NORTHWESTERN UNIVERSITY

Computer Science Department

Technical Report
Number: NU-CS-2025-39

September, 2025

Structured Hackathons: A Pedagogical Model for AI Education

Jacob Puthipiroj, Thepchai Supnithi, Nongnuch Ketui, and Nucharee Thongthungwong

Abstract.

Hackathons have gained traction as a compelling educational tool in data science and AI, yet evidence for their long-term effectiveness remains sparse. This paper investigates a structured hackathon curriculum, focusing on a medium-scale program in Thailand that enrolled 175 participants over two months. Seven weeks of Kaggle-style challenges were integrated into an intensive structure, blending online and with onsite sessions. Our results indicate sizeable learning gains: participants' mean exam scores improved post-program, and hackathon metrics proved to be moderate predictors of formal assessment outcomes. Demographic factors had minimal impact on performance, whereas behaviors such as rapid iteration and collaborative problem-solving stood out. Qualitative data further highlight how hackathons can develop critical soft skills, including leadership, adaptability, and communication. While questions of fairness and inclusivity persist, our findings suggest that well-orchestrated hackathons can enrich AI education by targeting industry-relevant competencies often underassessed in traditional exams.

Keywords

Hackathons · Artificial Intelligence Education · Data Science Education

Structured Hackathons: A Pedagogical Model For AI Education

Jacob Puthipiroj[1](✉), Thepchai Supnithi[2,4](✉), Nongnuch Ketui[3,4](✉), and Nucharee Thongthungwong[4]

[1] Northwestern University, Evanston, IL, USA
puthipiroj@u.northwestern.edu
[2] National Electronics and Computer Technology Center (NECTEC), Thailand
thepchai@nectec.or.th
[3] Rajamangala University of Technology Lanna (RMUTL), Thailand
nongnuchketui@rmutl.ac.th
[4] Artificial Intelligence Institute of Thailand
nuchhub@gmail.com

Abstract. Hackathons have gained traction as a compelling educational tool in data science and AI, yet evidence for their long-term effectiveness remains sparse. This paper investigates a *structured hackathon* curriculum, focusing on a medium-scale program in Thailand that enrolled 175 participants over two months. Seven weeks of Kaggle-style challenges were integrated into an intensive structure, blending online and with onsite sessions. Our results indicate sizeable learning gains: participants' mean exam scores improved post-program, and hackathon metrics proved to be moderate predictors of formal assessment outcomes. Demographic factors had minimal impact on performance, whereas behaviors such as rapid iteration and collaborative problem-solving stood out. Qualitative data further highlight how hackathons can develop critical soft skills, including leadership, adaptability, and communication. While questions of fairness and inclusivity persist, our findings suggest that well-orchestrated hackathons can enrich AI education by targeting industry-relevant competencies often underassessed in traditional exams.

Keywords: Hackathons · Artificial Intelligence Education · Data Science Education

1 Introduction

The term "hackathon" –a portmanteau of the words "hack" (referring to exploratory programming) and "marathon"– was first used during a cryptographic development event held in Calgary on June 4, 1999, where ten OpenBSD developers collaborated to address legal challenges associated with U.S. cryptographic software export regulations [2]. Over the past 20 years, hackathons have diversified dramatically in both format and purpose. They have been employed in educational settings, particularly as a strategy to engage students in software engineering practices [17], by companies to encourage internal innovation, improve employee engagement, and generate novel product ideas [1], and by governments to address civic challenges, particularly to encourage citizens to collaborate on public policy issues [9]. Additionally, the scope of hackathons has expanded far beyond their roots in computer science and entrepreneurship [18], encompassing fields as varied as healthcare, where participants tackle medical and public health challenges [14]; art, promoting digital creativity and collaboration among artists and technologists in the form of "culture hackathons" [19]; and environmental science, where interdisciplinary teams develop solutions to sustainability and climate issues [12].

The academic literature on hackathons in education remains sparse, with notable gaps. Early investigations in computing education noted that the few studies on collegiate hackathons were often conducted by the event organizers themselves [21], raising concerns about objectivity and leaving many research questions open. In a multidisciplinary literature review of 111 publications, Chau and Gerber categorize research on hackathons into four core areas –Purpose, Format, Processes, and Outcomes– with the current stage of scholarship placing greater emphasis on hackathon outcomes and post-event impacts [2]. In particular, they note that while many anecdotal reports praise hackathons for inspiration and community building, few studies rigorously assess how hackathon participation translates into learning gains or academic performance. Thus, empirical research is needed to examine whether structured hackathon experiences can produce significant improvements in knowledge and skills, comparable to or exceeding traditional instructional methods. A recent systematic review of the literature emphasized the importance of using structured approaches to orchestrate learning in hackathons [16], suggesting that with careful curriculum integration, hackathons could become effective educational interventions rather than one-time events. Indeed, some educators have even experimented with hackathons as a form of course assessment. For example, in software engineering education, a hackathon-style evaluation has been piloted as an alternative to a conventional exam to directly test students' applied skills on course learning outcomes [15]. These innovations point to the potential of hackathons within formal education, but their impact needs to be validated.

Given these gaps, our work investigates structured hackathons as a model for AI education through a formal study. We focus on three research questions:

- **RQ1.** What is the effect, if any, of a structured hackathon curriculum on students' learning of AI concepts and skills?
- **RQ2.** How does achievement in the hackathon (e.g. project performance or ranking) relate to traditional assessments?
- **RQ3.** What key factors influence learning outcomes in a hackathon-based program?

To answer these questions, we designed an educational program that integrates a hackathon in a curriculum-like structure, and we gathered both quantitative and qualitative data on student learning. By analyzing exam score improvements and correlating them with hackathon results, our goal is to provide evidence of learning effectiveness. We also explore the influence of participant backgrounds and team composition on both hackathon success and learning gains.

2 Design

2.1 Participants

Initially, a broad pool of 513 individuals took part in a series of six short, online Kaggle challenges, lasting 36 hours each and occuring on six consecutive weekends. Participants who performed reasonably well in these hackathons could then take a technical test and sign up for a behavioral interview to progress to the main program, but not all would be invited.

We aimed for a broad representation in the participant cohort in terms of gender, age, educational stage, and geographic location. There were 150 male and 20 female participants. The participants' ages range from 16 to 50, with a median of 21 and a mean of 22.4 years old. The standard deviation in age was 6.65 years. The interquartile range was 5 years (between 19-24). Over 70% of the cohort were students in secondary or tertiary education. 32 participants were employed and 13 were unemployed at the time of the program. Geographically, nearly half (90) were from Bangkok, Thailand's largest city, though other regions were also represented, including the Central (35), Northern (15), Southern (14), Eastern (7), Northeastern (6), and Western (3) regions.

Upon acceptance, the 175 participants were grouped into six large teams of approximately 30 members each. Assignment to these teams followed a two-way preference system: initially, participants indicated their top preferences for each group, and subsequently, group leaders—typically professors or lecturers who would serve as coaches—selected participants based on these preferences. This process ensured that each team was roughly balanced in terms of competence. Thematically, these teams had distinct foci; for example, one team concentrated on academic AI/ML research, another on business applications of AI, and yet another on brain-computer interfaces. The rationale behind this organization was to mirror real-world AI environments, where diverse skill sets and specialized project goals coexist, yet still require active cross-pollination of ideas.

Informed consent was obtained from participants on two occasions. Initially, consent was acquired as part of the conditions for joining the first round of six hackathons. Subsequently, the participants who were accepted into the program again gave their consent, explicitly agreeing to the collection and analysis of their data, including their code and other artifacts generated during the program.

2.2 Program Structure

The core program was structured around 8 week-long hackathons of varying formats, each leveraging Kaggle as a platform but with unique requirements. Each weekly challenge was accompanied by a series of lectures concerning either an AI technique, AI as pertains to some field, or a key infrastructure or tool. The first week remained online, with participants collaborating primarily over Discord. For four subsequent weeks, the entire cohort traveled to a dedicated hotel in central Thailand, which the organizers had fully booked for the program. During these weeks, participants lived on-site with mentors, instructors, and logistical staff, effectively replicating an intensive training camp environment. The program was then onsite for another 2 weeks, before asking participants to return for their final exam and other activities for three days. This residential component was chosen to foster community, ease collaboration, and promote a fully immersive learning experience. Attendance was tracked for both the online and onsite weeks through checkpoints multiple times per day, and through Zoom sessions for any remote lectures.

Table 1. Schedule of Weekly Tasks and Details

Week	Venue	AI Topic	Field	Team Size
1	Online	Image Captioning	General	5–6
2	Onsite	Question Answering	Retail	25–28
3	Onsite	Text Classification	Law	13–15
4	Onsite	Signal Classification	EEG	25–28
5	Onsite	Financial Modeling	Finance	4–5
6	Online	Image Detection/Classification	Medicine	4–5
7	Online	Tabular Classification	Agriculture	1 (Individual)
8	Onsite	Exams	General	1 (Individual)

In a typical week-long hackathon cycle, activities began on Monday and concluded on Friday. On Monday morning, instructors announced the specific Kaggle challenge—ranging from image classification tasks to financial modeling—and opened the corresponding competition on the Kaggle platform. Lectures or workshops typically occupied the remainder of Monday and continued into Tuesday and Wednesday, providing foundational knowledge and guidance relevant to that week's challenge (e.g., exploring relevant algorithms or domain considerations). There were generally no lectures on Thursday and Friday, leaving participants free to focus entirely on coding. The final submission deadline on Friday at noon served as the race to the finish, after which the private leaderboard scores were immediately revealed.

Although Kaggle standings played a central role, they were never the sole criterion for determining the weekly winners. Each week also included a pitching session—an in-person or Zoom-based forum—where top Kaggle-ranking teams or individuals showcased the reasoning behind their approach, the creativity of their solutions, and any unique technical innovations. A panel of judges (comprising instructors, industry experts, or sponsors) then evaluated both objective performance metrics and subjective considerations to name the victors. Winners would be rewarded with cash prizes.

Data were collected from multiple sources to capture both performance and behavioral metrics in order to obtain a holistic view. These sources included each participant's hackathon results (e.g., ranks and scores), final exam scores, frequency of code submissions, percentile standings within competitions, session attendance records, as well as compute usage and other relevant activities.

Fig. 1. A pitch on Friday of the 4th week.

2.3 Design Commentary

In this section, we would like to highlight and explain certain choices that were made in design of the program.

Kaggle Competitions as a Pedagogical Model Kaggle has long been recognized as a powerful platform for fostering practical skills in data science and AI. Studies such as those by Polak and Cook [13] have shown that integrating Kaggle-style challenges into coursework can significantly enhance student engagement and skill development. Kaggle's model thrives on its blend of competition, peer feedback, and open sharing of solutions, which creates an ecosystem where learners are continually exposed to state-of-the-art techniques and diverse problem-solving strategies.

An intriguing aspect of Kaggle is the blend of collaboration with competition. The platform allows individuals to collaborate by sharing code or forming teams, and at the same time, every individual or team is ultimately competing for a higher rank on the leaderboard. We believe this to be a key feature of Kaggle's pedagogical success: a feeling of community and mutual learning coupled with a drive to win. Inspired by these insights, we have sought to replicate the most effective aspects of Kaggle's environment by organizing our participants into teams. This enforced teamwork approach not only harnesses the positive forces of collaboration—where members share code, discuss strategies, and build on one another's strengths—but also preserves a sense of competition, as teams still strive for the best overall performance.

The presence of a live leaderboard in Kaggle competitions is a defining feature that significantly influences participant behavior and learning. The leaderboard ranks all submissions in real time (on a validation subset) which provides continuous, quantifiable feedback on performance. From a psychological perspective, the leaderboard taps into competitive instincts and the desire for achievement.

A further effect of the Kaggle leaderboard is that it externalizes a performance standard. Participants often learn implicitly from the leaderboard even without reading others' code: seeing someone achieve a score much higher than yours signals that better feature engineering or models exist. One can then deduce that certain techniques (perhaps indicated in discussion forums or kernels) are effective. The leaderboard thus sets a de-facto curriculum of techniques to learn. In one competition, for instance, participants realized that using an ensemble of models was necessary to reach the top spots; this pushed those unfamiliar with ensembling to learn about and implement it to stay competitive.

Our experience has highlighted the pitfalls of relying on a single performance metric to determine the "best" solution. A model might achieve outstanding accuracy, yet be so computationally expensive that it becomes impractical for real-world inference. Such factors—ranging from latency and resource usage to maintainability—are crucial to a model's overall viability. As a result, we augment our grading beyond a single leaderboard metric by inviting top-performing teams to pitch their solutions. By allowing a more holistic evaluation—one that weighs trade-offs like speed, scalability, and implementation complexity—we ensure that students learn to balance raw performance with practical considerations, reflecting the realities of industry demands.

Communities of Practice A final core theoretical underpinning of our program design comes from the concept of *Communities of Practice* (CoPs), as articulated by Lave and Wenger in their influential work, *Situated Learning* [10]. In essence, a community of practice is a group of individuals who share a common interest or domain of knowledge, and who deepen their expertise by interacting on an ongoing basis. Lave and Wenger propose that learning is an inherently social phenomenon, best understood not simply as the acquisition of abstract facts, but as a process of participation in a community—where novices learn through engagement with more experienced members. This perspective shifts the focus from classroom-style instruction to the dynamic process of learning through legitimate peripheral participation, in which newcomers gradually assume more complex roles over time and become integral members of the community.

Within our structured hackathon program, we deliberately foster such a community of practice to encourage both formal and informal learning pathways. Rather than relying solely on lectures or top-down instructions, we create multiple channels for peer-to-peer learning and mentorship. Novices working on a particular Kaggle challenge can observe, contribute to, and eventually co-lead the same tasks that advanced participants tackle. Over the course of several weeks, participants discuss code, share tips in online forums, and collaborate in brainstorming sessions, steadily building competence and confidence. In doing so, they naturally transition from novices to more seasoned community members, ultimately mentoring others themselves.

Importantly, our vision of a community of practice extends far beyond just the learners enrolled in the hackathon program. We strive to connect participants with a broader ecosystem of local developers, researchers, engineers, employers, and students across Thailand. This network allows each individual to "plug in" to the community at a level and in a mode that suits their personal aspirations. The 16-year-old participant seeking an extracurricular challenge, the 22-year-old recent graduate hoping to break into the tech industry, and the 35-year-old medical professional exploring AI for healthcare applications all find their niches within the shared space of the structured hackathon curriculum. Rather than prescribing a one-size-fits-all set of outcomes, we accommodate varied goals and skill levels, enabling each member to extract precisely what they need—be it hands-on coding practice, professional networking, or interdisciplinary collaboration.

This inclusive approach is pivotal to sustaining a vibrant CoP, as it reinforces what Lave and Wenger describe as the social fabric of learning. Newcomers see visible paths for growth and engagement, while more experienced members benefit from opportunities to pass on knowledge and stay connected to emerging talent. Industry sponsors and local employers also find value in scouting prospective hires, exchanging ideas, and aligning academic insights with real-world needs. Meanwhile, faculty, TAs, and alumni remain invested in the group's development, facilitating continuous learning cycles that enrich everyone involved.

Finally, our program consciously breaks away from the notion of a linear pipeline that ends once participants "graduate." We encourage former students to reintegrate into the community in various capacities, such as teaching assistants, guest lecturers, or even full-time staff. Many who initially joined as hackathon participants now return to lead workshops, mentor novices, or share their professional experiences. This cyclical process not only preserves institutional memory and perpetuates a strong culture of mentorship, but it also demonstrates that learning is never fully complete; it continues through active engagement in the community.

3 Data Analysis and Results

3.1 RQ1: Pre- vs. Post-Program Test Differences

To evaluate the effect of the structured hackathon program on participants' learning, we compared their exam scores before and after the program. The pre- and post-program exams were designed to be similar in scope and difficulty, ensuring a fair comparison. Each test consisted of 100 multiple-choice questions covering five key machine learning topics: Signal Processing, Image Processing, Data Science, Computer Vision, and IoT.

Analysis of the test scores indicated an improvement in participants' knowledge. The pre-test had an average score of 59.70 (SD = 11.24), while the post-test average increased to 63.94 (SD = 10.75). A paired t-test yielded a highly significant difference between pre- and post-program scores ($t = 4.0353$, $p < 0.001$). The effect size, measured by Cohen's $d = 0.3095$, falls into the small-to-moderate range, point to a meaningful improvement in participants' overall performance. Each test consisted of 100 multiple-choice questions covering five key machine learning topics: Signal Processing, Image Processing, Data Science, Computer Vision, and IoT.

3.2 RQ2: Relationship between Exam Scores from Hackathon Performance

To explore how hackathon performance relates to final exam outcomes, we developed a predictive model via the AutoGluon framework. AutoGluon is an automated machine learning (AutoML) toolkit designed to simplify the application of machine learning by automatically performing model selection, hyperparameter tuning, and feature engineering, thereby leveraging the best available algorithms to optimize predictive accuracy [4].

Our predictive model utilized multiple hackathon performance metrics, including team rank and the number of submissions, among others, as input features. Model evaluation employed 5-fold cross-validation, which yielded a root mean square error (RMSE) of 8.50 and a mean absolute percentage error (MAPE) of 10.27%. Additionally, the resulting coefficient of determination (R^2) was 0.22.

In the context of social science research, an R^2 value of 0.22 indicates a reasonably good model fit, as behavioral and educational phenomena typically involve significant inherent variability and complexity. Studies in education and the broader social sciences often report similar or lower R^2 values due to the multifaceted nature of human behavior and learning outcomes [3]. Therefore, achieving an R^2 of 0.22 suggests that hackathon performance metrics provide meaningful, albeit partial, explanatory power regarding exam performance.

3.3 RQ3: Key Factors Influencing Performance (Permutation Feature Importance)

After training the AutoGluon ensemble, we computed a feature importance via a permutation based-method, that is, measuring the marginal loss in predictive performance when a single feature is randomly permuted, providing an intuitive estimate of each variable's contribution to the model. In total, 117 predictors were considered, grouped into four conceptual families for interpretability:

- **Hackathon rank/score metrics**: 21 variables capturing team private scores and ranks in each hackathon;
- **Submission–activity metrics**: 23 variables describing how often, and in what order, students submitted solutions
- **Practice Exams**: 2 pre-hackathon exam scores;
- **Other controls**: 71 demographic covariates

The leading driver—*Team Private Score on Hackathon 3*—alone accounts for over 40 % of the cumulative importance within the top ten. Aggregating by thematic family:

Table 2. Feature–Family Breakdown Within the Top 10 Predictors

Family	# in Top 10	Share of Importance
Hackathon rank/score	4	60.5%
Practice Exams	2	21.2%
Submission activity	4	18.3%

These results suggest that team performance in competitive settings is the single strongest indicator of subsequent exam success, echoing findings that rank-based feedback is a powerful motivator in learning competitions [7].

Table 3. Top 10 Features by Permutation Importance

Rank	Feature	Perm. Importance
1	Team Private Score on Hackathon 3	0.690
2	Practice Exam (Trial 2)	0.192
3	Practice Exam (Trial 1)	0.169
4	Team Private Rank on Hackathon 1	0.140
5	Team Private Rank on Hackathon 2	0.137
6	Total Submissions on Hackathon 1	0.125
7	Nth Submission Order on Hackathon 1	0.076
8	Nth Submission Order on Hackathon 5	0.073
9	Team Private Rank on Hackathon 5	0.064
10	Total Submissions on Hackathon 4	0.038

4 Discussion

4.1 How Do Hackathon Performances Relate to Traditional Assessment?

Our own data underscore the partially distinct skill sets captured by hackathon performance versus exam results. Although we achieved a mean absolute percentage error (MAPE) of roughly 10% when predicting final exam scores from hackathon data, the alignment between these metrics remains imperfect. Hackathons probe a student's ability to design, iterate, and collaborate quickly, whereas exams often assess in-depth conceptual understanding and memory-based knowledge [5]. From a pedagogical standpoint, the discrepancy suggests that hackathons complement, rather than replicate, what exams measure. In particular, top-performing hackathon participants displayed strong capabilities in ad-hoc problem-solving and rapid adaptation—traits that do not necessarily manifest in a classic, written setting [8]. Educational research thus points to a combined approach: integrating hackathons alongside exams can provide a *multidimensional* picture of student competence, covering both theoretical understanding and practical execution [11] [5].

We also observe that not all hackathons were equally significant in predicting exam performance, and consequently, their utility as indicators of participant ability varied. Nonetheless, specific patterns emerged clearly: notably, the 4^{th}, 6^{th} and 7^{th} most influential features were derived from Hackathon 1, with "Team Private Rank" and "Total Submissions" respectively. This hackathon was an online event involving small groups of around 5 participants selected from larger teams of approximately 30 members. This finding aligns with existing research suggesting that a team size of approximately 5 members optimizes coordination while minimizing communication overhead [14].

However, the most important feature, team performance on Hackathon 3, was the exception that proved the rule. Here, the team size of 13–15 members was well-suited for tackling the task of text classification, specifically in the Thai language: an extremely narrow field with fewer pretrained language models and sparser documentation available. Teams that could combine domain expertise with an understanding of existing NLP libraries for Thai thrived, as did teams with the organizational capacity to split subtasks effectively. This stands in contrast to more universal domains (like image classification), where off-the-shelf solutions and well-documented frameworks abound. Participants and teams adept at balancing domain knowledge with generalizable AI techniques could easily demonstrate their competitive edge in such tasks [6].

4.2 Who Succeeds in a Hackathon Environment?

A striking result from our feature importance analysis is that demographic factors such as age, employment status, or prior educational level did not emerge as strong predictors of exam performance. Instead, participants' behaviors during the hackathon—particularly submission frequency and pitching—proved far more indicative of success.

One of the most significant findings is the high impact of submission frequency on hackathon outcomes. Three of the top ten features—specifically, "Total Submissions on Hackathon 1" (ranked 6th), "Nth Submission Order on Hackathon 1" (ranked 7th), and "Nth Submission Order on Hackathon 5" (ranked 8th)—directly relate to how often participants submitted solutions. This pattern underscores the importance of iterative experimentation in hackathons, aligning with the "fail fast, learn fast" mentality that these events encourage [8]. Unlike traditional exams, where students typically have a single attempt, hackathon participants can submit multiple iterations of their projects, experimenting with different models, hyperparameters, or deployment strategies, and the ability to do so

Additionally, pitch participation surfaced as an influential predictor of success. In many hackathon formats, teams must present and justify their work to judges or peers, rather than relying solely on a numeric leaderboard [20]. Effective pitches call for clarity in explaining design choices, highlighting real-world applicability, and demonstrating collective teamwork. Students who excel in pitching can better articulate their strategies, convince stakeholders of their solution's relevance, and build excitement around their project. This resonates with educational research showing that hackathons hone not only technical skills but also communication and collaboration [11]. Even a well-engineered solution can falter if the team cannot present its value and functionality persuasively.

Ultimately, hackathon success demands a broad spectrum of competencies that go beyond coding prowess. Students need to be adept at leadership, conflict resolution, and agile planning—especially under tight deadlines and uncertain project scopes [15]. Many of these traits fall outside the scope of conventional exams, yet they are vital in modern AI and data science roles. As our results suggest, participants who embraced teamwork, adaptiveness, and rapid experimentation fared best. From a pedagogical standpoint, this highlights the hackathon's distinctive ability to reveal facets of student potential—such as resilience and interpersonal acumen—that remain invisible in purely theoretical tests. In this sense, hackathons complement traditional assessments by capturing real-world readiness in a highly concentrated, interactive environment.

5 Summary and Future Directions

Our findings reveal three major insights. First, students experienced significant learning gains, reflected in higher exam scores and increased engagement. Second, hackathon performance—particularly frequent submissions, team rankings, and pitch participation—proved to be a reasonable predictor of formal assessment outcomes, albeit capturing a different, more application-centric skill set than traditional written exams. Third, demographic variables such as age or employment background were less influential than behaviors shown during the event, reinforcing the notion that hackathon success hinges on rapid iteration, adaptability, and effective communication.

Moreover, questions of *fairness* and *inclusivity* arise in time-intensive competitions. Hackathons may inadvertently favor participants with the freedom to engage in marathon coding sessions or who already possess a certain level of expertise. Future implementations could experiment with structured mentoring, team composition guidelines, or scaffolded tasks to ensure that participants of varying ability levels can meaningfully contribute and grow. Research on self-efficacy and motivation, particularly examining how hackathons foster or hinder confidence for underrepresented groups in AI, remains ripe for investigation.

Finally, there is scope for refining the assessment mechanisms within hackathons. While our study points to a positive correlation between team performance and exam success, further work could develop more nuanced grading rubrics that account for real-world trade-offs like latency, scalability, ethical considerations, and sustainability. Incorporating industry-relevant metrics in the judging process—such as carbon footprints of training models or cost-effectiveness of deployments—could better align hackathon challenges with emerging priorities in AI. Such approaches would enhance not only the educational value but also the practical relevance of hackathon-based curricula.

Taken together, our results affirm that well-structured hackathons can deliver meaningful pedagogical benefits, motivating students and reinforcing essential AI skill sets that traditional classroom methods may overlook. By continuously refining this model through experimentation, cross-institutional collaborations, and rigorous research, hackathons can become a cornerstone of effective AI education that bridges theoretical knowledge with the demands of a rapidly evolving industry.

Acknowledgements

This research has received funding support from the National Science, Research and Innovation Fund (NSRF) via the Program Management Unit for Human Resources & Institutional Development, Research and Innovation [grant number B13F670080].

References

1. Arrington, M.: Groupme, born at techcrunch disrupt, secures funding and launches (2010)
2. Chau, C.W., Gerber, E.M.: On hackathons: A multidisciplinary literature review. In: Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems. ACM, Hamburg, Germany (2023). https://doi.org/10.1145/3544548.3580940
3. Cohen, J.: The effect size. statistical power analysis for the behavioral sciences. Abingdon: Routledge pp. 77–83 (1988)
4. Erickson, N., Mueller, J., Shirkov, A., Zhang, H., Larroy, P., Li, M., Smola, A.: Autogluon-tabular: Robust and accurate automl for structured data. arXiv preprint arXiv:2003.06505 (2020)
5. Gama, K., Alencar, B., Calegario, F., Neves, A., Alessio, P.: A hackathon methodology for undergraduate course projects. In: 2018 IEEE Frontiers in Education Conference (FIE). pp. 1–9. IEEE, San Jose, CA, USA (2018). https://doi.org/10.1109/FIE.2018.8659281
6. Garcia, M.: Integrating industry practices into the academic curriculum via hackathons. ResearchGate (2020)
7. Hanna, C.: Tournaments, rankings, and time crunches: Exploring the use of competition technologies in the classroom. The University of Western Ontario (Canada) (2014)
8. Indra, R., Parthasarathy, P., Ambasana, J., Satavlekar, S.: Flashhack: Reflections on the usage of a micro hackathon as an assessment tool in a machine learning course. In: Proceedings of the 2024 on ACM Virtual Global Computing Education Conference V. 1. pp. 67–73 (2024)
9. Johnson, T.: A review of hackathon pedagogy in computer science education. ResearchGate (2020)
10. Lave, J., Wenger, E.: Situated learning: Legitimate peripheral participation. Cambridge university press (1991)
11. Oyetade, K., Harmse, A., Morze, N.V., Zuva, T.: Evaluation of the impact of hackathons in education. Cogent Education **11**(1), 2392420 (2024). https://doi.org/10.1080/2331186X.2024.2392420
12. Paul, S.: University environmental hackathons to further the sustainable development goals. Sustainable development goals and institutions of higher education pp. 131–140 (2020)
13. Polak, J., Cook, D.: A study on student performance, engagement, and experience with kaggle inclass data challenges. Journal of Statistics and Data Science Education **29**(1), 59–71 (2021). https://doi.org/10.1080/10691898.2021.1892554
14. Poncette, A.S., Rojas, P.D., Hofferbert, J., Valera Sosa, A., Balzer, F., Braune, K.: Hackathons as stepping stones in health care innovation: case study with systematic recommendations. Journal of medical Internet research **22**(3), e17004 (2020)

15. Sadovykh, A., Aslam, H., Marouf, R., Naumcheva, M., Mazzara, M., Bobrov, E.: Hackathon approach for students evaluation at a software engineering course – preliminary results. In: Proc. of Int. Conf. on Computational Intelligence in Security for Information Systems (CISSE) 2023. Lecture Notes in Networks and Systems, vol. 206, pp. 237–245. Springer, Cham, Switzerland (2024). https://doi.org/10.1007/978-3-031-53382-2_20
16. Schulten, C., Chounta, I.: How do we learn in and from hackathons? a systematic literature review. Education and Information Technologies **29**(15), 20103–20134 (2024). https://doi.org/10.1007/s10639-023-11627-0
17. Steglich, C., Salerno, L., Fernandes, T., Marczak, S., Dutra, A., Bacelo, A.P., Trindade, C.: Hackathons as a pedagogical strategy to engage students to learn and to adopt software engineering practices. In: Proceedings of the XXXIV Brazilian Symposium on Software Engineering. pp. 670–679 (2020)
18. Szymanska, I., Sesti, T., Motley, H., Puia, G.: The effects of hackathons on the entrepreneurial skillset and perceived self-efficacy as factors shaping entrepreneurial intentions. Administrative Sciences **10**(3), 73 (2020)
19. Theise, A.: Open cultural data hackathon coding da vinci–bring the digital commons to life (2017)
20. Towhidi, G., Pridmore, J.: Hackathons for experiential learning in is higher education. Issues In Information Systems **23**(1) (2022)
21. Warner, J., Guo, P.J.: Hack.edu: Examining how college hackathons are perceived by student attendees and non-attendees. In: Proceedings of the 2017 ACM Conference on International Computing Education Research (ICER). pp. 254–262. ACM, New York, NY, USA (2017). https://doi.org/10.1145/3105726.3106174