muñoz-Antón JM, Zumba-Gamboa J, Dillon C (2023) Remote Labs Meet Computational Notebooks: An Architecture for Simplifying the Workflow of Remote Educational Experiments. *IEEE Access* 11:132496–132515. <https://doi.org/10.1109/ACCESS.2023.3336287>
106. Vidal-Silva C, Barriga NA, Ortega-Cordero F, González-López J, Jiménez-Quintana C, Pezoa-Fuentes C, Veas-González I (2022) Developing Computing Competencies Without Restrictions. *IEEE Access* 10:106568–106580. <https://doi.org/10.1109/ACCESS.2022.3211973>
107. Vladis NA, Coleman BI (2021) Moving a Flipped Class Online To Teach Python to Biomedical Ph.D. Students during COVID-19 and Beyond. *J Microbiol Biology Educ* 22(2). 10.1128. /jmbe.00099 – 21
108. Voogt J, Roblin NP (2012) A comparative analysis of international frameworks for 21st century competences: Implications for national curriculum policies. *J Curriculum Stud* 44(3):299–321. <https://doi.org/10.1080/00220272.2012.668938>
109. Wagemann J, Fierli F, Mantovani S, Siemen S, Seeger B, Bendix J (2022) Five Guiding Principles to Make Jupyter Notebooks Fit for Earth Observation Data Education. *Remote Sens* 14(14). <https://doi.org/10.3390/rs14143359>
110. Wang Y, Li M, Wang X-S, Gildersleeve A, Turki N (2023) ATRP Kinetic Simulator: An Online Open Resource Educational Tool Using Jupyter Notebook and Google Colaboratory. *J Chem Educ* 100(7):2770–2775. <https://doi.org/10.1021/acs.jchemed.2c01250>
111. Wen Z, Pacherkar HS, Yan G (2022) VET5G: A Virtual End-to-End Testbed for 5G Network Security Experimentation. *Proceedings of the 15th Workshop on Cyber Security Experimentation and Test, CSET '22*, 19–29. <https://doi.org/10.1145/3546096.3546111>
112. Werth A, Oliver KA, West CG, Lewandowski HJ (2022) *Engagement in collaboration and teamwork using Google Colaboratory*. 481–487. <https://www.per-central.org/items/detail.cfm?ID=16280>
113. Wing JM (2006) Computational thinking. *Commun ACM*. <https://doi.org/10.1145/1118178.1118215>
114. Wing JM (2008) Computational thinking and thinking about computing. *Philosophical Trans Royal Soc A: Math Phys Eng Sci* 366(1881):3717–3725. <https://doi.org/10.1098/rsta.2008.0118>

115. Xiao T, Greenberg RI, Albert MV (2021) Design and Assessment of a Task-Driven Introductory Data Science Course Taught Concurrently in Multiple Languages: Python, R, and MATLAB. *Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education V. 1, ITiCSE '21*, 290–295.
<https://doi.org/10.1145/3430665.3456364>
116. Zabasta A, Kazymyr V, Drozd O, Verslype S, Espeel L, Bruzgiene R (2024) Development of Shared Modeling and Simulation Environment for Sustainable e-Learning in the STEM Field. *Sustainability* 16(5).
<https://doi.org/10.3390/su16052197>
117. Zhang Z, Gautam A, Lim S-M, Hilty C (2023) Analysis of Large Data Sets in a Physical Chemistry Laboratory NMR Experiment Using Python. *J Chem Educ* 100(10):4109–4113.
<https://doi.org/10.1021/acs.jchemed.3c00586>
118. Zhao L, Shin J, Kim IL, Song C, Kabuo C, Joseph J, Merwade V, Hosen J, Rajib A, Huang W (2025) Developing an Interactive Online Platform for Advanced Cyber Training and Adaptive Learning Paths. *Practice and Experience in Advanced Research Computing 2025: The Power of Collaboration, PEARC '25*, 1–5.
<https://doi.org/10.1145/3708035.3736078>
119. Zheng L, Zhen Y, Niu J, Zhong L (2022) An exploratory study on fade-in versus fade-out scaffolding for novice programmers in online collaborative programming settings. *J Comput High Educ* 34(2):489–516.
<https://doi.org/10.1007/s12528-021-09307-w>

Figures

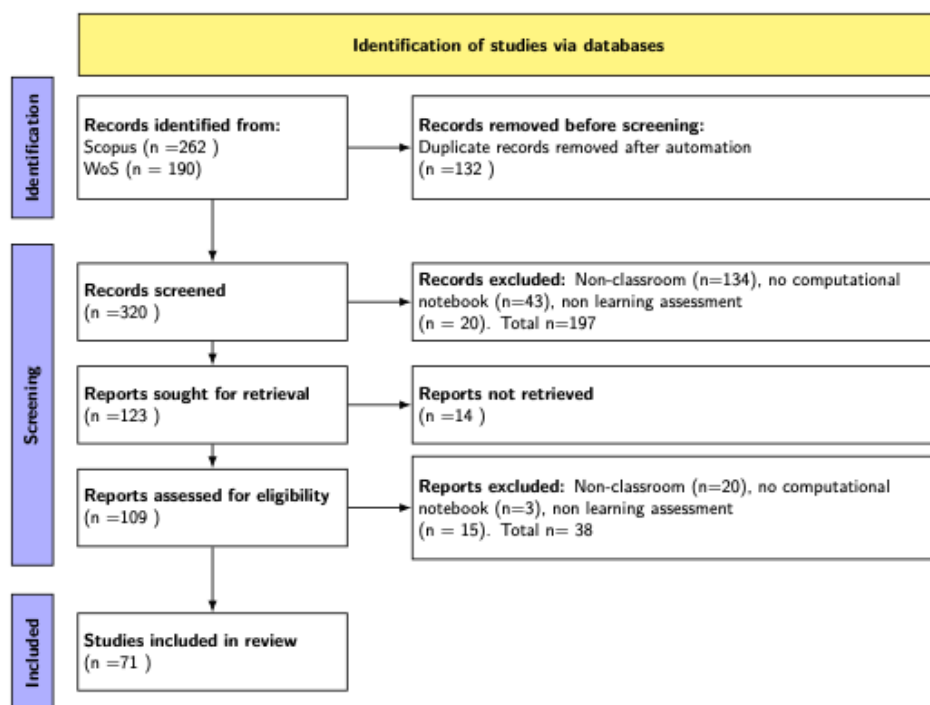


Figure 1

PRISMA diagram of Notebook implementation in the classrooms.

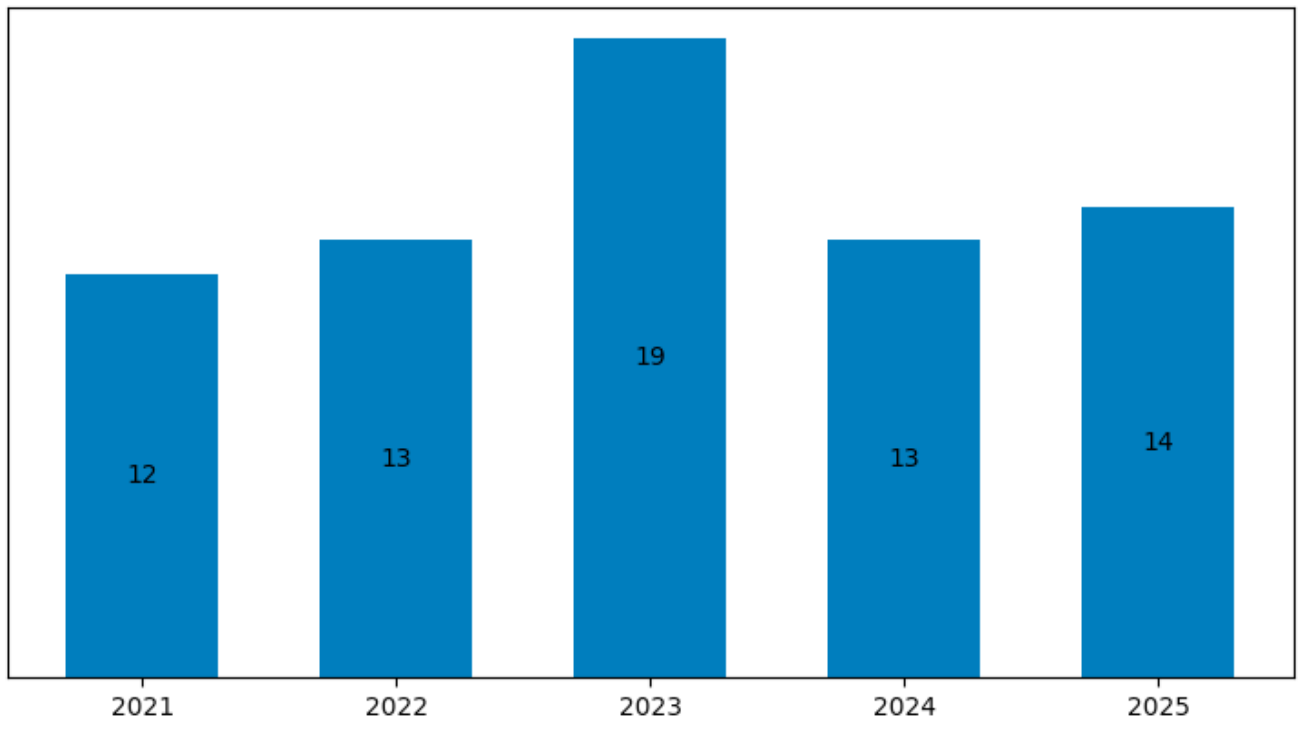


Figure 2
Number of publications 2021–2025

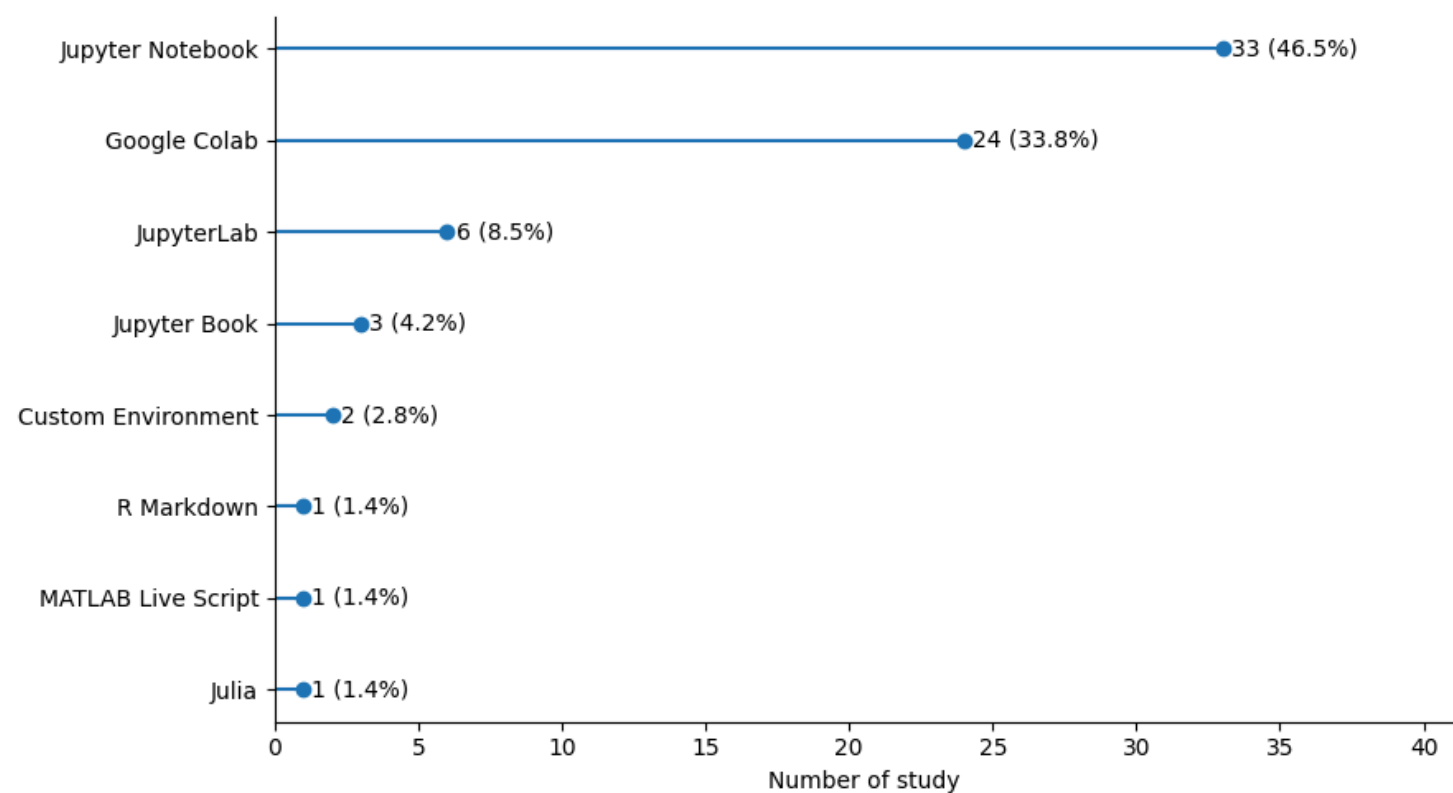


Figure 3
Platforms utilized

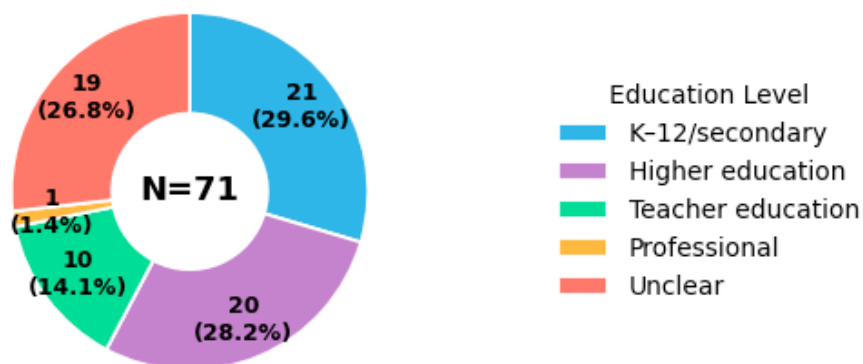


Figure 4

Distribution of educational levels

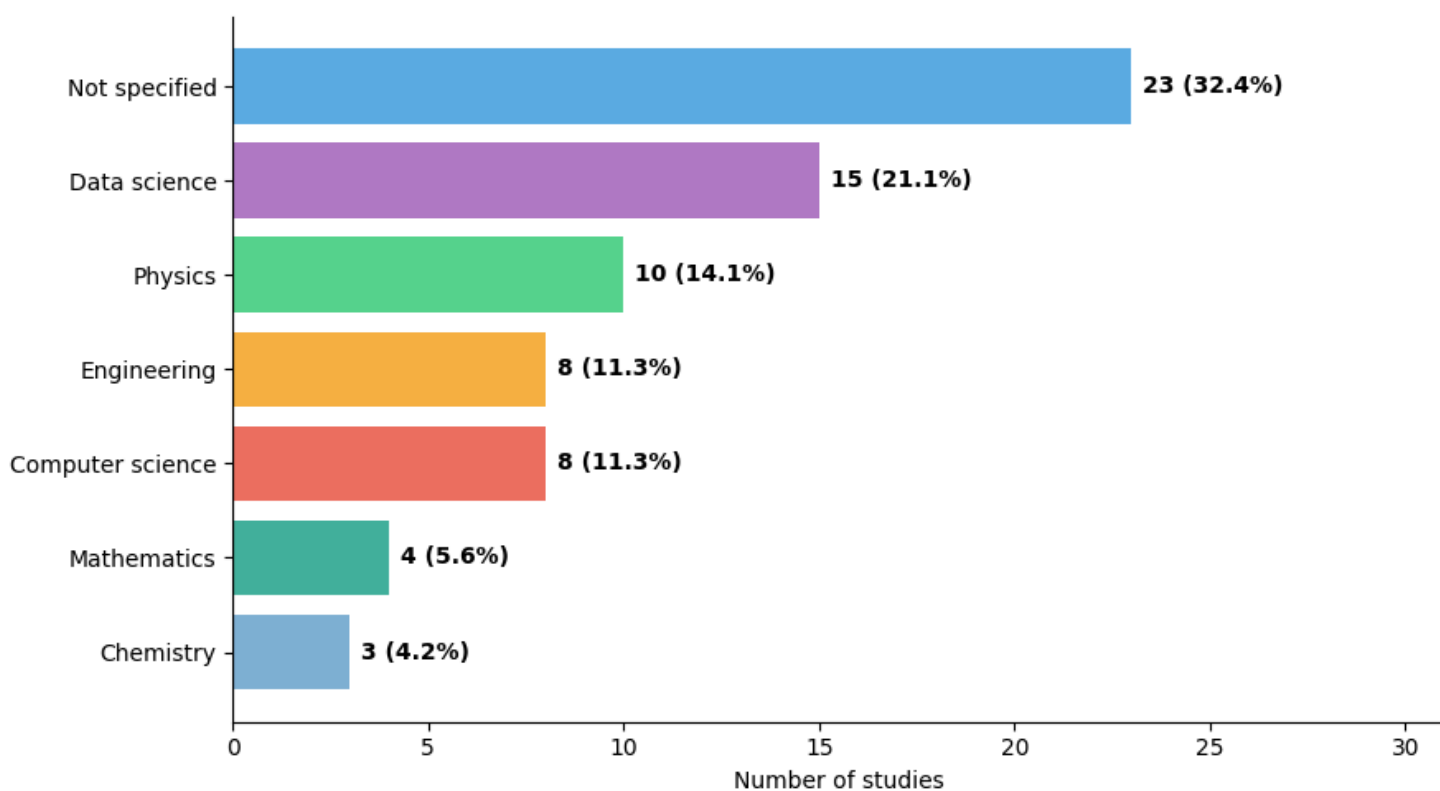


Figure 5

Notebook Topics

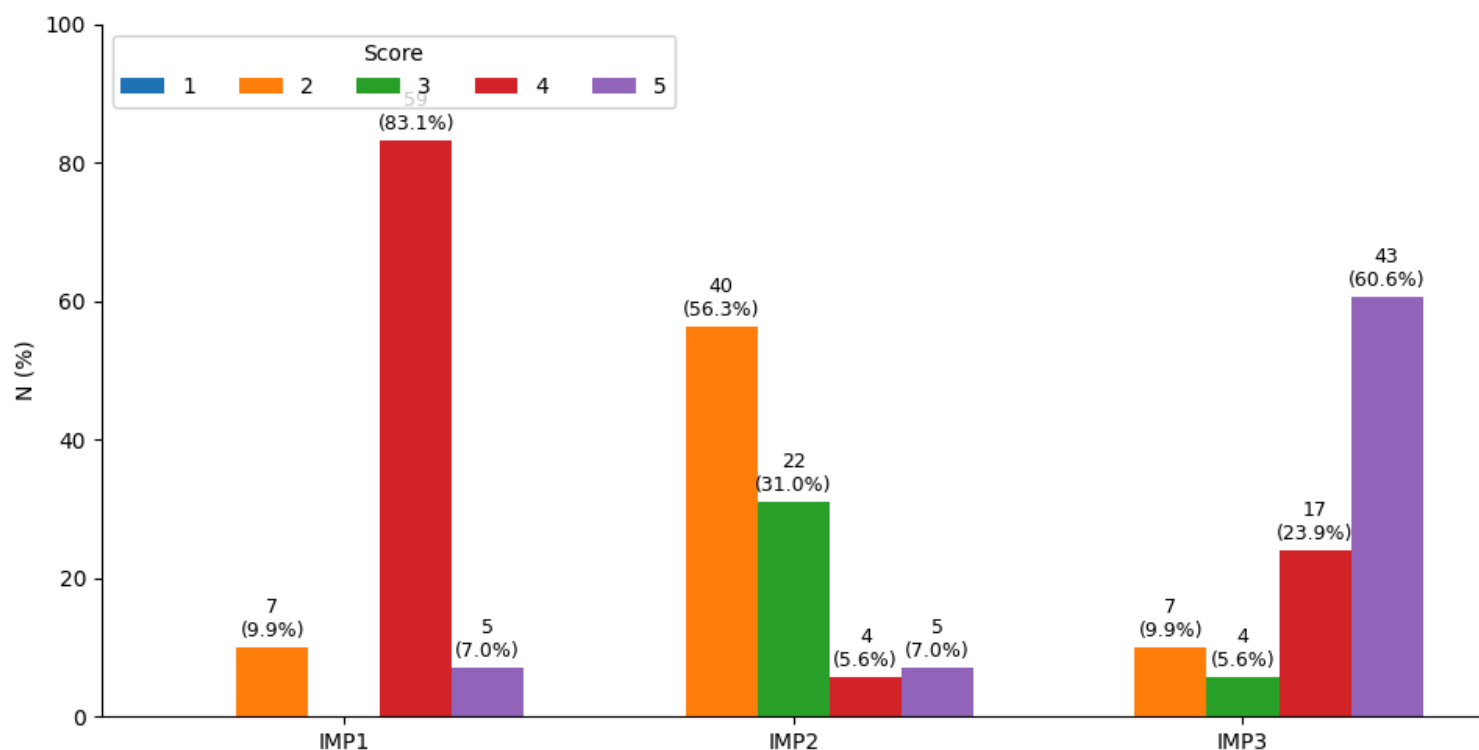


Figure 6

Notebook Implementation

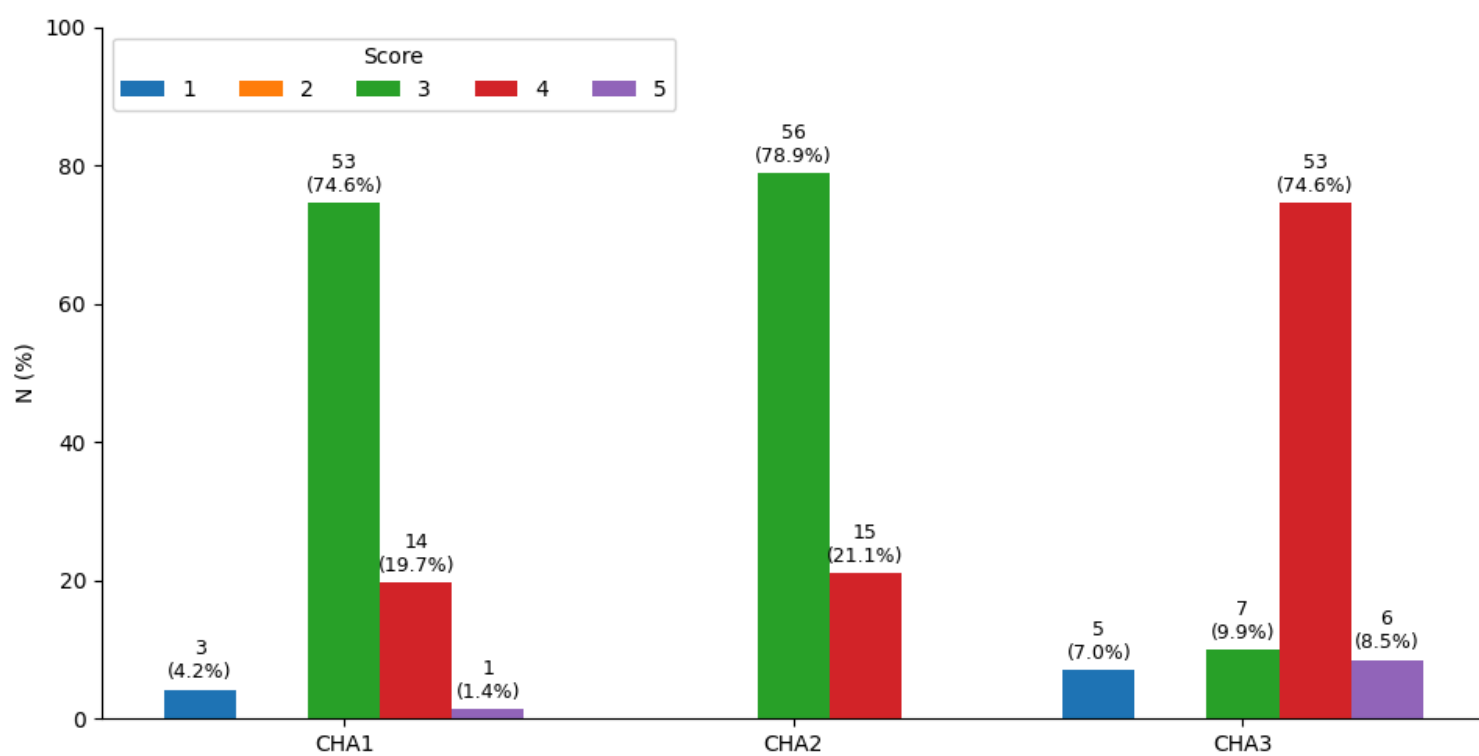


Figure 7

Challenges in Notebook Usage

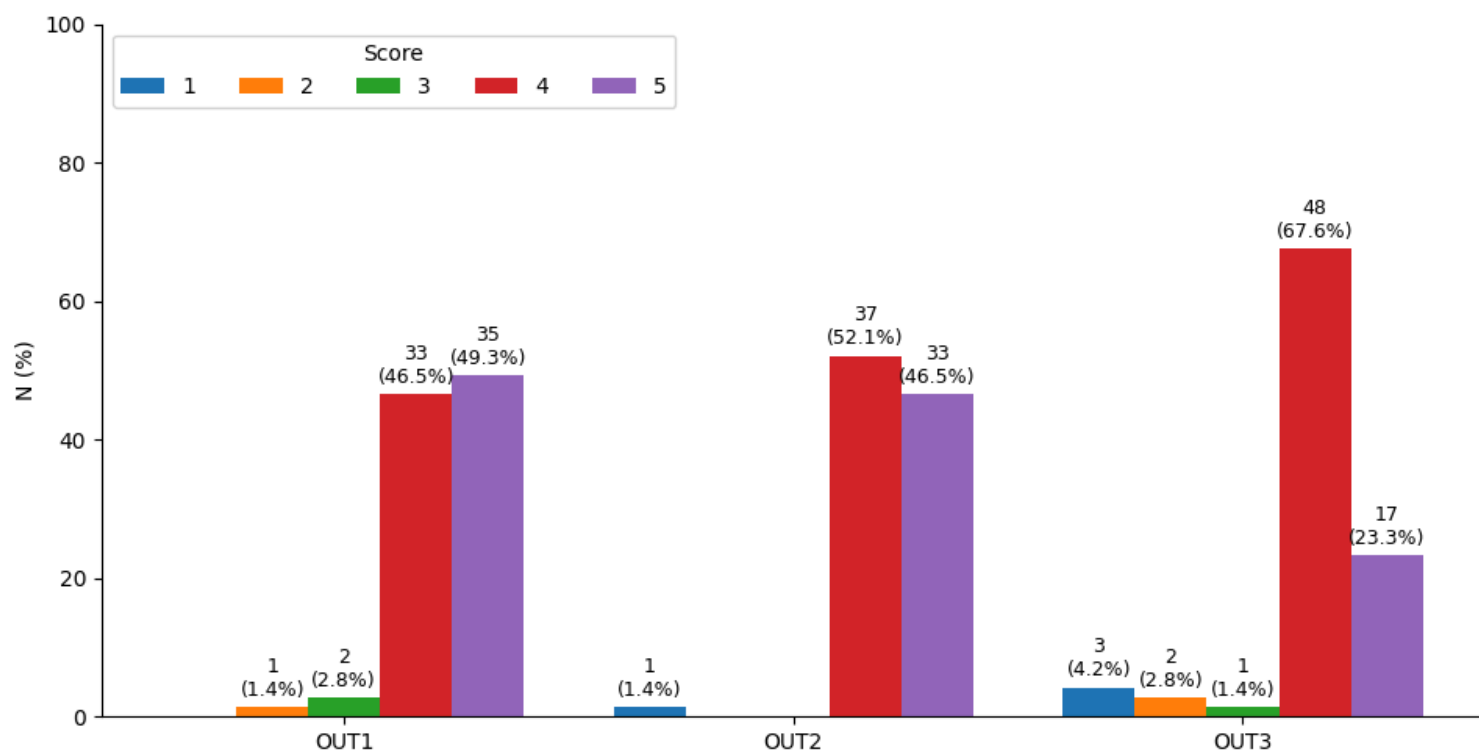


Figure 8

Impact distribution