Sequences in Data to Represent Learning Processes

What have you tried until now?

Sequences in Data to Represent Learning Processes

What have you tried until now?

Dissertation

for the purpose of obtaining the degree of doctor
at Delft University of Technology
by the authority of the Rector Magnificus, prof. dr. ir. H. Bijl,
chair of the Board for Doctorates
to be defended publicly on
Wednesday 7, October 2026 at 15:00

by

Manuel VALLE TORRE

This dissertation has been approved by the promotors.

Composition of the doctoral committee:

Rector Magnificus,	chairperson
Prof. Dr. M.M. Specht	Delft University of Technology, *promotor*
Dr. C.R.M.M. Oertel Genannt Bierbach	Delft University of Technology, *promotor*
Independent members:	
Prof. dr. ir. D.A. Abbink	Delft University of Technology
Prof. dr. M. Cukurova	University College London, UK
Prof. dr. ir. H. Drachsler	Leibniz Institute for Research and Information in Education (DIPF), Germany
Dr. A.T.A. Pereira	KTH Royal Institute of Technology, Sweden
Prof. dr. O. Viberg	KTH Royal Institute of Technology, Sweden
Prof. dr. ir. W.P. Brinkman	Delft University of Technology, *reserve member*

Keywords: Learning Analytics, Sequence Analysis, Artificial Intelligence, Problem Solving, Academic Help-seeking

Printed by: Ipskamp Printing

Cover: "Aquí nos tocó vivir" (2012) Lola Campbell de Torre.

ISBN 978-94-6518-416-6

An electronic copy of this dissertation is available at
https://repository.tudelft.nl/.

It’s not about arriving first, but knowing how to arrive.

Jose Alfredo Jiménez

Contents

Summary

Digital learning environments capture vast amounts of interaction data, yet educational assessment remains predominantly product-oriented, evaluating final answers while ignoring the learning journey. This disconnect is epitomized by the question "What have you tried?"—a pedagogical staple for human teachers that digital systems struggle to emulate. This doctoral dissertation investigates how sequential interaction data can be leveraged to bridge the gap between process-oriented pedagogy and digital learning environments, moving from retrospective analysis to real-time support.

The research follows a trajectory of increasing granularity and intervention capability across three phases. The Analytical Phase investigates the capabilities and limitations of coarse-grained data by developing ELAT (edX Log Analysis Tool), conducting a systematic review of learning sequences, and analyzing problem-solving patterns in data science notebooks. While scalable, this research highlights that macro-level logs and even detailed notebooks can capture what learners did, but lack the resolution to explain how they reason. To address this, the Technical Phase introduces JELAI (Jupyter Environment for Learning Analytics and AI), a modular architecture designed to capture and process fine-grained telemetry within coding environments and to guide students through conversational interventions.

Finally, the Pedagogical Phase deploys this infrastructure to address the "process collapse" introduced by Generative AI, where learners may offload essential metacognition. Through classroom interventions, this work demonstrates that while unguided access to AI can encourage unproductive "executive" help-seeking, data-driven scaffolding can successfully shift learners toward "instrumental" strategies. In the final study, we found an equifinality in performance between novice delegators and experienced learners. Generative AI enabled identical outputs through vastly different processes, making the final product a useless proxy for competence. Ultimately, this dissertation contributes a comprehensive framework for designing systems that value and support the learning process itself, prioritizing metacognitive engagement over short-term efficiency.

Samenvatting

Digitale leeromgevingen leggen enorme hoeveelheden interactiegegevens vast, maar onderwijsbeoordeling blijft overwegend productgericht, waarbij eindantwoorden worden geëvalueerd en het leerproces buiten beschouwing wordt gelaten. Deze kloof wordt treffend weergegeven door de vraag “Wat heb je geprobeerd?” – een pedagogisch standaardvraag voor menselijke docenten die digitale systemen moeilijk kunnen nabootsen. Dit proefschrift onderzoekt hoe sequentiële interactiegegevens kunnen worden gebruikt om de kloof tussen procesgerichte pedagogiek en digitale leeromgevingen te overbruggen, waarbij wordt overgestapt van retrospectieve analyse naar realtime ondersteuning.

Het onderzoek volgt een traject van toenemende granulariteit en interventiemogelijkheden in drie fasen. De analytische fase onderzoekt de mogelijkheden en beperkingen van grofkorrelige gegevens door ELAT (edX Log Analysis Tool) te ontwikkelen, een systematische evaluatie van leersequenties uit te voeren en probleemoplossingspatronen in datawetenschapsnotitieboeken te analyseren. Hoewel dit onderzoek schaalbaar is, benadrukt het dat logs op macroniveau en zelfs gedetailleerde notebooks weliswaar vastleggen wat leerlingen hebben gedaan, maar niet voldoende resolutie bieden om uit te leggen hoe ze redeneren. Om dit aan te pakken, introduceert de technische fase JELAI (Jupyter Environment for Learning Analytics and AI), een modulaire architectuur die is ontworpen om fijnmazige telemetrie binnen codeeromgevingen vast te leggen en te verwerken en om studenten te begeleiden door middel van conversationele interventies.

Ten slotte zet de pedagogische fase deze infrastructuur in om de “procesinstorting” aan te pakken die door generatieve AI wordt geïntroduceerd, waarbij leerlingen essentiële metacognitie kunnen afschuiven. Door middel van interventies in de klas laat dit werk zien dat onbegeleide toegang tot AI weliswaar kan leiden tot onproductief “uitvoerend” hulp zoeken, maar dat datagestuurde scaffolding leerlingen met succes kan doen overschakelen op “instrumentele” strategieën. In de laatste studie vonden we een equifinaliteit in prestaties tussen beginnende delegators en ervaren leerlingen. Generatieve AI maakte identieke outputs mogelijk via zeer verschillende processen, waardoor het eindproduct een nutteloze proxy voor competentie werd. Uiteindelijk levert dit proefschrift een uitgebreid kader voor het ontwerpen van systemen die het leerproces zelf waarderen en ondersteunen, waarbij metacognitieve betrokkenheid voorrang krijgt boven kortetermijnefficiëntie.

Acknowledgements

For years, I said I wouldn't do a PhD because it looked like hard, often lonely work that truly requires you to believe in what you are doing. Marcus told me it would be fun and that I could handle it. I guess we were both right.

I'm incredibly grateful to Marcus and Catha for your guidance, motivation, and support throughout this journey, both in my research and in my extracurricular activities. I was lucky to have you (see the chapter pictures as a reminder of what I am talking about). We had great deliberations and brainstorming sessions where I learned a lot about research, but also about myself. Thank you for the encouragement, the challenging questions, and the trust you put in me. For believing in the importance of good coffee, even when the bad one is free, and for knowing when it's time to stop and have a beer.

Thanks to the members of CEL and WIS for the great atmosphere and the many discussions, both inside and outside the lab. To all of WIS, thank you for setting a high bar for research and a fun environment to work in. The old Lambda team: Nirmal, Arthur, Felipe, Gustavo, Sara, Peide, and Claudia, thank you for the online support during lockdowns and my long, long literature review. To Xiaoling, Erna, Yoon, Shirley, Ali, Gabrielle, Bibeg, Nesse, Naomi, Sylvia, Vera, Noortje, Marlies, Marije, and other members of CEL: I learned a lot from each of you. To the new kids: Yuri, Ilinca, Merel, and Aleks, thank you for your energy and motivation; there is hope for educational research in Delft. And to the Hybrid Intelligence team, Morita, Paul, Deborah, Agnes, and Ruben, thank you for your tech support and research ideas.

Thanks to the Npuls Learning Analytics team, to Anouschka, Alan, Priyanka, Symen, Annie, Koen, and Dennis. Our work kept me inspired and grounded in the reality of LA and educational practice in the Netherlands. To Ioana and Jacqueline, you are an inspiration, and I constantly look up to you as researchers. To Fenia for always finding the positive side of things and the importance of enjoying what you do. To Gosia for stepping up and being there for everyone; TU Delft is lucky to have you. To Gitte and Vivian, we had some great times, and I think we learned and grew together as researchers and as people. Maybe someday we'll get that journal club actually going, or maybe the jenever distillery. Thanks to the Tech Support team for helping me navigate and resolve endless situations. I feel like I either had special treatment or you guys are superhuman if everyone gets the same attention. To Selma, the Extension School, and the Teaching Team: thank you for your patience and collaboration. To the students I supervised for your theses and projects, I had fun and learned a lot; I hope you did, too. And special thanks to Daphne and Nadia for always being in my corner when dealing with administration, bureaucracy, and other (not so) fun parts of life; I could do my work thanks to you. This PhD was a long journey involving many, many people, so I'm sure I'm missing important mentions here. This is due only to my forgetfulness, not because their impact was small. Thank you all.

To my friends in the Netherlands, especially my *bois*, Mike, Pietro, and Max, thanks for all the memories and for always being there for me. Life is just better with you guys around. To all of my friends in Mexico, especially Ro, Kay, Juls, and Isai, I feel so lucky to count on you and to call you, or come back home, and feel like I never left. Thank you for your encouragement and for always trying to understand a little bit of what I was doing, even when I couldn't explain it properly. To my Dutch family, Olga, Guido, Jurre, Annemieke, Liza, Bas, Madou, and the kids, thank you for always keeping up with me and for making me feel like part of the family; you make the Netherlands truly feel like home. Dan, I'm so lucky you decided to come to this side of the world, and we could spend time together. Thank you for the dinners, the construction help, for your interest in my work, and for always asking me how I was doing. A mi familia en casa, y en especial a Ma, Pa y Belel: los extraño todos los días y sé lo orgullosos que están. Todo esto es gracias a ustedes; siempre me han motivado a ser mejor.

And to my little family, Lieke and Yuna, thank you for everything. I love you *hasta la luna y de vuelta*.

Manuel Valle Torre
Utrecht, The Netherlands
2026

1 Introduction

It is not the answer that enlightens, but the question.

Eugène Ionesco

I came to do my Master's degree in Computer Science at TU Delft with minimal software development background. This meant learning advanced analytics and machine learning concepts while simultaneously teaching myself to program—like putting tracks in front of a moving train. When I inevitably hit roadblocks—and there were many—I learned that getting help required more than showing incomplete code and error messages. **What have you tried until now?** was always the first question my programming tutors and teachers would ask when I approached them during a lab session.

This question initially frustrated me. I wanted answers, not more questions, but I eventually understood the purpose: by forcing me to articulate my process, tutors could understand my reasoning, identify knowledge gaps, and guide me toward patterns I hadn't yet recognized. More importantly, putting my approach into concrete steps helped me become aware of my own mistakes. Eventually, just by explaining what I had tried and what I expected the code to do, I would realize the solution myself.

And yet, when it came to assessment—homework, automated learning environments, exams—only the product mattered: **did you get the right answer?** The process, the struggle, the learning journey—none of it was visible or valued in those final evaluations. This disparity between process-focused teaching and product-focused assessment was jarring, so it became the center of my research.

1.1. Learning Analytics: The Promise and the Problem

The field of Learning Analytics (LA) emerged with a compelling vision: to leverage the data traces generated in digital learning environments to understand and improve education [1]. As technology became increasingly integrated into learning—from learning management systems to automated tutoring platforms—every student interaction began generating data. Clicks, submissions, forum posts, video pauses, code executions: each one a digital footprint that could provide insights into the learning path [2].

Early Learning Analytics research focused primarily on course or cohort-scale outcomes: course completion rates, quiz scores, and average time spent on videos [3, 4]. These aggregate metrics provided valuable insights for institutional decision-making and identifying at-risk students [5]. However, they shared a fundamental limitation with traditional assessment—they measured products or macro-level behavioral patterns rather than cognitive processes [6–8].

At TU Delft, Massive Open Online Courses (MOOCs) already enrolled millions of learners, generating gigabytes of interaction data per course. With our edX Log Analysis Tool (ELAT) [9], we successfully demonstrated the feasibility of sequence analysis with massive datasets—visualizing learner sequences at scale, tracking navigation paths, video engagement, and forum participation. This macro-level analysis was effective for understanding course design and student retention. We tracked how learners moved through the course materials, but monitoring the learning process at this scale remained a challenge [10]. We could see what happened (a student dropped out, scored poorly on a quiz), but not how or why (what strategies did they try? where did they struggle? what misconceptions emerged?) [11].

1.1.1. The Gap Between Data and Action

A subset of LA researchers pursued process-oriented approaches—analyzing problem-solving strategies or modeling self-regulated learning behaviors [12, 13]. These investigations revealed rich insights about how students learn. Yet they remained retrospective, labor-intensive efforts, conducted in controlled settings with dozens of students rather than thousands. The question seemed clear: how do we support learning processes at the scale of modern digital education?

The COVID-19 pandemic intensified this question [14]. As universities and schools worldwide shifted online, digital learning moved from supplementary to essential. The amount of available trace data exploded—every student interaction, every click, every submission was now captured. But a troubling pattern emerged: while content delivery had scaled, the feedback loops essential for formative assessment had not [15]. Entire courses were reduced to self-study, multiple-choice quizzes, and proctored exams—placing even more weight on the final product.

Meanwhile, a parallel disruption was already underway: the rise of Large Language Models (LLMs) and Generative AI (GenAI) as tools for learning and problem-solving [16, 17]. Scale remained important, but the more pressing concern became the nature of *how* students interact with AI-powered tools—often in individual, small-scale settings where the quality of each interaction matters more than the volume of data. Understanding and supporting LLM-mediated collaboration thus required not only large-scale analytics but also close, fine-grained investigation of individual learning processes.

1.1.2. The Persistent Focus on Outcomes

Even with unprecedented access to behavioral data, most systems still reduced this rich sequential information to aggregated outcome metrics [18]. We were capturing the process, but evaluating it as a product. The outcome-centric paradigm remained unchanged. A student who struggled for hours, tried multiple approaches, consulted resources, and finally achieved understanding received the same "correct" as a student who solved it effortlessly or one who simply guessed. A student who demonstrated sophisticated problem-solving strategies but made a small syntax error received the same "incorrect" as one who had no idea how to begin.

This approach created three critical problems—problems that would only deepen with the arrival of Generative AI (GenAI):

1. Missed opportunities for personalized support: By the time a failing grade revealed a student's struggle, it was too late to intervene. Process data could reveal struggle or the need for a challenge as it happened [19].

2. Invisible learning: Students who improved their strategies, deepened their understanding, or overcame misconceptions weren't recognized unless those improvements manifested in better scores. The learning journey remained invisible [6].

3. Misaligned incentives: When only products matter, students optimize for answers rather than understanding. Why struggle to comprehend when you can find or generate a solution? [20]. The emergence of LLMs would soon make this shortcut effortless—capable of producing polished code, essays, and solutions on demand—collapsing process into an instant product.

These developments—the pandemic's exposure of fragile feedback loops, and GenAI's radical acceleration of product generation—revealed the real challenge: not scale, but actionability. We finally had the technological capability to capture learning processes—the very thing teachers valued when they asked **"What have you tried?"**—yet our systems still primarily measured the final product. The question wasn't how to analyze more data, but how to use process data to support learning *as it happened*.

1.2. From Data to Learning: The Research Gap

This contradiction between what we could measure and what we did measure, between valuing process and rewarding products, led to the guiding question for this dissertation:

Can we use sequential interaction data to understand and actively support the learning process?

This question has three parts, reflecting three distinct challenges:

- Analyzing: Can we develop methods to analyze sequences of learner actions at scale, moving beyond aggregated metrics to reveal the how and why of learning?
- Understanding: Can we translate these insights into real-time interventions that shape productive learning behaviors, not just identify struggling students after the fact?
- Supporting: How can we integrate interventions in learning environments to support teachers and learners in more process-oriented assessment and regulation?

The first challenge is one of methods and measurement. The second is pedagogical. The third is both technical and pedagogical—and it would become far more complex than we anticipated.

1.3. The Disruption: when Large Language Models became the Default Help Provider

Just as we were developing systems to support learners through process-aware analytics, a disruption occurred: the mainstream explosion of Large Language Models, catalyzed by ChatGPT's release in November 2022 [21, 22].

At first, this seemed like the right technological opportunity [17]. Imperfect as they were, LLMs could provide the flexible, natural-language interface between system and student we needed for real-time support. Students could simply have a conversation with their tutoring system; we could leverage this to deliver context-aware tutoring [23]. We could design pedagogical conversational systems and evaluate their effect on learning.

But the accessibility and quick progress that made LLMs powerful also made them problematic.

1.3.1. The Metacognitive Collapse

Within months, LLMs became more accurate and ubiquitous. They were convincing enough for users to believe they could answer all types of questions and problems. They required no technical knowledge and were accessible to everyone, with no guidelines or rules for their use in education [24]. Research quickly revealed a pattern: without explicit scaffolding, students frequently defaulted to using LLMs to bypass the learning process rather than support it [25, 26]. We term this phenomenon Process Collapse: students bypassed the critical stages of diagnosis and reflection, defaulting to a direct final product request. Instead of articulating their reasoning (**"This is what I've tried"**), they would paste error messages or entire tasks and receive immediate solutions.

Early research on LLMs rushed to report performance gains, often ignoring how the tools were actually used or the learning process itself [27]. The assignments that teachers designed to support the learning process and promote metacognitive awareness were rendered obsolete. The LLM didn't need to ask what you'd tried; it just gave you the answer. The final product that we used to assess—the essay, the solution, the code—was instantly available without effort or understanding [28].

The fall of the product as a proxy for learning forced us to confront the process directly. If AI can generate the destination instantly, the journey becomes even more valuable for supporting learning. This realization shifted my research focus from understanding processes to shaping them, from observation to intervention. The LLM became not just the problem but also a potential solution: students were already using conversational AI, and we could embed scaffolding within those conversations to guide the metacognitive processes that LLMs had displaced.

1.4. Research Journey: A Three-Phase Arc

This journey unfolds in three phases, building on foundational work where we explored learner sequences at scale.

Chapter 2: ELAT for edX Log Analysis (LAK 2020) [9] Before the main research arc began, we developed ELAT to explore the feasibility of process analysis in Massive Open Online Courses. This browser-based tool demonstrated that sequence visualization at scale was possible—but also revealed that coarse-grained MOOC data lacked the granularity we needed for individualized, real-time support. This foundational experience shaped the research questions that followed: if we wanted to move beyond visualization toward real-time support, we first needed to understand the state of the art.

1.4.1. PHASE I: Understanding Sequences (Analyze)

The limitations of ELAT raised a fundamental question: were we alone in this gap between analysis and action, or was this a field-wide challenge? And if this was a field-wide challenge, were there established methods and terminology to address it? To answer this, we conducted a systematic literature review examining how learning sequences were being analyzed within tasks and problems, and whether current methods were sufficient to support intervention.

Chapter 3: Systematic Literature Review (LAK 2024) [29] We systematically reviewed 74 articles (2010–2023) on sequence analysis in learning analytics. The review revealed a critical gap: only 2 of the 74 studies had implemented actual interventions based on sequence insights. Most work remained retrospective, and real-time support was still largely conceptual. The field had developed robust methods for analysis but lacked frameworks (and a common terminology) for translating those insights into action.

With this field-wide gap confirmed, we turned to authentic problem-solving data to understand what fine-grained sequences could reveal about learning processes.

Chapter 4: Problem-solving in Data Science [30] We conducted a multi-level sequence analysis of 440 Kaggle notebooks, using Kaggle tiers as a proxy for expertise. The analysis revealed that experts used shorter, iterative workflows compared to novices' longer, linear processes, and that expertise was distinguished by overall structure and fine-grained action patterns rather than transitions between data science phases. Critically, we found that even detailed notebooks captured only a final product—the polished, submitted version—missing the iterative trial-and-error of the actual problem-solving process. Real-time detection and intervention remained impossible with static artifacts alone.

Phase I confirmed both the opportunity and the gap: sequence analysis could reveal context-specific learning processes, but we lacked the frameworks and infrastructure to act on these insights. This demanded a shift from analysis to engineering.

1.4.2. PHASE II: Building Infrastructure (Understand)

We needed to move from observation to action—from analyzing what students did to intervening while they were doing it. This required systems that could capture fine-grained actions, analyze them in real-time, and deliver personalized support within authentic learning environments.

To address this challenge, we needed to integrate learning analytics with tutoring practices that could provide context-aware, in-the-moment support. This was not merely a software integration problem but an architectural one: how to tightly couple high-frequency, low-latency telemetry (keystrokes) with context-specific and pedagogical scaffolding, all within the constraints of a student's private coding environment.

Chapter 5: JELAI Architecture (AIED 2025) We designed and built JELAI (Jupyter Environment for Learning Analytics and AI), a modular platform integrating fine-grained telemetry, local LLM integration for conversational tutoring, and a pedagogical scaffolding layer—all within Jupyter Notebooks. The platform captured code edits, executions, errors, and chat interactions at sub-second granularity, enabling research on student-AI interaction patterns. The technical infrastructure now existed, leading to the question: how could we use it to support students to use GenAI productively?

With JELAI operational, we had the technical capability to observe *and* intervene. But early deployments revealed a troubling pattern: students were abusing the provided help at the cost of their own learning.

1.4.3. PHASE III: Shaping Behavior (Support)

The infrastructure worked, but the pedagogy was missing. Students had access to powerful AI tutors, yet lacked the metacognitive skills to use them for learning rather than task completion.

We explored what pedagogical interventions could shift students from unproductive (executive) to productive (instrumental) help-seeking.

Chapter 6: High School Intervention Study (SAC 2026) We conducted a Design-Based Research study with Dutch high school programming students (N=37), discovering that executive help-seeking was strongly negatively correlated with performance. After explicit teaching of instrumental versus executive help-seeking, the proportion of executive queries decreased significantly (large effect size) and sequential patterns shifted from reactive ("Error → AI → Paste") to proactive ("Success → AI → Clarify"). However, grade improvements were not statistically significant despite the behavioral change. The key insight was clear: behavior can be changed, but may not overcome foundational knowledge gaps.

Building on this, we investigated the nature of productive inquiry with GenAI and whether simple metacognitive scaffolding could prevent process collapse.

Chapter 7: Help-Seeking with GenAI and Metacognitive Scaffolding (Submitted - IJHCS) University students and professionals (N=40) worked on data science tasks with JELAI v2, which featured a multi-agent architecture with adaptive scaffolding. Using Hidden Markov Model analysis and cluster analysis, two distinct behavioral archetypes emerged: "Structured Coders" (higher experience, relying on tool acquisition and editing) and "Novice Delegators" (lower prior knowledge, relying on delegation and AI copying). Both archetypes achieved statistically equivalent performance—an equifinality paradox demonstrating that delegation served as a valid compensatory strategy for novices. A probe intervention using positive reinforcement did not change performance or behavior, but significantly increased persistence: participants who received reinforcement after executive help-seeking were more likely to return for Session 2. The intervention acted as a safety net, managing frustration without restructuring the underlying process. Simple conversational nudges significantly increased persistence, suggesting that motivational scaffolding is beneficial for engagement even when it doesn't immediately improve performance or behavior.

1.5. Thesis Outline

The remainder of this dissertation is organized as follows:

Chapter 2: Introducing ELAT:
An open-source, privacy-aware and browser-based edX log data analysis tool
Published in *LAK 2020* [9].
Demonstrates browser-based MOOC sequence and process visualization at scale.

Chapter 3: The Sequence Matters
A Systematic Literature Review
Published in *LAK 2024* [29].
Investigates how learning sequences are analyzed in the field, revealing methods for retrospective analysis but a critical gap in frameworks for real-time intervention.

Chapter 4: What Makes an Expert?
Comparing Problem-solving Practices in Data Science Notebooks Preprint in

ArXiv [31].
Examines problem-solving strategies in data science notebooks at multiple granularity levels, demonstrating how expertise manifests in workflow structure but highlighting the limitations of static artifacts.

Chapter 5: JELAI: Integrating AI and Learning Analytics in Jupyter Notebooks
Published in *AIED 2025* [32].
Presents modular platform architecture integrating fine-grained LA with context-aware LLM tutoring.

Chapter 6: LLM Chatbots in High School Programming
Exploring Behaviors and Interventions
Published in *SAC 2026* [33].
Investigates how help-seeking behavior with LLMs correlates with programming performance, and whether explicit teaching can shift students from executive to instrumental help-seeking.

Chapter 7: Process Collapse or Compensatory Strategy?
Profiling and Scaffolding Help-Seeking with an AI Tutor
Submitted to International Journal of Human-Computer Studies [34].
Examines the behavioral patterns and archetypes that emerge in AI-supported problem-solving, and how metacognitive scaffolding affects persistence and engagement.

Chapter 8 concludes this dissertation by discussing the overarching findings and implications for the field of learning analytics and artificial intelligence in education.

2

edX Log Data Analysis Made Easy

Introducing `ELAT`: an open-source, privacy-aware and browser-based edX log data analysis tool

Manuel Valle Tore, Esther Tan and Claudia Hauff

It is a capital mistake to theorize before one has data.

Sherlock Holmes (Arthur Conan Doyle)

*Massive Open Online Courses (MOOCs), delivered on platforms such as edX and Coursera, have led to a surge in large-scale learning research. MOOC platforms gather a continuous stream of learner traces, which can amount to several Gigabytes per MOOC, that learning analytics researchers use for exploratory analyses and to evaluate deployed interventions. edX has proven to be a popular platform for such experiments, as the data generated by each MOOC is easily accessible to the institution running it. One issue researchers face is preprocessing, cleaning, and formatting large-scale learner traces. It is a tedious process that requires considerable computational skills. To reduce this burden, several tools have been proposed and released to simplify the process. Those tools, though, still have a significant setup cost, are already out of date, or require preprocessed data as a starting point. In this paper, we introduce `ELAT`, the **e**dX **L**og file **A**nalysis **T**ool, which is browser-based, keeps the data local, and takes edX data dumps as input. `ELAT` not only processes raw data but also generates semantically meaningful units (learner sessions rather than just click events) that are visualized in various ways (learning paths, forum participation, video-watching sequences). We report on two evaluations we conducted: (i) a technological evaluation and (ii) a user study with potential end users of `ELAT`. `ELAT` is open-source and available at https://mvallet91.github.io/ELAT/.*

This chapter was published in LAK20 [1].

ectionIntroduction

Educational research into MOOCs has taken off in recent years, as—among others—evident in the creation of a whole conference series[1] dedicated to it. In contrast to early predictions of MOOCs as being a revolution of higher and life-long education, reality has proven to be more complex. The retention rates in MOOCs remain very low [2] and interventions designed to improve MOOC learners' outcomes are often failing when deployed on MOOC platforms with thousands of learners [3]. This is most often the case when topic-agnostic MOOC platforms such as edX and Coursera are investigated, while platforms specifically designed for a particular domain (e.g. language learning platforms, programming platforms) tend to yield more positive results [3].

Despite these problems, there are distinct advantages to exploring learning phenomena in MOOCs: they are large-scale (often with tens of thousands of learners), platforms typically log all possible interactions the learners have with the material, they attract participants from a wide range of backgrounds (and thus insights are not limited to a particular type of cohort) and they often allow researchers to deploy their own interventions such as an interactive learning planner [4], a webcam-based attention detector [5], a next-step recommender [6] or a collaborative chat [7].

Besides intervention-based research, a large number of studies have also been dedicated to the post-hoc analysis of MOOC learner behaviours, e.g., [8–12]. What both the intervention-based and posthoc analyses-based works have in common is their reliance on the data traces logged by MOOC platforms in their quest to analyze learners' behaviour. We here focus on the *log traces produced by the edX platform*, as the data each MOOC generates is easily accessible to the institution running the MOOC. Those log traces, though detailed, are typically not at the *semantic* level necessary to answer a particular research question. As a concrete example, consider the edX log entry in Figure 2.1: it contains a single click event (in this case, video pausing) of a single learner being active in a single course. Many of these log entries need to be aggregated to even compute basic statistics such as the number of seconds a learner watched a particular video. This typically requires the writing of a number of scripts (often in Python or R), in order to extract the desired information from the logs. This is not only repetitive and inefficient (as researchers duplicate their efforts to clean, preprocess and aggregate the logs), it also severely limits the insights researchers without a computational background can gain, as they have to rely on the precomputed statistics provided by the `edX Insights`[2] platform.

A number of prior works have started to tackle this issue. In 2014, [13] introduced `MOOCdb`, a shared data model which revolves around the types of interactions (observing, submitting, collaborating and feedback) learners can have on a MOOC platform. The accompanying Python implementation converts log traces into a relational schema that can be queried through SQL. A year later, `moocRP` [14] was introduced, which incorporates data models such as the `MOOCdb` one and offers a number of out-of-the-box visualization and analytics functionalities, and thus also caters to researchers without programming experience. At the same time, though, the

[1] The *ACM Learning @ Scale* conference series was started in 2014.
[2] `https://insights.edx.org/`

```
{"username": "USER", "event_type": "pause_video", "ip":
↪ "000.000.000.000", "agent": "Mozilla/5.0 (Windows NT 6.3; WOW64)
↪ AppleWebKit/537.36 (KHTML, like Gecko) Chrome/42.0.2311.90
↪ Safari/537.36", "host": "courses.edx.org", "session":
↪ "4098c9df34a6668e5a352951ceee407d", "referer":
↪ "https://courses.edx.org/courses/InsituteX/EX2015/courseware/a7⌋
↪ 9124bd10c042cc886fcac3b6f09ee7/9e5e8017/", "accept_language":
↪ "en-US;q=0.6, en;q=0.4", "event":
↪ "{\"id\":\"i4x-InstituteX-EX-video-ea41fb\",\"currentTime\":46.⌋
↪ 44,\"code\":\"H_RWt3rxrWE\"}", "event_source": "browser",
↪ "context": {"user_id": USER_ID, "org_id": "InsituteX",
↪ "course_id": "InsituteX/EX/1T2015", "path": "/event"}, "time":
↪ "2015-04-21T19:29:42.524601+00:00", "page":
↪ "https://courses.edx.org/courses/InsituteX/EX/1T2015/courseware⌋
↪ /a79124bd10c042cc886fcac3b6f09ee7/9e5e8017/"}
```


Figure 2.1: Example of a raw edX log entry. The complex nested structure makes manual inspection difficult.

moocRP implementation was last updated four years ago and requires a number of installation steps that are not trivial to execute, especially considering the age of the software.

As an answer to these challenges, we introduce ELAT, a browser-based tool that (i) requires zero installation efforts beyond that of a modern web browser such as Google Chrome, (ii) is privacy-aware in the sense that all computations happen on the user's local machine and no data is send to the cloud, and (iii) incorporates a number of analytics and visualization modules (based on feedback we acquired from end user interviews) that can be arranged into a personalized dashboard. ELAT is written in JavaScript, builds upon MOOCdb's data model and makes exclusive use of the modern web browser's functionalities, such as a built-in database, to access functionalities that formerly required a client-server architecture. Longevity of our software is ensured due to the browser vendors' careful browser updating policies—browser updates are designed to *not* break existing web applications.

Having implemented ELAT, we conducted two evaluations—a system evaluation and a user study with seven participants, each of them a potential end user of our tool. We find that even large MOOCs (65K learners) can be processed in a reasonable amount of time (less than six hours) on a standard laptop. Our user evaluation shows that ELAT's envisioned purpose—to remove a major burden for learning researchers (to setup tooling and to preprocess the data) and to provide meaningful insights of learner behaviour at various levels of granularity—has been achieved.

In the remainder of this paper, we first present related literature and tooling (2) and then turn to ELAT and the description of its architecture and capabilities (2.1). In 2.2 we outline our evaluation—both from the technical perspective *(is a standard browser able to handle Gigabytes of log data?)* and the user perspective *(is the tool useful to our targeted end users?)*. Finally, we conclude with an outlook into ELAT's future in 2.3.

ectionBackground

In recent years, research on MOOCs has witnessed a proliferation of empirical studies on learner attrition rate [15–17], learner engagement patterns [9, 18–23] and self-regulation [24–26]. These research foci often demand more than simple statistical analyses of survey or self-report measures to draw conclusions on learners' online learning experience and learning engagement. In addition, they may also require daily updates to determine for instance whether or not to extend or deactivate an intervention deployed in an A/B test setup [27].

As a result, researchers have been leveraging different tools for log analysis for years now, but it appears to remain an obscure and repetitive process. Often, there is little information beyond *"All log parsing was performed using standard modules in Python and R."* [33] which makes it impossible to fully reproduce the findings as low-level log events have to be aggregated into semantically meaningful units (e.g., at what amount of inactivity does a new learner session start? at what level of activity is a learner considered to be an active MOOC learner—is one visit to the course enough?). This issue has been recognized and addressed by a number of prior works, the most important and relevant ones for us—i.e., open-source tooling that processes edX log files—are listed in Table 2.1: apart from our own tool `ELAT`, we consider `MOOCdb`[3] (one of the first efforts to build process log data into a session-based data model, which they defined), `moocRP`[4] (a data processing and visualization tool similar in spirit to `ELAT` with emphasis on server-side security measures), `visMOOC`[5] (initially a video analysis tool to inspect the aggregated learner activities on course videos but later extended to include forum behavior [34], dropout analysis visualizations [35], and more[6]), `ANALYSE`[7] (developed as an OpenEdX plugin to address its lack of analytics [36]) and `edx2bigquery`[8] (a tool to import edX log files into Google's BigQuery, a web service for the interactive analysis of large datasets).

We set up our comparison along three dimensions: (i) the tool itself, (ii) the data that goes in and comes out, and, (iii) the available visualizations of learners' behavioural data. For all tools, we first determined whether they can still be installed according to the available instructions—not surprisingly, for the tools whose last software update was a few years ago (`MOOCdb`, `ANALYSE` and `moocRP`) this is no longer possible, due to outdated software libraries[9]. We expect `ELAT`, even if not developed any further, to remain working for years, due to the browser vendors' policies of not releasing new browser versions that break existing web applications. While for most tools the user platform (i.e., how the end user accesses the tool) is indeed the browser, for all but `ELAT` (which requires no setup) significant computational knowledge is required in order to set up the tooling as they all employ a traditional client-server architecture. As a concrete example, consider the setup requirements of `moocRP`: knowledge of Node.js, mySQL, Redis and Sails is required.

[3] `https://github.com/MOOCdb/Translation_software`
[4] `https://github.com/CAHLR/moocRP`
[5] `https://github.com/HKUST-VISLab/vismooc`
[6] `https://elearning.hkustvis.org/`
[7] `https://github.com/jruiperezv/ANALYSE`
[8] `https://github.com/mitodl/edx2bigquery`
[9] For the subsequent categories of tools that are no longer working, we relied on the tools' available documentation and demonstration videos to gauge their visualization and data capabilities.

criptsize

Table 2.1: Overview of edX-based data processing and visualization tools.

	`ELAT`	`MOOCdb` [28]	`visMOOC` [29]	`ANALYSE` [30]	`moocRP` [31]	`edx2bigquery` [32]
MOOC Platform	edX	edX & others‡	edX	Open edX	edX & others‡	edX
Tool						
Working	✓	✗	✓	✗	✗	✓
Open Source	✓	✓	✓	✓	✓	✓
Last Update	09/2019	06/2015	04/2018	07/2015	11/2015	08/2018
Dev Platform	JavaScript	Python, Matlab	Python, TypeScript	Python	Python, JavaScript	Python, BigQuery
User Platform	browser	Python + SQL	browser	browser	browser	Python/R + SQL
Knowledge Required for Setup	✗	Docker, Python, MongoDB, MySQL	Shell, Python, MongoDB, MySQL	Open edX dev, Django Server	Node, MySQL, Redis, Sails	Python, mySQL, BigQuery
Data						
Output (dl)	sessions	sessions	✗	events	events	events & sessions
Storage	local	server	server	server	server	cloud (BigQuery)
Incr. Update	manual	manual	✗	automatic	automatic	automatic
Visualizations						
Customization	◐	✗	○	◐	○	✗
Downloadable	✓	✗	✗	✗	✗	✗
Knowledge Required for Operation	✗	Python, R	✗	✗	✗	Python, R
Types	8 VIS, 2 SAM	3 SAM	3 VIS	20 VIS	4 VIS	7 SAM

‡ We note that besides edX, multiple data models... found no information to this effect in the accessible code.

In terms of data, most tools allow the downloading of the aggregated data, with `ELAT`, `MOOCdb` and `edx2bigquery` aggregating the logs on a learner session level, instead of an individual event level. In terms of storage, only `ELAT` stores the data locally (i.e., inside the end user's browser), whereas all other tools require the use of an additional server or the cloud (`edx2bigquery`). The use of a service such as Google's BigQuery seems tempting, as all processing costs reside in the cloud; however, not only does it require a substantial pipelining effort for an institution to move its data to the cloud, it also leads to issues with respect to data privacy. Apart from `visMOOC` for which no information is available, all tools allow an incremental update of the logs—as edX logs are released in 24 hour cycles, it would be wasteful to process all data again. Due to `ELAT` residing completely in the browser, new edX log files have to be uploaded manually; in contrast, `ANALYSE`, `moocRP` and `edx2bigquery` provide an integration with Amazon S3 (the cloud service edX uses to store and serve all edX data[10]) and thus any new log file added to the correct S3 folder is automatically being processed.

Lastly, let us consider the visualization category: while all tools have at least some visualization capabilities (either by providing visualizations out of the box or by providing sample code of how to generate visualizations from the data), for `edx2bigquery` and `MOOCdb` accessing those capabilities requires at least some knowledge of programming languages. `ELAT` allows the downloading of the generated charts and plots for further processing by the end user. Customization of the visualizations provided is possible for all but `visMOOC` and `moocRP`. Finally, we note that `ELAT` includes eight interactive visualizations while `ANALYSE` offers by far the most visualizations, but as already stated is no longer being updated.

Overall, we argue that in this comparison `ELAT`'s strengths lie in the complete lack of setup costs, the local processing of log files into semantically meaningful learner sessions and the provision of a range of visualizations based on recent research works.

2.1. ELAT

In this section, we first outline our design requirements, describe the edX log traces and then turn to `ELAT`'s system architecture. Lastly, we showcase the analytics and visualization functionalities `ELAT` incorporates.

2.1.1. Design Requirements

In response to the challenges outlined so far, we have developed `ELAT`. It is a browser-based application that provides a host of information extracted from edX log traces. After an initial interview with six stakeholders from our local institution who are all involved in the edX MOOC content creation process, we identified a number of requirements:

1. The tool can be used on different operating systems.

[10] https://edx.readthedocs.io/projects/devdata/en/latest/access/download.html

2. It requires minimal or no installation efforts.

3. It requires no data preprocessing of the edX log traces.

4. It requires no configuration.

5. No data leaves the end user's machine as course data contains sensitive information including learners' names, email addresses and so on.

6. It supports the visualization of controlled trials—especially important when interventions are deployed in an A/B test setup to gauge whether they have an effect.

7. Aggregated data and visualizations can be downloaded easily for further processing.

8. As edX releases the daily log files in 24 hour cycles, it should be possible to simply add another log file to an already existing dataset without having to restart the processing pipeline from scratch.

Due to requirements (1) and (2) we settled on the modern web browser, specifically Chrome and Firefox[11], as runtime environment for our tool. To the end user, `ELAT` has the look and feel of a regular web application (though in contrast to most web applications, it will be running all data processing on the client). Initially this may seem like an odd choice, as we are dealing with potentially Gigabytes of course data as seen a number of edX MOOC examples in Table 2.2: e.g., the 2-year run of the self-paced *Solar Energy* MOOC attracted more than 64,000 learners (0.9% of those successfully completed the MOOC). In that time period, more than 845,000 learning sessions[12] were recorded, yielding more than 5 Gigabytes of processed data. Due to requirement (5) we could not develop a traditional client-server application (where data is send to a server that does the heavy lifting in terms of processing), and thus had to rely on the modern browser's web APIs to enable a similar processing pipeline within the browser. All modern browsers support `IndexedDB`, a low-level web API that enables client-side (i.e., *inside the browser*) storing of large amounts of structured data. As already noted, another advantage of a web-browser based tool is the fact that web browser are designed to run code that was created many years ago, quite a contrast to the multitude of tools and frameworks shown in Table 2.1 where a single outdated library or a breaking change in a framework can mean a tool to no longer work (unless significant effort is expended to resolve library/framework issues).

Before describing the system architecture of `ELAT`, let us briefly discuss the makeup of the edX data traces and how to translate those to a semantically meaningful data model (`MOOCdb`'s data model in our case).

[11] Other browsers such as Safari or Edge present compatibility issues with some of the utilized libraries as of now.

[12] A learning session starts when a learner enters the MOOC and ends after 30 minutes of inactivity.

2.1.2. From Low-level Logs to Relational Schemas

Every edX course is defined by two types of data traces: *metadata* files and *clickstream* files, as visible at the end of the chapter in Figure 2.8. The former contains information on the individual learners as well as the course makeup. The latter is a click-by-click *log* of learners' activities across *all courses* running at a particular institution *on a particular day*. For the *Solar Energy* MOOC (cf. Table 2.2) this means that `ELAT` had to handle the processing of 731 files (that amounts to 34.5 GB compressed files) in order to extract the clickstream data relevant for that particular MOOC.

As the `MOOCdb` data model provides meaningful semantic units (including forum interactions, forum sessions, survey responses, quiz questions and so on) we decided to use it as a basis for `ELAT`. Given that the original `MOOCdb` Python implementation[13] is by now five years old, we started off with a modified version of it[14] that had been updated to be usable with a more recent version of Python.

Python is not natively supported in the browser, a fact that can be remedied in two ways: we can either make use of compilers that compile Python code into JavaScript in an "offline step" (e.g., `Transcrypt`[15]) or we employ Python interpreters that themselves are written in JavaScript (e.g., `Brython`[16]). We experimented with both `Transcrypt` and `Brython` and found the latter not suitable for our purposes as (i) we cannot access the JavaScript code that is generated on the fly and thus cannot manually optimize it, and (ii) the unoptimized version takes too long to process even small log files (17 minutes for a file of 10 Megabytes—our optimized code now processes the same file in less than 6 seconds). Using `Transcrypt`, we were able to first generate the JavaScript code, analyze its function and performance, and optimize it manually to fix any issues and increase the speed of conversion. We note that an extensive manual code review was required (approx. 100 hours overall), as `Transcrypt`'s automatically translated code could not be executed as-is due to the fact that many Python to JavaScript data structure conversions had issues[17].

2.1.3. System Architecture

With the JavaScript code in place to convert data traces to a relational schema, we can now discuss the overall architecture of `ELAT`, which is depicted in Figure 2.2. The user interface component enables the user to load the metadata (which determines for which course the clickstream logs are extracted) and clickstream files into the browser's storage. The `FileReader` module is responsible for parsing and extracting the data in a efficient manner, even at Gigabyte sizes. For the compressed

[13] https://github.com/MOOCdb/Translation_software
[14] https://github.com/AngusGLChen/DelftX-Daily-Database
[15] https://www.transcrypt.org/
[16] https://github.com/brython-dev/brython
[17] For instance: to check if an element belongs to a set, the command in Python is `if element in set`; `Transcrypt` translated the code into JavaScript as `if set.includes(element)`. This translation works for a JavaScript array, but the correct notation for a JavaScript set is `if set.has(element)`.

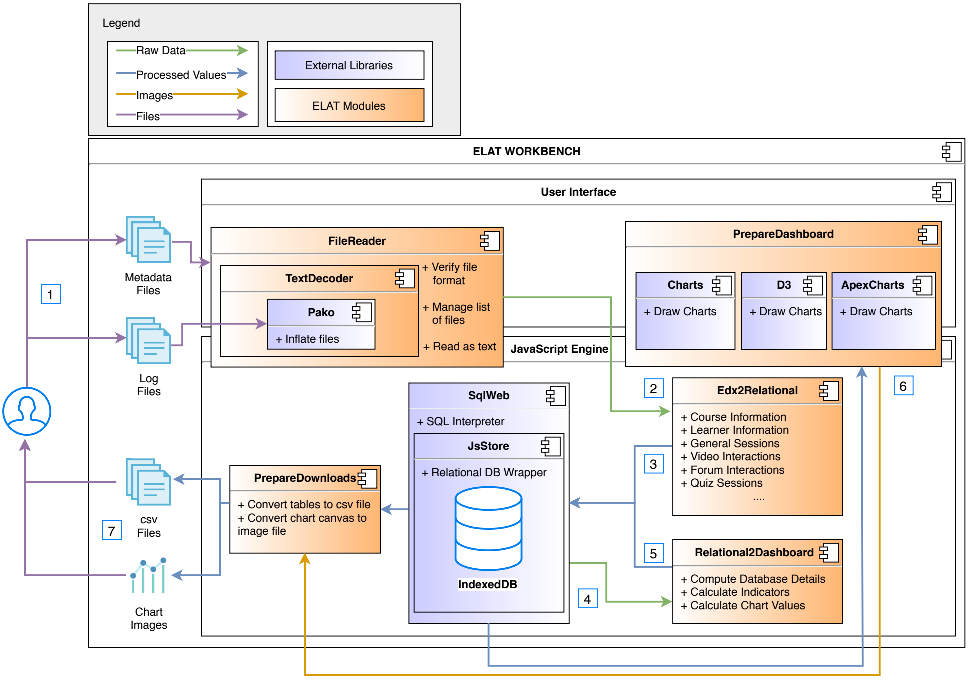

Figure 2.2: ELAT Architecture

log files (in gzip), we employ the pako[18] library, a zlib port to JavaScript.

Once the data has been read, it is now processed by the Processing module which translates the log files into MOOCdb's data model as already outlined in Section 2.1.2. The relational data is moved to the browser's persistent storage, that is, IndexedDB. As IndexedDB does not support fixed-column tables (like a relational database) out-of-the-box, we employ JsStore[19] as a relational wrapper for IndexedDB and the accompanying SqlWeb as a SQL query interpreter. This in turn allows us to formulate SQL queries to extract the desired learner information and aggregates.

As our runtime environment is the browser, we also need to discuss storage limits. There is the *global limit* the browser has, which is half of all available disk space on the local machine. Secondly, there is also the *group limit* for websites (to the browser, ELAT is just a website) of the same origin or domain—this limit is 20% of the global limit. As an example, if a machine has 500 GB of free disk space, the browser can use 250 GB for persistent storage (global limit) and 50 GB (group limit) for a single origin or domain such as ELAT.

Once the database instance has been set up, the Dashboard component is responsible for rendering a number of charts. The popular visualization libraries

[18] https://github.com/nodeca/pako
[19] https://jsstore.net/

`apexcharts.js`[20], `Chart.js`[21], and `D3`[22] are employed to populate the user's dashboard.

Lastly, the `Downloader` component is responsible for processing the data into `csv` format, which our end users can use as input to another program (R, SPSS, etc.). For instance, it is possible to plot the distribution of video watching sessions using the video interactions csv file, and even go a step further and use the course learner file to compare the plot of certified and uncertified learners. This sample is included in `ELAT`'s documentation [23].

2.1.4. User Interface Choices

The user interface consists of two parts: (i) the tabular overview (Figure 2.3) which provides basic information extracted from the metadata and log files and (ii) eight visualizations, which provide more specific insights and were chosen based on the recent literature and our interview with our stakeholders.

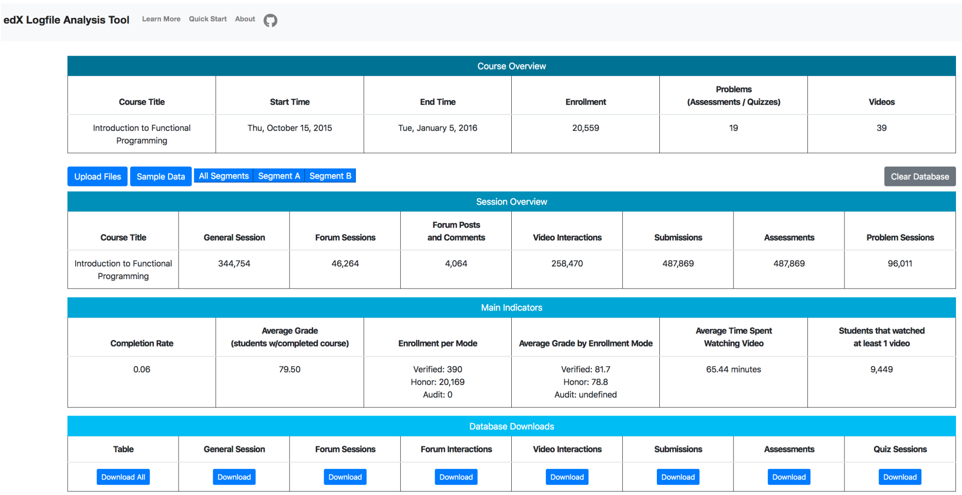

Figure 2.3: `ELAT`'s user interface: data can be uploaded to the browser for processing. Sample data is available as well. Once data has been processed basic course statistics are presented. Data can be filtered according to different conditions (e.g., based on learner IDs to evaluate randomized controlled trials). Data can be filtered and downloaded according to session type.

[20] https://github.com/apexcharts/apexcharts.js
[21] https://github.com/chartjs/Chart.js
[22] https://github.com/d3/d3
[23] https://mvallet91.github.io/ELAT/docs/home/

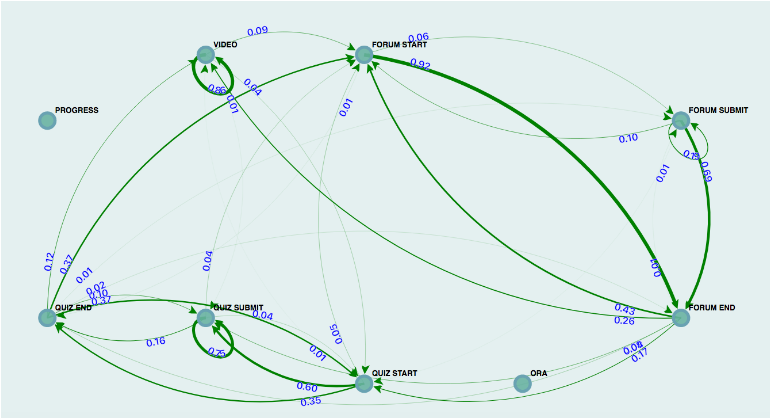

Figure 2.4: ELAT's learning path visualization for the *Functional Programming* course, based on [12].

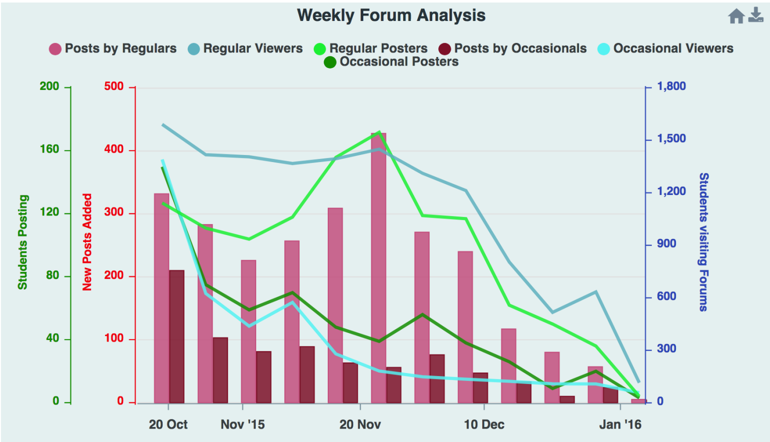

Figure 2.5: ELAT's forum analysis for the *Functional Programming* course, based on [37]

Due to space constraints, we here only discuss three of the eight visualizations (those based on recent research works):

- The learning path visualization (Figure 2.4) [12] provides an overview of the paths learners take through the different *types* of MOOC components. Each edX MOOC has a limited number of components (video, forum submit, quiz start, quiz submit and so on) and the transition probabilities from one to the next are computed based on all learner traces (though this can also be filtered according to successful/non-successful learners). In Figure 2.4, for instance,

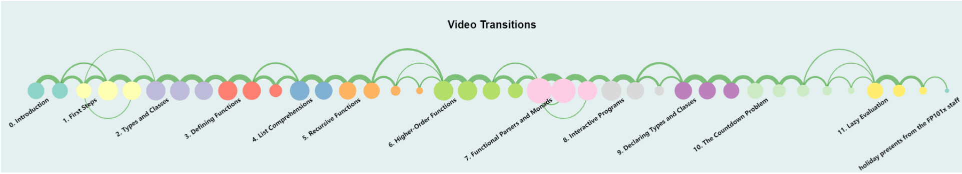

Figure 2.6: ELAT's video watching sequence visualization for the *Functional Programming* course, based on [12].

we observe that nearly 40% of learners transition to the forum once they have ended their quiz (presumably because they have questions and are looking for answers).

- The forum analysis chart (Figure 2.5) shows different types of forum posters (regular posters, regular forum readers, etc.) and how their numbers develop across the course weeks [37]. We find forum posts by regular posters to peek midway, while most occasional posters have contributed posts in the first weeks of the MOOC.

- The video watching sequence visualization (Figure 2.6) [12] presents information to what extent learners that pass the course follow the prescribed video watching learning paths. Each dot represents a video, colored by course unit, and the layout from left to right is according to the designed (i.e., instructor-provided) learning path. The width of the edges indicate the percentage of learners moving from one node (i.e., video) to the next.

The visualizations themselves are interactive and users can use mouse hovers, or select different segments, *such as passing or failing learners*, and date ranges, to receive more information on the different plots. In addition, as for the table view it is possible to filter learners according to certain criteria, such as separating learners into segments by their edX id (this is useful if interventions were randomly deployed to learners based on their edX ids). This allows learning researcher to for instance find out whether the inclusion of an intervention led to a higher passing rate.

2.2. Evaluation

Having described ELAT's goals and design, we now describe the two types of evaluations we conducted: a technical evaluation (§2.2.1) and a user study with potential end users (§2.2.2).

2.2.1. System Evaluation

In order to investigate to what extent ELAT can handle MOOC data, we employed ELAT on four edX MOOCs (cf. Table 2.2)—we selected those due to their different learner sizes (between 2.7K and 65K learners), the different years they ran (between 2015 and 2019) and the vastly different time interval they were open for enrollment (between 83 and 731 days). Recall that edX logs are organized by day, so the size of the log file depends on the number of MOOCs running concurrently by the institution. On average, the log file size is 50 Megabytes compressed and more than 1 Gigabyte uncompressed, with hundreds of thousands of click records.

Processing Times & Disk Space

We evaluated the processing time on two different machines with the most recent version of Google Chrome[24]:

Machine A Windows 10, 16GB RAM, Intel i7 @ 2.80 GHz

Machine B Windows 10, 8GB RAM, Intel i5 @ 2.90 GHz

and found the processing time and disk space use very similar. All functionalities of ELAT were successfully evaluated on macOS during the tool's development, but the technical testing for this paper was performed on Windows simply due to equipment availability. For each of the MOOCs we report the processing times (in minutes) and occupied disk space (in Gigabytes) in Table 2.2.

As expected, the disk space usage depends strongly on enrollment and course duration, with the largest MOOC (both in terms of days open and enrollment) requiring about 5 Gigabytes of disk space, easily manageable by any modern computer, according to the storage limits described in §2.1.3. On the other hand, the processing time is not only dependent on the duration of the course and the enrollment numbers but also on the *other MOOCs of the same institution* that are running at the same time (last column in Table 2.2) due to the way the edX log files aggregate all clickstream data of all running MOOCs in one file. Concretely, our smallest MOOC in terms of enrollment (*Robots in society*) has by far the most concurrently running MOOCs (184 over the course of the year) and thus its processing time is almost twice as long as those of the two MOOCs with only 40-50 concurrently running MOOCs. ELAT has to uncompress every log file and read each record in it to determine whether it matches the wanted course identifier. More concretely, the processing time of our four MOOCs varied between one and four hours.

Table 2.2: Overview of the courses employed in our technical evaluation.

Course	**Time Period**	**Total Days**	**Total Enr.**	**Cert. Stud.**	**Total Sess.**	**Proc. Time (min)**	**Disk (GB)**	**Concur. Courses**
Robots in society	04/18-03/19	322	2,668	18	20,371	129	0.06	184
Creating powerful political messages	01/16-01/17	353	15,002	238	114,500	77	0.41	49
Functional programming	10/15-01/16	83	20,559	1,149	344,754	54	1.23	40
Solar energy	08/16-08/18	731	64,667	582	845,263	324	5.08	78

[24] ELAT also works well on Firefox. As ELAT makes use of some rather recently introduced web APIs not all browsers support all web APIs completely as of the time of writing.

Data Issues

In the process of developing and evaluating ELAT, we encountered a number of data issues, two of which we now describe in greater detail: (i) mobile, and, (ii) open response assessment. They can be attributed to the facts that our MOOCdb code base was developed a number of years ago and the lack of documentation on edX's parts.

The sharing of log data is not one of edX's priorities, as reflected by the small amount of provided documentation. For instance, the changes in the logs' structure is shared with researchers in the edX Release Notes[25], but the last update (at the time of writing this paper) is March 2017. Consequently, special cases in the records have to be handled as they appear.

Concretely, edX mobile (a native app for Android/iOS) was introduced in 2015. Learners using the mobile app can be distinguished by the slightly different logs they generate (one may expect a particular mobile flag in the logs, but this is not the case). Specifically (as we found by digging into the log file format), the format of the `details` field of a video record is a string for a non-mobile interaction, while for a mobile record it is a nested record (JSON)—and thus, ELAT has to identify the type of video record and process the detail accordingly. This is just one example of the edX log file format (slightly) changing over time.

Our second issue refers to the open response assessment (ORA). This type of assessment exists in edX but was not part of the MOOCdb code base. As in our initial interview with potential end users the importance of ORA data came up repeatedly, we extended the MOOCdb data model to incorporate ORA events as well. In this manner, ELAT also aggregates information on the number of times the learner saved his/her solution during the session, if they submitted a solution during the session, how many peers they reviewed, etc.

2.2.2. User Evaluation

For our user evaluation, we recruited seven participants through internal mailing lists and academic contacts that were required to have some experience in the area of learning analytics, preferably having worked with edX data logs before. The participants completed the study online—we provided each participant with a link to ELAT they could open in their browser as well as a link to the online form we used for the pre/post questionnaire. Each participant was asked to complete three steps:

1. Fill in an initial questionnaire: basic demographics, background and experience.
2. Execute a task: the ELAT version we provided included edX log files of the *Functional Programming* MOOC (cf. Table 2.2) and participants were asked to explore the data with ELAT and write down 10 insights they gained.
3. Complete an exit questionnaire: we asked participants about the major insights they gained about the MOOC, their opinions about specific features of ELAT and employed the User Experience Questionnaire (UEQ) [38].

[25] https://edx.readthedocs.io/projects/edx-release-notes/en/latest/

We did not pay our participants and advised them that the experiment would take about an hour of their time. Our participants spent on average 33 minutes in step (2).

Participants

All seven participants (four females and three males ranging in age between 25 and 44 years) assume a major role in learner research at their respective universities. Four of whom are data analysts and consider themselves as advanced in analyzing MOOC learner data whereas the other three consider themselves as novices in this respect. Except for two participants, all have experience in working with edX data.

Our participants work with MOOC learner data for various reasons: ranging from designing feedback interventions to investigating learners' engagement, learning behavior and learning outcomes. Participants cited the understanding of the log data format, cleaning and preparing data for analysis, identifying relevant insights in the sea of data and visualizing the results as major challenges in the analysis of learner data.

User Experience Questionnaire

For a standard measure of the participants' experience with `ELAT`, we applied the well-known UEQ [38]. It consists of 26 bipolar items rated on a 7-point Likert scale, to measure 6 factors of user experience: *attractiveness, perspicuity, efficiency, dependability, stimulation, and novelty*. The results obtained from the questionnaire are shown in Figure 2.7. Considering the small number of participants, this is not a conclusive result, however the UEQ has been shown to be applicable to small groups [39], and it is useful to form an idea of `ELAT`'s strengths and weaknesses. `ELAT` scores particularly high on the *stimulation* and *novelty* factors; this is very motivating for the future development of `ELAT` as we envision `ELAT` not to be the final step in a learning research effort but instead as a means to conduct this research more efficiently, develop research hypotheses and explore the data. The least positive results on *perspicuity* (i.e., clarity of presentation) can be explained by the relatively small amount of time (33 minutes on average) participants spend on the tool in comparison to the amount of information (both in tabular as well as plotted form) available to them, as well as the fact that `ELAT` is a prototype and in need of a few more iterations to improve the visualizations' intuitiveness. The high error margins (portrayed by the black lines in Figure 2.7) can be explained by the difference in self-reported learning data analysis experience: four participants consider themselves advanced while three self-identify as novices.

Insights gained with `ELAT`

In the exit questionnaire we asked our participants to list their ten most important insights obtained about the MOOC. The three recurring themes were forum interaction, video transition and learning paths:

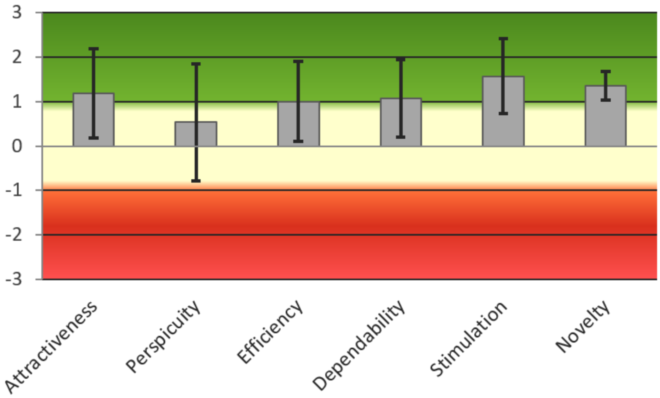

Figure 2.7: Average UEQ Results for `ELAT`

Forum interaction Participants found the forum participation analytics (regularity of viewing, posting and size of posts) and the graphic visualizations very useful to see correlations between regular viewers/posters, occasional viewers/posters and their final grades. Further, participants appreciated the customization of the visual analytics according to desired time frame e.g., weekly, monthly which afforded further investigation on learners' allocation of time in viewing and posting activities and its implications on their final grades.

Video transition The provision of video transition analytics for passing and failing learners enabled participants to gain in-depth information on how and when the two groups of learners adhere/deviate from the prescribed path in video watching. Participants were able to state some core results, e.g., one participant wrote that *"learners who watched the videos sequentially have a higher chance of passing"* and another found the video interaction graph very useful as *"it shows the difference between the instructor views on the sequence of videos and the reaction/perception of learners to this sequence in practice."*

Learning paths Participants found the learning paths analytics useful for comparing how (and when) passing and failing learners interact with the course elements such as video, forum and quiz. For instance, one participant commented that the transition between forum and quiz is higher for failing learners which might imply that they are seeking help in the forum. Another participant found the learning path feature useful in investigating how and when failing learners deviate from the designed learning path which could enable possible interventions or remediations before learners dropout or fail the course.

Features and Functionalities

A number of questions in the exit questionnaire pertained to the features and functionality of `ELAT` as an analysis tool. Participants' comments chiefly revolved around the customizable and downloadable graphs, downloadable session database and the main indicators table (as seen in Figure 2.3). Most of the participants were positive except for one who found the features confusing, in particular, the graphs. Five participants found the overview and main indicators table helpful to quickly access descriptive statistics and course information. Two participants commented that the downloadable session database facilitates the transformation of data logs into comma-separated values (csv) files which afford further in-depth analysis and another found ELAT to complement the current instrument he uses in the analysis of MOOC data well. Five participants found the customizable and downloadable graphs for forum interaction, video transition and learning paths were most useful and effective amongst all the available features of ELAT.

Recommendations

Most of the participants' suggestions for the improvement of ELAT centered on the visual analytics of the forum participation, video interactions and learning paths. Participants requested for provision of more in-depth information e.g., the distribution of failing learners in their engagement with the various course components. One participant recommended the inclusion of content analysis and social network analysis features to afford a more wholesome investigation of the patterns and trends within the forum activities. Other suggestions involved format aspects such as labeling, legends and colour schemes of the line graphs and tables, as well as a provision of course structure elements in the place of the timeline on the x-axis. The latter will afford an evaluation of learners' behavioral responses in relation to the course elements. Likewise, the type of enrollment (audit, honor or verified) apart from passing and failing could better inform learners' behavior in their interactions with the various course elements. Overall, most of the participants see the potential in ELAT and express keen interest in using ELAT for future analysis of MOOC learner data.

2.3. Conclusions

This paper presents ELAT, an open-source, privacy-aware and browser-based edX log data analysis tool. The primary goal in the design and development of ELAT is to equip and to empower researchers with less data processing experience to effectively and efficiently analyse large MOOC data sets—in contrast to existing tools, ELAT requires no computational knowledge to set it up.

Our system evaluation has shown that edX MOOCs with tens of thousands of learners can be analyzed within the browser, within a few hours of time on a standard machine.

From our user study, it is evident that the value-added features of ELAT (forum interaction, video transition and learning paths) have enabled researchers to tease out important findings on learners' learning progress as well as learning experience easily and effectively.

Most of our user study participants found the customizable visualizations of learner engagement and the downloadable session data for further analysis particularly useful. On the same note, these features would also be useful for instructional designers to make appropriate design decisions for various target learners and learning objectives. Likewise, for course facilitators, `ELAT` could be instrumental to monitor learning progress and to appropriate effective measures of interventions when learners show signs of prolonged disengagement.

At the time of writing, we are exploring compatibility with other major MOOC platforms, first Coursera then OpenEdX, to facilitate more wide-spread adoption and evaluation of `ELAT`. In the future, we plan to take up our participants' suggestions and conduct a more thorough user study. Specifically, we are planning a replication study by asking learning researchers to reproduce (part of) their previous work on edX log data with `ELAT`, allowing us to assess their performance and experience in a more natural study setup.

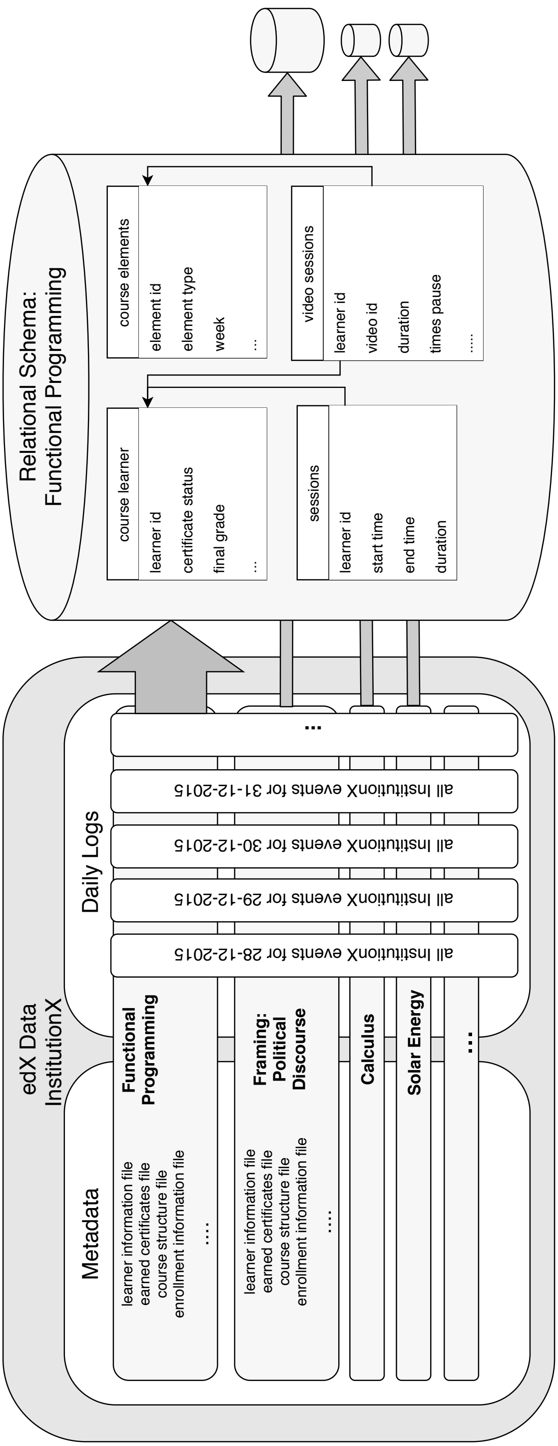

Figure 2.8: edX data traces are converted to a relational schema which is based on MOOCdb. edX log files aggregate all clickstream data generated on a single day across all running MOOCs offered by an institution. The relational schema is generated for each course separately.

3

The Sequence Matters

A Systematic Literature Review

Manuel Valle Torre, Catharine Oertel and Marcus Specht

A sequence works in a way a collection never can

George Murray

Describing and analysing learner behaviour using sequential data and analysis is becoming more and more popular in Learning Analytics. Nevertheless, we found a variety of definitions of learning sequences, as well as choices regarding data aggregation and the methods implemented for analysis. Furthermore, sequences are used to study different educational settings and serve as a base for various interventions. In this literature review, the authors aim to generate an overview of these aspects to describe the current state of using sequence analysis in educational support and learning analytics. The 74 included articles were selected based on the criteria that they conduct empirical research on an educational environment using sequences of learning actions as the main focus of their analysis. The results enable us to highlight different learning tasks where sequences are analysed, identify data mapping strategies for different types of sequence actions, differentiate techniques based on purpose and scope, and identify educational interventions based on the outcomes of sequence analysis.

This chapter was published in LAK24 [40].

3.1. Introduction

Educational Technologies (EdTech) have rapidly scaled teachers' capacity for instruction, for example through live, virtual and recorded lectures [41, 42]. EdTech, has also increased the opportunity for learners to actively engage with learning materials, from practice problems in electronic environments to automated assessments, or annotation tools [43, 44]. However, some processes that link teaching and learning in a continuous cycle, such as formative assessment, and personalized support and feedback, have not scaled at the same rate [45]. From its origins, Learning Analytics (LA) intends to close this cycle using data generated by learners in educational systems, and processing it to obtain insights grounded in learning theory and provide (semi) automated learning support and feedback [46].

Existing implementations of LA, such as dashboards or nudges, are often based on descriptive analytics of performance metrics [1], designed for summative assessment and (self) monitoring [47]. However, analytics of the learning process with the detail level to enable timely assessment and support are an ongoing development [45]. Learning is a process that unfolds and changes over time, and aggregating or counting the actions the learner executes is limited in its narrative power [48]. For that reason, it is critical to understand the execution of learning tasks as a temporal and sequential notion [49]. Using sequential data to observe, understand, and support learning is not exclusive to LA, but has a history in fields such as Educational Data Mining (EDM), Intelligent Tutoring Systems (ITS), and Artificial Intelligence in Education (AIED) [50]. This results in a lack of common definitions of concepts such as learning sequences or sequence units. Furthermore, there are no established best practices of the analysis methods that can be applied for specific purposes, educational settings, or scenarios [51]. As a consequence, research is often in the exploratory stage, using data from scenarios where the analysed course or task is over, without intervention [52].

In this article, we work towards a broader understanding of the variety of definitions, data points, methods, and interpretations used for the analysis of learning as a sequence. We aim to enable a broader understanding of the learner context to be taken into account for intelligent feedback and analytics interventions. In the following, we will first introduce the state-of-the-art of learning sequence analytics and then present the analysis of 74 articles considering the given dimensions.

3.2. Sequence Analysis and Learning Sequences

Sequence Analysis (SA) of human behaviour, where a sequence is an ordered list of events, allows the contextualization of these events, their relationships with each other, and the effects of part of or the whole sequence [53]. SA is especially useful to research phenomena that occur as a process across time, where the use of static attributes and frequency rates may not be informative enough to understand their development [53]. The analysed sequences can include actions executed by a person as well as reactions of others in their environment, i.e. a peer or an intelligent agent.

Sequences can have different interpretations in education, in this review, we focus only on the sequences of actions that learners execute during a learning task. For example, the steps followed to solve a problem [50] or the sequences of answers that learners provide in guided practice environments, and how certain sequences may relate to learners' behaviours [54] or characteristics [55].

3.2.1. Sequences of Actions in Learning Tasks

Learning tasks refer to situations when learners actively engage with the learning materials which require them to integrate domain-specific knowledge and procedures with more general critical thinking and analysis skills [56], such as solving complex problems [57] or scientific inquiry tasks [50]. For tasks in different domains, there are behavioural patterns that appear with different levels of practice. For example, experts in a domain may show deeper categorization of a problem, use forward-thinking strategies in their execution of the solution or iteratively evaluate intermediate solutions [56]. In these tasks, learners have a large number of operators or actions available in the problem space, or learning environment, making the combinations leading to a solution difficult to observe. Furthermore, defining and describing such combinations is not trivial, since they are "automated", or stored in one's memory as procedural learning, making it difficult to explain by experienced individuals [56]. As a consequence, the assessment and monitoring of learners are particularly difficult for teachers, especially considering the scale provided by educational technologies and learning platforms [58]. The data produced by these systems, however, can be leveraged to scale the capabilities of teachers, by analysing sequences of learner's data records to identify behaviours and reveal learning processes [59].

3.2.2. Trace Data Sequences and Analysis

The reliability of trace data comes from its ability to express dynamic events, more than as aggregated, static aspects of a measure [50]. Using analytics to find patterns in the sequences that indicate specific tactics or common issues can help practitioners obtain insights into the use of different strategies. Such patterns can be a set of learning actions that occur in succession for a proportion of the learners, from partial segments to complete sequences. In addition, given that the patterns are extracted from sequences of actions, they can be interpreted by the teacher and even learners themselves [60]. For example, in a 3D geometry game where learners can insert, remove, and rotate shapes to solve puzzles, teachers cannot assess nor give feedback to learners based on the gameplay records without assistance. Common patterns that lead to a failed assignment can be identified as mistakes and teachers can prioritize which to address [61].

Analysing sequences of trace data throughout a learning task offers practical advantages as well [62]: when systems process data traces as sequences of learner actions, there is a constant input of information to incrementally update the learner model, enabling the system with the flexibility to constantly adapt to the learner's level or intervene whenever necessary [63]. This allows practitioners, and eventually, an intelligent system, to predict if the current path is trending towards an undesirable outcome, to determine which action patterns are leading to that result and how to leverage these insights to optimise learning [64, 65].

3.2.3. Related Work

The analysis of learning sequences is emerging from several fields, and while there are no surveys specifically on the topic, there are related works from the EDM, LA and AIED communities. Regarding the use of educational data to observe learner behaviour and predict performance, Xiao et al. [66] analyse EDM methods in the age of big data to discover the trends around the complete data mining pipeline, such as data collection and preprocessing, feature engineering, classifier and ensemble method selection, and interpretation of results. While they mention techniques that analyse sequential data, there are no definitions of learning sequences or sequence units. On the other hand, the survey by Bogarín et al. [67] focuses on Educational Process Mining (EPM), a field that studies the process-oriented view of EDM, particularly as an end-to-end process. The authors describe Virtual Learning Environments (VLE) and the Event Logs they produce as valuable data sources for sequential analysis, as they are detailed records of learners' actions. They mention methods handling learning as sequences, including Sequential Pattern Mining, but the definitions outside of process mining are not addressed.

Regarding the use of trace data to observe learners' actions, the review by Wang et al. [50] analyses research on the use of log data from Open-Ended Learning Environments (OELEs) to understand scientific inquiry and problem-solving. Their conceptual framework consists of 4 points: the competencies measured, mainly problem-solving and inquiry; the observations used, ranging from multiple-choice questions to log data; the feature extraction and modelling processes and the interpretation of the analysis. While the focus is not only on sequential methods, almost half of the 70 reviewed papers use sequential features for analysis, with 29 applying pattern mining to identify behaviours. The review by Nadimpally et al. [68] surveys Artificial Intelligence techniques for adaptive learning systems, identifying two main types: to model learner behaviour and to organize learning content. Concerning modelling learner behaviour, they found that the most common methods are Similarity-based Pattern Recognition, Probabilistic Graphical Models, and Deep Learning: all of them capable of processing sequential data. Existing literature explores the growing use of temporal data to identify, classify or predict learner behaviours, which often involves sequence analysis. However, a focused study of when to use sequence analysis, the data units that can be leveraged and which methods to implement, is less covered.

3.3. Research Questions

Addressing the shortcomings from the previous section, this study intends to establish a common ground for Sequence Learning Analytics from the literature, so researchers can use this to find similar works, compare their approach to the state-of-the-art, and leverage insights of existing work, allowing the community to also focus on the intervention design. In this review, a **learning sequence** is defined as an ordered list of context-specific actions, with each **unit** being a systematically mapped learning action from one or more data traces. The following questions guide our analysis:

1. How are learning sequences analyzed to describe and support learners' actions in computer systems?
    a) Which types of *learning tasks* are analysed as sequences?
    b) What is the motivation or *purpose* when using learning sequences?
    c) Which *actions* are used as the learning sequence units?
    d) Which *methods* are used to analyse learning sequences? And what are the interpretations of the patterns that result from them?
    e) Which *educational interventions* are designed with the obtained insights? Which learning principles are used to support and evaluate them?

3.4. Method

In this review, we followed the Preferred Reporting Items for Systematic reviews and Meta-Analyses (PRISMA) [69], to answer the questions above, in Section 3.3. The queries, information sources and eligibility criteria are described in this section, followed by the document collection and inclusion process, and the data extraction method.

The queries were executed in SCOPUS and Web of Science (WoS), for the domains of Educational Data Mining (EDM), Artificial Intelligence in Education (AIED), Learning Analytics (LA), Intelligent Tutors and Learner Modeling, exploring the fields of title, abstract and author-defined keywords. SCOPUS and WoS were selected as they include most of the top publications in such domains, such as the proceedings of the conferences on Learning Analytics and Knowledge, Educational Data Mining, and the International Conference on Computers in Education, as well as the Journal of Learning Analytics, British Journal of Educational Technology, Journal of Educational Data Mining, and Computers and Education: Artificial Intelligence. The queries were executed on September 2023, including articles from 2010 to 2023, and they focus on the use of sequential analysis in an educational setting, where the obtained patterns are leveraged for an educational intervention. Two queries were executed, one per database, using the query below, with the results described in Section 3.4.2.

("data mining" OR "artificial intelligence" OR "learning analytics" OR "intelligent tutor" OR "learner model*") AND (sequen* OR procedur* OR series) AND (educatio* OR student OR teacher OR instructor) AND (feedback OR assessment OR predict*)*

3.4.1. Study Selection Process and Eligibility Criteria:

The results were included if they were full-length articles, published in a peer-reviewed source, and written in English. This was followed by two rounds of screening, the first one based on the abstracts and the second on full text for those that were unclear from the first screening. To be included, the publication had to be implemented in an educational setting, where the learning sequences refer to an ordered series of actions during a learning task. Non-empirical works are excluded, as are works that only mention education as part of the institution name or a possible use case. Additionally, publications were excluded if they analysed unrelated sequencing, like proteins, or where sequences were part of the methodology, such as "a sequence of trials" or "a series of interviews". Finally, works exploring learner patterns across multiple tasks were excluded, such as navigation patterns on the elements accessed in MOOC or the courses in a curriculum.

3.4.2. Data Collection Process and Data Items

Reference lists were exported from SCOPUS and WoS, and then transferred to Google Sheets for screening. The reason for exclusion or inclusion was recorded for each article, and the values are shown in Figure 3.1. Of the articles obtained from the queries, 953 were unique, then in the abstract screening 717 were excluded since they were not in an educational setting, not empirical work, the use of learner action sequences was not part of the analysis or did not focus on a single learning task. The full text of the remaining 236 articles was screened, and 162 were excluded based on the criteria described above, resulting in 74 articles included. Document collection and inclusion were performed by the first author, using the criteria previously defined with the rest of the team.

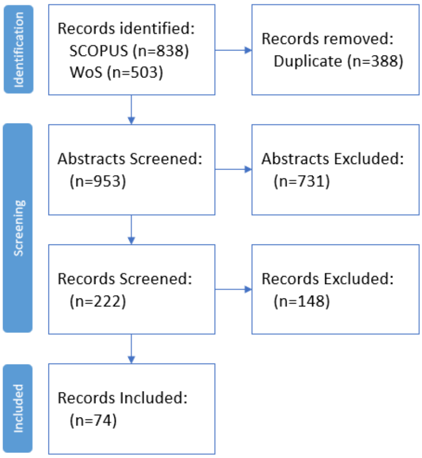

Figure 3.1: Document Flow Diagram

3.4.3. Analysis Procedure

The included studies were coded by the first author using ATLAS.ti (a qualitative research software) with the Research Questions from Section 3.3) as a guide, following negotiated consensual validation with the rest of the authors to iteratively review the codes [70]. For each question, we determined 3 or 4 possible categories from the related work in Section 3.2.3, and then the full text of the articles was reviewed and coded using the questions and categories. If a code answered one of the questions but did not fit any of the categories, a new category was created. Additionally, a second coder reviewed a sample of 10% of the articles and the results were compared to ensure the reliability of the coding process, which showed an inter-rater reliability kappa of 0.922, where any differences were discussed and addressed.

For the first question, we identified the learning tasks considering the educational construct they are based on: deliberate practice, solving a problem or scientific inquiry [50]. The first one refers to tasks where learners practice a skill through repetition, usually with a system tracking their progress and adjusting accordingly. The second one refers to tasks where learners have to solve a complex problem in a virtual environment, and the third one refers to tasks where learners have to obtain and aggregate information.

On the other hand, the purpose of sequence analysis can be defined similarly to the goal of data mining and analytics tasks, such as clustering, classification or prediction [66], in line with the findings in [50]. While all the reviewed works identify patterns using sequence analysis, some of them go further and compare the patterns between groups of learners, for example, to identify behaviour differences among learners who received different experimental treatments, or to classify them by the use of strategies. Finally, some works use the identified patterns to predict the next action of a sequence, the success of such action, or a learner's state.

Regarding the units of the sequences, the reviewed research uses data sources such as log files or event records [71], which have to be mapped to representations of meaningful actions, to be analysed and understood as part of the learning sequence [72]. These sequence units can categorised as learning actions, which are domain or system-specific actions such as selecting and using a system tool in an OELE [50], or metacognitive actions, usually defined based on an educational theory, such as self-explanation [73]. The third category is problem attempts, the series of submissions in a guided practice system [68, 74].

For the implemented methods, we obtained the categories of methods that use sequential data from related literature on AI in Education and Process Mining [67, 68]: Probability Modeling, Deep Learning, Sequential Pattern Mining, Educational Process Mining and Knowledge Tracing. Additionally, we identify the scope of the resulting sequences as the reference for interpretation, from system-specific tactics to general strategies for a complete task [67].

Regarding interventions and educational theories, we seek to identify which works go beyond the observation and exploration stage, which is why we included *assessment, prediction and feedback* in our query, in Section 3.4. However, considering the challenge of implementing and evaluating an educational intervention [52, 75], we classify if articles implement an intervention, or if they explicitly mention a potential intervention or not. Additionally, the impact of interventions relies on a strong theoretical foundation [76], so we identify the educational theories used to support or interpret the results of sequential analysis [77].

3.4.4. Limitations

The search was limited to Web of Science and SCOPUS, while these libraries include a wide range of publications, the results are not exhaustive. Additionally, there are no established practices to mention sequence analysis in the title, keywords, or abstract, in consequence, some articles may not have appeared in our queries. Furthermore, abstracts were used to assess the first round of inclusion, there is the possibility that articles were erroneously excluded. Regarding the analysis, the articles were coded only by the first author, however, we addressed this limitation by following negotiated consensual validation with the author team [70] and assessing the agreement of a sample with a second coder.

3.5. Results

Our analysis included 74 articles, with the number of publications doubling in the second half of our search range, suggesting a constant increase in interest in the field, shown in Figure 3.2. The articles were analysed following the Research Questions in Section 3.3, to establish the trends of the different types of learning tasks, the different purposes of the sequence analysis, the types of units and analysis methods used, and any implemented interventions. The following sections address each of these questions, with the results available as an open-access dataset [78], and as an interactive interface at https://mvallet91.github.io/sequence-matters/.

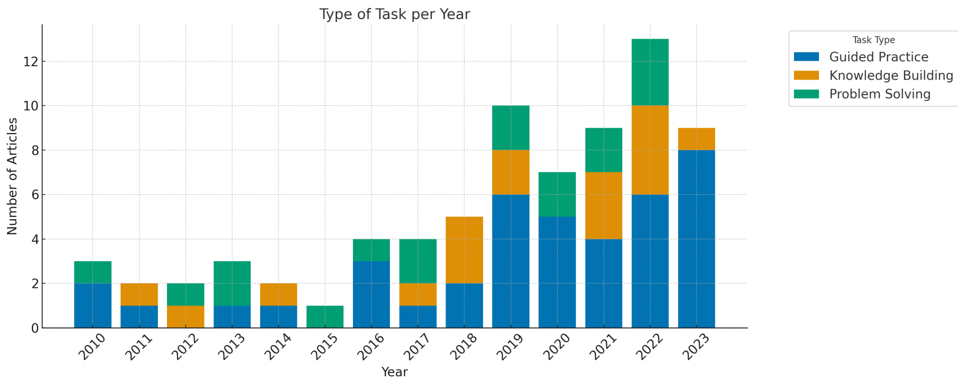

Figure 3.2: Articles, and Task Types, per Year

3.5.1. Which types of learning tasks are analysed using sequences?

This review focuses on tasks that require active engagement from the learner, which instructional design models, such as the 4C/ID model [57], consider as authentic problems and practices. However, they can follow different educational designs. We identify three main types of tasks: guided practice, problem-solving, and knowledge building; the yearly number of publications is shown in Figure 3.2. The first type is *guided practice*, characterized by a set of practice questions that are part of a topic, where a system uses the sequence of answers to constantly measure and adapt to the learner's knowledge level. We registered 40 articles of this type, 2 of them using dialogue [79, 80], and 2 as a quiz [81, 82].

The second type of task is *problem-solving*, a complex task integrating domain knowledge with domain-specific and general skills. In these tasks, the learner's interactions with the system can be interpreted as the operations executed towards a solution state [83]. While these tasks generally have a defined objective, they can be satisfied with different sequences of actions or applied operators, from a wide variety of actions available within the educational system. The 17 articles using data from a variety of problem-solving tasks, including programming assignments [84], mechanical engineering problems [85] or procedures in a biotechnology virtual lab [65, 86].

Finally, some of the tasks require that the learner collects and processes information, from video lectures or reading materials, and then generates and evaluates a representation of knowledge, such as a concept map [87]. In these tasks, the final result may be similar for learners who followed different strategies, or sequences of steps, and would benefit from personalized feedback. We identified 17 articles that can be classified as *knowledge-building* tasks, for example, where a learner interacts with articles on climate change and then generates a causal model to explain the relationships between the learned concepts [88].

In summary, we identified 40 guided practice tasks, 17 problem-solving and 17 knowledge-building tasks, with the first type gaining ground in recent years, shown in Figure 3.2. This may be due to the introduction of increasingly sophisticated deep learning methods or the availability of public datasets [74, 89, 90], eliminating the need to design new systems or collect data. Additionally, it might be related to the research objectives, described below.

3.5.2. What is the motivation or purpose of using learning sequences for analysis?

There are many affordances to the use of learning sequences and sequence analysis methods in EDM and LA, such as the ability to extract the sequential patterns from data [91], to identify them as tactics or strategies [92], or to include temporal structures in behaviour analysis [44]. Most articles identify the patterns as a learning behaviour, tactic or strategy, however, some of them go further and use these patterns for classification or for prediction. We found 8 cases of *pattern identification*, for example, as the self-regulatory processes that occur before a help-seeking event. These patterns can be used to anticipate the struggle of learners who may not use help-seeking tools correctly [93]. Another case of identification is when groups of learners receive different experimental treatments, such as a feedback strategy [94], and then the patterns of the groups are compared. This means that in these 6 cases, the identified sequences are used as part of the evaluation of an educational intervention [95].

Going a step further, in the 24 articles where the purpose of the sequence is to serve for *classification*, researchers first separate the learners by a given characteristic, such as proficiency in the task [96] and then identify the patterns per group. In these cases, like in traditional data mining classification tasks, a small percentage of the data is used to evaluate the accuracy of the method, or how well can it classify a learner by their behaviour [96–98]. In only one article the grouping is done by applying clustering algorithms on the sequences of all learners, and then samples of the groups are reviewed and labelled as a certain strategy, such as *Efficient Completion* [92].

Regarding *prediction*, the purpose is to use a sequence as input but there is a variety of outputs: in 8 cases it is the learner's performance, such as success or failure on the task [85, 99], in 2 is the learner's state such as emotional or cognitive states [55, 63] and for 1 it is the next action they might do [65]. For those works focusing on prediction, this is often with the vision of an early warning system or even automated interventions. Finally, in 24 cases, the purpose is to predict the success of a learner's next action, often to determine their knowledge level for a given topic [100].

3.5.3. Which actions are used as the units of learning sequences?

In the analysed articles we found a broad variety of data records logged, including clicks on a web interface, sensor data from haptic devices, and text input in command line interfaces. A single case uses the data as-is, as the authors consider them detailed enough for the sensor data collected from a dentistry virtual reality simulator [97]. In the rest of the publications, we found three different types of combining data records into traces: learning actions, problem attempts, and meta-cognitive actions. We identified 26 articles that use *learning actions* as units, which usually represent the different ways that learners apply operators in *problem-solving tasks*, defined in Section 3.5.1, such as editing code, executing a command, or using a tool. Mapping the data records into such actions allows researchers, teachers and students to easily analyse and interpret the results of sequence methods. The downside is that this mapping usually requires expertise with the domain and system, to know which actions are possible, and it also limits the possibility to generalize the analysis and results.

In the second case, identified in 37 of the reviewed works, the collected data records are better portrayed as *problem attempts* because they are always delimited by a submission event: when a student submits a response to be evaluated by the system, usually named Intelligent Tutoring System (ITS), and they correspond to those using the *guided task* type. Considering that there is a clear definition of where one unit starts and ends, there is little focus on the process of mapping data records to units.

Finally, for the 10 articles using *meta-cognitive actions*, we found that they introduced a conceptual background from different learning theories, and then use the selected theory to construct the sequence units. Eight of them use a Self-Regulated Learning (SRL) model to map certain records into a meta-cognitive action. From the rest, one maps the data to meta-cognitive states, which they refer to as on-task and off-task states [94], and the other to a variety of meta-cognitive and cognitive behaviours, such as evaluating, help-seeking, and skipping video [101]. Using conceptually mapped actions reduces the granularity and length of the sequences to be analysed since they usually aggregate data records into just a few possible options, compared with the simpler learning actions above. Additionally, this means less variability during analysis and a more direct interpretation.

The sequence units used for analysis are strongly related to the type of learning task, the 26 articles with learning actions tend to be on problem-solving, using specific actions in specific problems, and the 37 using problem attempts are generally on guided practice tasks, focusing more on the accuracy of the learner's model. Finally, the 10 cases of meta-cognitive actions are usually in the knowledge-building tasks, focusing more on self-regulated learning processes.

3.5.4. Which methods are used for the analysis of learning sequences?

The methods in the reviewed articles usually take a set of sequences as input, where each one corresponds to learning actions executed by one learner during a task. The methods that we found range from the manual analysis of game logs [102] to deep learning models capable of predicting the performance of the learner [85], with many others among them. In our analysis, we classify the different methods into 5 categories: Sequential Probability Models, Educational Process Mining, Deep Learning, Knowledge Tracing, and Sequential Pattern Mining (SPM).

Sequential Probability Models estimate the probabilities of transition between elements in a sequence, based on the previous one or more elements [103]. On the other hand, *Educational Process Mining* refers to the EDM technique to discover, visualise, and compare process models, focusing on the complete sequence [96]. *Deep Learning* methods use Artificial Neural Networks (ANN) to predict the outcome of a sequence, mainly Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) networks, as they are better suited to handle sequential data than other ANNs [99]. While *Knowledge Tracing* approaches leverage probabilistic, machine learning or deep learning techniques, their main focus is to model the knowledge evolution of a learner throughout their interaction with an educational system [104]. *Sequential Pattern Mining* methods refer to a group of data mining techniques used to find statistically significant patterns—or subsequences—in sequence datasets, such as actions that happen frequently in the same order across learners' records [95]. *Manual Analysis* methods are often used in combination with other methods but are also used on their own in a few cases, for instance by reviewing and classifying the actions of individual learners in a small group [55]. These methods often relate to the scope of the analysis, from the full-task strategies shown by a learner population to short tactics that certain students apply depending on context: the Macro-level Patterns identify strategies that span across the complete task and are analysed for a population or group. The Meso-level refers to behaviours that appear in large part of the task, but the focus is on the individual

Scope	Method Category	Analysis Method	References
Macro-level Patterns	Educational Process Mining	Fuzzy mining, PetriNet, Process Maps	[81, 96, 105]
	Other Methods	N-gram processing, Network Visualization	[58, 91, 106]
	Sequential Probability Model	Markov Chain Model, Markov Chain Process, Hidden Markov Model, Microsoft Sequence Clustering, Markov Logic Networks, Probabilistic Automata and Hidden Conditional Random Fields	[43, 60, 65, 73, 79, 86, 88, 97, 103, 107, 108]
Meso-level Patterns	Deep Learning	GRU, MLP, CNN, LSTM, SAE	[41, 63, 82, 85, 99, 109–111]
	Knowledge Tracing	PFA, ModPFA, BKT, DKT, DKVMN, AKT, and many more	[74, 89, 90, 100, 104, 112–129]
	Other Methods	Dynamic Time Warping, Model-free bounded optimization, Course-agnostic Student Performance Model	[92, 130, 131]
Micro-level Patterns	Sequential Pattern Mining	SPAM, Differential, Temporal Interestingness of Patterns in Sequences (D-TIPS), Pex-SPAM, Generalized Sequential Pattern Mining, Frequent Item set,	[54, 61, 62, 80, 87, 95, 98, 132–138]
	Manual Analysis	Manual Labeling, Sequence of Action Tables, State Permutations	[55, 84, 94, 139]
	Other Methods	Visualization, Lag Sequential Analysis, GSEQ (Generalized Sequential Querier), Time-embedded n-gram	[44, 101, 140, 141]

Table 3.1: Methods per Category and Scope

level. Finally, the Micro-level patterns are short tactics that occur in a particular context. Shown in Table 3.1, we group the examples of each method and the scope that they belong to.

3.5.5. Educational Interventions - What is the final intervention designed using the obtained insights?

Educational interventions provide students with the support required to acquire the skills and knowledge being taught. However, interventions remain one of the main challenges in the learning analytics cycle [75]. This is reflected in our findings, where most reviewed publications are exploratory studies with potential interventions, for instance, as an early warning system that can notify teachers about learners using tactics that lead to poor performance [133], or automated systems that can detect if a learner might go into a blocking state [63]. There were also 5 prototype teacher dashboards [61, 81, 134, 140, 142], with one of them receiving a positive response from a usability survey with 56 participants [106].

In our review, we found 2 articles that design and implement an intervention. In the first, the authors integrate the patterns obtained to provide suggestions to students in a programming tutoring system. The suggestions included reading the manual or asking a peer for help when an error state is identified, to nudge the learners into a self-regulation strategy, for example, *reflection-then-success* after a sequence that could lead to failure [54]. The impact of this intervention, however, is not evaluated. The second one implemented and evaluated the use of learning sequences, by creating a “good-learner path" model using a Markov Decision Process in a guided practice system, so it can automatically generate the content sequence for first-time users. The model was then evaluated by comparing the test results of learners who followed the created path and learners who were provided with exercises one by one based on their prior results. Their approach produced 19.5% more good learners and 20.5% lower standard deviation than the control group [108].

3.6. Discussion

3.6.1. Learning Tasks and Analysis Purpose

Regarding the *purpose* of sequence analysis, the common objective is to extract patterns from the data and identify the real-world behaviours they describe. In some cases, these patterns are then used to classify the learner between groups or to predict the learner’s state, their subsequent action, or the outcome of the task. In most research using problem-solving tasks, patterns are closely related to the scenario, for example, how novices implement functions in programming. A few cases use these scenario-specific patterns to understand more general skills, for instance, identifying if learners follow linear or iterative processes for problem-solving. Similarly, for the knowledge-building tasks, the focus is to find information processing strategies and the meta-cognitive processes executed by the learners [73, 95]. Using an educational lens to determine the purpose of a study, such as complex problem-solving or self-regulation, facilitates the generalisation of the methods and results, regardless of the task.

On the other hand, the purpose of many guided practice tasks is to improve the prediction of the learners' performance, often to evaluate a proposed method against the state-of-the-art, as is the case for most of the 16 articles using the ASSISTments[1] dataset. In these studies, the sequential patterns receive less attention, and educational theory is leveraged in the prediction function, such as learning and forgetting, or while calculating the difficulty of a learning component. While this field has seen important growth with the popularity of Neural Networks (see figure 3.2), the real-world application and explainability of modern techniques are still key challenges [143].

3.6.2. Learning Actions and Methods

From our analysis, there seem to be increasing similarities in how researchers of different domains use sequences for the analysis of learner behaviour. For example, we found that 25 out of the 37 articles using Problem Attempts as the units of sequence, also integrate contextual information about the learner and their actions before submitting a problem, such as temporal features or their previous knowledge. Another common case is the use of statistical methods in combination with sequential methods, such as using logistic regression with features extracted from a sequential technique. Furthermore, researchers often implement more than one sequence analysis technique, using one to find case-specific action units, and then another to explore how these actions are sequenced in a full task strategy [60]. This may be a consequence of recent efforts towards the transparency and explainability of data-driven analysis, as well as the increasing accessibility of technology and analysis techniques.

The use of machine learning and deep learning methods to power automated or semi-automated decisions has recently started to see widespread implementation. This may be due to the recent efforts to mitigate risks associated with AI, for instance by pursuing transparency, causality, privacy, fairness, trust, usability, and reliability; some of the goals of Explainable AI (XAI) [118]. Explainability is especially important when applied to education, as the explanations can also have an educational component for learners, and it can aid teachers in their assessment of learners and their courses [118]. Regarding accessibility, different techniques are available via tools and libraries, we found several articles that use the Waikato Environment for Knowledge Analysis (Weka[2]) for multiple machine learning classifiers and predictors, [80, 86, 87, 98, 102, 124, 144], 2 more that apply Sequential Pattern Mining with TraMineR for R[3] [113, 136], as well as Markov Models via MeTA Toolkit[4] [64], and the Microsoft Sequence Clustering Algorithm [86]. It is important to note that these tools were explicitly mentioned in their respective

[1] https://sites.google.com/site/assistmentsdata/datasets
[2] https://www.cs.waikato.ac.nz/ml/weka/
[3] http://traminer.unige.ch/
[4] https://github.com/meta-toolkit/meta

articles, further facilitating reproducibility and comparison. This allows researchers to work with known methods throughout their analysis process, without having to develop, test, and evaluate them from scratch. Finally, we provide the dataset [78] of the reviewed works as an interactive interface [5] where researchers can filter the methods depending on the scope of their study, as shown in Table 3.1.

3.6.3. Interventions

Some of the reviewed works mention potential interventions that can be designed using the obtained patterns. For example, adaptive scaffolding in problem-solving tasks, personalized feedback for a knowledge-building system, or detection of the learners' level in guided practice. However, only 2 implemented an intervention, with 1 of them evaluating it, while 20 articles had no mention of a potential intervention at all. This is in line with recent publications such as "LAK of Direction", where they found that 11% of their reviewed articles introduced an intervention into a learning environment [52].

The use of learning principles, such as cognitive load [80] or learning and forgetting curves [121], is often present when describing or framing the problem. Very few use learning principles in the interpretation of patterns or interventions. This may be due to the focus on task- or system-specific tactics more than general learning behaviours. However, actionable results in LA research require a strong theoretical foundation, present in the analysis and the interpretation of the findings [76]. The theories mentioned in the framing of the problem should remain throughout the complete process, such as the use of Neo-Piagetian theory to classify patterns as programming strategies related to expertise [84]. This was the case for the 10 articles that aimed to identify meta-cognitive strategies, with 8 of them using a Self-Regulated Learning model to map data records into activities, interpret patterns after analysis, and describe a potential intervention [77].

3.7. Conclusion

In this document, we reviewed the different learning tasks, research purposes, action units, and methods of learning sequence analysis in educational systems, as well as the interventions that researchers design and propose based on the insights obtained. We described the major trends that can help to establish an overview of the use of Learning Sequence Analysis, aggregated in an open repository [78] and an interactive interface available in https://mvallet91.github.io/sequence-matters/. This is to assist researchers in finding, comparing, and using others' efforts as a foundation and move past the exploratory stage; as noted in one of the reviewed articles [65], "the community is missing more generally applicable results."

[5] https://mvallet91.github.io/sequence-matters/

4

What makes an Expert?

Comparing Problem-solving Practices in Data Science Notebooks

Manuel Valle Torre, Marcus Specht and Catharine Oertel

Problem-solving expertise does not come from superior problem-solving ability but rather from domain learning.

John. R. Anderson

The development of data science expertise requires tacit, process-oriented skills that are difficult to teach directly. This study addresses the challenge of empirically understanding how experts' and novices' problem-solving processes differ. We apply a multi-level sequence analysis to 440 Jupyter notebooks from a public dataset, mapping low-level coding actions to higher-level problem-solving practices. Our findings reveal that experts do not follow fundamentally different transitions between data science phases than novices (e.g., Data Import, EDA, Model Training, Visualization). Instead, expertise is distinguished by the overall workflow structure from a problem-solving perspective and cell-level, fine-grained action patterns. Novices tend to follow long, linear processes, whereas experts employ shorter, more iterative strategies enacted through efficient, context-specific action sequences. These results provide data science educators with empirical insights for curriculum design and assessment, shifting the focus from final products toward the development of the flexible, iterative thinking that defines expertise—a priority in a field increasingly shaped by AI tools.

4.1. Introduction

Data Science Education (DSE) is widely recognized as a complex, interdisciplinary field that requires more than just technical proficiency ([145]). The consensus in the literature is that DSE should focus on cultivating creative problem-solving skills, prioritizing a learner's ability to navigate ambiguity over rote memorization of tools and algorithms ([146]). Consequently, data science is increasingly taught as a process—a structured yet flexible workflow that guides practitioners from problem definition to final solution ([147]).

While we teach data science as a process, we lack a deep, empirical understanding of how experts and learners actually enact this process ([148]). The crucial procedural skills involved are often tacit—second nature to experts but opaque and misunderstood by novices ([56]). This makes them difficult to observe and, therefore, difficult to teach directly ([83]). The recent rise of Generative AI tools only increases the urgency of this problem. While powerful, these tools can encourage a superficial, task-completion approach, potentially allowing learners to offload critical thinking and reinforcing the very novice-like behaviors we seek to overcome ([149]). Understanding and supporting the process of developing expertise is therefore more critical than ever ([49]).

This study addresses that gap by using computational methods to make these learning processes visible. We apply a range of sequence analysis techniques, guided by a multi-level framework for educational research ([40]), to investigate the problem-solving process in a large dataset of Jupyter notebooks. By doing so, we aim to answer the following research questions:

- What characteristic problem-solving strategies can be identified in data science notebooks at different granularity levels?
- At which of these levels, if any, can these strategies effectively differentiate novices from experts?

The findings from this work will provide a deeper understanding of how data science expertise develops. This knowledge can be used to inform pedagogical practices, design more effective learning materials, and develop intelligent support systems that help learners develop the flexible, adaptive skills of an expert practitioner, allowing for better human-AI collaboration in the future ([150, 151]).

4.2. Related Work

4.2.1. Analyzing the Data Science Process

Teaching and learning data science is challenging, as it requires integrating statistics, programming, and domain-specific problem-solving ([145, 146, 152]). In education, this is often structured around a well-defined workflow, such as the Cross-Industry Standard Process for Data Mining (CRISP-DM), which outlines key phases like data preparation, modeling, and evaluation ([153]). This workflow provides a necessary scaffold for novices ([145]). However, a key challenge in Data Science Education (DSE) is moving learners beyond a rigid, step-by-step execution of this workflow toward the more fluid, iterative, and adaptive process that characterizes expert practice ([148]).

The recent introduction of Generative AI tools only magnifies this challenge. These tools can automate code writing and debugging, assist with explanations and examples, and generate learning materials for data science ([154]). However, research suggests they can also promote a dependency that stifles self-regulation and encourages a superficial, task-completion approach, potentially leading to 'metacognitive laziness', as described by [149].

This makes it critical to shift the focus from the outcomes to understand and support learning as a process ([49]). To do so, researchers need not only authentic data from real-world tasks, but data that is also rich enough to capture the sequence of actions that constitute that process ([155]). Computational notebooks, such as those from platforms like Kaggle, offer a rich source of such data, capturing the sequenced actions of thousands of practitioners solving complex, real-world problems ([156, 157]). Prior educational research using notebook data has focused primarily on the final product or static code quality, analyzing, for example, cell-level mistakes by [158], overall code metrics by [159], or documentation quality by [160]. While valuable, these approaches overlook the dynamic, sequential nature of the work ([40]). They can tell us what was produced, but not how the author arrived at the solution ([161]). The critical gap in the literature, therefore, is an understanding of the sequential process itself—the strategies and habits that differentiate novice and expert problem-solvers ([56, 148, 150]).

4.2.2. A Multi-Level Framework for Data Science Process Analysis

To analyze these complex workflows, we adopt the principled structure outlined by [40] in their five-component framework for educational sequence analysis. This guides our work by ensuring a clear purpose, grounding in theory, and appropriate selection of techniques. Following this structure, we first establish a theoretical lens through the literature of problem-solving, which treats complex tasks as a series of cognitive and metacognitive practices ([162]). Specifically, we adopt the five-stage model of authentic problem-solving practices identified by [163]:

1. Problem definition and decomposition: Practices to understand and simplify a problem, such as breaking it down into smaller subproblems.
2. Data collection: Actions and decision-making to collect the data needed to solve the problem.
3. Data recording: How problem solvers keep track of the data they have collected.
4. Data interpretation: Applying domain knowledge to make sense of collected data and reach a solution.
5. Reflection: Processes to monitor progress and evaluate the quality of a solution.

This framework allows us to abstract the technical data science actions into a higher-level, cognitively-grounded process, providing a perspective to understand the overall strategic shape of a workflow.

While this high-level framework provides an essential starting point, expertise often manifests at different levels of granularity. An expert might follow the same high-level stages as a novice but execute the steps within those stages with far greater efficiency. A single analytical method is often insufficient to capture this complexity ([164]). Therefore, this study adopts a multi-level sequence analysis approach to investigate the data science process at different resolutions:

- Overall Process Structure: Analyzing the sequence of broad problem-solving stages to identify high-level strategic archetypes.
- Phase-Level Transitions: Modeling the probabilistic transitions between specific DS phases (e.g., from EDA to Model Training) to understand the workflow's order of operations.
- Fine-Grained Action Patterns: Discovering recurring, low-level sequences of actions within a given problem-solving stage (e.g., the specific steps taken during Data Collection).

This multi-level approach, which has been used to understand complex learning in other domains ([165]), is essential for deconstructing the data science process. It allows us to investigate not only what stages practitioners follow, but also how they transition between them and what specific actions they take to enact their strategies. This study applies this approach to investigate the differences between the processes of novices and experts ([83]), addressing a critical need to better understand and teach the process of data science. Such an understanding is a prerequisite for designing learning tasks and support systems able to assist learners in regulating their own work, especially when using modern AI tools ([166]).

4.3. Dataset and Methods

4.3.1. Dataset

Our study uses Code4ML by [167], a set of labelled notebooks from Kaggle, to analyze patterns in Data Science (DS) by treating DS challenges as problem-solving tasks. We focus on a single Kaggle competition to minimize variability caused by different data modalities (e.g., image recognition vs. text processing). We selected the competition with the most notebooks available in Code4ML: the *Quora Insincere Questions Classification*. This competition was chosen because it involves a common data science task (text classification), provides a sufficient number of notebooks for analysis, and requires a range of typical data science activities, including data cleaning, feature engineering, classification, visualization, and evaluation ([168]).

To mitigate the impact of outliers, we removed the top and bottom 5% of the notebooks based on length ([165]), resulting in a dataset of 440 notebooks.

The classification of participants into novice and expert categories is based on their Kaggle progression tier. While imperfect, these tiers are based on provable metrics and achievements (competition medals, votes by experts, etc.) [1]. Novices are defined as users in the *Novice or Contributor* tiers (n=339, M length=20.3 cells, SD=10.2), while experts are those in the *Expert, Master, or Grandmaster* tiers (n=101, M length=16.7 cells, SD=6.6). Data preprocessing was performed using `Python`, integrating files from Code4ML and the KGTorrent database to filter the corresponding authors, notebooks, and cell labels ([157, 167]).

4.3.2. Methodology

We follow the five-component framework for applying sequence analysis in education by [40]: defining the purpose, identifying units and scope, grounding the analysis in theory, selecting appropriate techniques, and interpreting the results for educational intervention.

The **purpose** of our analysis is to identify discernible problem-solving strategies within the notebook data and determine if these strategies can differentiate between novice and expert data scientists. This aligns with our overarching goal of understanding how expertise develops in data science to better inform pedagogy.

For the **units** and **scope** of analysis, we treat each labelled code cell as a single action and each Jupyter notebook as a complete, self-contained problem-solving sequence.

Our analysis is guided by the **educational principle** of data science as a form of problem-solving ([83, 146]). We use the framework from [163] to map the 11 technical DS phases from Code4ML to five general problem-solving practices, based on the actions within each phase from [167]. The mapping is detailed in Table 4.1. While conventional data science process models like CRISP-DM are useful for outlining an idealized workflow, they are less effective at capturing the enacted *cognitive process* of problem-solving. By adopting this cognitive lens, we can interpret actions like 'model training' as a crucial 'data collection' practice where the practitioner is actively gathering information about feature effectiveness. This perspective is essential for revealing the underlying problem-solving strategies that a standard workflow model would otherwise obscure.

For the **analysis**, we employ a suite of sequence analysis techniques to investigate the notebooks at different levels of granularity. This multi-level approach allows us to build a comprehensive picture of the problem-solving processes.

1. **Overall Process Structure:** To identify high-level workflow patterns, we use Agglomerative Hierarchical Clustering (AHC) on the full sequences of problem-solving practices. We use Optimal Matching (OM) to calculate the dissimilarity between sequences and Ward's method for clustering ([169]). This approach groups notebooks that share a common overall structure or 'narrative'([170]).

[1] https://www.kaggle.com/progression

Table 4.1: Mapping Between Problem-Solving Practices and Code4ML Data Science Phase

Practice	DS Phase	Explanation
Problem Definition and Decomposition	Environment	Setting up the environment (e.g., `install` and `import modules`) represents an initial decomposition of the problem into a set of solvable tasks that will be handled by specific libraries.
	Data Extraction	Extracting data from various sources implies understanding the problem's requirements.
	[Data] Debug	Debugging the data involves identifying and correcting errors, which requires understanding the problem and the expected data characteristics.
Data Collection	Data Transform	Data transformation practices, such as `feature engineering`, `drop columns`, or `merge`, are direct actions to shape the raw data into the necessary clean and relevant dataset required to solve the problem.
	Exploratory Data Analysis	Understanding patterns and characteristics of a dataset, with `count unique` or `count duplicate`, is a crucial part of "collecting data to solve the problem".
	Model Training	Actions within training, like `choose model class` and `predict on train`, can be seen as "actions to collect data," for example, about the relationships between variables.
Data Recording	Data Visualization	Creating a visualization (`plot metrics`, `heatmap`, `distribution`) produces a persistent artifact that records and summarizes the state of the data or model at a specific point, allowing the practitioner to track their findings.
Data Interpretation	Model Interpretation	Actions such as `print shapley coeffs` and `features selection` are explicit methods for understanding what the model has learned and using that understanding to refine the solution.
	Hyperparameter Tuning	Tuning hyperparameters, by defining `search model` or `search space`, is guided by the interpretation of the model's performance on the data.
	Model Evaluation	Evaluating model performance is a key step in interpreting the model's effectiveness and making sense of the data.
Reflection	Data Export	This encompasses the process of "evaluating the quality of the solution." Actions like `save to csv` or `prepare output` often represent a final assessment and synthesis of the findings, signifying the end of a cycle and reflecting a judgment on the result.

2. **Phase-Level Transitions:** To model the entire workflow as a process and examine transitions between phases, we use Process Mining (PM) and Markov Models (MM). *Process Mining* provides a holistic view of the most common pathways through the DS workflow, assuming the author is following a plan ([171]). We specifically use the technique of *Process Discovery* on the `bupaR` package in R, which provides similar results to Fuzzy Miner, the most popular algorithm for Educational Process Mining ([172]). To ensure clarity, process maps visualize the top 80% most frequent transitions. *Markov Models*, using `seqHMM` in R, focus on the probabilistic transitions between states (i.e., DS phases), allowing us to find differences between Novices and Experts based on their transition profiles ([173]). Furthermore, we can use the initial models and the Expectation Maximization algorithm to cluster the sequences, potentially finding major representations of Experts or Novices per cluster.

3. **Within-Phase Action Patterns:** To find common, low-level action sequences within a specific problem-solving phase, we use Sequential Pattern Mining (SPM). Using `PrefixSpan` in Python, we search for frequently occurring sub-sequences of specific DS actions (e.g., import_modules, count_missing, save_to_csv) per problem-solving stage. This is ideal for discovering fine-grained procedures in the specific context of each stage ([174]). We set a minimum support threshold of 5% due to the density of information in sequences within stages ([138]).

Following the clustering analyses (AHC and MM), we use a Chi-square test of independence to assess whether cluster membership is significantly associated with the author's expertise level.

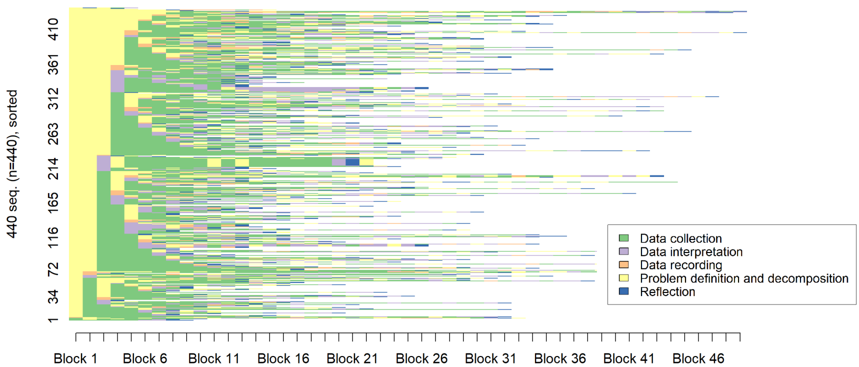

Figure 4.1: Visualization of all Problem-solving Sequences

Finally, while designing a specific **educational intervention** is beyond this study's scope, we discuss the implications of our findings for DSE curriculum, teaching practices, and the design of support tools in the Discussion section (4.5).

4.4. Results

This section presents the results of our sequence analysis, structured according to the multi-level framework described in the Methodology. For each level of granularity—overall process structure, phase-level transitions, and within-phase actions—we first identify characteristic patterns and strategies (RQ1) and then assess whether these patterns differentiate authors by their expertise level (RQ2).

4.4.1. Exploratory Analysis and Overall Process Structure

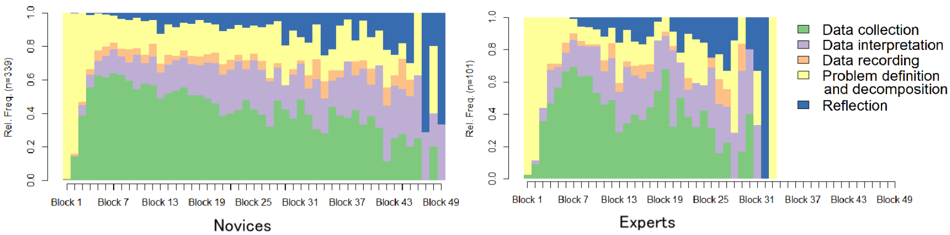

Figure 4.2: Relative Frequency visualization of Novice and Expert Sequences

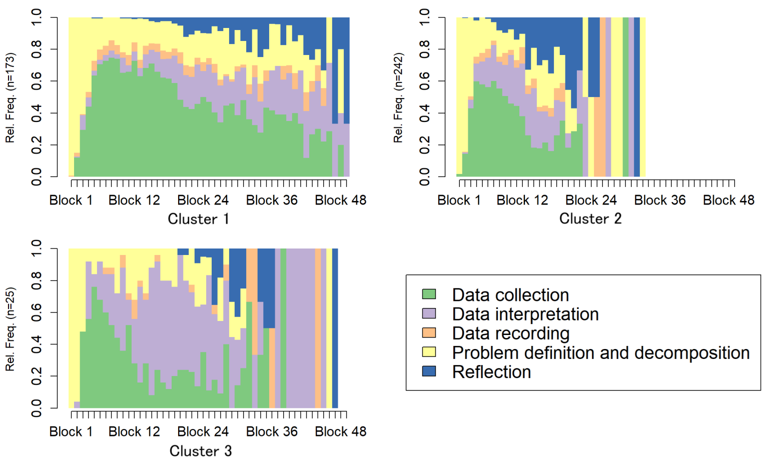

Figure 4.3: Relative Frequency visualization of Clustered Sequences

A visual inspection of all 440 sequences mapped to problem-solving stages, in Figure 4.1, reveals common structural characteristics (RQ1). Most sequences begin with one to three *Problem Definition and Decomposition* actions (yellow) and conclude with *Reflection* (blue). The phase of *Data Collection* (green) is prevalent throughout the process. Beyond this initial structure, the sequences show high variability, suggesting multiple pathways through the problem.

The relative frequency of actions over time (Figure 4.2) shows that novice notebooks tend to be longer than expert notebooks. Furthermore, experts exhibit a second peak of *Data Collection* activity later in their process (around step 19), whereas for novices, *Data Collection* peaks early and then steadily declines.

Agglomerative Hierarchical Clustering (AHC) of the problem-solving sequences yielded an optimal solution of three clusters with an Average Silhouette Width = 0.28, indicating a discernible but weak cluster structure ([169]). The clusters display distinct patterns (Figure 4.3):

1. Cluster 1 (n=173): Characterized by long sequences that generally follow the problem-solving stages in a linear order.
2. Cluster 2 (n=242): Contains shorter sequences with an earlier and more prominent second peak of *Data Collection* and earlier instances of *Reflection*.
3. Cluster 3 (n=25): A small cluster distinguished by a significantly higher proportion of *Data Interpretation* actions.

Addressing RQ2, a Chi-square test of independence revealed a significant association between cluster membership and expertise level ($\chi^2 = 11.85$ and $p = 0.0027$). Novices were overrepresented in Cluster 1 (Observed=148, Expected=133), while experts were overrepresented in Cluster 2 (O=68, E=55) and Cluster 3 (O=8, E=5).

4.4.2. Phase-Level Transitions: Process Mining and Markov Models

To examine the workflows with greater detail, we shift our analysis from the five high-level problem-solving stages to the 11 more granular DS phases. This shift is motivated by the fact that initial analyses using the five broad stages yielded nearly identical process models for novices and experts for PM and MM.

Process Mining (PM) models were generated for novice and expert groups at the DS phase level. The thickness of the arrows is proportional to the frequency of transitions between phases, and the percentages inside the nodes represent the proportion of all actions that occurred in that phase. The resulting process maps (Figures 4.4 and 4.5) identify largely similar workflows (RQ1). However, subtle differences emerge in the transitions between novices and experts (RQ2). Novices transition to *Data Transform* actions more frequently overall (35.9 vs 28.4%), while experts show a slightly higher tendency to begin with *Environment setup* (see the line from *Start* to *Environment*) and exhibit more frequent transitions involving *Model Train* and *Model Evaluation*.

A direct comparison of Markov Models (MM) built for novices and experts also revealed highly similar transition probabilities between DS phases. The most notable difference was a slightly higher probability for experts to transition from *EDA* to *Data Transform*. Given these subtle differences at the individual transition level, we proceeded with clustering to identify more holistic strategic patterns.

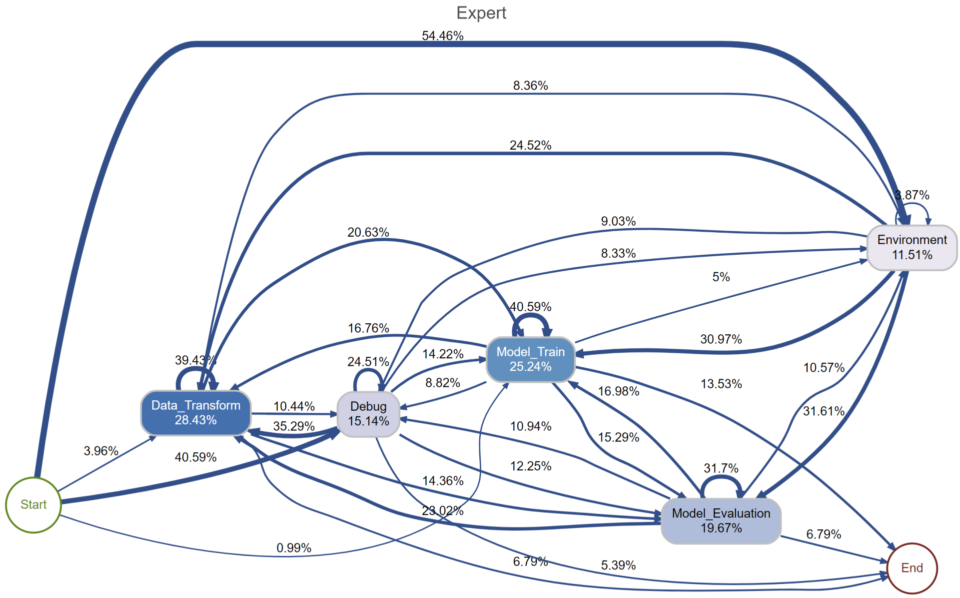

Figure 4.4: Process Visualisation for Experts

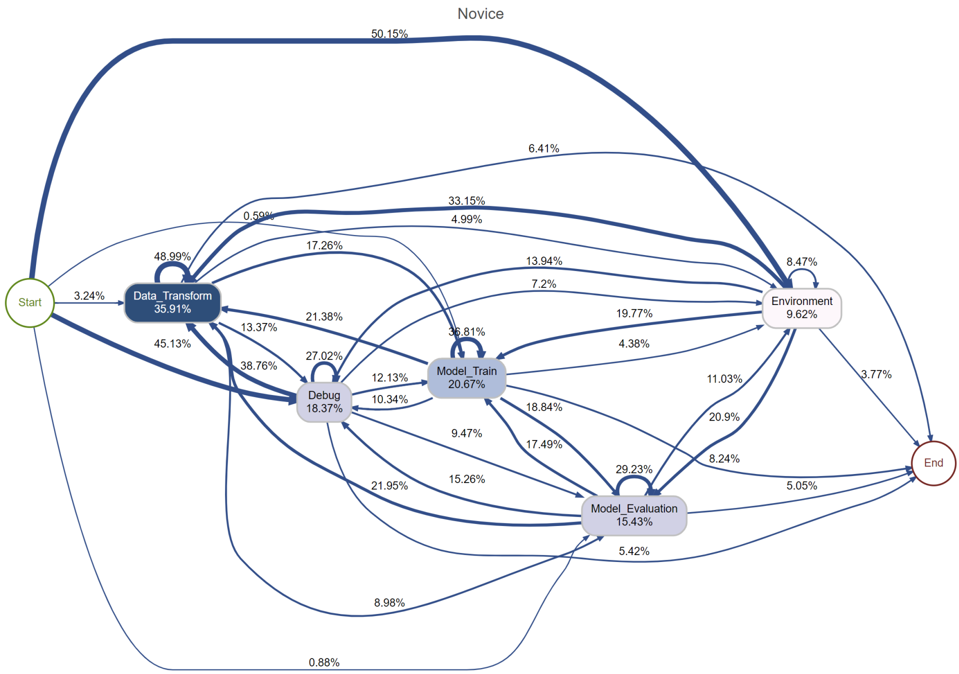

Figure 4.5: Process Visualisation for Novices

The MM clustering analysis revealed three distinct strategic clusters based on their transition probabilities (RQ1) in Figures 4.6 and 4.8. We label these based on their dominant patterns:

1. A Data-driven cluster, characterized by frequent self-loops on *Data Transform* and many incoming transitions to it.
2. An Iterative Model Improvement cluster, defined by strong self-transitions (the thick, round arrows with 0.5) in *Model Training* and *Model Evaluation*, as well as from *Model Interpretation* back to *Model Training*.
3. A Problem Decomposition cluster, which showed a high probability of starting (the circle around the node) with and repeating *Environment* actions before moving deeper into the workflow.

However, when testing if these strategies were associated with expertise (RQ2), a Chi-square test found no significant association between the MM-based clusters and expertise level ($\chi^2 = 0.053$ and $p = 0.97$).

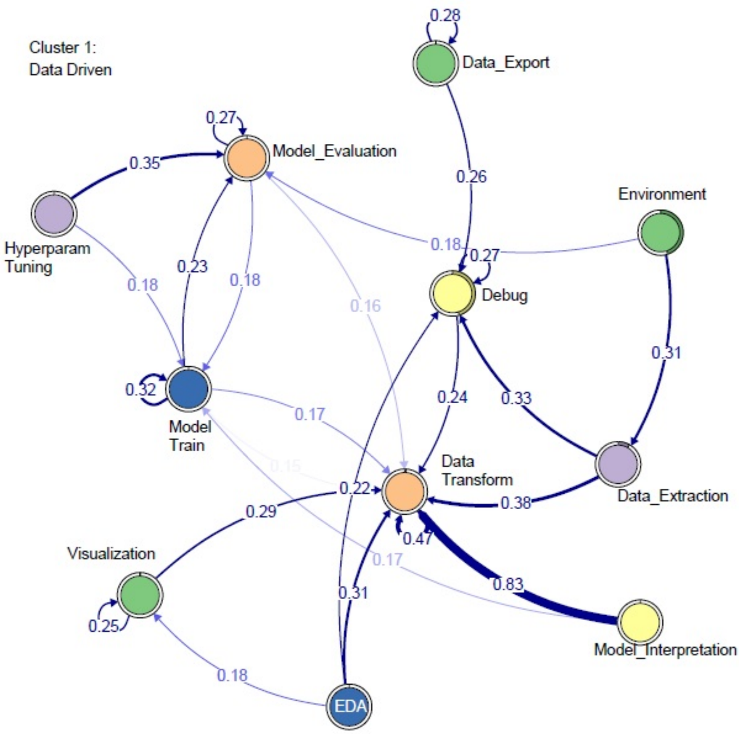

Figure 4.6: Transition Probabilities of Sequences for Cluster 1

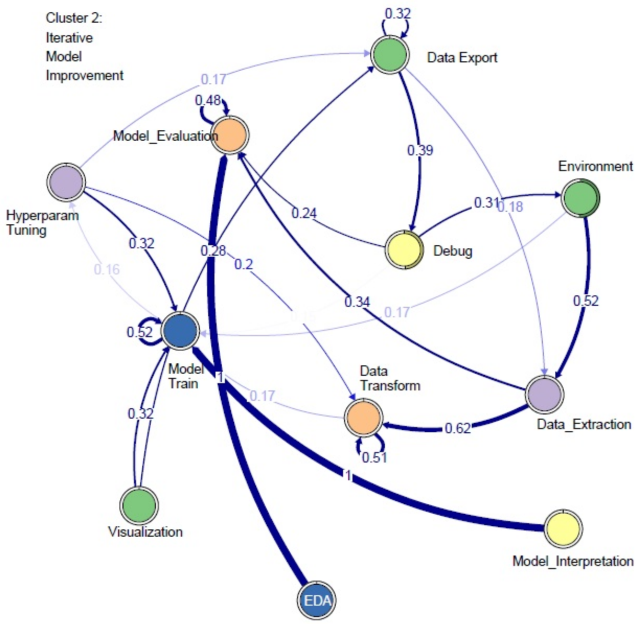

Figure 4.7: Transition Probabilities of Sequences for Cluster 2

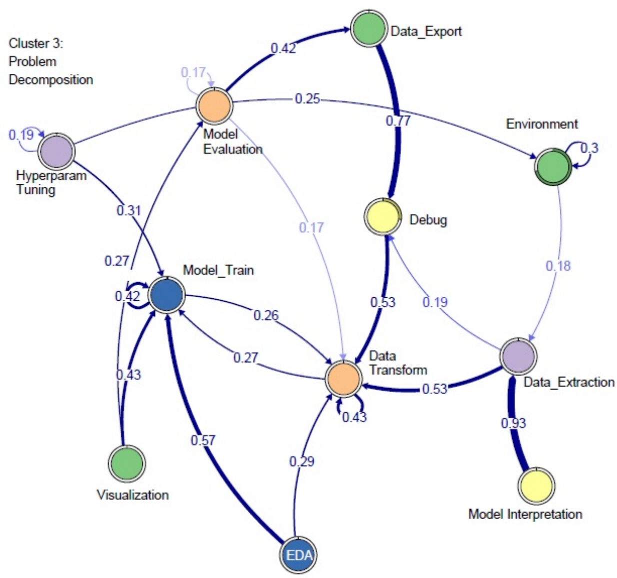

Figure 4.8: Transition Probabilities of Sequences for Cluster 3

4.4.3. Action Patterns Within Problem-Solving Stages

For our final level of analysis, we return to the five high-level problem-solving stages to provide the context for discovering fine-grained action patterns. This approach allows us to identify which specific DS actions are frequently used in sequence to achieve a broader goal (e.g., Data Collection), rather than analyzing patterns within a single, more narrow DS phase. To identify these procedures (RQ1) and compare their usage by expertise (RQ2), we used Sequential Pattern Mining (SPM). Table 4.2 summarizes notable patterns where frequencies differed between novices and experts. Some notable examples are:

- Problem Definition: Experts more frequently showed patterns involving repeated import_modules actions. Novices, in contrast, uniquely showed a [load_from_csv, show_table] pattern (7% support).
- Data Collection: Novices frequently exhibited a [feature_engineering, feature_engineering] pattern (7%). Expert patterns were dominated by model-related actions like [model_class, fit] (6%) and [model_class, compile] (5%).
- Data Recording: A [distribution, distribution] pattern was common among novices (15%), while experts showed patterns like [model_coefficients, learning_history] (6%).
- Reflection: Experts showed a [save_to_csv, save_to_csv] pattern (5%) not present among novices.

Table 4.2: Top Sequential Patterns by Problem-Solving Phase and Expertise

Phase	Pattern	Novices	Experts
Problem Definition and Decomposition	[import modules, load from csv]	11%	17%
	[list files, load from csv]	10%	12%
	[load from csv, show table]	7%	-
	[import modules, import modules]	-	9%
Data Collection	[feature engineering, feature engineering]	7%	-
	[model class, fit]	-	6%
	[model class, compile]	-	5%
Data Recording	[distribution, distribution]	15%	-
	[model coefficients, learning history]	-	6%
Data Interpretation	Same patterns for both groups	-	-
Reflection	[save to csv, save to csv]	-	5%

4.5. Discussion and Implications

This study used a multi-level sequence analysis to compare the problem-solving processes of novice and expert data scientists. Our central aim was to identify characteristic strategies that could differentiate expertise levels and, ultimately, inform data science education. Our findings confirm that discernible strategies exist (RQ1) and that the nature of these strategies depends heavily on the level of analytical granularity. While high-level phase transitions are similar across skill levels, the overall process structure and, most notably, the fine-grained action patterns within those processes reveal significant differences between novices and experts (RQ2).

4.5.1. Interpreting Strategies Across Different Granularities

By analyzing the notebooks at three distinct levels, we constructed a detailed picture of the differences in workflows. At the overall process level, our analysis shows that experts and novices structure their entire workflow differently. Novices were significantly more likely to belong to a cluster characterized by a long, linear, step-by-step process, suggesting they adhere closely to a canonical workflow. In contrast, experts were overrepresented in a cluster featuring shorter, more iterative workflows. This aligns with classic literature on expertise, which finds that experts possess flexible mental schemas that allow them to adapt their strategies and work more efficiently ([83, 175]). However, the low silhouette score for this clustering (0.28) suggests that these "strategies" are not rigid archetypes but rather overlapping tendencies, highlighting the inherent flexibility and variability in the data science process, even at different skill levels.

At the phase-transition level, the analysis revealed a crucial insight: the high-level order of operations is not a strong differentiator of expertise. Our initial analysis using the five broad problem-solving stages showed no discernible differences between groups. Even when "zooming in" to the 11 more granular DS phases, the non-significant result from the Markov Model clustering confirms that experts do not follow a fundamentally different sequence of phases than novices.

At the action-pattern level, the procedures identified by SPM provide the concrete evidence that explains the high-level differences. The expert tendency toward shorter, iterative workflows (identified by AHC) is likely enacted through efficient modeling patterns like `[model_class, fit]`. In contrast, the novices' linear, step-by-step approach is reflected in more exploratory patterns like `[feature_engineering, feature_engineering]` and `[load_from_csv, show_table]`. These fine-grained patterns are the tangible evidence of the abstract strategies, explaining how experts achieve their efficiency and iterative nature.

4.5.2. Implications for Data Science Education

The patterns identified in this study have direct implications for designing learning experiences and support systems in data science.

- Curriculum Design: The curriculum should explicitly teach and model the iterative nature of data science ([146]). This focus on process becomes even more critical in an era of Generative AI ([154]). As AI tools automate the generation of code snippets, the core human skill shifts from writing code to strategically guiding the problem-solving process—a skill our analysis identifies as a key component of expertise. Project-based learning activities that require students to cycle back through the workflow can help move them beyond a rigid, linear mindset ([176]).

- Teaching Practices: Instructors can use these findings to provide targeted feedback. A notebook that shows a linear process with many trial-and-error transformations could prompt a discussion about planning and iterative refinement. Instructors can also use worked examples or live coding to model expert practices, explicitly narrating why they are returning to an earlier stage of the process ([145]).
- Navigating Generative AI: Our findings highlight a potential pitfall of Generative AI in education; its use can inadvertently reinforce the linear, non-iterative process characteristic of novices. A student who simply prompts an AI for the "next step" may get correct code but fails to develop the strategic, reflective, and iterative thinking that defines expertise. Educators must therefore focus on teaching students how to use these tools as a partner for brainstorming and refinement, not as a replacement for critical thought ([149]).
- Intelligent Support Systems: The patterns identified here can form the basis for automated feedback systems. These systems can be significantly enhanced by Generative AI. An AI-powered assistant, informed by our findings, could not only detect novice patterns but also provide expert-like scaffolding. For example, by prompting a student to reflect on model performance before attempting more feature engineering, it could actively encourage an expert-like iterative cycle ([166]).
- Assessment: This process-oriented analysis suggests new forms of assessment. Instead of only grading the final notebook or model accuracy, instructors could assess the process itself. The structure of a student's workflow, as captured by their sequence of actions, could become a valuable formative assessment tool to gauge their problem-solving maturity ([49]).

4.5.3. Limitations and Future Work

A primary limitation of this study is its intentional focus on a single, comprehensive Kaggle competition in text classification [168]. While this case study approach provides the depth necessary to analyze a wide spectrum of authentic procedures, the specific, low-level action patterns may not generalize to other contexts. We contend, however, that the core practical implications for education are broadly transferable. The analytical framework presented here can be readily applied to other data science domains, such as image analysis or time-series forecasting, as well as to different scopes like short in-class work or scaffolded assignments, to investigate which problem-solving strategies are universal and which are task-specific.

A second limitation is that we analyze only the final state of notebooks, which misses the real-time process of editing, deleting, and debugging cells. Future research should address these limitations by capturing more fine-grained, real-time interaction data from within notebook environments across a wider variety of tasks, learner expertise levels, and educational contexts ([177, 178]). Such data would enable a more dynamic analysis of the learning process, paving the way for adaptive support systems that can provide real-time ([179]), contextualized scaffolding to learners as they navigate the complexities of data science problem-solving ([150, 180]).

4.6. Conclusion

This study set out to make the tacit problem-solving processes of data science visible, using a multi-level sequence analysis to compare the workflows of novices and experts. While our analysis of phase-to-phase transitions showed that novices and experts follow a similar order of operations, telling differences emerged at other levels of granularity. We found a statistically significant difference in the overall structure of their workflows, with novices tending toward longer, more linear processes, while experts demonstrated shorter, more iterative approaches. These high-level strategic differences were explained by the fine-grained action patterns within each phase, where experts employed more technically efficient procedures.

The central contribution of this work is the demonstration that data science expertise lies not in following a different or unique sequence of steps, but in how the process is enacted—with flexibility, iteration, and efficiency. This insight has critical implications for education, especially with the integration of Generative AI into data science tools and environments. Educational approaches that only teach a canonical, step-by-step workflow risk reinforcing the very novice-like behaviors we identified. Instead, assignments, materials, and intelligent support tools must be carefully designed to cultivate the adaptive, reflective, and iterative thinking that truly defines an expert practitioner, ensuring that new AI assistants become partners in developing expertise, not crutches that stifle it.

5

JELAI: Integrating AI and Learning Analytics in Jupyter Notebooks

Manuel Valle Torre, Thom van der Velden, Marcus Specht, and Catharine Oertel

I shall either find a way or make one

Hannibal

Generative AI offers potential for educational support, but often lacks pedagogical grounding and awareness of the student's learning context. Furthermore, researching student interactions with these tools within authentic learning environments remains challenging. To address this, we present JELAI, an open-source platform architecture designed to integrate fine-grained Learning Analytics (LA) with Large Language Model (LLM)-based tutoring directly within a Jupyter Notebook environment. JELAI employs a modular, containerized design featuring JupyterLab extensions for telemetry and chat, alongside a central middleware handling LA processing and context-aware LLM prompt enrichment. This architecture enables the capture of integrated code interaction and chat data, facilitating real-time, context-sensitive AI scaffolding and research into student behaviour. We describe the system's design, implementation, and demonstrate its feasibility through system performance benchmarks and two proof-of-concept use cases illustrating its capabilities for logging multi-modal data, analysing help-seeking patterns, and supporting A/B testing of AI configurations. JELAI's primary contribution is its technical framework, providing a flexible tool for researchers and educators to develop, deploy, and study LA-informed AI tutoring within the widely used Jupyter ecosystem.

This chapter was published in AIED25 [177].

5.1. Introduction

Generative AI (GenAI) tools like ChatGPT offer potential for on-demand student support, but are generally designed to solve problems rather than support learning pedagogically [181]. They typically lack insight into students' learning context (e.g., task, progress, challenges) and can foster reliance on quick answers over deeper understanding [149]. Novices struggle to provide effective context [182], leading to AI responses that may be unhelpful or misaligned with instructional goals [183]. Furthermore, research on GenAI in education often relies on self-reports or high-level comparisons, lacking detailed analysis of student-AI interactions needed to understand learning mechanisms and design effective, pedagogically-grounded support [184, 185].

Integrating Learning Analytics (LA) with GenAI offers a path forward [186]. Digital environments like Jupyter Notebooks capture rich interaction logs [187]. Analyzing these logs alongside chat interactions can provide the context needed for adaptive, real-time interventions [40, 188] and enable research into student cognitive and metacognitive behaviours [149, 189]. While Jupyter is widely used [190], existing extensions typically focus on isolated functions like grading (NBGrader[1]) or provide basic logging/tutoring without deep LA integration [191]. Recognizing the need for pedagogically sound AI in programming, researchers have developed specialized systems like CodeTutor [192], CodeAid [193], and CodeHelp [194], which use techniques like targeted prompting and avoiding direct solutions. However, these systems often operate separately from the primary coding environment, and their design typically does not focus on the analysis of integrated student workflows within that environment. To enable such integrated analysis, there is a need for systems that tightly couple fine-grained LA (capturing detailed coding processes and errors) with context-aware conversational AI tutoring *within* the notebook environment. This integration could bridge the gap between flexible LLMs and more structured learning systems, enabling personalized feedback based on student models and promoting more effective help-seeking [195].

To address this gap, we present JELAI (*Jupyter Environment for Learning Analytics and AI*), an experimental platform integrating LLM-based tutoring with an LA pipeline inside Jupyter notebooks (Figure 5.1). JELAI is designed as a modular, extensible framework to:

1. Capture and process fine-grained student activity (code edits, executions, errors) and chat interactions into meaningful learning traces.
2. Enrich LLM prompts with real-time context (e.g., recent code, errors, objectives, chat history) for adaptive scaffolding.
3. Provide a platform for educational research on student-AI interaction patterns and the impact of different AI configurations (e.g., prompting strategies), leveraging the integrated nature of the interaction data.

This paper details JELAI's architecture, implementation, and demonstrates its feasibility through preliminary system performance data and two proof-of-concept

[1] https://nbgrader.readthedocs.io/en/stable/

use cases. The primary contribution is the technical design and implementation of a platform enabling LA-informed AI tutoring within a widely used educational environment. The open-source repository (BSD-3-Clause license) is available for use and collaboration [2].

5.2. System Design and Implementation

The design of JELAI was guided by several key requirements necessary for integrating LA with AI tutoring effectively within an educational context:

- **Granular Data Logging:** Capture fine-grained interaction data (code edits, executions, errors, chat) as the foundation for LA and context-aware AI [186, 187].
- **Real-time Context Enrichment:** Leverage logged data to provide LLMs with immediate context (recent code, errors, conversation history, task goals) to improve relevance and mitigate generic responses [183].
- **Pedagogical Alignment & Scaffolding:** Allow instructors to configure AI behaviour (via system prompts, intervention strategies) to align with learning objectives and enable adaptive scaffolding based on student activity [189, 195].

[2] https://github.com/mvallet91/JELAI

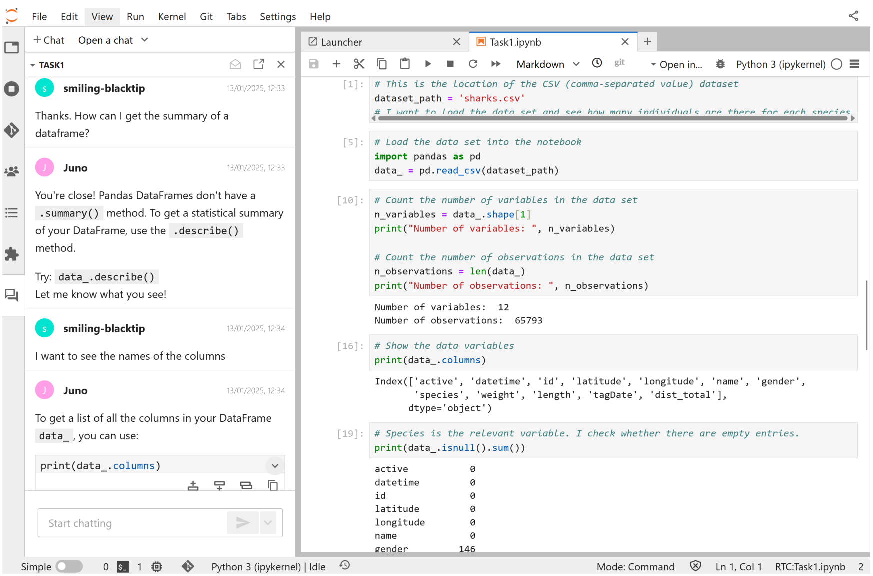

Figure 5.1: JELAI Interface: Jupyter Notebook with integrated AI tutor (Juno).

- **Modularity and Extensibility:** Design components (logging, LA, LLM interaction) independently to facilitate updates, integration of new techniques, and research experimentation.
- **Scalability and Privacy:** Support multiple concurrent users efficiently while ensuring data isolation and privacy, particularly when using local LLMs.

To meet these requirements, JELAI is implemented as a modular, containerized system using open-source technologies (Docker, JupyterHub, JupyterLab, Ollama) and custom Jupyter extensions (`Jupyter-Chat`, `JupyterLab-Pioneer`). JELAI is deployed using `docker-compose`, simplifying setup and ensuring user isolation for scalability and privacy. The architecture (Figure 5.2) comprises four main components:

1. **User Notebook Container:** Hosts the student's JupyterLab instance. It includes the `Jupyter-Chat`[3] extension for the AI tutor interface and the `JupyterLab-Pioneer`[4] extension for capturing detailed telemetry (keystrokes, cell executions, errors, UI interactions). An `Interaction Handler` within this container preprocesses student chat messages, enriching them with immediate context (e.g., recent edits and errors, task objective) before forwarding them.
2. **Middleware Container:** Acts as the central orchestration hub. It contains the `Learning Analytics Module`, which processes incoming telemetry logs and chat data, potentially storing them or making them available for real-time analysis. It also houses the `LLM Handler`, responsible for retrieving relevant conversation history and context from the LA module, applying instructor-defined pedagogical rules or scaffolding logic (e.g., modifying prompts based on recent student errors or help-seeking patterns), constructing the final prompt, and managing communication with the LLM Server.
3. **JupyterHub Container:** Manages user authentication, spawns individual User Notebook Containers, and handles proxying, enabling multi-user deployment.
4. **LLM Server:** Provides the interface to the chosen language model. JELAI uses Ollama to support various local open-source LLMs, ensuring data privacy and control, but can also be configured to connect to external APIs compatible with the OpenAI standard.

A typical interaction workflow proceeds as follows: (1) A student interacts within their JupyterLab notebook (e.g., writing code, running cells). (2) `JupyterLab-Pioneer` logs these actions and sends them asynchronously to the Middleware's LA Module. (3) The student sends a message to the AI tutor via `Jupyter-Chat`. (4) The local `Interaction Handler` intercepts the message, adds immediate context (e.g., code from the current cell), and sends it

[3] https://github.com/jupyterlab/jupyter-chat
[4] https://jupyterlab-pioneer.readthedocs.io/en/latest/

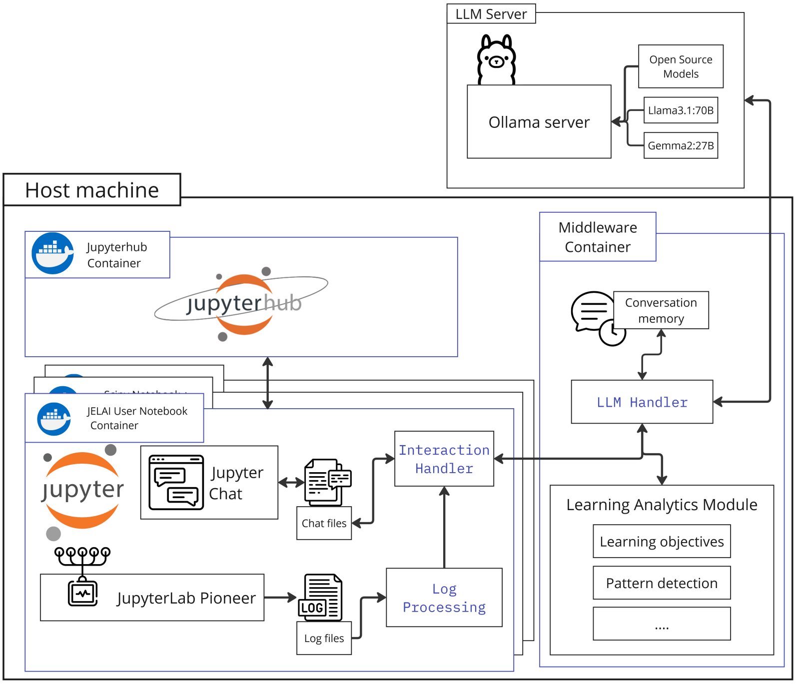

Figure 5.2: JELAI System Architecture.

to the Middleware. (5) The Middleware's `LLM Handler` retrieves the message, fetches relevant context from the LA Module, applies any configured pedagogical logic or prompt engineering, and sends the final, context-enriched prompt to the LLM Server. (6) The LLM generates a response, which is sent back through the Middleware to the `Jupyter-Chat` interface for the student. This architecture ensures that AI responses are informed by both immediate actions and broader interaction history, allowing for more adaptive and pedagogically relevant support.

5.3. Proof-of-Concept Demonstration

To demonstrate JELAI's feasibility and capabilities for capturing interaction data and enabling comparative studies, we conducted preliminary evaluations focusing on system performance and two use cases. These serve as proof-of-concept illustrations rather than conclusive studies, given the small sample sizes inherent in initial system testing.

5.3.1. System Performance

JELAI demonstrated stable performance on an A40 GPU. With local LLMs, such as Llama3.1-70b and Gemma2-27b, average response latencies were 5-9s and 2-3s, respectively, acceptable for interactive use. Telemetry processing occurred in near real-time. The JupyterHub deployment supported 20+ concurrent users on a moderate server (6 CPU cores, 16GB RAM) with low per-user resource consumption (CPU < 0.25 cores, RAM 400MB steady state), indicating potential for scalability.

5.3.2. Use Case Examples

Two small-scale studies illustrate JELAI's data collection and experimental capabilities.

1. Help-Seeking in Python Course (N=18): JELAI was used over 7 weeks in an introductory Python course. It logged student code interactions and chat messages with an LLM tutor (Llama3.1-70b). We categorized chat prompts into instrumental vs. executive help-seeking [196]. *Demonstration:* JELAI successfully captured granular, integrated data streams (code activity, chat logs, help-seeking types). This capability allows for exploring relationships between coding behaviours, help-seeking patterns (e.g., relative frequency of executive vs. instrumental requests), and external measures like grades, as well as identifying specific interaction sequences (e.g., help-avoidance despite errors, post-completion verification checks) for qualitative analysis.

2. Prompt Comparison Pilot (N=19): This study demonstrated JELAI's use for A/B testing AI configurations. Novice programmers worked on 4 data science tasks using JELAI with either a generic "helpful assistant" prompt or a "pedagogical" prompt (Gemma2-27b). *Demonstration:* JELAI facilitated the comparison by logging interactions for both groups and automated classification of learners. The pedagogical prompt increased dialogue length (17.7 vs. 10.7 messages) and engaged in primarily instrumental (27.9% vs. 59.6%), rather than executive requests. Programming logs allowed us to hypothesize: unprompted students ran more code (12.8 vs. 8.3 executions) and encountered more errors (7.4 vs. 5.3), a behaviour consistent with productive struggle.

Together, the pilots confirm that JELAI sustains real-time responsiveness, surfaces actionable help-seeking signals, and enables real-time behaviour analysis and rapid A/B testing of prompt designs without infrastructure changes.

5.4. Discussion and Future Work

This paper presented JELAI, a novel, open-source technical platform integrating LA and AI tutoring within Jupyter Notebooks. Its modular architecture allows context-aware AI interaction and, crucially, supports research into student behaviour with GenAI by enabling flexible configuration and data capture. The proof-of-concept demonstration confirmed system stability and its capability to capture rich, multi-modal data for analyzing interaction patterns and comparing pedagogical strategies, although the preliminary studies require validation with larger samples.

Key limitations include the computational cost of powerful local LLMs (though rapidly improving models mitigate this) and the technical expertise currently needed for configuration. Future work will focus on: 1) Enhancing AI interaction with multi-step processing for real-time help classification and proactive interventions [197]. 2) Simplifying configuration via templates and higher-level indicators [198]. 3) Integrating RAG and AI Agents for improved tutoring and tool use. 4) Improving interoperability via standards like xAPI and LTI [187]. 5) Enabling student collaboration features. Research will further investigate how JELAI can foster instrumental help-seeking and whether guided AI dialogue improves learning outcomes. In summary, JELAI's main contribution is its flexible technical framework, designed to enable pedagogically-informed, LA-driven AI tutoring research and development within a widely used educational environment, with ongoing work focused on enhancing its usability, capabilities, and interoperability.

6

LLM Chatbots in High School Programming

Exploring Behaviors and Interventions

Manuel Valle Torre, Marcus Specht, and Catharine Oertel

He who asks a question is a fool for five minutes;
he who does not ask a question remains a fool forever.

Chinese Proverb

This study uses a Design-Based Research (DBR) cycle to refine the integration of Large Language Models (LLMs) in high school programming education. The initial problem was identified in an Intervention Group where, in an unguided setting, a higher proportion of executive, solution-seeking queries correlated strongly and negatively with exam performance. A contemporaneous Comparison Group demonstrated that without guidance, these unproductive help-seeking patterns do not self-correct, with engagement fluctuating and eventually declining. This insight prompted a mid-course pedagogical intervention in the first group, designed to teach instrumental help-seeking. The subsequent evaluation confirmed the intervention's success, revealing a decrease in executive queries, as well as a shift toward more productive learning workflows. However, this behavioral change did not translate into a statistically significant improvement in exam grades, suggesting that altering tool-use strategies alone may be insufficient to overcome foundational knowledge gaps. The DBR process thus yields a more nuanced principle: the educational value of an LLM depends on a pedagogy that scaffolds help-seeking, but this is only one part of the complex process of learning.

This chapter was published in the ACM Symposium on Applied Computing (SAC '26). [1]

6.1. Introduction

As technology becomes increasingly integral to education and society, the need for widespread computational skills and programming grows [2]. Learning to program, however, presents unique challenges: Students frequently struggle with abstract concepts such as loops and conditionals, find error messages cryptic, and may default to memorizing syntax rather than understanding programming logic [3]. These difficulties often lead to frustration, reduced motivation, and high dropout rates, highlighting a critical need for timely, individual support [4, 5].

Recent advancements in Large Language Models (LLMs) present promising solutions for overcoming these educational barriers [6]. By providing personalized, real-time assistance, LLMs have the potential to help students better manage cognitive load, clarify confusing syntax, and focus more deeply on programming concepts [7]. At the same time, the integration of LLMs in educational settings carries risks. Over-reliance on AI assistance may weaken students' independent problem-solving skills, encourage superficial learning strategies, and even hinder the development of key cognitive skills like critical thinking when compared to non-AI approaches [8–10]. Furthermore, potential inaccuracies or misleading responses from LLMs raise concerns about their reliability as educational assistants [6].

While the importance of guiding student use of GenAI is recognized and there are opportunities to foster metacognitive processes [11], there is a lack of research on structured interventions designed to promote effective use of LLMs. Some prior work has explored data-driven guidance systems for help features in tutoring systems [12], but applying these ideas to the context of GenAI requires further investigation. Moreover, most studies focus on university-level learners, leaving high school students less explored [13]. In short, more empirical research is needed to understand the effects and underlying mechanisms of LLM use in education [14].

To address these gaps, this study employs a Design-Based Research (DBR) cycle to investigate and refine the integration of LLMs in a high school programming course. The study is centered on an Intervention Group (IG), where we first identified a core problem: in an unguided setting, a higher frequency of LLM use—particularly for executive, solution-seeking queries—negatively correlated with midterm exam performance [15]. This critical insight informed the design of a targeted pedagogical intervention, which was then implemented and evaluated in the second phase of the course. Data from a contemporaneous Comparison Group (CG) provides additional context for our findings.

The following research questions guide this study:

- **RQ1 - Baseline**: How does the frequency of student interactions with an LLM correlate with their programming performance?
- **RQ2 - Baseline**: How do instrumental vs. executive query types impact learning outcomes in an unguided setting?
- **RQ3 - Intervention**: How does a targeted intervention influence students' help-seeking behaviors and learning outcomes?

The following sections build the case for our questions by reviewing prior work, detailing our research design, presenting the results of our analysis, and discussing the implications for supporting high school programming learners.

6.2. Related Work

Effective pedagogical support extends beyond providing direct solutions; it also involves guiding students toward appropriate help-seeking strategies. Help-seeking theory offers a valuable framework for understanding how students request and utilize assistance, which is highly relevant when introducing AI helpers [16]. Help-seeking has been framed as a positive, strategic part of learning, rather than a sign of failure, and distinguished between executive and instrumental help-seeking [17].

Executive help-seeking is considered non-adaptive; the student aims to minimize effort by having someone else (or an AI) complete the task, for example, by asking, "Can you just give me the answer to this problem?" In contrast, instrumental help-seeking is an adaptive, learning-focused strategy where the student requests just enough help (e.g., a hint or explanation) to overcome an impasse and move forward independently [18]. Instrumental help-seeking is strongly associated with better retention and skill transfer [19]. Ultimately, help-seeking behavior is often considered a key aspect of the broader construct of self-regulated learning (SRL) [16].

This active role is vital for learning, as even well-crafted instructional explanations can be ineffective if the learner does not engage with them to develop their own understanding [20]. By asking instrumental questions, students are prepared to engage in cognitive activities, such as generating inferences or revising their mental models. This process helps them integrate new information with prior knowledge,

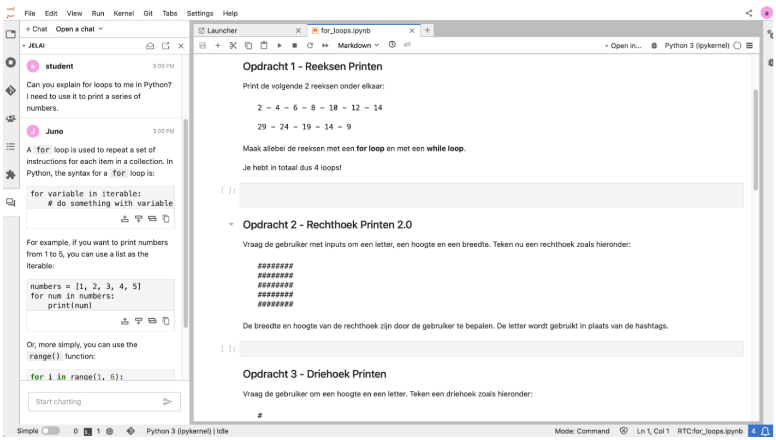

Figure 6.1: JELAI interface with a sample task in the Jupyter Notebook on the right and a task-specific LLM-based tutor on the left

thereby constructing a more robust understanding of the material [21].

LLM-based tutors make this process even more relevant. Whereas prior research on ITS often studied students requesting incremental hints [22], LLMs typically provide complete explanations, akin to bottom-out hints [23]. This fundamental shift from minimal guidance to comprehensive answers makes it essential to reevaluate whether established help-seeking principles remain, and if students can be guided towards more productive learning behaviors [13].

6.2.1. Programming Education and Large Language Models

Novice programmers must simultaneously learn new syntax and logic while also interpreting error messages [24, 25]. Studies comparing novice and expert coders highlight these differences, showing novices make more errors and struggle to achieve program correctness [26]. Effective help-seeking behaviors are crucial for overcoming these obstacles, yet novices often lack the metacognitive skills or confidence to ask for appropriate assistance [12, 19, 27]. These challenges underscore the need for timely, individualized, and scalable educational support.

LLMs offer scalable, personalized support in programming education, potentially replicating the benefits of human tutors by explaining code and reducing stress. Recent advancements of LLMs offer potential solutions for providing this in programming education [6, 7]. LLMs can generate and explain code, answer questions immediately, potentially reduce stress, and improve task efficiency [28]. Furthermore, some students may feel more comfortable asking basic or repetitive questions to an LLM compared to a human instructor [7, 29].

However, the integration of LLMs like ChatGPT also presents significant challenges and risks [6]. Firstly, these tools can complicate the programming learning experience, giving confusing or advanced information [30]. If students fail to see the usefulness of these systems, they may stop using them before getting any benefits from them [31]. Standard LLMs are not designed as pedagogical tools; they typically prioritize task completion over fostering deeper conceptual understanding [8, 9]. This can lead to superficial engagement and potentially hinder the development of independent problem-solving and debugging skills [28].

Recognizing these complexities, researchers have developed specialized LLM-based systems for programming education, aiming to provide help that supports learning. Some prominent examples are: **CodeTutor**: A chatbot using system prompts for educational responses; its interface is similar to that of ChatGPT and allows students to rate feedback at both the message and conversation levels [32]. **CodeAid**: An AI tutor providing targeted support, guiding the student via specific feature selection while avoiding direct solutions [33]. And **CodeHelp**: A web interface using multiple prompts to ensure an appropriate response from the LLM, which also avoids giving away solutions [28].

While these systems represent important advances, the studies evaluating them often rely on post-task surveys or analysis of conversations in a separate chat window, rather than logging interactions that occur seamlessly within the coding environment itself. The system used in this study, JELAI, is designed to address this by integrating the LLM chatbot directly into the notebook environment, similar to modern development interfaces (like GitHub Copilot or Cursor), potentially making interaction more intuitive.

6.3. Methodology

6.3.1. Research Design and Procedure

Integrating LLMs into authentic classroom settings presents a complex, ill-defined problem for which Design-Based Research (DBR) is an ideal methodological framework. DBR enables us to move beyond efficacy questions ("does it work?") to develop a deeper, theoretically grounded understanding of how, why, and under what conditions an intervention functions in a real classroom. Our study follows a classic DBR cycle: we first used help-seeking theory to analyze an existing implementation and identify a core pedagogical problem, then designed and implemented a targeted intervention to address it, with the ultimate goal of generating shareable design principles. The aim of this DBR cycle, therefore, is not statistical generalization to a wider population, but rather the development of a local theory and shareable design principles grounded in the complex, authentic context of this specific classroom.

The design centers on a pre-post analysis of an Intervention Group (IG, N = 18) and a non-equivalent Comparison Group (CG, N = 19) to contextualize the impact of the intervention. We maintained authentic class sizes to preserve ecological validity and avoid confounding variables (e.g., teacher effects) associated with merging disparate cohorts.

Participants in both groups were recruited from elective Informatica classes at two different Dutch high schools, where they were introduced to foundational programming concepts (e.g., variables, conditionals, loops). The non-equivalent Comparison Group was enlisted to provide context on how unguided help-seeking behaviors evolve over a similar timeframe, rather than for direct statistical comparison. In both settings, students used the JELAI environment during class assignments under their teacher's supervision. All participants and their parents provided informed consent.

The IG's course was divided into two six-week phases. Participants were asked to complete two short surveys rating their previous experience with programming and with LLM chatbots. In the pre-intervention phase, the unguided use of JELAI by students was logged, and a midterm exam was administered to identify baseline help-seeking behaviors and their correlation with performance. Based on this initial analysis, we implemented a two-part pedagogical intervention. This included a class-wide discussion on the distinction between instrumental and executive help-seeking, followed by brief, one-on-one sessions where students reflected on their own query patterns with a researcher [21]. In the post-intervention phase, students continued the course with JELAI access, and the intervention's effects were evaluated via logs and a final exam.

The CG used JELAI without any specific guidance for their entire course. The data collected included all interactions with the system and their final exam scores; no midterm exams or interventions were administered in this group.

All exams for both groups tested the concepts taught during the lessons, using a mix of syntax and coding tasks in JELAI, but with access to Juno deactivated.

6.3.2. JELAI Environment and Data Classification

The primary learning environment for both groups was JELAI [34], an open-source platform that integrates an LLM-based chatbot, Juno, directly into a Jupyter Notebook interface (see Figure 6.1). Juno was powered by a Llama3.1:70b[1] open-weights model, guided by a system prompt[2] to act as a Socratic tutor that avoids providing direct solutions. JELAI logged all student interactions—including chat messages, code cell executions, and errors—providing a fine-grained dataset of the learning process[3]. These logs, along with student exam scores, served as the primary data for this study.

To analyze student queries, we employed a multi-step classification process. Drawing from help-seeking theory [17, 18], we first manually coded the IG's pre-intervention queries into three high-level categories: Instrumental (seeking understanding), Executive (seeking solutions), and Other. This manually coded data was used to train and validate a few-shot LLM classifier using DSPy[4], which achieved an 84% accuracy and was subsequently used to classify all queries from the CG. For a more detailed analysis within the IG, queries were also manually coded into finer-grained sub-categories [28, 35] (see Figure 6.2).

6.3.3. Data Analysis

For RQ1, we used Spearman rank correlation (ρ) due to the small sample sizes of our groups and potential non-normality of the data. Similarly, for RQ2, we used Spearman rank correlation (ρ) to assess the relationship between instrumental/executive queries and the grades, as well as a detailed analysis per query type.

For RQ3, investigating the intervention's influence, we employed non-parametric tests: the Wilcoxon signed-rank test was used to compare paired data (e.g., proportion of query types before and after the intervention for the same students, and midterm vs. final exam scores). All statistical tests were conducted with a significance level of $\alpha = 0.05$.

Based on coding and dialog logs, we describe the learning activities of several participants to contextualize our findings and illuminate specific behaviors. We also conduct sequence analysis on the entire group to further explore the behavioral changes. We describe these insights in the Discussion (in Section 6.5).

6.4. Results

This section presents the findings organized by our research questions, first establishing the baseline behaviors observed in both groups and then evaluating the

[1] https://ollama.com/library/llama3.1
[2] https://github.com/mvallet91/JELAI/blob/celdelta/history_app.py#L88
[3] The dataset is available at https://data.4tu.nl/
[4] https://dspy.ai/

impact of the intervention. Students in the Intervention Group (IG, N=18) submitted 480 queries to the LLM, with a mean of 25.4 queries per student. The Comparison Group (CG, N=19) submitted 403 queries, with a similar mean of 21.21 queries per user.

Based on pre-course survey for IG, the students were largely novice programmers, reporting a low average prior programming experience ($M = 1.89, SD = 1.02$) on a 5-point scale. Their self-reported prior usage of LLM chatbots was moderate, corresponding to an average use of 1-2 times per week ($M = 3.06, SD = 1.11$) on a 5-point scale.

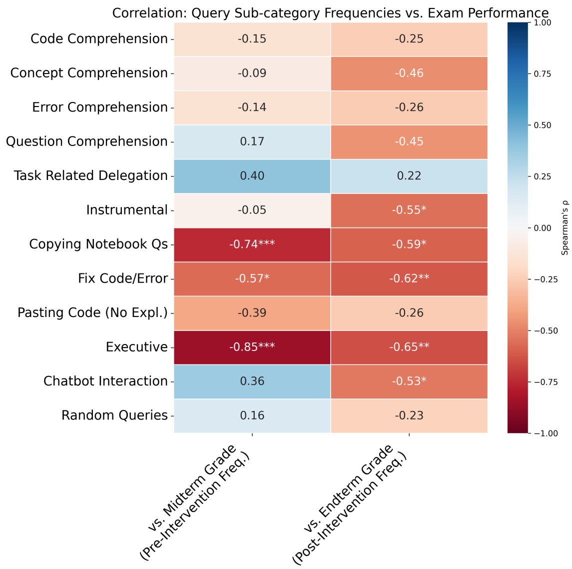

Figure 6.2: Correlations between student interaction types and grades, left column is data until the midterm correlated with the midterm grades, right column is the data after the midterm, correlated with the final exam grades. (* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$)

RQ1: Correlation Between Interaction Frequency and Performance In the IG's pre-intervention phase, we found a significant, negative correlation between the total number of LLM queries and midterm exam scores ($\rho = -0.502, p = 0.034$), indicating that higher interaction frequency was associated with lower performance. This is consistent with prior ITS research suggesting that frequency often signals a struggling student [15]. Further analysis suggested this was a selection effect; after controlling for students' self-reported prior programming experience, which showed a negative trend with interaction frequency ($\rho = -0.442, p = 0.067$), the direct relationship between frequency and grades became non-significant ($p = 0.084$). In the CG, no significant correlation was found between total query frequency and final exam grades ($\rho = -0.059, p = 0.811$)

RQ2: Impact of Instrumental vs. Executive Queries Analysis of query types in

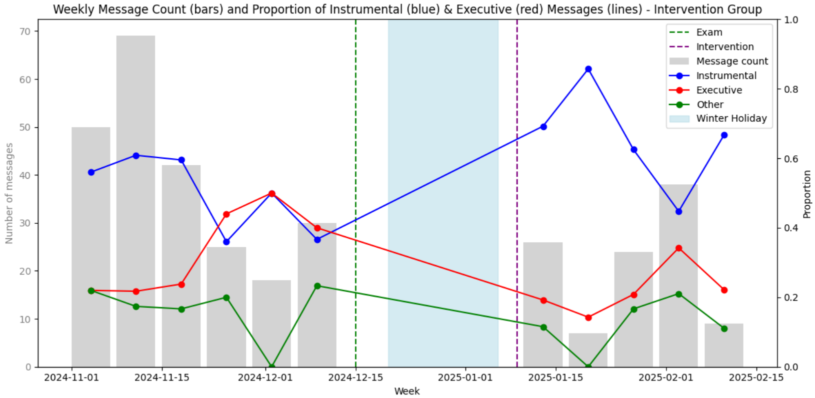

Figure 6.3: Weekly message counts and proportions per type for the Intervention Group

the IG's pre-intervention phase revealed the core problem for our DBR cycle. The most frequent query type was `Concept Comprehension` (24.7%), followed by `Copying Notebook Questions` (19.2%), a clear executive behavior. This was reflected in the correlations: there was a clear and significant negative correlation between the proportion of executive queries and midterm grades ($\rho = -0.851, p < 0.001$), while instrumental queries showed no significant correlation ($\rho = -0.052, p = 0.838$). The heatmap in Figure 6.2 illustrates these relationships across all sub-categories.

In the CG, a similar directional trend was observed, with a weak, non-significant negative correlation between executive queries and final grades ($rho = -0.260, p = 0.281$). As shown in Figure 6.4, this group's engagement declined after an initial novelty period. While the proportion of executive queries shows a downward trend over time, the pattern is marked by high fluctuation and sparse data in later weeks, preventing any firm conclusions about self-correction. This finding suggests that without guidance, unproductive help-seeking patterns do not reliably resolve on their own.

RQ3: Influence of the Pedagogical Intervention The intervention in the IG served as a clear inflection point, prompting a significant shift in behavior (Figure 6.3). A Wilcoxon signed-rank test on paired data (N = 14) showed that the proportion of executive queries, which had been rising, decreased significantly post-intervention, with a large effect size ($W = 15.0, p = 0.036, r = 0.560$). This behavioral change was accompanied by a reduction in overall query volume. While the average final exam grade (M=7.56, SD=1.97) was higher than the midterm grade (M=7.28, SD=2.00), this improvement was not statistically significant ($W = 34.0, p = 0.083$), though the medium-to-large effect size ($r = 0.409$) suggests a potentially meaningful change. This contrasts with the fluctuating, directionless patterns in the CG in Figure 6.4, reinforcing that the observed shift in the IG was a direct result of the intervention.

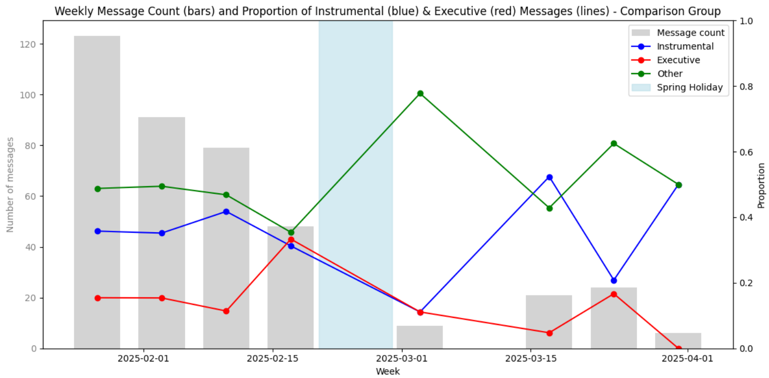

Figure 6.4: Weekly message counts and proportions per type for the Comparison Group

6.4.1. Qualitative Cases of LLM Interaction

Qualitative analysis of the IG's interaction logs provides crucial context for our findings, revealing distinct learner profiles and the nuanced impact of our intervention. High-achieving students demonstrated a variety of effective strategies. Some used the LLM sparingly, relying on strong independent skills, while others effectively used it as a conceptual partner, asking instrumental questions (e.g., "what is the difference between = and ==?") instead of requesting direct fixes.

The intervention's potential is best illustrated by contrasting two students. Mary, who initially relied on executive queries and pasting errors, shifted to asking for general algorithms after the intervention, leading to improved grades (7.30 $\rightarrow$ 8.65). Mary had a high number of total interactions with Juno and was one of the most active students in terms of code cell executions. Conversely, Sam markedly improved their help-seeking strategy—shifting from 58% executive queries to 64% instrumental—but saw their grades decline (5.50 $\rightarrow$ 4.37). Sam's case critically suggests that while behavior can be changed, this may not be sufficient to overcome foundational knowledge gaps.

This behavioral shift was mirrored across the group. To understand changes in student workflows, we analyzed sequential patterns of log events using a fixed-window approach, examining the two actions preceding and two actions following each LLM interaction. Pre-intervention, a dominant pattern was reactive and superficial: students would encounter an error, immediately ask the AI for a fix, and then paste the provided code (`Edit` → `Error` → `Interaction` → `Edit` → `Success`). After the intervention, this pattern became less frequent. In its place, a more proactive and engaged workflow emerged, where students would successfully execute code and then consult the AI, likely for planning or conceptual clarification, before proceeding with another successful action (`Edit` → `Success` → `Interaction` → `Edit` → `Success`). This shift from using the AI as a reactive code-fixer to a proactive conceptual partner reinforces the positive behavioral change observed in the quantitative data.

6.5. Discussion

Addressing RQ1, our finding of a significant negative correlation between unguided LLM usage and exam scores is consistent with prior research [36]. However, our analysis suggests this is not a simple causal relationship, as it contradicts results from similar studies with a more guided LLM programming tool [28, 33]. A plausible explanation is a selection effect, whereby students with less prior programming experience—who are already more likely to struggle—are the ones self-selecting into higher usage of the tool [15]. Controlling for the self-reported prior experience showed that the initial negative correlation is better explained by this selection effect. This trend is echoed in our qualitative observations; the highest-performing students used the LLM sparingly, whereas students who struggled, such as Sam, demonstrated a higher query volume.

For RQ2, our analysis revealed a sharp contrast: executive help-seeking was detrimental to performance, whereas instrumental help-seeking had a more complex and non-significant correlation. This suggests that sustained high frequency of any help-seeking, even theoretically beneficial instrumental queries, likely indicates persistent difficulty that impacts overall performance [12]. When a struggling student turns to the LLM, a tendency to seek direct solutions rather than explanations will naturally hinder learning and exam performance. This pattern of superficial engagement is supported by our finding that 19.2% of queries in the IG involved copying questions. Such a pattern may then lead to a breakdown in the learning process, where the learner fails to actively process the information provided and just pastes the solution into the notebook [20]. Furthermore, the integration of the LLM may be changing the process from writing code to a more complex co-creation and verification, increasing the challenge for less experienced students [30].

The Comparison Group provides important context, illustrating that these unproductive patterns do not self-correct. Without guidance, the CG's engagement with the chatbot was marked by fluctuating help-seeking proportions and an overall decline in use after an initial novelty period (Figure 6.4). Instead of converging on productive strategies, students engaged in less-focused conversations, eventually diminishing the perceived usefulness of the tool and reverting to asking their human teacher for assistance—a finding consistent with prior work [31, 32]. This waning engagement likely explains the lack of a clear correlation between their usage and final grades.

Our mid-course intervention for RQ3 successfully guided students away from detrimental executive behaviors and toward more effective instrumental strategies [13]. Post-intervention data indicated success in shifting relative usage: the percentage of executive questions decreased significantly for the group, with a large effect size. And while there was a reduction in use after the novelty of the first 3 weeks, it was not as marked as in the Comparison Group or similar studies [32]. This behavioral shift toward more instrumental approaches was also demonstrated in the post-intervention query patterns of both Mary and Sam (Section 6.4.1).

6.5.1. Practical Implications for Educators and System Designers

The findings from this study provide several actionable recommendations for practitioners seeking to effectively integrate LLMs into programming education.

For Educators:

- Teach Help-Seeking as a Skill: Do not assume students know how to use LLMs effectively. Our intervention's success suggests that educators should explicitly teach the distinction between instrumental ("how does this concept work?") and executive ("fix my code") queries.

- Design for Metacognition: Assignments should be designed to discourage simple solution-seeking. Instead of tasks that can be solved with a single block of code, educators can create multi-part problems or require students to submit a brief "learning journal" with their code, explaining a key concept they learned from the LLM or a particularly helpful interaction. This incentivizes and makes the learning process visible for students and instructors.

- Use Analytics for Formative Feedback: The methods used in this study—analyzing query types and error rates—can serve as a powerful tool for formative assessment. An instructor could review interaction logs to identify groups of students who rely heavily on executive help and provide targeted, individual guidance long before a summative exam.

For System Designers:

- Prioritize Pedagogy over Task Completion: Educational chatbots should not be neutral answer-bots. The underlying system prompts must be engineered to be pedagogical agents that prioritize learning. This means adding guardrails to the LLM to prevent it from providing direct solutions and instead guiding the student toward a conceptual understanding, much as a human tutor would.

- Implement Adaptive Scaffolding: Systems should be designed to detect patterns of unproductive learning behaviors, like executive help-seeking or help-avoidance. For example, if a student submits consecutive "fix-it" queries, the system could automatically intervene with a metacognitive prompt, such as, "It looks like you're stuck, can you tell me what you were trying to do with this code?" This automates the kind of guidance that proved effective in our intervention.

- Make Learning Visible to the Learner: Future iterations of educational programming environments could include a learner-facing dashboard that provides simple analytics (e.g., "This week, most of your questions were productive—great job focusing on improving your understanding!"). This feedback loop empowers students to monitor and regulate their own learning behaviors, fostering the kind of self-awareness demonstrated by our high-achieving participants.

6.5.2. Contributions

This study makes several distinct contributions. Methodologically, it provides a concrete example of a DBR cycle applied to the integration of generative AI, demonstrating its value in moving from a vaguely defined problem ("students might misuse LLMs") to a specific, evidence-based diagnosis: the negative impact of unguided executive help-seeking. The iterative process of analysis, design, and evaluation refined our understanding, yielding tangible design principles rooted in authentic classroom practice.

Furthermore, the JELAI integration serves as a novel research instrument, capturing granular workflow data (e.g., the sequence of execution after a query) that is lost in studies relying on external chat interfaces. The analytical approach of classifying help-seeking intent (instrumental vs. executive) and analyzing error rates provides a replicable framework for future research.

Empirically, our exploratory study further strengthens evidence that simply providing access to an LLM does not guarantee positive learning outcomes and can, in fact, correlate with lower performance if students adopt non-adaptive strategies. Critically, the qualitative case studies—particularly the divergent outcomes of Mary and Sam—demonstrate that while help-seeking behaviors can be improved through direct intervention, overcoming foundational knowledge gaps presents a significant challenge.

Theoretically, this research extends help-seeking theory into the context of modern generative AI, confirming that the instrumental-executive framework is a powerful lens for understanding student behavior with these tools. By connecting the observed behaviors to established learning science concepts such as the "illusion of understanding" [20], this work highlights the critical need for educational LLMs to be designed not as answer providers, but as pedagogical tools that actively foster student metacognition and critical thinking [10].

6.5.3. Limitations

This study has several limitations that warrant consideration. First, as an exploratory study, the small sample sizes restrict the generalizability of our findings. The observed negative correlation for specific instrumental comprehension subcategories (as shown in Figure 6.2) or the lack of significant correlation for total instrumental frequency may be heavily influenced by the specific usage patterns of the lowest- and highest-performing students.

Second, our analysis was confined to interactions with the integrated JELAI chatbot; unmonitored use of external LLMs (like ChatGPT) or other help sources (peers or family) could have confounded the relationship between measured usage and learning outcomes. While students were asked to only use Juno during class, there was no strict enforcement, and they were allowed to chat with each other or ask the teacher for help.

Third, the qualitative analysis of sequential patterns, while indicative, is based on aggregated top frequencies and does not capture workflow changes with rigor. Finally, other individual factors, such as motivation and self-regulation, as well as variations in exam difficulty, may have also influenced the observed trends.

Finally, while this study focused on student help-seeking behaviors, it did not strictly evaluate the accuracy of the LLM's responses. The risk of hallucinations or bias remains a critical challenge in educational AI, one that pedagogical interventions alone cannot resolve, necessitating further technical safeguards [6].

6.5.4. Future Work

Future work should aim to address these limitations. Replicating this study with larger, more diverse samples is necessary to enhance generalizability and obtain more robust correlation estimates. Research designs could incorporate methods to account for external tool usage and other forms of help-seeking, such as self-reporting or a more detailed log analysis.

Further investigation into sequential patterns could involve more advanced sequence mining techniques on larger datasets to identify macro-trends, or potentially linking specific micro-patterns to learning outcomes at the individual student level [37]. Furthermore, employing experimental designs that control for prior knowledge, motivation, and other individual differences would help isolate the specific impact of different LLM interaction types and frequencies. Investigating the learning trajectories associated with specific help-seeking patterns, particularly for mid-range performers [16], and exploring optimal LLM design features that effectively scaffold learning remain important avenues for further research [14].

6.6. Conclusion

This study investigated the role of LLMs in introductory high school programming, finding that they can act as both a learning partner and a hindrance. Our initial results confirm the risks: increased interaction, particularly driven by executive help-seeking, correlated with lower exam performance. While a targeted pedagogical intervention reduced detrimental behaviors, it did not lead to a statistically significant improvement in average grades. Ultimately, this research concludes that the value of an LLM is not inherent to the technology but is critically dependent on pedagogy. Providing access alone is insufficient and can be counterproductive. To ensure LLMs serve as a "friend" and not a "foe," educators and system designers must co-develop learning environments that explicitly teach and scaffold productive, instrumental learning strategies.

7

Process Collapse or Compensatory Strategy?

Profiling and Scaffolding Help-Seeking with an AI Tutor

Manuel Valle Torre, Ruben Weijers, Marcus Specht, and Catharine Oertel

The real danger is not that computers will begin to think like men, but that men will begin to think like computers.

Sydney Harris

Generative AI offers immediate educational support but risks fostering "metacognitive offloading," in which learners bypass critical error diagnosis in favor of quick solutions. While concerns about help abuse are widespread, empirical evidence characterizing the behavioral processes of unguided GenAI interaction remains limited. This study addresses this gap by investigating process collapse—the breakdown of self-regulated learning cycles—through an exploratory experiment ($N = 40$) involving a diverse cohort of university students and professionals performing data science tasks with an AI tutor. By applying Hidden Markov Models (HMMs) to granular trace data, we identified a pervasive "Tester-Dominant" loop characterized by rapid code iteration. Cluster analysis revealed two distinct archetypes within this pattern: Structured Coders, who use the AI for targeted syntax recall, and Novice Delegators, who rely more on the AI to get solutions. Pretest data revealed that the first group had higher prior self-reported knowledge in Python and data analysis. However, the performance and engagement of both groups were comparable, suggesting that the AI tutor supported multiple viable learning strategies. Furthermore, we tested an adaptive metacognitive intervention to guide participants towards instrumental help-seeking behavior via conversational nudges. The intervention acted as a safety net: while it did not restructure

This chapter was submitted to the Journal of Human-Computer Studies [199].

the underlying process or affect performance, it produced a significant increase in persistence. These findings suggest that for AI-supported novices, "offloading" may be a necessary coping strategy to fill knowledge gaps, and that interventions can have a positive effect by preserving motivation rather than focusing on immediate process optimization.

7.1. Introduction

The rapid integration of Generative Artificial Intelligence (GenAI) systems is shifting our relationship with information; reshaping how knowledge is created, accessed, and applied across all sectors of society [200, 201]. Within this transformation, students have emerged as key adopters, integrating Large Language Models (LLMs) like ChatGPT into their academic lives as a constant source of support in their learning process [202, 203]. This adoption is particularly relevant in the post-pandemic landscape, where reduced face-to-face interaction and an emphasis on autonomy place more responsibility on learners to manage their own progress [204].

When confronted with academic challenges, students increasingly utilize GenAI tools as a learning aid, in a process analogous to help-seeking [205]. Effective help-seeking is a cornerstone of Self-Regulated Learning (SRL), requiring learners to monitor their understanding, identify gaps, and assess the assistance they received [197, 206]. However, this process is traditionally fraught with cognitive and social barriers. Novice learners often lack the domain knowledge to articulate the gaps in their knowledge or process, resulting in vague or insufficient queries even with human tutors [207, 208]. Additionally, the fear of appearing incompetent can deter students from seeking help from instructors or peers [209, 210].

LLMs seem to neutralize these barriers. Cognitively, LLMs will answer a question regardless of poor formulation or unclear intent [211]. Socially, they offer a private, non-judgmental resource [192]. While this accessibility promotes their popularity, the "path of least resistance" encourages help abuse—obtaining solutions with minimal effort [212]. This bypasses essential reflection, leading to *metacognitive laziness* and potentially damaging the development of critical thinking skills [149, 181, 213]. Consequently, there is a risk of widening the divide, with skilled learners leveraging LLMs for support while others use them to avoid learning altogether [197].

However, a binary "productive vs. abusive" framing is insufficient for problem-solving learning tasks. A student asking "How do I import a CSV?" is not necessarily avoiding the cognitive work of data analysis; they may simply be acquiring a necessary tool to proceed [83]. Unlike a human tutor, a chatbot may be seen as a replacement for documentation or search engines for such queries, fundamentally changing what constitutes "appropriate" help-seeking [214]. Distinguishing such *tool acquisition* queries from true *task offloading* is critical for accurately assessing how students use AI support, as one facilitates the learning process, while the other short-circuits it.

This transforms the effective use of LLMs into a new, essential skill: productive inquiry with LLMs [215, 216]. This emerging skill has the unique potential to be scaffolded in the moment of need by embedding pedagogical guidance directly into the design of the AI agents themselves [183]. By creating systems that guide reflection rather than provide direct answers, we can understand help-seeking in conversational contexts and establish design principles for technologies that foster self-regulation. Ultimately, the goal of the help-seeking process is to advance learning, not merely complete a task [217].

This study aims to understand the nature of productive inquiry with GenAI. While this emerging skill can be scaffolded, doing so effectively depends on a clear definition of what it looks like in an authentic learning context and how traditional help-seeking frameworks apply. Our investigation confronts this definitional challenge directly: the line between constructive, instrumental help (e.g., "How do I import a file?") and executive "help abuse" (e.g., "Give me the code to read the data") becomes blurred when the constructive explanation is a single line of code that is also the quick solution.

To investigate this, we conducted an exploratory two-session study designed to profile and scaffold help-seeking. We situated the experiment within a learning environment featuring programming and data science tasks explicitly designed to leverage AI interaction using the 4-Component Instructional Design (4C/ID) model [218]. This longitudinal design allows us to elicit and contrast the help-seeking patterns that emerge as students tackle new tasks over time [185, 219]. By applying Hidden Markov Models (HMMs) to the resulting trace data, we address the following research questions:

- RQ1: What behavioral patterns characterize student help-seeking in AI-supported data science tasks?
- RQ2: Do distinct behavioral archetypes emerge from these patterns, and how do their learning outcomes compare?
- RQ3: How does adaptive metacognitive scaffolding affect student persistence and engagement compared to unguided support?

7.2. Related Work

7.2.1. Theoretical Framework: Self-Regulated Help-Seeking

Academic help-seeking is a fundamental component of self-regulated learning (SRL), understood as a metacognitive cycle involving monitoring, diagnosis, and action [205]. Theoretically, this process consists of five distinct steps: (1) becoming aware of a need for help; (2) deciding to seek help; (3) identifying a potential help source; (4) employing strategies to elicit help; and (5) processing the help received [220]. While conceptually linear, learners often navigate these steps recursively or bypass them entirely under cognitive load [221].

A critical distinction in this process is the learner's intent. **Instrumental help-seeking** involves seeking explanations or hints to enable independent problem-solving, a strategy linked to deep processing and mastery goals [222]. In contrast, **executive help-seeking** is an expedient approach aimed at obtaining a quick answer, often reducing immediate effort but hindering long-term retention [209].

The choice between these strategies is influenced by underlying cognitive traits and prior knowledge [223]. Students with a high *Need for Cognition*—a stable preference for engaging in effortful cognitive tasks—are more likely to employ the processing strategies required for an instrumental approach [224, 225]. Conversely, students experiencing high cognitive load tend toward maladaptive patterns like help abuse or complete help-avoidance [195, 226].

Furthermore, the social context plays a decisive role. Students are more likely to engage in adaptive instrumental strategies when they perceive emotional support and receive explicit praise for asking good questions [227, 228]. This motivational dimension is critical: possessing effective strategies is insufficient if learners lack the motivation to activate them [229]. Indeed, mastery-oriented motivation has been linked not only to deep processing, but also to *persistence* in the face of challenge [228].

7.2.2. The Disruption: Generative AI and Process Collapse

Generative AI represents a paradigm shift in educational support, and students are increasingly using it as a primary source of assistance [185, 214]. This includes extensive applications, ranging from automated content generation to personalized feedback, identifying unprecedented opportunities to democratize access to one-on-one tutoring [230]. By offering immediate, high-fluency support, these tools promise to resolve long-standing barriers in scaling educational assistance [202].

However, this technological affordance raises critical questions about the nature of the learning it promotes [203]. By offering immediate, non-judgmental assistance, LLMs alleviate the social and cognitive costs of asking for help [221]. This creates a "convenience trap" that disrupts the self-regulated learning framework: without guidance, students drift toward *metacognitive offloading*, relying on the AI to bypass cognitive effort rather than scaffold it [149, 181].

Empirical findings suggest that learners engaging in unguided GenAI interaction can fall into a process collapse, where they bypass critical stages of the help-seeking process (specifically steps 1 (diagnosis) and 5 (processing))leading to a copy-paste loop [197, 213]. This creates a paradox: the learner receives a rich explanation, but because they have outsourced the metacognitive work of diagnosis and verification, they fail to internalize the information [231]. Furthermore, this passivity leaves students vulnerable to hallucinations, as they lack the deep engagement necessary to critically evaluate the AI's output [185].

7.2.3. From Static Scaffolding to Dynamic Process Analysis

To address this collapse, we look to instructional design frameworks, primarily 4C/ID, and to Intelligent Tutoring Systems. 4C/ID posits that complex skills require scaffolding that helps learners manage cognitive load during whole-task practice [218]. Managing cognitive load mitigates frustration and abandonment [232], and it may reduce instances in which students ask for a single-step solution rather than decomposing the problem and working progressively [182].

Intelligent Tutoring Systems (ITS) represented the first major effort to automate and scale personalized tutoring [195]. Successful ITS designs implemented this by providing strategic help (e.g., conceptual hints or prompts for self-explanation) rather than direct step-by-step solutions, a method proven to be significantly more effective for deep learning [233]. Classic systems like Betty's Brain demonstrated that adding explicit self-regulation scaffolding and mentor-provided hints not only improved immediate learning outcomes but also prepared students to transfer skills to new contexts [234].

While standard LLMs are not inherently pedagogical, they can function as interactive tutors if constrained to provide this strategic support [235]. For example, providing students with explicit metacognitive prompts while they use GenAI has been shown to improve SRL skills [166], and pedagogical interventions can successfully reduce students' executive queries [236]. However, effective implementation requires a robust learner model to profile student needs and drive these personalization decisions [237]. Recent implementations like SRLbot, LLM-integrated AutoTutor, and CodeRunner Agent demonstrate that when guided by such sound teaching strategies, AI agents can successfully improve engagement and knowledge compared to standard baselines [183, 238, 239].

Finally, to implement adaptive metacognitive support in real-time, we must move beyond static usage metrics to dynamic trace analysis [40]. Self-regulation is a temporal phenomenon, best captured by analyzing the sequence of events (traces) a student generates [49, 240]. Hidden Markov Models (HMMs) have proven effective for this purpose in both structured and Open-Ended Learning Environments (OELEs) [241]. By decoding latent activity patterns from log data—such as monitoring, reflecting, or unproductive wheel-spinning—HMMs allow agents to trigger targeted dialogue strategies precisely when a learner enters a maladaptive phase [55, 242]. The present study adopts this methodology, using HMMs to characterize the process as students learn with AI support.

7.3. Research Methodology

This study adopts an exploratory approach to characterize help-seeking behaviors in AI-supported learning environments. Our first objective is to diagnose and understand the natural patterns of interaction that emerge when learners navigate complex tasks with an AI tutor (RQ1). Secondarily, we investigate if we can identify groups based on their help-seeking behaviors and if there are differences in performance and persistence (RQ2). Finally, we experiment with an adaptive motivational scaffolding and the subsequent impact on behavior, performance, and persistence (RQ3). We first describe the pilots that informed our exploratory focus and the operationalization of student behavior.

Running Example

To make the interaction between components more concrete, we will use a running example of two hypothetical participants, Alex and Sam.

7.3.1. Pilot Study

We conducted two pilot studies to calibrate the system and the tasks.

Pilot 1 (High School Students - Introductory Python): The first pilot involved 18 high school students learning introductory Python through regular lectures, followed by work sessions on a Jupyter environment with an AI tutor. Students had extensive code skeletons in the notebooks, together with instructions and conceptual information. The AI tutor used Llama3.1:70b, prompted as a Socratic tutor (see Appendix), to answer questions and provide guidance. During the 12 weeks of work sessions, students were free to interact with the LLM AI tutor for support. This pilot revealed a natural tendency toward executive help-seeking: students frequently asked "fix this" or "what should I do?" rather than seeking to understand the underlying concepts. This tendency persisted until the teacher's explicit pedagogical intervention reversed it [243], suggesting that instrumental help-seeking behavior can be influenced by metacognitive scaffolding. Since students lacked basic knowledge, they often omitted critical context or asked vague questions, resulting in generic responses.

Pilot 2 (Graduate Students - Data Science Tasks): In a second pilot, we tested a version of the current data science tasks with 7 graduate students with low or no programming experience. These tasks used simple guiding questions without conceptual explanations or code scaffolding. The AI tutor was powered by Gemma2:27b and also served as a Socratic tutor (see Appendix) to answer questions and provide guidance specifically for data science tasks. To eliminate noise caused by poor prompting (observed in Pilot 1), we enhanced the AI tutor's access to context, such as user code and errors, and task objectives. In a few cases, the participant's question was buried among the error logs and chat history, leading the AI tutor in circles instead of answering. Participants became unmotivated and abandoned tasks early.

Implications for Main Study Design: Based on these pilots, we implemented several key design improvements:

1. **Multi-Agent Architecture:** We developed a multi-agent system (described in detail in Section 7.3.5) consisting of an expert and a tutor agent. This architecture enables the real-time generation of adaptive responses and improves the quality when using local LLMs [244]. Furthermore, it allows us to manage the information that the LLM receives in each step, mitigating the risk of getting lost, as in pilot 2.

2. **Task Scaffolding:** Adopting the 4C/ID framework ensured that observed behaviors were due to learner strategy, not insurmountable task difficulty.

3. **Adaptive Intervention Design:** The system should be able to detect conditions and trigger responses in real time. This enables us to evaluate if the change in behavior observed in Pilot 1 could be shifted by the agent itself.

7.3.2. Experimental Design and Procedure

This study was designed as an exploratory investigation into student help-seeking behaviors with AI tutors. We employed a two-session design to assess the stability of help-seeking strategies over time and participants' willingness to return for the second session [185, 219].

- **Session 1 (Exploration)**: Participants engaged with introductory data science tasks (Tasks 1-2).
- **Session 2 (Persistence)**: A follow-up session (Tasks 3-4) two weeks later served to measure voluntary re-engagement and the stability of strategies.

Participants were instructed to engage for at least 1 hour per session, but there was no automatic cutoff. The sessions were separated by approximately two weeks, they took place online, unsupervised, with participants keeping their own time. For analysis, we defined a "work-sprint" as the first continuous period of activity before any inactivity gap exceeding 30 minutes, ensuring we captured focused engagement rather than sporadic check-ins.

Running Example

Alex and Sam are two participants in the study.

7.3.3. Participants

Participants were recruited through open invitations in the institution's forums, mentoring courses for new bachelor students, and mailing lists for individuals interested in participating in research studies. This yielded a diverse cohort of university students and working professionals. Participation was voluntary and driven by a genuine interest in learning; the study was not part of any mandatory coursework. The invitation offered them the opportunity to learn the basics of data science, receive instant AI feedback, and a 15-euro gift card for their time. The reward was not contingent on completing the tasks, but rather on working for 1 hour in each of the two sessions.

From an initial pool of 45 participants, we derived an analytic sample of **N = 40** using engagement filters. These filters removed participants who opted out, those who did not use the system as designed, and those with minimal engagement (less than 10 minutes of interaction). This sample size aligns with similar exploratory studies examining student-AI interaction patterns [166, 182, 245, 246]. The study was approved by our institution's Ethics Committee.

Running Example

Alex, a university student, consents to the study. After completing the questionnaire, they log in to the system and start session 1. Sam, a project manager, follows the same procedure.

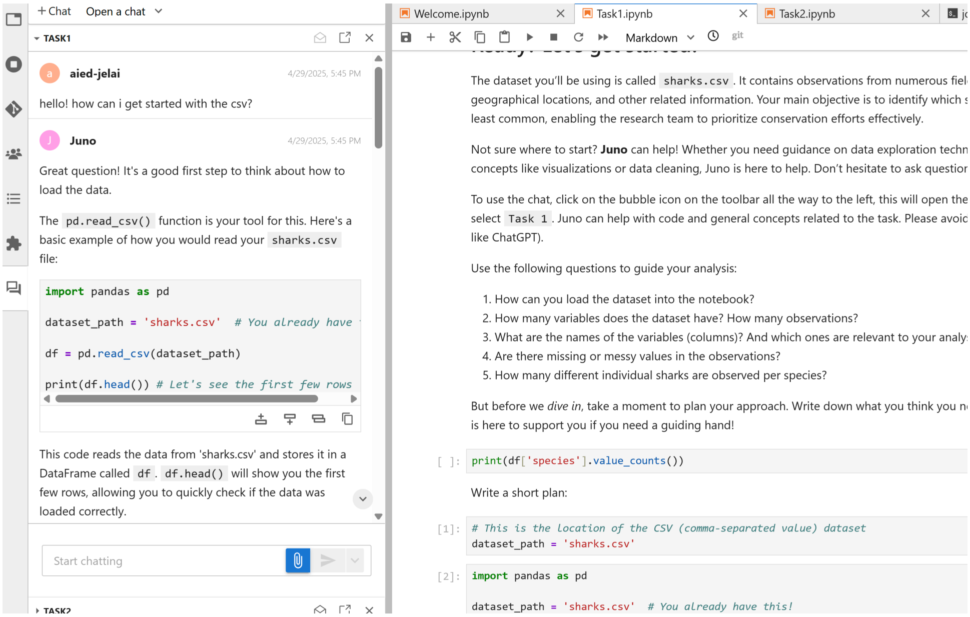

Figure 7.1: The JELAI interface showing a student interaction with the AI tutor Juno. Left: chat panel with conversation history. Right: Jupyter notebook with task instructions and code cells.

7.3.4. Materials: JELAI and Data Science Tasks

The experiment was conducted within **JELAI** (Jupyter Environment for Learning Analytics with AI), an open-source, multi-agent LLM tutor system operating in a JupyterHub environment [177]. The system captures fine-grained trace data from all student activities. These logs can be used by the system for contextual information or to infer underlying cognitive strategies during analysis [240]. Figure 7.1 shows the student-facing interface.

For this study, JELAI's architecture consisted of multiple agents, mainly an expert and a tutor, shown in Figure 7.2 (left). This Mixture of Agents (MoA) architecture enabled us to improve the performance of the open-source LLMs we run locally [244]. This ensures that all data and interactions remain within our servers.

The *expert* agent focuses solely on data science and Python programming. It processes the student's message and the technical context: executed code, error messages, task description, and learning objectives to generate a factual answer. The *tutor* agent integrates the conversation history with expert information to generate an appropriate answer for the student. The tutor is instructed to provide short and concise answers, offering guidance rather than full solutions[1]. The classifier component in the figure is described below in Section 7.3.5, but it does not affect the system in this stage. The workflow was powered by Gemma3 (4B for the expert and 27B for the tutor). Responses took 5-6 seconds to complete; students saw status messages (e.g., 'Let me think for a second') during processing.

The **learning tasks** were designed following the 4C/ID framework [218], which structures learning around whole-task practice with decreasing support. The core of 4C/ID comprises a series of *learning tasks* that form the backbone of the educational experience. The conceptual information and progression of the tasks were in line with Data Science education materials [247, 248]. In this study, there were four data science tasks of increasing complexity, centered on an exploratory data analysis of a realistic "shark scenario" dataset:

- **Session 1 Tasks:** Task 1 (species inventory) and Task 2 (Physical characteristics).
- **Session 2 Tasks:** Task 3 (geolocation analysis) and Task 4 (Movement patterns).

According to the 4C/ID model, these tasks are supported by other components. The *procedural information* included small exercises, with demonstrations and code skeletons provided in the notebooks to reduce the cognitive load of routine syntax recall. The *supportive information* was integrated with conceptual explanations and examples to help learners build mental models (germane load). The AI tutor was designed to complement both, with students requesting further information as needed. The *part-task practice* was achieved by reproducing learned procedures within and across tasks. By providing structured code skeletons and clear sub-goals, this design aims to minimize the initial difficulty that may drive novices to delegate entire tasks to AI, encouraging them instead to engage with specific sub-problems.

Running Example

Both Alex and Sam start session 1 in the JELAI environment. As a 4C/ID learning task, it begins with a scenario: they are junior marine biologists working with a dataset of shark sightings. The first task involves inventorying individuals per species, starting with a brief explanation of Pandas.

7.3.5. Experimental Probe: Adaptive Metacognitive Scaffolding

For a subset of participants (assigned randomly during registration), we activated a motivational intervention. The system identifies when participants ask an executive question followed by an instrumental question, and it positively reinforces the

[1]The full prompt will be added after submission.

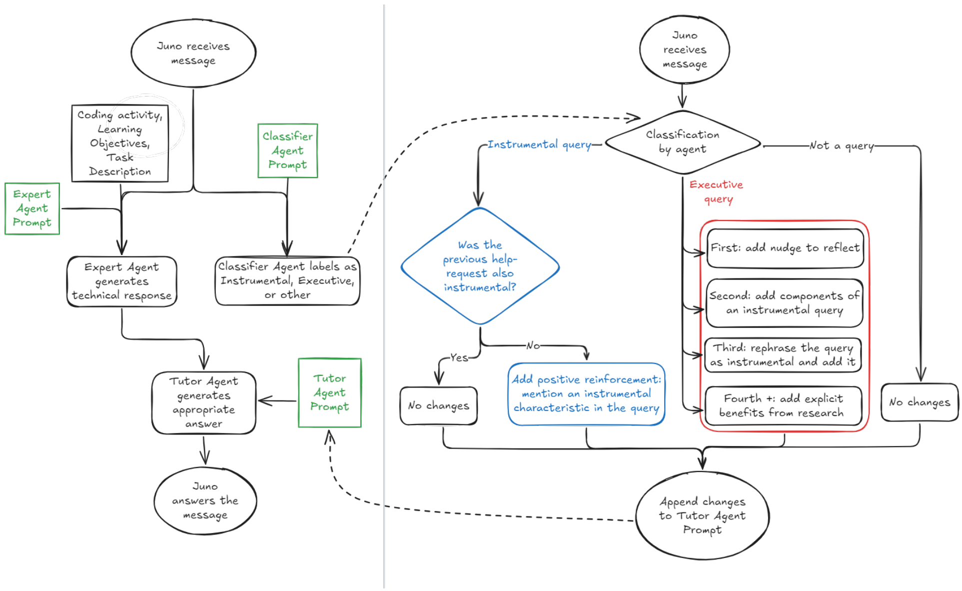

Figure 7.2: JELAI system workflow. Left: General agent pipeline where Juno receives a message, classifies help-seeking intent, generates expert content, and produces a tutoring response. Right: Adaptive intervention logic, showing positive reinforcement for instrumental queries.

student for asking the instrumental question: "That's a great way to formulate a question to understand the content," or "This last question was a good way to approach the problem."

We use a classifier (Figure 7.2) as a third agent to identify the student's intent The *classifier* agent was developed using the MIPROv2 optimizer[2] from the Declarative Self-improving Python (DSPy) framework to categorize student queries according to their Intent (instrumental, executive, or other). The classifier was trained and tested on data from our Pilot 1 with 86% accuracy.

This intervention was based on the adaptive tutoring agent logic displayed in Figure 7.2 (right side of the figure).

1. **Help-Seeking Classification:** Upon receiving a student message, an LLM-classifier first categorizes it as instrumental, executive, or other (greetings, unrelated, answer to a follow-up question by Juno).

2. **Adaptive Response Logic:** Juno's response is shaped by this classification. A shift from executive to instrumental behavior is met with explicit positive reinforcement. This design choice is grounded in research showing that perceived support and praise for good questions are linked to greater achievement gains and encourage the very behaviors we aim to foster [228].

[2] https://dspy.ai/api/optimizers/MIPROv2/

Everything else remains the same, including the base agents, system prompts, and task knowledge base.

Running Example

After importing the data, both Alex and Sam encounter a `KeyError` when filtering species. Both type "fix this error" to their AI tutor, an executive query. The AI tutor responds to each one: "You're getting a KeyError because Pandas is case-sensitive..." Sam moves on to the next step, while Alex asks "why is it called a key error?" The AI tutor responds "It's great that you want to understand the reasoning behind it! In Pandas..."

7.3.6. Measures and Analysis

Intent Classification

For help-seeking **intent**, we developed a **five-class pedagogical taxonomy** using a classifier with a 2-turn context window:

- **Tool Acquisition:** Queries requesting specific syntax or library functions.
- **Instrumental:** Questions aimed at deepening conceptual understanding.
- **Executive:** High-level requests for task completion without conceptual engagement.
- **Response to Bot:** Direct responses to the AI tutor's scaffolding prompts.
- **Other:** Social interaction or off-topic chat.

These categories emerged via manual labelling of 398 messages by 2 coders, with disagreements resolved through discussion. These messages were divided as 70% for training, 20% for testing and 10% for evaluation of the classifier (Weighted F1=0.83, $\kappa = 0.78$).

7.3.7. Behavioral Process Modeling (RQ1)

For the help-seeking *process*, we operationalized behavioral patterns as sequences of trace log events (errors, queries, code edits) [197, 221]. For each participant's study session, we identified the first continuous "work-sprint" and analyzed activity within each task separately.

We trained a single unified HMM on a 12-dimensional observation vector for each event, utilizing all data to learn a shared state representation. The feature vector included standardized measures of:

- **Mechanical Action:** Binary flags for error generation, code editing, and code execution.
- **Intent Class:** One-hot encoding of the 5-class taxonomy.

- **Paste Source:** Binary flags for internal code copy-paste, copy-paste from AI-chat, or pasting content from external sources. We use clipboard logs to infer the source of paste events.
- **Temporal Dynamics:** Log-transformed time since the previous event.

We used Gaussian HMMs with diagonal covariance, trained with the Baum-Welch algorithm (50 iterations, random initialization with seed=42), consistent with prior work on modeling latent behaviors in open-ended learning environments [241, 242]. This approach treats features as conditionally independent given the hidden state, which is appropriate for our mixed feature types. We determined the optimal number of states using the Bayesian Information Criterion (BIC) on the full dataset.

7.3.8. Behavioral Clustering (RQ2)

To identify latent behavioral archetypes, we applied K-means clustering to the participants' HMM state-occupancy vectors. To focus specifically on interaction strategies rather than coding proficiency, we filtered the feature vectors to include only the occupancy of "Help-Seeking" states (e.g., delegating, asking for tools, responding to questions), excluding the dominant "Code-Manipulation" states (testing, debugging, reorganizing code), which are common to all users. We determined the optimal number of clusters based on the Silhouette Score and interpretability of the resulting profiles. We compared the archetype profiles using Welch's t-tests for continuous variables, considering the unequal sample sizes.

Pre-Measures and Baseline Equivalence

The **pre-measures** include basic demographics, self-reported familiarity with Python and data analysis to gauge prior knowledge, and the short form of the Need for Cognition scale (NCS-6) to assess intrinsic motivation for cognitive effort [225].

Performance Measures

Subtask completion is assessed across 42 skills distributed over the four tasks. Scoring was automated via regex patterns matching expected outputs (validated against trace logs):

- **Not Attempted (0 pts):** No evidence in logs.
- **Attempted (1 pt):** Valid attempts (code/cell edits) but no successful solution.
- **Completed (2 pts):** Successful code execution matching solution patterns.

We report two metrics derived from this scoring: *Total Points* (sum of all scores, including partial credit for attempts) and *Completion Rate* (percentage of subtasks where the full 2 points were awarded).

For **Persistence**, we analyze the attrition rate as the proportion of participants returning for Session 2 using Fisher's exact test to compare between groups. We compared the archetype profiles using Welch's t-tests for continuous variables and Chi-squared tests for categorical variables.

Experimental Probe Evaluation (RQ3)

For participants who experienced the experimental intervention, we compared them to the rest of the population using all pre-measures and performance measures described above.

Running Example

The system logs Alex's and Sam's messages, classifying each as "executive" or "instrumental." Their interaction sequences (error → query → edit → success) become HMM observations. Alex returned for Session 2, but Sam did not, contributing to persistence statistics.

7.4. Results

This section presents the results of our experiment, organized by research question. The analysis is based on an analytic sample of $N = 40$ participants, using the first work-sprint per session as described in Section 7.3.6. The trace data and classified messages are published in [243].

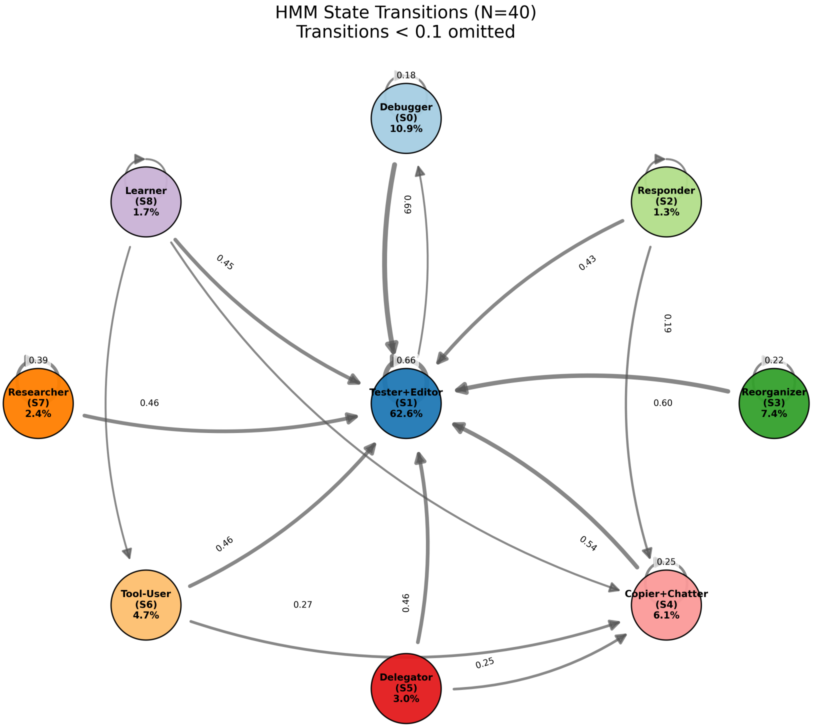

Figure 7.3: Occupancy and Transition Network of the Hidden Markov Model (N=40 participants)

7.4.1. RQ1: Diagnosis - Baseline Strategies

Analysis of the full cohort ($N = 40$ students, $N_{tasks} = 105$ task-segments) reveals the natural problem-solving strategies of students in open-ended data science tasks. These task-segments include all activities per task within the first continuous work-sprint of each participant: if a student worked on tasks 1 and 2 during their first work sprint, they generated two task-segments.

Identified Behavioral States

The HMM trained on the full dataset ($N = 11,925$ events) identified 9 latent behavioral states. The Bayesian Information Criterion (BIC) confirmed this as the optimal configuration for granular yet interpretable separation:

- **State 0 (Debugger) - 10.9%**: Dominated by error events ($+2.9\sigma$). Represents struggling with code execution.
- **State 1 (Tester+Editor) 62.6%**: The most frequent state, characterizing the standard loop of editing and running code.
- **State 2 (Responder) - 1.3%**: Direct interaction with the AI tutor ($+8.8\sigma$ Juno). Represents engaged dialogue.
- **State 3 (Reorganizer) - 7.4%**: Internal code pasting ($+3.5\sigma$). Represents rearranging code blocks.
- **State 4 (Copier+Chatter) - 6.1%**: Pasting AI code ($+3.5\sigma$) and off-topic chat ($+1.7\sigma$). Represents accepting help and possibly thanks or confirmation.
- **State 5 (Delegator) - 3.0%**: Executive commands ($+5.7\sigma$). High-level requests without conceptual engagement.
- **State 6 (Tool-User) - 4.7%**: Tool acquisition queries ($+4.5\sigma$). Seeking syntax or procedural help.
- **State 7 (Researcher/Cheater) - 2.4%**: External pasting ($+6.4\sigma$). Bringing in code from outside the environment.
- **State 8 (Learner) - 1.7%**: Instrumental queries ($+7.7\sigma$). Active conceptual engagement.

Dominant Behavioral Patterns

The occupancy rates and transitions reveal that the "Tester+Editor" state acts as a gravitational center for all students, accounting for the majority of session time, in the center of Figure 7.3. Transitions out of this state typically lead to "Debugger" (when errors occur) or "Tool-User" (when syntax help is needed), before returning to the coding loop. Notably, the "Delegator" state (executive help-seeking) often transitions to "Copier+Chatter" or directly back to "Tester," confirming the *request code* → *paste* → *run* usage pattern.

7.4.2. RQ2: Behavioral Archetypes and Outcomes

Using the clustering methodology described in Section 7.3.8, we identified two distinct behavioral archetypes:

- **Cluster 0: The Delegator** ($N = 13$): Characterized by lower coding autonomy and higher reliance on AI dialogue.
- **Cluster 1: The Structured Coder** ($N = 27$): Characterized by higher autonomy, systematic testing, and specific tool inquiries.

Table 7.1: HMM State Occupancy by Cluster (Session 1)

State Label	Cluster 0	Cluster 1	Diff	p-value
Tester+Editor	52.9%	68.3%	-15.4%	0.001
Debugger	6.3%	12.2%	-5.9%	0.003
Copier+Chatter	14.6%	3.2%	+11.5%	0.002
Delegator	5.2%	1.7%	+3.5%	0.008
Tool-User	7.0%	4.1%	+2.9%	0.022
Reorganizer	5.6%	6.4%	-0.8%	0.490

Table 7.2: Help-Seeking Intent Distribution by Cluster

Intent Category	Cluster 0	Cluster 1	Diff	p-value
Executive	30.1%	27.9%	+2.2%	0.74
Tool Acquisition	31.9%	42.3%	-10.4%	0.62
Instrumental	13.4%	13.9%	-0.5%	0.64
Answer Juno	13.2%	8.5%	+4.7%	0.90
Other	11.4%	7.3%	+4.1%	0.25

Archetype Intent and Behavior Differences

The divergence in strategy is best captured by the HMM State Occupancy profiles (Table 7.1), which reveal significant differences in problem-solving mechanics. Cluster 0 is defined by significantly higher occupancy in passive states: `Copier+Chatter` and `Delegator`, confirming a strategy of delegation and code copying. Cluster 1, conversely, is *Tester+Editor* dominant and spends more time in the `Debugger` state, reflecting a strategy of active construction and self-correction with some trial-and-error. While differences in classified Intent (Table 7.2) follow similar trends (higher Tool Acquisition for Cluster 1), the high variance renders them statistically indistinct, suggesting that the *process* (HMM states) differentiates these users more reliably than the frequencies of individual queries.

Archetype Profiling

To understand the composition of these archetypes, we compared the demographic and pre-test characteristics of the students in each cluster (Table 7.3). A clear distinction emerged based on prior experience: Cluster 1 (Structured Coder) students had significantly higher self-reported Python experience compared to Cluster 0 and significantly higher general data analysis experience. Notably, while there were visible differences in Gender (53.8% vs 33.3%) and Education (30.8% vs 59.3%), these did not reach statistical significance according to Chi-squared tests, likely due to the smaller sample size of Cluster 0 ($N = 13$). There were also no significant differences in Age, Motivation, or Need for Cognition. This confirms that the behavioral archetypes capture a "Novice vs. Intermediate" split, where novices (Cluster 0) compensate for low prior knowledge by relying on the AI for code generation (Executive help-seeking), while intermediates (Cluster 1) use the AI for targeted syntax recall (Tool Acquisition).

Table 7.3: Demographic Profile by Behavioral Archetype

Variable	C0 - Delegators	C1 - Coders	p-value
Continuous			
Python Experience (1-5)	1.83 ($SD = 0.83$)	2.85 ($SD = 1.06$)	**0.003**
Data Analysis Exp (1-5)	1.33 ($SD = 0.65$)	2.56 ($SD = 1.12$)	**<0.001**
Age (1:16-24, ... 5:55+)	2.08 ($SD = 1.51$)	2.30 ($SD = 1.44$)	0.684
Motivation (1-5)	2.67 ($SD = 0.89$)	2.41 ($SD = 0.89$)	0.409
Need for Cognition	3.29 ($SD = 0.50$)	3.30 ($SD = 0.41$)	0.978
Categorical			
Gender (% Female)	53.8%	33.3%	0.305
Education (Bachelor+)	30.8%	59.3%	0.337

Arechetype Engagement and Outcomes

Table 7.4 compares the engagement and performance metrics between the two archetypes across both sessions.

Despite their divergent strategies, both groups invested similar time in Session 1 and achieved statistically equivalent learning outcomes. This suggests a phenomenon of *equifinality*: the "Novice Delegator" strategy of delegating code generation to the AI was nearly as effective for task completion as the "Structured Coder" strategy of manual coding and debugging, at least in the short term. However, the behavioral mechanisms differed fundamentally. As shown in Table 7.1, Cluster 0 relied significantly more on `Copier+Chatter` and `Delegator` states, while Cluster 1 dominated the `Tester+Editor`.

Longitudinally, a divergence in strategic adaptation emerged. Cluster 0 displayed behavioral stability: among returning participants ($N = 6$ pairs), there were no significant shifts in state occupancy between sessions, suggesting they maintained their delegation strategy even as tasks became more complex. In contrast, Cluster 1 ($N = 14$ pairs) demonstrated significant adaptation. In Session 2, they significantly reduced their time in the `Tester+Editor` state and increased their reliance on `Delegation` and `Copying`. This suggests that capable learners adaptively shifted towards AI-offloading strategies to manage increased cognitive load, whereas novices (still novices) remained in their initial pattern.

Persistence was also similar between the two clusters. The return rate for Session 2 was 46.2% (6/13) for Cluster 0 and 51.9% (14/27) for Cluster 1. This demonstrates that the "Novice Delegation" strategy is not inherently less sustainable or more frustrating than the "Structured Coder" strategy. Both pathways are equally viable in an LLM-supported environment.

Table 7.4: Outcomes by Behavioral Archetype (Session 1 & 2)

Metric	**C0 - Delegators**	**C1 - Coders**	**p-value**
Session 1			
Points	23.9 ($SD = 8.0$)	28.3 ($SD = 5.5$)	0.09
Completion Rate	56.3%	63.6%	0.15
Time on Task (min)	62.8	105.2	0.40
Executive Intent (%)	12.5%	8.5%	0.02^{*}
Instrumental Intent (%)	57.4%	73.4%	$< 0.001^{**}$
Session 2 (Persisters)			
Points	17.0 ($SD = 10.3$)	18.9 ($SD = 8.9$)	0.70
Completion Rate	68.5%	72.6%	0.80
Time on Task (min)	52.4	78.2	0.15
Executive Intent (%)	5.9%	12.2%	0.09
Instrumental Intent (%)	69.5%	62.9%	0.24
Rate (Returning for S2)	46.2% (6/13)	51.9% (14/27)	1.00

7.4.3. RQ3: Intervention Mechanism and Persistence

To understand whether the intervention influenced student behavior, we analyzed the mechanism of "Positive Reinforcement." This mechanism was triggered for selected participants when they exhibited the eligible behavior: upon exiting an "Executive" help-seeking state by asking an "Instrumental" question, the system provided a positive metacognitive scaffold (reinforcement). The intervention typically triggered 26.1 minutes (median) into the session, serving as an early-to-mid session reinforcement of effective behavior.

Impact on Help-Seeking Intent

We examined whether receiving the intervention altered the overall help-seeking intent of the participants. Comparing the help-seeking intent distribution of participants who received the reinforcement ($N = 12$) against the non-reinforced population ($N = 28$), we found no significant differences in strategy. The Reinforced group was statistically indistinguishable from the general population in terms of Executive intent (24.6% vs 25.3%, $p = 0.89$) and Instrumental intent (17.1% vs 12.7%, $p = 0.22$). The "Tool Acquisition" intent also showed no significant difference

(34.3% vs 46.0%, $p = 0.12$). Thus, the intervention did not shift the fundamental help-seeking profile of the users.

Impact on Performance and Engagement

In Session 1, participants who received reinforcement achieved statistically equivalent scores ($M = 28.2$ vs $M = 26.2$, $p = 0.40$) and completion rates (64.8% vs 59.6%, $p = 0.28$) compared to the non-reinforced population. However, they were significantly more engaged with the AI chat interface, asking roughly twice as many questions ($M = 41.1$ vs $M = 21.4$, $p = 0.012$). While Reinforced students spent more time on the task (Median 141 min vs 96 min), this difference was not statistically significant due to high variance.

This gap in message volume disappeared in Session 2. Among persisting students, there were no significant differences in the number of questions asked ($M = 18.5$ vs $M = 18.2$, $p = 0.94$), time on task ($p = 0.56$), or performance metrics ($p = 0.53$) between the groups.

Impact on Persistence

The most significant effect of the intervention was on participant retention.To understand the mechanism, we analyzed how many participants actually engaged in the target behavior (Executive → Instrumental shift).

In summary, approximately 35% of the population ($N = 14$) did not trigger the intervention condition, typically by avoiding Executive help-seeking. From those 14, 8 were in the intervention-eligible group, and 6 in the non-eligible group. For the remaining 65% ($N = 26$) who *did* exhibit the target behavior, the intervention defined the outcome:

- **Reinforced ($N = 12$):** Eligible participants received praise and persisted at **83.3%** (10/12).
- **Ignored ($N = 14$):** Non-eligible participants showed the exact same adaptive behavior but received no feedback; only **42.9%** (6/14) returned.

This demonstrates that while the behavior itself (questions and strategies) was equally common in both groups, the *validation* of that behavior was the decisive factor for retention.

Table 7.5: Persistence Rates by Interaction Depth

Group	N	Persisted (S2)	Rate
Reinforced (Eligible, Triggered)	12	10	83.3%
Not-reinforced (Eligible, Untriggered)	8	4	50.0%
Not Eligible (All)	20	6	30.0%
Total	40	20	50.0%

Comparison: Reinforced vs Not Eligible ($p = 0.005$, OR=11.6)

Moreover, this *safety net* effect was universal across behavioral archetypes. The intervention did not influence cluster formation (Independence χ^2, $p > 0.05$) and was equally effective for both groups (Interaction $p = 0.48$). Notably, while the Novice Delegators in the non-reinforced group experienced low persistence (16.7%), those who received reinforcement persisted at the same high rate (83.3%) as the Structured Coders. The intervention thus effectively closed the retention gap between struggling novices and experienced students.

7.5. Discussion

This study aimed to characterize student help-seeking behaviors and find discernible patterns in AI-supported programming tasks. We also sought to determine if an adaptive intervention could influence these patterns. Our findings reveal a nuanced reality: while all students gravitate toward incremental coding cycles (RQ1), novices successfully employed delegation as a compensatory strategy to achieve equifinal outcomes to the more experienced participants (RQ2). Regarding the intervention, while it did not alter these underlying strategies, it acted as a safety net that significantly increased learner persistence (RQ3).

7.5.1. The Universality of the Tester Loop (RQ1)

Our diagnosis of the learner behaviors (RQ1) reveals that in an open-ended coding environment, the `Tester+Editor` loop is the center of all student activity. All participants spent the majority of their time in this state, confirming that the primary mode of interaction with the AI tutor in this context is not conversation but iterative co-construction [249]. On the other hand, the prominent transition from `Delegator` state (asking for code) directly to `Tester` (running it) may represent a form of *metacognitive offloading*, where the assessment of the received help is missing [213].

Furthermore, a tension emerged between the instructional design and the AI's affordances. While the 4C/ID task structure successfully restricted the *scope* of help-seeking to specific sub-problems (preventing full-task offloading), the LLM encouraged shallow *depth* of processing within those subtasks [218]. Instead of engaging in conceptual diagnosis and understanding, students treated the AI as a syntax dictionary or a debugger, entering iterative cycles of tool acquisition and coding that may prioritize immediate function over understanding. Students used the AI to offload the cognitive load of syntax recall and error diagnosis [197].

7.5.2. Two Paths to the Same Outcome (RQ2)

A key finding of this study is the *equifinality* of the observed strategies. Despite the radically different approaches of the Novice Delegator and the Structured Coder, both groups achieved statistically equivalent performance outcomes in terms of points and completion rates. This challenges the binary of "productive vs. abusive" help-seeking [195] in the context of learning with Generative AI. For novices, the

`Delegator` state (asking for code/fixes) may not represent a failure of learning, but a functional compensatory strategy that allowed them to perform at the same level as their more experienced peers.

Crucially, this equivalence was not driven by cognitive disposition—both clusters exhibited virtually identical Need for Cognition scores and similar task engagement—ruling out a "lazy student" interpretation. Our results confirm that prior knowledge drives the choice of strategy [223]: novices gravitated toward delegation while intermediates used targeted tool acquisition. In traditional frameworks, this executive pattern is viewed as maladaptive because it bypasses the cognitive effort required for schema construction [209], and the model predicts it should lead to worse outcomes.

Rather than a failure of self-regulation, however, these participants employed delegation as a *compensatory survival strategy*. Lacking the syntactic fluency to construct code from scratch or ask for specific syntax, they utilized the AI to bridge the gap between their conceptual intent and valid Python syntax. This aligns with recent findings on metacognitive offloading [197], but adds a critical nuance: for novices without prior programming knowledge, this offloading may be the only mechanism that prevents the friction of syntax errors from overwhelming their cognitive load and driving abandonment [195]. Our data suggest that, in the AI-supported context, the AI disrupts the presumed causal chain between help-seeking type and learning outcomes. Delegation produced equifinal performance, not the degraded outcomes the traditional model predicts.

This performance equivalence should not be confused with learning equivalence. While the outcomes were comparable, the process data reveals a risk of stagnation. Longitudinal analysis showed that Structured Coders adaptively shifted their strategies in Session 2 to manage increased task complexity, while Novice Delegators remained in their delegation loops. This suggests that while AI-enabled delegation is an effective scaffold for immediate performance—allowing novices to produce results comparable to their more experienced peers—it risks creating a *competence illusion*: the user feels capable but remains entirely dependent on the tool [149]. Whether this delegation strategy can serve as a stepping stone toward independent mastery, or whether it entrenches dependence, remains an open question that requires longitudinal investigation beyond the scope of this study.

7.5.3. Scaffolding as a Persistence Safety Net (RQ3)

While the adaptive intervention did not change participants' process or help-seeking strategies, it had a profound impact on their persistence in the learning environment (RQ3). The implementation of a "positive reinforcement" for instrumental questions acted as an **agency-preserving safety net**. By offering strategic prompts integrated into the conversation [233], the system likely managed the frustration that leads to help avoidance [195]. Crucially, this effect was achieved without a significant change in performance scores, highlighting the content-independent value of motivational scaffolding [227, 228].

7.5.4. Implications for AI in Education

Our findings carry several implications for the design of AI-supported learning environments. First, regarding **Task Design and AI Scaffolding (RQ1)**, the observed tension between task structure and AI affordances implies that neither instructional design nor AI assistance alone is sufficient. Effectively integrating GenAI requires a dual-layer approach: structured task design to restrict the problem space, paired with active conversational scaffolding to maintain germane load during the interaction. In our study, the 4C/ID framework successfully confined help-seeking to specific sub-problems, preventing full-task offloading, while the AI tutor sustained engagement within those sub-tasks. However, the AI's affordance for shallow processing within sub-tasks remained unaddressed, indicating that future designs must also scaffold the *depth* of interaction, not just its scope.

Second, regarding **Scaffolding the "Novice Strategy" (RQ2)**, the phenomenon of equifinality suggests that delegation is a functional compensatory strategy for novices. Rather than blocking code generation—which risks immediate dropout—educational agents should initially support this strategy, recognizing that it keeps novices productive and engaged. However, our longitudinal data reveal the risk: without intervention, novices exhibited behavioral stagnation, maintaining the same delegation patterns even as tasks grew more complex. To prevent the *competence illusion* from solidifying, systems must be designed to progressively fade AI code generation, gradually transitioning the user from delegating solutions to constructing them; for example, by shifting from complete code responses to pseudocode hints as the learner demonstrates growing proficiency.

Third, regarding **Agency Preservation (RQ3)**, the discrepancy between persistence and performance highlights that "efficiency" is a poor metric for novice support. High-accuracy answers that resolve a bug immediately may fail to preserve the learner's sense of agency. Our most striking finding is that the decisive factor for retention was not the *content* of the scaffolding, but the *validation* of the learner's own adaptive behavior: students who exhibited the same executive-to-instrumental shift but received no acknowledgment persisted at half the rate of those who were reinforced. This implies a concrete design principle: AI tutors should detect and explicitly praise moments when learners self-correct their help-seeking strategy, rather than focusing solely on delivering accurate domain content. Preserving the learner's willingness to engage is a prerequisite for any subsequent cognitive gain. While our intervention targeted help-seeking specifically, this validation mechanism is unlikely to be domain-specific: the same principle of detecting and reinforcing adaptive behavioral shifts in real time could extend to other self-regulation processes such as time management, metacognitive monitoring, or persistence under difficulty, where learners similarly benefit from confirmation that their self-corrective efforts are productive.

7.5.5. Limitations

Our findings must be interpreted within specific boundaries. First, the study's voluntary nature strengthens the interpretation of delegation as a survival strategy rather than academic dishonesty, but this dynamic may differ in high-stakes assessments where performance incentives override learning goals. Second, the observed equifinality appears limited to introductory tasks; as complexity increased in Session 2, the Delegator strategy showed a lower cognitive ceiling than the Structured approach, suggesting delegation may not scale to problems requiring deep decomposition. Third, the intervention dosage was intentionally conservative to preserve user experience; a more aggressive trigger might have forced behavioral adaptation but at the risk of user frustration. Additionally, our sample of 40 participants, while sufficient for the process-level modeling employed, limits the generalizability of between-group comparisons. Finally, while we demonstrated that the safety net preserves agency (persistence), we cannot confirm it promotes mastery. The risk of a *competence illusion* remains significant for learners who rely on the AI scaffold without engaging in independent transfer tasks.

7.6. Conclusion

This study moved beyond performance metrics to characterize how learners actually engage with Generative AI, revealing three core findings. First, in open-ended programming tasks, a rapid generate-and-test loop is the default mode of engagement. Second, within this mode, novices who delegated code generation to the AI achieved outcomes equivalent to more experienced peers who used the system instrumentally, reframing delegation from a failure of self-regulation to a compensatory survival strategy. Third, an adaptive intervention that validated learners' own behavioral shifts dramatically increased persistence, without altering the underlying strategy, demonstrating that agency preservation is an appropriate lever for retention.

These findings expose a fundamental tension in AI-supported education. Delegation works: it closes performance gaps, keeps novices engaged, and allows them to participate in tasks that would otherwise exceed their abilities. But delegation alone does not teach. Our longitudinal data show that novices who relied on this strategy did not adapt as task complexity grew, risking a competence illusion where the feeling of capability masks continued dependence on the tool.

The challenge for the field is therefore not whether to permit AI-enabled delegation—our data suggest that blocking it risks abandonment—but how to design the transition from assisted performance to independent mastery. This requires educational AI systems that can recognize where a learner sits on this trajectory and respond accordingly: supporting the compensatory strategy when it is needed, validating progress when it occurs, and progressively fading the scaffold as competence develops.

8

Conclusion

I don't know where I'm going from here, but I promise it won't be boring.

David Bowie

When a novice learner who delegates every problem to AI achieves the same score as an experienced peer who struggles through it independently, the score tells us nothing. Generative AI has made the final product a useless proxy for competence, and with it, the entire paradigm of outcome-based assessment in digital learning.

This dissertation began with a simple pedagogical question—"What have you tried?"—and arrived at an uncomfortable answer: we built systems that could finally capture what students tried, only to discover that AI had made trying optional. Through three phases—**Understanding Sequences**, **Building Infrastructure**, and **Shaping Behavior**—this work demonstrates that the learning process itself must become the primary object of measurement and support, and that persistence in the learning journey may be more valuable than any outcome we currently measure.

8.1. Summary

This dissertation progressed through three phases. In Phase I, we mapped the landscape of sequence analysis through a systematic review of 74 studies (Chapter 3), built a browser-based tool for MOOC sequence visualization (Chapter 2), and analyzed fine-grained problem-solving sequences in 440 Kaggle notebooks (Chapter 4). In Phase II, we designed JELAI, a modular platform that integrates learning analytics with conversational AI in Jupyter Notebooks (Chapter 5). In Phase III, we conducted two intervention studies: a Design-Based Research study with high school programming students (Chapter 6, N=37) and a behavioral analysis with metacognitive scaffolding for university learners using JELAI v2 (Chapter 7, N=40).

8.2. General Discussion

The individual chapters of this dissertation each address specific research questions, but together they reveal broader patterns that no single study could establish. This section synthesizes six overarching insights that emerged throughout this research.

8.2.1. The Analysis-to-Action Gap

Across this dissertation, a consistent pattern emerged: the field knows how to analyze learning processes, but rarely acts on those analyses. Our systematic review (Chapter 3) found that only 2 of 74 studies implemented actual interventions based on sequence insights. ELAT (Chapter 2) demonstrated that visualizing learning sequences at scale was feasible, yet the resulting insights remained descriptive. Our Kaggle notebook analysis (Chapter 4) showed that process data could meaningfully distinguish novice from expert strategies, but the data came from static artifacts analyzed after the fact.

The gap between understanding and action persisted at every level of granularity we examined. Coarse-grained MOOC logs, fine-grained notebook interactions, and even our own initial deployments of JELAI (Chapter 5) followed the same pattern: rich data, retrospective analysis, no intervention. This suggests that the barrier to closing this gap was never purely technical [151]. The field lacked frameworks that treated process data as something to act on in the moment, and the infrastructure to support such action within authentic learning environments. Furthermore, every system or domain required proprietary data-collection solutions, making it difficult to build on previous work. Bridging this gap required building JELAI as a platform for concurrent analysis and intervention, with an open-source system already popular across educational domains.

8.2.2. The Pedagogy Debt

Building the right infrastructure proved necessary but insufficient. JELAI successfully captured fine-grained interactions and delivered context-aware AI tutoring (Chapter 5), yet early unguided deployments revealed that students used the AI to bypass learning rather than support it. Our high school study (Chapter 6) confirmed this: without pedagogical guidance, students defaulted to executive help-seeking strategies [220] that correlated strongly with lower performance. Chapter 7 reinforced the finding that the raw availability of conversational AI led to syntax-level delegation rather than genuine diagnosis, even within a carefully designed task structure.

Every AI-powered educational system carries a hidden cost: the pedagogical design effort required to prevent the tool from undermining the learning it is supposed to support. We can call this the *pedagogy debt*. A system that answers questions correctly is trivial to build; a system that answers questions *productively* requires understanding the learner's intent, their current knowledge state, and the appropriate level of scaffolding. Our multi-agent architecture in JELAI v2, with its classifier, expert, and tutor pipeline, represents one approach to paying this debt. Commercial AI providers, optimized for user satisfaction and task completion, currently externalize this debt entirely onto educators and students.

8.2.3. The Self-Correcting Myth

A persistent assumption in educational technology holds that with enough exposure and practice, students will naturally develop productive strategies. Two independent datasets in this dissertation challenge this assumption. In the comparison group of Chapter 6, where high school students used JELAI without intervention, unproductive help-seeking patterns did not diminish over time; engagement declined after initial novelty, but the proportion of executive queries remained constant. In Chapter 7, "Novice Delegators" maintained stable behavioral patterns across two sessions separated by two weeks, showing no spontaneous shift toward the "Structured Coder" archetype.

In every sample we examined, bad habits with AI did not self-correct. Students who began with executive delegation strategies continued with them unless explicitly guided otherwise [250]. This finding has practical urgency: it means that the widespread approach of providing AI tools and assuming students will "figure it out" is likely to entrench unproductive patterns rather than resolve them.

8.2.4. The Novice Paradox

Our earliest work established that novice and expert problem-solving processes differ substantially. Kaggle notebooks (Chapter 4) showed that novices follow longer, linear workflows, whereas experts iterate more efficiently. In classroom settings (Chapters 6 and 7), novices defaulted to trial-and-error with executive help-seeking while experienced learners used AI for tool acquisition and conceptual clarification.

Historically, process differences also manifested in the product: novices produced lower-quality work, incomplete solutions, or more errors. Generative AI has severed this link [83]. In Chapter 7, "Novice Delegators" achieved statistically equivalent performance to "Structured Coders" despite fundamentally different problem-solving processes. Delegation served as an effective compensatory strategy, allowing novices to produce expert-quality outputs without developing expert-level understanding.

This creates a paradox for education: the students who need the most support are now much harder to identify, because their products have become indistinguishable from those of experienced learners. Process data, captured through systems like JELAI, becomes essential for identifying these students before they disengage or develop a false sense of competence.

8.2.5. From Rubber Duck to Solution Vending Machine

The introduction of this dissertation recounts how human tutors use the question "What have you tried?" to prompt articulation, diagnosis, and reflection. This mirrors the well-known *rubber duck* debugging principle: explaining a problem clearly is often sufficient to reveal the solution (or at least the right question to ask), because the act of articulation forces structured thinking and reveals knowledge gaps [251]. In both cases, the cognitive benefit comes from the student explaining their reasoning *to* an interlocutor.

Large Language Models have inverted this dynamic. Instead of students articulating their problem to a system that listens, the system articulates solutions to students who receive them. The direction of explanation has flipped, and with it, the locus of cognitive effort. Chapter 7 provides empirical support for this shift: despite having access to a conversational AI tutor, participants primarily engaged in "Tester+Editor" loops, treating the AI as a debugging tool rather than a reasoning partner. The conversational potential of LLMs went largely unused. Students preferred receiving targeted fixes to engaging in the kind of reflective dialogue that both human tutoring and rubber duck debugging encourage.

Our intervention studies suggest that reversing this default requires deliberate scaffolding. In Chapter 6, explicit instrumental help-seeking instruction shifted conversational patterns from reactive (Error → AI → Paste) to proactive (Success → AI → Clarify), partially restoring the student-as-explainer dynamic. In Chapter 7, even simple positive reinforcement for productive inquiry increased engagement. The technology for rich, reflective dialogue exists; the challenge is designing interactions that activate it.

8.2.6. Persistence as a Promising Signal

Each phase of this dissertation progressively undermined confidence in traditional outcome measures. Chapter 4 showed that polished notebooks hide the problem-solving process. Chapter 6 showed that behavioral improvement does not reliably translate to grade improvement. Chapter 7 showed that students with fundamentally different levels of understanding can produce identical outputs.

If the product no longer reflects what the student has learned, the question becomes which alternative metrics might. Our probe intervention in Chapter 7 offers one candidate: positive reinforcement after executive help-seeking did not change performance or behavior, but it did increase the likelihood that participants returned for a second session. This was exploratory (N=12), but it connects to the argument that what matters is whether institutional practices lead students to *want* to persist [252]. If persistence proves robust under dedicated investigation, it would reframe educational scaffolding around sustained engagement rather than immediate performance [228].

8.3. Implications for Practice

Beyond the insights discussed above, this dissertation offers concrete implications for three audiences: researchers, educators, and system designers.

8.3.1. For Researchers

Conceptual contributions. This work provides a systematic framework for sequence analysis in learning analytics, mapping trace data to learning actions to learning processes. It extends help-seeking theory [209, 220] to GenAI interactions, confirming that the instrumental-executive distinction applies to LLM use and identifying tool acquisition as an additional category in which students use LLMs to replace search engines and reference materials.

The concept of lightweight metacognitive scaffolding generalizes beyond programming: any context in which learners interact with GenAI could benefit from simple reflective prompts that encourage articulation before solution-seeking becomes habitual [166, 233]. Curriculum designers could adopt the instrumental-executive distinction as a design heuristic, explicitly teaching students *how* to ask productive questions, just as we teach information literacy for web search.

Methodological contributions. We contribute open datasets, an interactive dashboard for the systematic literature review, and an HMM-based method for characterizing behavioral states in human-AI collaboration. This approach offers a reusable template applicable beyond education to any domain where process quality matters as much as outcomes, such as clinical decision support or software engineering. Broader adoption will require simplified tooling and partnerships with LMS providers to make fine-grained data capture the standard rather than the exception.

Empirical findings and open questions. Executive help-seeking correlates strongly with lower performance, unproductive patterns do not self-correct, and prior knowledge predicts delegation behavior. Behavioral change is achievable with large effect sizes but does not guarantee performance gains, and our exploratory evidence suggests that persistence may matter more than immediate performance for at-risk learners.

These findings open several avenues. The equifinality result demands new process-based assessment instruments that can operate at scale, moving beyond product rubrics toward evaluating the quality of the learning journey. The persistence effect was measured over weeks; longitudinal studies spanning semesters are needed to determine whether early metacognitive scaffolding translates into sustained self-regulation. Finally, extending these methods to essay writing, mathematics, and scientific reasoning will reveal whether the dynamics we observed are domain-specific or generalizable. The JELAI platform and the datasets we release provide a concrete starting point for this work.

8.3.2. For Educators

Teaching productive inquiry as an explicit skill. Our high school study demonstrated that a single focused intervention shifted help-seeking behavior with a large effect size. Teachers introducing AI tools could dedicate one or two class sessions early in the term to distinguish between "fix this for me" and "help me understand why this doesn't work," using concrete examples from student interactions. At the university level, instructors could integrate help-seeking literacy into course syllabi by including a required module on productive AI use, with periodic reflection assignments that ask students to categorize their own AI interactions.

Using process data for formative feedback. Rather than relying solely on grades, educators can use process data to intervene earlier. Reviewing aggregated help-seeking logs before summative assessments can identify students who are predominantly copy-pasting solutions. For larger cohorts, automated dashboards could flag behavioral patterns suggesting disengagement or unproductive delegation.

Designing tasks that resist shortcutting. Task design is a critical lever. Multi-part problems in which each step builds on the previous one, tasks that require students to explain their reasoning alongside their solution, and open-ended projects where no single correct answer exists all encourage instrumental help-seeking because the AI cannot produce a final product without the student's contextual judgment.

8.3.3. For System Designers

Paying the pedagogy debt/Bridge the pedagogy gap Most commercial AI assistants optimize for the right solution or task completion. Our findings show that this design choice undermines learning. System designers building AI features for educational contexts should configure LLM behavior to prioritize understanding over solution generation: guiding questions before complete solutions, hints before answers, and engagement checks before revealing results. The JELAI multi-agent architecture demonstrates that this is feasible through prompt engineering and agent pipelines, without modifying the underlying language model. For commercial coding assistants or chatbots, this manifests as an optional "learning mode" that scaffolds suggestions rather than auto-completing entire code blocks. These systems, however, still put the responsibility on the student to use them productively.

Implementing real-time adaptive scaffolding. Unproductive help-seeking patterns do not self-correct. System designers should implement real-time classification of user interactions to detect unproductive patterns and trigger appropriate scaffolding. This scaffolding need not be heavy-handed: our most effective intervention was a simple positive reinforcement message when students shifted toward instrumental queries. The key technical challenge is latency: scaffolding must be delivered within the flow of the interaction, not as post-hoc feedback.

Making learning visible. System designers should build features that make the learning process visible to both learners and instructors: dashboards showing help-seeking patterns over time for metacognitive self-awareness, and aggregated views of class-wide behavior for instructor oversight. Production-ready implementations will need to address usability, privacy compliance (GDPR, FERPA), scalability, and integration with existing institutional systems. The systems presented here are research prototypes; achieving broad impact will require partnerships with educational institutions and technology providers.

8.4. Limitations and Scope

We acknowledge several limitations that affected this research. The sample sizes in our intervention studies were relatively modest: Chapter 6 involved 37 high school students from a single school in the Netherlands, while Chapter 7 included 40 university students and professionals who were self-selected and highly motivated. These relatively small samples limit the generalizability of findings and intervention effects when applied to broader populations.

The studies were also temporally and contextually constrained. All interventions were short-term, spanning weeks rather than months or years, which prevented us from measuring long-term skill retention or sustained behavioral change. The research focused exclusively on programming and data science domains within computational notebook environments. While the principles may transfer, our empirical work does not extend to essay writing, mathematics, or other domains where LLMs are equally prevalent.

Cultural and institutional contexts also limit generalizability. The high school study was conducted within the Dutch education system, which may have specific pedagogical norms and student expectations that differ from other contexts. All participants were self-selected volunteers, potentially introducing selection bias toward students already interested in improving their learning strategies.

Finally, the systems presented are open-source research prototypes rather than production-ready platforms. Considerations such as privacy regulations, institutional integration, scalability to thousands of concurrent users, and long-term maintenance remain important challenges for future work.

8.5. Final Remarks

The "What have you tried?" question—once a pedagogical tool that teachers used to understand student thinking—is now an educational necessity. When the final answer reveals nothing about the learning that occurred to produce it, the process is all we have left.

This dissertation provides the tools to capture that process: the analytical methods to identify productive and unproductive patterns, the infrastructure to observe them in real time, and the frameworks to intervene before disengagement becomes permanent. But the challenge ahead is not technical—it is cultural. It requires convincing students, educators, institutions, and policymakers that in the age of AI, the journey matters more than the destination, and that keeping students engaged in productive struggle is more valuable than optimizing for performance metrics that AI has rendered meaningless.

Bibliography

[1] M. Valle Torre, E. Tan and C. Hauff. 'edX log data analysis made easy: introducing ELAT: An open-source, privacy-aware and browser-based edX log data analysis tool'. In: *Proceedings of the Tenth International Conference on Learning Analytics & Knowledge*. LAK '20. New York, NY, USA: Association for Computing Machinery, Mar. 2020, pp. 502–511. DOI: 10.1145/3375462.3375510.

[2] J. Reich and J. A. Ruipérez-Valiente. 'The MOOC pivot'. In: *Science* 363.6423 (2019), pp. 130–131.

[3] D. Davis, G. Chen, C. Hauff and G.-J. Houben. 'Activating learning at scale: A review of innovations in online learning strategies'. In: *Computers & Education* 125 (2018), pp. 327–344.

[4] D. Davis, V. Triglianos, C. Hauff and G.-J. Houben. 'Srlx: A personalized learner interface for moocs'. In: *European conference on technology enhanced learning*. Springer, 2018, pp. 122–135.

[5] T. Robal, Y. Zhao, C. Lofi and C. Hauff. 'IntelliEye: Enhancing MOOC learners' video watching experience through real-time attention tracking'. In: *Proceedings of the 29th on hypertext and social media*. ACM, 2018, pp. 106–114.

[6] Z. A. Pardos, S. Tang, D. Davis and C. V. Le. 'Enabling real-time adaptivity in MOOCs with a personalized next-step recommendation framework'. In: *Proceedings of the fourth (2017) ACM conference on learning@ scale*. ACM, 2017, pp. 23–32.

[7] O. Ferschke, D. Yang, G. Tomar and C. P. Rosé. 'Positive impact of collaborative chat participation in an edX MOOC'. In: *International conference on artificial intelligence in education*. Springer, 2015, pp. 115–124.

[8] M. Wen, D. Yang and C. Rose. 'Sentiment analysis in MOOC discussion forums: What does it tell us?' In: *Educational data mining 2014*. Citeseer, 2014.

[9] P. J. Guo, J. Kim and R. Rubin. 'How video production affects student engagement: An empirical study of MOOC videos'. In: *Proceedings of the first ACM conference on Learning@ scale conference*. ACM, 2014, pp. 41–50.

[10] J. Kim, P. J. Guo, D. T. Seaton, P. Mitros, K. Z. Gajos and R. C. Miller. 'Understanding in-video dropouts and interaction peaks inonline lecture videos'. In: *Proceedings of the first ACM conference on Learning@ scale conference*. ACM, 2014, pp. 31–40.

[11] N. Li, Ł. Kidziński, P. Jermann and P. Dillenbourg. 'MOOC video interaction patterns: What do they tell us?' In: *Design for teaching and learning in a networked world*. 2015, pp. 197–210.

[12] D. Davis, G. Chen, C. Hauff and G.-J. Houben. 'Gauging MOOC learners' adherence to the designed learning path.' In: *International Educational Data Mining Society* (2016).

[13] K. Veeramachaneni, S. Halawa, F. Dernoncourt, U.-M. O'Reilly, C. Taylor and C. Do. 'Moocdb: Developing standards and systems to support mooc data science'. In: *arXiv preprint arXiv:1406.2015* (2014).

[14] Z. A. Pardos and K. Kao. 'moocRP: An open-source analytics platform'. In: *Proceedings of the second (2015) ACM conference on learning @ scale*. L@S '15. Vancouver, BC, Canada, 2015, pp. 103–110.

[15] S. I. De Freitas, J. Morgan and D. Gibson. 'Will MOOCs transform learning and teaching in higher education? Engagement and course retention in online learning provision'. In: *British Journal of Educational Technology* 46.3 (2015), pp. 455–471.

[16] J. A. Greene, C. A. Oswald and J. Pomerantz. 'Predictors of retention and achievement in a massive open online course'. In: *American Educational Research Journal* 52.5 (2015), pp. 925–955.

[17] K. S. Hone and G. R. El Said. 'Exploring the factors affecting MOOC retention: A survey study'. In: *Computers & Education* 98 (2016), pp. 157–168.

[18] R. F. Kizilcec, C. Piech and E. Schneider. 'Deconstructing disengagement: analyzing learner subpopulations in massive open online courses'. In: *Proceedings of the third international conference on learning analytics and knowledge*. ACM, 2013, pp. 170–179.

[19] C. Milligan, A. Littlejohn and A. Margaryan. 'Patterns of engagement in connectivist MOOCs'. In: *Journal of Online Learning and Teaching* 9.2 (2013), pp. 149–159.

[20] T. Phan, S. G. McNeil and B. R. Robin. 'Students' patterns of engagement and course performance in a massive open online course'. In: *Computers & Education* 95 (2016), pp. 36–44.

[21] A. Ramesh, D. Goldwasser, B. Huang, H. Daumé III and L. Getoor. 'Modeling learner engagement in MOOCs using probabilistic soft logic'. In: *NIPS workshop on data driven education*. Vol. 21. 2013, p. 62.

[22] S. L. Watson, W. R. Watson, J. H. Yu, H. Alamri and C. Mueller. 'Learner profiles of attitudinal learning in a MOOC: An explanatory sequential mixed methods study'. In: *Computers & Education* 114 (2017), pp. 274–285.

[23] J. Wintrup, K. Wakefield and H. C. Davis. 'Engaged learning in moocs: a study using the UK engagement survey'. In: (2015).

[24] D. Davis, G. Chen, I. Jivet, C. Hauff and G.-J. Houben. 'Encouraging metacognition & self-regulation in moocs through increased learner feedback.' In: *Lal@ lak*. 2016, pp. 17–22.

[25] R. F. Kizilcec, M. Pérez-Sanagustín and J. J. Maldonado. 'Recommending self-regulated learning strategies does not improve performance in a MOOC'. In: *Proceedings of the third (2016) ACM conference on learning@ scale*. ACM, 2016, pp. 101–104.

[26] A. Littlejohn, N. Hood, C. Milligan and P. Mustain. 'Learning in moocs: Motivations and self-regulated learning in moocs'. In: *The Internet and Higher Education* 29 (2016), pp. 40–48.

[27] J. Renz, D. Hoffmann, T. Staubitz and C. Meinel. 'Using A/B testing in MOOC environments'. In: *Proceedings of the sixth international conference on learning analytics & knowledge*. ACM, 2016, pp. 304–313.

[28] K. Veeramachaneni, F. Dernoncourt, C. Taylor, Z. Pardos and U.-M. O'Reilly. 'Moocdb: Developing data standards for mooc data science'. In: *AIED 2013 workshops proceedings volume*. Vol. 17. Citeseer, 2013.

[29] C. Shi, S. Fu, Q. Chen and H. Qu. 'VisMOOC: Visualizing video clickstream data from massive open online courses'. In: *2015 IEEE Pacific visualization symposium (PacificVis)*. IEEE, 2015, pp. 159–166.

[30] H. J. Pijeira Díaz, J. Santofimia Ruiz, J. A. Ruipérez-Valiente, P. J. Muñoz-Merino and C. Delgado Kloos. 'A demonstration of ANALYSE: a learning analytics tool for open edX'. In: *Proceedings of the third (2016) ACM conference on learning@ scale*. ACM, 2016, pp. 329–330.

[31] Z. A. Pardos, A. Whyte and K. Kao. 'moocRP: enabling open learning analytics with an open source platform for data distribution, analysis, and visualization'. In: *Technology, Knowledge and Learning* 21.1 (2016), pp. 75–98.

[32] G. Lopez, D. T. Seaton, A. Ang, D. Tingley and I. Chuang. 'Google BigQuery for education: framework for parsing and analyzing edX MOOC data'. In: *Proceedings of the fourth (2017) ACM conference on learning@ scale*. ACM, 2017, pp. 181–184.

[33] D. T. Seaton, Y. Bergner, I. Chuang, P. Mitros and D. E. Pritchard. 'Who does what in a massive open online course?' In: (2014).

[34] S. Fu, J. Zhao, W. Cui and H. Qu. 'Visual analysis of MOOC forums with iForum'. In: *IEEE Transactions on Visualization and Computer Graphics* 23.1 (Jan. 2017), pp. 201–210.

[35] Y. Chen, Q. Chen, Mingqian Zhao, S. Boyer, K. Veeramachaneni and H. Qu. 'DropoutSeer: Visualizing learning patterns in massive open online courses for dropout reasoning and prediction'. In: *2016 IEEE conference on visual analytics science and technology (VAST)*. Oct. 2016, pp. 111–120.

[36] J. S. Ruiz, H. J. P. Díaz, J. A. Ruipérez-Valiente, P. J. Muñoz-Merino and C. D. Kloos. 'Towards the development of a learning analytics extension in open edX'. In: *Proceedings of the second international conference on technological ecosystems for enhancing multiculturality*. ACM, 2014, pp. 299–306.

[37] O. Poquet, V. Kovanović, P. de Vries, T. Hennis, S. Joksimović, D. Gašević and S. Dawson. 'Social presence in massive open online courses'. In: *International Review of Research in Open and Distributed Learning* 19.3 (2018).

[38] B. Laugwitz, T. Held and M. Schrepp. 'Construction and evaluation of a user experience questionnaire'. In: *HCI and usability for education and work*. Ed. by A. Holzinger. Berlin, Heidelberg, 2008, pp. 63–76.

[39] M. Schrepp, A. Hinderks and J. Thomaschewski. 'Applying the user experience questionnaire (UEQ) in different evaluation scenarios'. In: *International conference of design, user experience, and usability*. Springer, 2014, pp. 383–392.

[40] M. Valle Torre, C. Oertel and M. Specht. 'The Sequence Matters in Learning - A Systematic Literature Review'. In: *Proceedings of the 14th Learning Analytics and Knowledge Conference*. LAK '24. New York, NY, USA: Association for Computing Machinery, Mar. 2024, pp. 263–272. DOI: 10.1145/3636555.3636880.

[41] H. El Aouifi, M. El Hajji, Y. Es-Saady and H. Douzi. 'Predicting learner's performance through video sequences viewing behavior analysis using educational data-mining'. en. In: *Education and Information Technologies* 26.5 (Sept. 2021), pp. 5799–5814. DOI: 10.1007/s10639-021-10512-4.

[42] A. A. Mubarak, H. Cao and S. A. Ahmed. 'Predictive learning analytics using deep learning model in MOOCs' courses videos'. en. In: *Education and Information Technologies* 26.1 (Jan. 2021), pp. 371–392. DOI: 10.1007/s10639-020-10273-6.

[43] D. A. Dever, N. A. Sonnenfeld, M. D. Wiedbusch and R. Azevedo. 'Pedagogical Agent Support and Its Relationship to Learners' Self-regulated Learning Strategy Use with an Intelligent Tutoring System'. In: *Artificial Intelligence in Education*. Ed. by M. M. Rodrigo, N. Matsuda, A. I. Cristea and V. Dimitrova. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2022, pp. 332–343. DOI: 10.1007/978-3-031-11644-5_27.

[44] S. Li, J. Zheng and S. Lajoie. 'Temporal structures and sequential patterns of self-regulated learning behaviors in problem solving with an intelligent tutoring system'. In: *Educational Technology & Society* 25 (Mar. 2022), pp. 1–14. URL: https://eric.ed.gov/?id=EJ1366535.

[45] G. Sedrakyan, J. Malmberg, K. Verbert, S. Järvelä and P. A. Kirschner. 'Linking learning behavior analytics and learning science concepts: Designing a learning analytics dashboard for feedback to support learning regulation'. en. In: *Computers in Human Behavior* 107 (June 2020), p. 105512. DOI: 10.1016/j.chb.2018.05.004.

[46] D. Clow. 'The learning analytics cycle: Closing the loop effectively'. In: *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge*. LAK '12. New York, NY, USA: Association for Computing Machinery, Apr. 2012, pp. 134–138. DOI: 10.1145/2330601.2330636.

[47] J. Wong, M. Khalil, M. Baars, B. B. de Koning and F. Paas. 'Exploring sequences of learner activities in relation to self-regulated learning in a massive open online course'. In: *Computers & Education* 140 (Oct. 2019), p. 103595. DOI: 10.1016/j.compedu.2019.103595.

[48] J. Saint, D. Gašević, W. Matcha, N. A. Uzir and A. Pardo. 'Combining analytic methods to unlock sequential and temporal patterns of self-regulated learning'. In: *Proceedings of the Tenth International Conference on Learning Analytics & Knowledge*. LAK '20. New York, NY, USA: Association for Computing Machinery, Mar. 2020, pp. 402–411. DOI: 10.1145/3375462.3375487.

[49] I. Molenaar and A. F. Wise. 'Temporal aspects of learning analytics - grounding analyses in concepts of time'. In: *The handbook of learning analytics*. Ed. by C. Lang, G. Siemens, A. F. Wise, D. Gašević and A. Merceron. 2nd ed. Vancouver, Canada: SoLAR, 2022, pp. 66–76. URL: https://www.solaresearch.org/publications/hla-22/hla22-chapter7/.

[50] K. D. Wang, J. M. Cock, T. Käser and E. Bumbacher. 'A systematic review of empirical studies using log data from open-ended learning environments to measure science and engineering practices'. en. In: *British Journal of Educational Technology* 54.1 (2023), pp. 192–221. DOI: 10.1111/bjet.13289.

[51] S. Knight, A. Wise, B. Chen and B. Cheng. 'It's about time: 4th international workshop on temporal analyses of learning data'. In: vol. 16-20-March-2015. 2015, pp. 388–389. DOI: 10.1145/2723576.2723638.

[52] B. A. Motz, Y. Bergner, C. A. Brooks, A. Gladden, G. Gray, C. Lang, W. Li, F. Marmolejo-Ramos and J. D. Quick. 'A LAK of Direction: Misalignment Between the Goals of Learning Analytics and its Research Scholarship'. In: *Journal of Learning Analytics* (Mar. 2023), pp. 1–13. DOI: 10.18608/jla.2023.7913.

[53] A. Abbott. 'Sequence Analysis: New Methods for Old Ideas'. In: *Annual Review of Sociology* 21 (1995), pp. 93–113.

[54] R. Venant, K. Sharma, P. Vidal, P. Dillenbourg and J. Broisin. 'Using sequential pattern mining to explore learners' behaviors and evaluate their correlation with performance in inquiry-based learning'. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 10474 LNCS (2017), pp. 286–299. DOI: 10.1007/978-3-319-66610-5_21.

[55] R. S. Baker, A. Mitrović and M. Mathews. 'Detecting gaming the system in constraint-based tutors'. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 6075 LNCS (2010), pp. 267–278. DOI: 10.1007/978-3-642-13470-8_25.

[56] T. J. Nokes, C. D. Schunn and M. T. H. Chi. 'Problem Solving and Human Expertise'. In: *International Encyclopedia of Education (Third Edition)*. Ed. by P. Peterson, E. Baker and B. McGaw. Oxford: Elsevier, Jan. 2010, pp. 265–272. DOI: 10.1016/B978-0-08-044894-7.00486-3.

[57] P. Kirschner and J. J. G. van Merrienboer. 'Ten Steps to Complex Learning: A New Approach to Instruction and Instructional Design'. en. In: *21st Century Education: A Reference Handbook*. 2455 Teller Road, Thousand Oaks California 91320 United States: SAGE Publications, Inc., 2008, pp. I–244–I–253. DOI: 10.4135/9781412964012.n26.

[58] N. Hurtig, J. Hollingsworth, S. Blankenship, E. Kraemer, M. Sitaraman and J. O. Hallstrom. 'Network Visualization and Assessment of Student Reasoning About Conditionals'. In: *Proceedings of the 27th ACM Conference on on Innovation and Technology in Computer Science Education Vol. 1*. Dublin Ireland: ACM, July 2022, pp. 255–261. DOI: 10.1145/3502718.3524793.

[59] W. Matcha, D. Gašević, N. Ahmad Uzir, J. Jovanović, A. Pardo, J. Maldonado-Mahauad and M. Pérez-Sanagustín. 'Detection of Learning Strategies: A Comparison of Process, Sequence and Network Analytic Approaches'. en. In: *Transforming Learning with Meaningful Technologies*. Ed. by M. Scheffel, J. Broisin, V. Pammer-Schindler, A. Ioannou and J. Schneider. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2019, pp. 525–540. DOI: 10.1007/978-3-030-29736-7_39.

[60] Y. Tang, Z. Li, G. Wang and X. Hu. 'Modeling learning behaviors and predicting performance in an intelligent tutoring system: A two-layer hidden Markov modeling approach'. In: *Interactive Learning Environments* 0.0 (Dec. 2021), pp. 1–13. DOI: 10.1080/10494820.2021.2010100.

[61] M. J. Gomez, J. A. Ruipérez-Valiente, P. A. Martínez and Y. J. Kim. 'Applying Learning Analytics to Detect Sequences of Actions and Common Errors in a Geometry Game'. en. In: *Sensors* 21.4 (Jan. 2021), p. 1025. DOI: 10.3390/s21041025.

[62] M. Taub and R. Azevedo. 'How Does Prior Knowledge Influence Eye Fixations and Sequences of Cognitive and Metacognitive SRL Processes during Learning with an Intelligent Tutoring System?' In: *International Journal of Artificial Intelligence in Education* 29.1 (Mar. 2019), pp. 1–28. DOI: 10.1007/s40593-018-0165-4.

[63] J.-P. Corbeil, M. Gagnon and P. R. Richard. 'Probabilistic Approaches to Detect Blocking States in Intelligent Tutoring System'. In: *Intelligent Tutoring Systems: 16th International Conference, ITS 2020, Athens, Greece, June 8–12, 2020, Proceedings*. Berlin, Heidelberg: Springer-Verlag, June 2020, pp. 79–88. DOI: 10.1007/978-3-030-49663-0_11.

[64] X. Tang, Z. Wang, J. Liu and Z. Ying. 'An exploratory analysis of the latent structure of process data via action sequence autoencoders'. In: *British Journal of Mathematical and Statistical Psychology* 74.1 (2021), pp. 1–33. DOI: 10.1111/bmsp.12203.

[65] D. Riofrío-Luzcando and J. Ramírez. ‘Student Action Prediction for Automatic Tutoring for Procedural Training in 3D Virtual Environments’. In: *E-Learning, E-Education, and Online Training*. Ed. by G. Vincenti, A. Bucciero and C. Vaz de Carvalho. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering. Cham: Springer International Publishing, 2016, pp. 200–207. DOI: 10.1007/978-3-319-28883-3_25.

[66] W. Xiao, P. Ji and J. Hu. ‘A survey on educational data mining methods used for predicting students’ performance’. In: *Engineering Reports* 4.5 (2022), e12482. DOI: 10.1002/eng2.12482.

[67] A. Bogarín, R. Cerezo and C. Romero. ‘A survey on educational process mining’. In: *WIREs Data Mining and Knowledge Discovery* 8.1 (2018), e1230. DOI: 10.1002/widm.1230.

[68] V. K. Nadimpalli, F. Hauser, D. Bittner, L. Grabinger, S. Staufer and J. Mottok. ‘Systematic Literature Review for the Use of AI Based Techniques in Adaptive Learning Management Systems’. In: *Proceedings of the 5th European Conference on Software Engineering Education*. Seeon/Bavaria Germany: ACM, June 2023, pp. 83–92. DOI: 10.1145/3593663.3593681.

[69] M. J. Page, J. E. McKenzie, P. M. Bossuyt, I. Boutron, T. C. Hoffmann, C. D. Mulrow, L. Shamseer, J. M. Tetzlaff, E. A. Akl, S. E. Brennan, R. Chou, J. Glanville, J. M. Grimshaw, A. Hróbjartsson, M. M. Lalu, T. Li, E. W. Loder, E. Mayo-Wilson, S. McDonald, L. A. McGuinness, L. A. Stewart, J. Thomas, A. C. Tricco, V. A. Welch, P. Whiting and D. Moher. ‘The PRISMA 2020 statement: An updated guideline for reporting systematic reviews’. In: *Systematic Reviews* 10.1 (Mar. 2021), p. 89. DOI: 10.1186/s13643-021-01626-4.

[70] M. Sandelowski and J. Barroso. *Handbook for synthesizing qualitative research*. New York, NY: Springer Publishing Company, 2007.

[71] P. H. Winne. ‘Improving Measurements of Self-Regulated Learning’. In: *Educational Psychologist* 45.4 (Oct. 2010), pp. 267–276. DOI: 10.1080/00461520.2010.517150.

[72] J. Nesbit, M. Zhou, Y. Xu and P. Winne. ‘Advancing log analysis of student interactions with cognitive tools’. In: *2th biennial conference of the european association for research on learning and insruction (EARLI)*. 2007, pp. 2–20.

[73] J. Kinnebrew and G. Biswas. ‘Modeling and measuring self-regulated learning in teachable agent environments’. In: *Journal of E-Learning and Knowledge Society* 7.2 (2011), pp. 19–35.

[74] X. Xiong, S. Zhao, E. Van Inwegen and J. Beck. ‘Going deeper with deep knowledge tracing’. In: Proceedings of the 9th International Conference on Educational Data Mining, EDM 2016. 2016, pp. 545–550.

[75] B. Rienties, S. Cross and Z. Zdrahal. 'Implementing a Learning Analytics Intervention and Evaluation Framework: What Works?' In: *Big Data and Learning Analytics in Higher Education: Current Theory and Practice*. Ed. by B. Kei Daniel. Cham: Springer International Publishing, 2017, pp. 147–166. DOI: 10.1007/978-3-319-06520-5_10.

[76] A. F. Wise and D. W. Shaffer. 'Why Theory Matters More than Ever in the Age of Big Data'. In: *Journal of Learning Analytics* 2.2 (Dec. 2015), pp. 5–13. DOI: 10.18608/jla.2015.22.2.

[77] M. Khalil, P. Prinsloo and S. Slade. 'The use and application of learning theory in learning analytics: A scoping review'. In: *Journal of Computing in Higher Education* (Oct. 2022). DOI: 10.1007/s12528-022-09340-3.

[78] M. Valle Torre, M. Specht and C. Oertel. *Coded Dataset of a Literature Review on Learning Sequence Analysis*. en. May 2023. DOI: 10.4121/10FC3A10-37AA-4292-818D-468B940B2BB7.V1.

[79] J. González-Brenes, W. Duan and J. Mostow. 'How to classify tutorial dialogue? Comparing feature vectors vs. Sequences'. In: *EDM 2011 - Proceedings of the 4th International Conference on Educational Data Mining*. 2011, pp. 169–178. URL: https://dblp.org/rec/conf/edm/Gonzalez-BrenesMD11.

[80] M. Myers. 'Automatic detection of a student's affective states for intelligent teaching systems'. In: *Brain Sciences* 11.3 (2021), pp. 1–15. DOI: 10.3390/brainsci11030331.

[81] L. Juhaňák, J. Zounek and L. Rohlíková. 'Using process mining to analyze students' quiz-taking behavior patterns in a learning management system'. In: *Computers in Human Behavior* 92 (Mar. 2019), pp. 496–506. DOI: 10.1016/j.chb.2017.12.015.

[82] S. Qu, K. Li, B. Wu, S. Zhang and Y. Wang. 'Predicting student achievement based on temporal learning behavior in MOOCs'. In: *Applied Sciences (Switzerland)* 9.24 (2019). DOI: 10.3390/app9245539.

[83] J. R. Anderson. 'Problem solving and learning'. In: *American Psychologist* 48 (1993), pp. 35–44. DOI: 10.1037/0003-066X.48.1.35.

[84] M. Kesselbacher and A. Bollin. 'Discriminating programming strategies in scratch: Making the difference between novice and experienced programmers'. In: *Proceedings of the 14th workshop in primary and secondary computing education*. 2019. DOI: 10.1145/3361721.3361727.

[85] A. Smith, W. Min, B. W. Mott and J. C. Lester. 'Diagrammatic student models: Modeling student drawing performance with deep learning'. In: *User modeling, adaptation and personalization*. Ed. by F. Ricci, K. Bontcheva, O. Conlan and S. Lawless. Cham: Springer International Publishing, 2015, pp. 216–227. DOI: 10.1007/978-3-319-20267-9_18.

[86] D. Riofrío-Luzcando, J. Ramírez and M. Berrocal-Lobo. 'Predicting student actions in a procedural training environment'. In: *IEEE Transactions on Learning Technologies* 10.4 (2017), pp. 463–474. DOI: 10.1109/TLT.2017.2658569.

[87] F. Bouchet, R. Azevedo, J. S. Kinnebrew and G. Biswas. 'Identifying Students' Characteristic Learning Behaviors in an Intelligent Tutoring System Fostering Self-Regulated Learning'. In: *International Educational Data Mining Society*. Chania, Greece: International Educational Data Mining Society, June 2012.

[88] Y. Omae, T. Furuya, K. Mizukoshi, T. Oshima, N. Sakakibara, Y. Mizuochi, K. Yatsushiro and H. Takahashi. 'Data mining for discovering effective time-series transition of learning strategies on mutual viewing-based learning'. In: *Journal of Advanced Computational Intelligence and Intelligent Informatics* 22.7 (2018), pp. 1046–1055. DOI: 10.20965/jaciii.2018.p1046.

[89] T. Wu and Q. Ling. 'Fusing hybrid attentive network with self-supervised dual-channel heterogeneous graph for knowledge tracing'. In: *Expert Systems with Applications* 225 (Sept. 2023), p. 120212. DOI: 10.1016/j.eswa.2023.120212.

[90] Q. Ni, T. Wei, J. Zhao, L. He and C. Zheng. 'HHSKT: A Learner–Question interactions based heterogeneous graph neural network model for knowledge tracing'. In: *Expert Systems with Applications* 215 (Apr. 2023), p. 119334. DOI: 10.1016/j.eswa.2022.119334.

[91] E. Lundgren. 'Latent program modeling: Inferring latent problem-solving strategies from a PISA problem- solving task'. In: *Journal of Educational Data Mining* 14.1 (June 2022), pp. 46–80. DOI: 10.5281/zenodo.6686443.

[92] F. Morshed Fahid, X. Tian, A. Emerson, J. B. Wiggins, D. Bounajim, A. Smith, E. Wiebe, B. Mott, K. Elizabeth Boyer and J. Lester. 'Progression Trajectory-Based Student Modeling for Novice Block-Based Programming'. In: *Proceedings of the 29th ACM Conference on User Modeling, Adaptation and Personalization*. New York, NY, USA: Association for Computing Machinery, June 2021, pp. 189–200. URL: http://doi.org/10.1145/3450613.3456833.

[93] S. Lajoie, E. Poitras, T. Doleck and A. Jarrell. 'Modeling metacognitive activities in medical problem-solving with bio world'. In: *Intelligent Systems Reference Library* 76 (2015), pp. 323–343. DOI: 10.1007/978-3-319-11062-2_13.

[94] J.-Y. Sun, S.-J. Yu and C.-H. Chao. 'Effects of intelligent feedback on online learners' engagement and cognitive load: The case of research ethics education'. In: *Educational Psychology* 39.10 (2019), pp. 1293–1310. DOI: 10.1080/01443410.2018.1527291.

[95] J. Kinnebrew, D. Mack, G. Biswas and C.-K. Chang. 'A differential approach for identifying important student learning behavior patterns with evolving usage over time'. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 8643 (2014), pp. 281–292. DOI: 10.1007/978-3-319-13186-3_27.

[96] R. Rajendran, A. Munshi, M. Emara and G. Biswas. 'A temporal model of learner behaviors in OELEs using process mining'. In: *ICCE 2018 - 26th International Conference on Computers in Education, Main Conference Proceedings*. 2018, pp. 276–285. URL: https://apsce.net/icce/icce2018/wp-content/uploads/2018/12/C3-07.pdf.

[97] P. Rhienmora, P. Haddawy, P. Khanal, S. Suebnukarn and M. N. Dailey. 'A virtual reality simulator for teaching and evaluating dental procedures'. In: *Methods of Information in Medicine* 49.4 (2010), pp. 396–405. DOI: 10.3414/ME9310.

[98] J. Herold, A. Zundel and T. Stahovich. 'Mining meaningful patterns from students' handwritten coursework'. In: Proceedings of the 6th International Conference on Educational Data Mining, EDM 2013. 2013.

[99] Y.-W. Chu, E. Tenorio, L. Cruz, K. Douglas, A. S. Lan and C. G. Brinton. 'Click-Based Student Performance Prediction: A Clustering Guided Meta-Learning Approach'. In: *2021 IEEE International Conference on Big Data (Big Data)*. Orlando, FL, USA, Dec. 2021, pp. 1389–1398. DOI: 10.1109/BigData52589.2021.9671729.

[100] J.-J. Vie and H. Kashima. 'Knowledge Tracing Machines: Factorization Machines for Knowledge Tracing'. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 33.01 (July 2019), pp. 750–757. DOI: 10.1609/aaai.v33i01.3301750.

[101] H.-M. Chen, B.-A. Nguyen, C.-R. Dow, N.-L. Hsueh and A.-C. Liu. 'Exploring Time-Related Micro-Behavioral Patterns in a Python Programming Online Course.' In: *Journal of Information Science & Engineering* 38.6 (2022).

[102] E. Owen, G. Anton and R. Baker. 'Modeling user exploration and boundary testing in digital learning games'. In: *UMAP 2016 - Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization*. 2016, pp. 301–302. DOI: 10.1145/2930238.2930271.

[103] D. A. Dever, N. A. Sonnenfeld, M. D. Wiedbusch, S. G. Schmorrow, M. J. Amon and R. Azevedo. 'A complex systems approach to analyzing pedagogical agents' scaffolding of self-regulated learning within an intelligent tutoring system'. en. In: *Metacognition and Learning* (May 2023). DOI: 10.1007/s11409-023-09346-x.

[104] Y. Guo, T. He, A. Li, Z. Li, Y. Rong and G. Liu. 'Deep Knowledge Tracking Method Based on DKVTMN-DTCN Model'. In: *Computer Science and Education*. Ed. by W. Hong and Y. Weng. Communications in Computer and Information Science. Singapore: Springer Nature, 2023, pp. 623–636. DOI: 10.1007/978-981-99-2446-2_57.

[105] A. Neyem, J. Diaz-Mosquera, J. Munoz-Gama and J. Navon. 'Understanding Student Interactions in Capstone Courses to Improve Learning Experiences'. en. In: *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*. Seattle Washington USA: ACM, Mar. 2017, pp. 423–428. DOI: 10.1145/3017680.3017716.

[106] N. Hurtig, J. Hollingsworth and O. Scrivner. 'Visualization of Students' Solutions as a Sequential Network'. In: *2022 IEEE Global Engineering Education Conference (EDUCON)*. Mar. 2022, pp. 1189–1194. DOI: 10.1109/EDUCON52537.2022.9766502.

[107] E. Y. Ha, J. P. Rowe, B. W. Mott and J. C. Lester. 'Goal recognition with Markov logic networks for player-adaptive games'. In: *Proceedings of the Seventh AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*. AIIDE'11. Stanford, California, USA: AAAI Press, Oct. 2011, pp. 32–39.

[108] S. Meliana and D. Nurjanah. 'Adopting Good-Learners' Paths in an Intelligent Tutoring System'. In: *2018 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*. Dec. 2018, pp. 877–882. DOI: 10.1109/TALE.2018.8615176.

[109] A. Mathrani, T. Susnjak, G. Ramaswami and A. Barczak. 'Perspectives on the challenges of generalizability, transparency and ethics in predictive learning analytics'. In: *Computers and Education Open* 2 (Dec. 2021), p. 100060. DOI: 10.1016/j.caeo.2021.100060.

[110] M. Geden, A. Emerson, J. Rowe, R. Azevedo and J. Lester. 'Predictive Student Modeling in Educational Games with Multi-Task Learning'. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 34.01 (Apr. 2020), pp. 654–661. DOI: 10.1609/aaai.v34i01.5406.

[111] G. Zhou, T. Umada and S. D'Mello. 'What do Students' Interactions with Online Lecture Videos Reveal about their Learning?' In: *Proceedings of the 30th ACM Conference on User Modeling, Adaptation and Personalization*. UMAP '22. New York, NY, USA: Association for Computing Machinery, July 2022, pp. 295–305. DOI: 10.1145/3503252.3531315.

[112] S. Banno, H. Murad and M. Sallal. 'Automated Cognitive Analyses for Intelligent Tutoring Systems'. In: *2020 IEEE/ACM International Conference on Big Data Computing, Applications and Technologies (BDCAT)*. Leicester, UK, Dec. 2020, pp. 171–178. DOI: 10.1109/BDCAT50828.2020.00007.

[113] T. Tuti, C. Paton, M. English and N. Winters. 'Building a Learner Model for a Smartphone-Based Clinical Training Intervention in a Low-Income Context: A Pilot Study'. In: *Transforming Learning with Meaningful Technologies*. Ed. by M. Scheffel, J. Broisin, V. Pammer-Schindler, A. Ioannou and J. Schneider. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2019, pp. 55–68. DOI: 10.1007/978-3-030-29736-7_5.

[114] Y. Gong, J. E. Beck and N. T. Heffernan. 'Comparing Knowledge Tracing and Performance Factor Analysis by Using Multiple Model Fitting Procedures'. In: *Intelligent Tutoring Systems*. Ed. by V. Aleven, J. Kay and J. Mostow. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2010, pp. 35–44. DOI: 10.1007/978-3-642-13388-6_8.

[115] X. Zhang, Y. Shang, Y. Ren and K. Liang. 'Dynamic multi-objective sequence-wise recommendation framework via deep reinforcement learning'. en. In: *Complex & Intelligent Systems* 9.2 (Apr. 2023), pp. 1891–1911. DOI: 10.1007/s40747-022-00871-x.

[116] Z. Liu, Q. Liu, J. Chen, S. Huang, B. Gao, W. Luo and J. Weng. 'Enhancing Deep Knowledge Tracing with Auxiliary Tasks'. In: *Proceedings of the ACM Web Conference 2023*. WWW '23. New York, NY, USA: Association for Computing Machinery, Apr. 2023, pp. 4178–4187. DOI: 10.1145/3543507.3583866.

[117] J. Mandlazi, A. Jadhav and R. Ajoodha. 'Evaluating Deep Sequential Knowledge Tracing Models for Predicting Student Performance'. In: *2021 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE)*. Dec. 2021, pp. 1–6. DOI: 10.1109/CSDE53843.2021.9718405.

[118] Y. Lu, D. Wang, P. Chen, Q. Meng and S. Yu. 'Interpreting Deep Learning Models for Knowledge Tracing'. In: *International Journal of Artificial Intelligence in Education* 33.3 (Sept. 2023), pp. 519–542. DOI: 10.1007/s40593-022-00297-z.

[119] Y. He, X. Hu, Z. Xu and a. G. Sun. 'KT-XL: A Knowledge Tracing Model for Predicting Learning Performance Based on Transformer-XL'. In: *Proceedings of the ACM Turing Celebration Conference - China*. ACM TURC '20. New York, NY, USA: Association for Computing Machinery, Oct. 2020, pp. 175–179. DOI: 10.1145/3393527.3393557.

[120] X. Jiang, Y. Ouyang, Z. Liu, W. Rong and Z. Xiong. 'MAKT: Multichannel Attention Networks based Knowledge Tracing with Representation Learning'. In: *2022 IEEE International Conference on Teaching, Assessment and Learning for Engineering (TALE)*. Dec. 2022, pp. 292–299. DOI: 10.1109/TALE54877.2022.00055.

[121] W. Gan, Y. Sun, X. Peng and Y. Sun. 'Modeling learner's dynamic knowledge construction procedure and cognitive item difficulty for knowledge tracing'. In: *Applied Intelligence* 50.11 (2020), pp. 3894–3912. DOI: 10.1007/s10489-020-01756-7.

[122] Z. Xia, N. Dong, J. Wu and C. Ma. 'Multi-Variate Knowledge Tracking Based on Graph Neural Network in ASSISTments'. In: *IEEE Transactions on Learning Technologies* (2023), pp. 1–12. DOI: 10.1109/TLT.2023.3301011.

[123] M. Eagle, A. Corbett, J. Stamper and B. Mclaren. 'Predicting Individualized Learner Models Across Tutor Lessons'. In: (2018).

[124] S. Sahebi, Y. Huang and P. Brusilovsky. 'Predicting student performance in solving parameterized exercises'. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 8474 LNCS (2014), pp. 496–503. DOI: 10.1007/978-3-319-07221-0_62.

[125] C. Schatten and L. Schmidt-Thieme. 'Student Progress Modeling with Skills Deficiency Aware Kalman Filters:' in: *Proceedings of the 8th International Conference on Computer Supported Education*. Rome, Italy: SCITEPRESS - Science, 2016, pp. 31–42. DOI: 10.5220/0005737200310042.

[126] S. Sahebi, Y.-R. Lin and P. Brusilovsky. 'Tensor factorization for student modeling and performance prediction in unstructured domain'. In: *Proceedings of the 9th International Conference on Educational Data Mining, EDM 2016*. 2016, pp. 502–506.

[127] J. Geng, H. Yang and S. Hu. 'Noise-Filtering Enhanced Deep Cognitive Diagnosis Model for Latent Skill Discovering'. In: *Intelligent Automation & Soft Computing* 37.2 (2023), pp. 1311–1324. DOI: 10.32604/iasc.2023.038481.

[128] Z. Yang, S. Xu, C. Wang and G. He. 'Skill-Oriented Hierarchical Structure for Deep Knowledge Tracing'. In: *2022 IEEE 34th International Conference on Tools with Artificial Intelligence (ICTAI)*. Oct. 2022, pp. 425–432. DOI: 10.1109/ICTAI56018.2022.00069.

[129] J. Cook, C. F. Lynch, A. G. Hicks and B. Mostafavi. 'Task and Timing: Separating Procedural and Tactical Knowledge in Student Models'. In: *International Educational Data Mining Society*. International Educational Data Mining Society, June 2017.

[130] B. Chen, M. West and C. Zilles. 'Towards a Model-Free Estimate of the Limits to Student Modeling Accuracy'. en. In: vol. 11. Raleigh, NC, USA, July 2018.

[131] R. Schmucker and T. M. Mitchell. 'Transferable Student Performance Modeling for Intelligent Tutoring Systems'. In: *Proceedings of the 30th International Conference on Computers in Education. Asia-Pacific Society for Computers in Education*. Kuala Lumpur, Malaysia: The Asia-Pacific Society for Computers in Education, 2022.

[132] K. Akhuseyinoglu and P. Brusilovsky. 'Data-Driven Modeling of Learners' Individual Differences for Predicting Engagement and Success in Online Learning'. In: *Proceedings of the 29th ACM Conference on User Modeling, Adaptation and Personalization*. UMAP '21. New York, NY, USA: Association for Computing Machinery, June 2021, pp. 201–212. DOI: 10.1145/3450613.3456834.

[133] Y. Wang, T. Li, C. Geng and Y. Wang. 'Evaluating Student Learning Effect Based on Process Mining'. en. In: *Applied Informatics*. Ed. by H. Florez, M. Leon, J. M. Diaz-Nafria and S. Belli. Communications in Computer and Information Science. Cham: Springer International Publishing, 2019, pp. 59–72. DOI: 10.1007/978-3-030-32475-9_5.

[134] M. J. Gomez, J. A. Ruipérez-Valiente, P. A. Martinez and Y. J. Kim. 'Exploring the Affordances of Sequence Mining in Educational Games'. en. In: *Eighth International Conference on Technological Ecosystems for Enhancing Multiculturality*. Salamanca Spain: ACM, Oct. 2020, pp. 648–654. DOI: 10.1145/3434780.3436562.

[135] Y.-C. Norm Lien, W.-J. Wu and Y.-L. Lu. 'How well do teachers predict students' actions in solving an ill-defined problem in STEM education: A solution using sequential pattern mining'. In: *IEEE access : practical innovations, open solutions* 8 (2020), pp. 134976–134986. DOI: 10.1109/ACCESS.2020.3010168.

[136] Lin, J., Singh, S., Sha, L., Tan, W., Lang, D., Gasevic, D. and Chen, G. 'Is it a good move? Mining effective tutoring strategies from human-human tutorial dialogues'. null. In: *Future Generation Computer Systems* 127 (2022), pp. 194–207. DOI: 10.1016/j.future.2021.09.001.

[137] V. Latypova. 'Work with Free Response Implementation Process Analysis Based on Sequential Pattern Mining in Engineering Education'. In: *2022 VI International Conference on Information Technologies in Engineering Education (Inforino)*. Apr. 2022, pp. 1–4. DOI: 10.1109/Inforino53888.2022.9782969.

[138] K. Akhuseyinoglu and P. Brusilovsky. 'Exploring Behavioral Patterns for Data-Driven Modeling of Learners' Individual Differences'. In: *Frontiers in Artificial Intelligence* 5 (Feb. 2022), p. 807320. DOI: 10.3389/frai.2022.807320.

[139] H. Duong, L. Zhu, Y. Wang and N. Heffernan. 'A prediction model uses the sequence of attempts and hints to better predict knowledge: Better to attempt the problem first, rather than ask for a hint'. In: Proceedings of the 6th International Conference on Educational Data Mining, EDM 2013. 2013.

[140] A. Supianto and M. Hafis. 'GTRAS: Graphical tracking activity system for problem-posing learning process insights'. In: *2018 International Conference on Advanced Computer Science and Information Systems, ICACSIS 2018*. IEEE, 2019, pp. 231–235. DOI: 10.1109/ICACSIS.2018.8618216.

[141] M.-C. Liu, C.-H. Yu, J. Wu, A.-C. Liu and H.-M. Chen. 'Applying learning analytics to deconstruct user engagement by using log data of MOOCs'. In: *Journal of Information Science and Engineering* 34.5 (2018), pp. 1175–1186. DOI: 10.6688/JISE.201809_34(5).0004.

[142] S. Bryfczynski, R. P. Pargas, M. M. Cooper, M. Klymkowsky and B. C. Dean. 'Teaching data structures with beSocratic'. In: *Proceedings of the 18th ACM conference on Innovation and technology in computer science education*. ITiCSE '13. New York, NY, USA: Association for Computing Machinery, July 2013, pp. 105–110. DOI: 10.1145/2462476.2465583.

[143] X. Song, J. Li, T. Cai, S. Yang, T. Yang and C. Liu. 'A survey on deep learning based knowledge tracing'. In: *Knowledge-Based Systems* 258 (Dec. 2022), p. 110036. DOI: 10.1016/j.knosys.2022.110036.

[144] G. Padrón-Rivera and G. Rebolledo-Mendez. 'Identifying Affective Trajectories in Relation to Learning Gains During the Interaction with a Tutoring System'. In: *Artificial Intelligence in Education*. Ed. by C. Conati, N. Heffernan, A. Mitrovic and M. F. Verdejo. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2015, pp. 756–759. DOI: 10.1007/978-3-319-19773-9_109.

[145] O. Hazzan and K. Mike. *Guide to Teaching Data Science: An Interdisciplinary Approach*. en. Springer Nature, Mar. 2023.

[146] T. Donoghue, B. Voytek and S. E. Ellis. 'Teaching Creative and Practical Data Science at Scale'. In: *Journal of Statistics and Data Science Education* 29.sup1 (Jan. 2021), S27–S39. DOI: 10.1080/10691898.2020.1860725.

[147] A. Schwab-McCoy, C. M. Baker and R. E. Gasper. 'Data Science in 2020: Computing, Curricula, and Challenges for the Next 10 Years'. In: *Journal of Statistics and Data Science Education* 29.sup1 (Jan. 2021), S40–S50. DOI: 10.1080/10691898.2020.1851159.

[148] D. Ramasamy, C. Sarasua, A. Bacchelli and A. Bernstein. 'Workflow analysis of data science code in public GitHub repositories'. en. In: *Empirical Software Engineering* 28.1 (Nov. 2022), p. 7. DOI: 10.1007/s10664-022-10229-z.

[149] Y. Fan, L. Tang, H. Le, K. Shen, S. Tan, Y. Zhao, Y. Shen, X. Li and D. Gašević. 'Beware of metacognitive laziness: Effects of generative artificial intelligence on learning motivation, processes, and performance'. en. In: *British Journal of Educational Technology* n/a.n/a (Dec. 2024). DOI: 10.1111/bjet.13544.

[150] J. Zhao, A. Gal and S. Krishnan. 'Data Makes Better Data Scientists'. en. In: *Proceedings of the Workshop on Human-In-the-Loop Data Analytics*. Seattle WA USA: ACM, June 2023, pp. 1–3. DOI: 10.1145/3597465.3605228.

[151] S. López-Pernas, M. Saqr and O. Viberg. 'Putting It All Together: Combining Learning Analytics Methods and Data Sources to Understand Students' Approaches to Learning Programming'. en. In: *Sustainability* 13.9 (Jan. 2021), p. 4825. DOI: 10.3390/su13094825.

[152] M. A. Posner, A. Kerby-Helm, A. Unfried, D. Whitaker, M. E. Bond and L. Batakci. 'A Cross-disciplinary Review of Introductory Undergraduate Data Science Course Content'. In: *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 2*. SIGCSE 2024. New York, NY, USA: Association for Computing Machinery, Mar. 2024, p. 1937. DOI: 10.1145/3626253.3635352.

[153] F. Martínez-Plumed, L. Contreras-Ochando, C. Ferri, J. Hernández-Orallo, M. Kull, N. Lachiche, M. J. Ramírez-Quintana and P. Flach. 'CRISP-DM Twenty Years Later: From Data Mining Processes to Data Science Trajectories'. In: *IEEE Transactions on Knowledge and Data Engineering* 33.8 (Aug. 2021), pp. 3048–3061. DOI: 10.1109/TKDE.2019.2962680.

[154] A. R. Ellis and E. Slade. 'A New Era of Learning: Considerations for ChatGPT as a Tool to Enhance Statistics and Data Science Education'. In: *Journal of Statistics and Data Science Education* 31.2 (May 2023), pp. 128–133. DOI: 10.1080/26939169.2023.2223609.

[155] M. Saqr. *Temporal network analysis: Introduction, methods and detailed tutorial with R*. July 2023. DOI: 10.48550/arXiv.2307.12339.

[156] J. F. Pimentel, L. Murta, V. Braganholo and J. Freire. 'A Large-Scale Study About Quality and Reproducibility of Jupyter Notebooks'. en. In: *2019 IEEE/ACM 16th International Conference on Mining Software Repositories (MSR)*. Montreal, QC, Canada: IEEE, May 2019, pp. 507–517. DOI: 10.1109/MSR.2019.00077.

[157] L. Quaranta, F. Calefato and F. Lanubile. 'KGTorrent: A Dataset of Python Jupyter Notebooks from Kaggle'. In: *2021 IEEE/ACM 18th International Conference on Mining Software Repositories (MSR)*. May 2021, pp. 550–554. DOI: 10.1109/MSR52588.2021.00072.

[158] A. Singh, A. Fariha, C. Brooks, G. Soares, A. Z. Henley, A. Tiwari, C. M, H. Choi and S. Gulwani. 'Investigating Student Mistakes in Introductory Data Science Programming'. In: *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1*. SIGCSE 2024. New York, NY, USA: Association for Computing Machinery, Mar. 2024, pp. 1258–1264. DOI: 10.1145/3626252.3630884.

[159] H. Nguyen, M. Lim, S. Moore, E. Nyberg, M. Sakr and J. Stamper. 'Exploring Metrics for the Analysis of Code Submissions in an Introductory Data Science Course'. In: *LAK21: 11th International Learning Analytics and Knowledge Conference*. LAK21. New York, NY, USA: Association for Computing Machinery, Apr. 2021, pp. 632–638. DOI: 10.1145/3448139.3448209.

[160] A. Y. Wang, D. Wang, J. Drozdal, X. Liu, S. Park, S. Oney and C. Brooks. 'What Makes a Well-Documented Notebook? A Case Study of Data Scientists' Documentation Practices in Kaggle'. In: *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*. CHI EA '21. New York, NY, USA: Association for Computing Machinery, May 2021, pp. 1–7. DOI: 10.1145/3411763.3451617.

[161] S. Knight, A. Friend Wise and B. Chen. 'Time for Change: Why Learning Analytics Needs Temporal Analysis'. en. In: *Journal of Learning Analytics* 4.3 (Dec. 2017). DOI: 10.18608/jla.2017.43.2.

[162] K. D. Wang, S. Salehi, M. Arseneault, K. Nair and C. Wieman. 'Automating the Assessment of Problem-solving Practices Using Log Data and Data Mining Techniques'. en. In: *Proceedings of the Eighth ACM Conference on Learning @ Scale*. Virtual Event Germany: ACM, June 2021, pp. 69–76. DOI: 10.1145/3430895.3460127.

[163] K. D. Wang, S. Salehi and C. Wieman. ‘Applying Log Data Analytics to Measure Problem Solving in Simulation-Based Learning Environments’. en. In: *Unobtrusive Observations of Learning in Digital Environments: Examining Behavior, Cognition, Emotion, Metacognition and Social Processes Using Learning Analytics*. Ed. by V. Kovanovic, R. Azevedo, D. C. Gibson and D. Ifenthaler. Advances in Analytics for Learning and Teaching. Cham: Springer International Publishing, 2023, pp. 31–52. DOI: 10.1007/978-3-031-30992-2_3.

[164] J. Samuelsen, W. Chen and B. Wasson. ‘Integrating multiple data sources for learning analytics—review of literature’. en. In: *Research and Practice in Technology Enhanced Learning* 14.1 (Aug. 2019), p. 11. DOI: 10.1186/s41039-019-0105-4.

[165] W. Matcha, D. Gašević, N. A. Uzir, J. Jovanović, A. Pardo, L. Lim, J. Maldonado-Mahauad, S. Gentili, M. Pérez-Sanagustín and Y.-S. Tsai. ‘Analytics of Learning Strategies: Role of Course Design and Delivery Modality’. en. In: *Journal of Learning Analytics* 7.2 (Sept. 2020), pp. 45–71. DOI: 10.18608/jla.2020.72.3.

[166] X. Xu, L. Qiao, N. Cheng, H. Liu and W. Zhao. ‘Enhancing self-regulated learning and learning experience in generative AI environments: The critical role of metacognitive support’. In: *British Journal of Educational Technology* n/a.n/a (May 2025). DOI: 10.1111/bjet.13599.

[167] A. Drozdova, E. Trofimova, P. Guseva, A. Scherbakova and A. Ustyuzhanin. ‘Code4ML: a large-scale dataset of annotated Machine Learning code’. en. In: *PeerJ Computer Science* 9 (Feb. 2023), e1230. DOI: 10.7717/peerj-cs.1230.

[168] S. Chakraborty, M. Wilson, S. Assi, A. Al-Hamid, M. Alamran, A. Al-Nahari, J. Mustafina, J. Lunn and D. Al-Jumeily OBE. ‘Quora Insincere Questions Classification Using Attention Based Model’. en. In: *Data Science and Emerging Technologies*. Ed. by Y. B. Wah, M. W. Berry, A. Mohamed and D. Al-Jumeily. Singapore: Springer Nature, 2023, pp. 357–372. DOI: 10.1007/978-981-99-0741-0_26.

[169] M. Saqr, S. López-Pernas, S. Helske, M. Durand, K. Murphy, M. Studer and G. Ritschard. ‘Sequence Analysis in Education: Principles, Technique, and Tutorial with R’. en. In: *Learning analytics methods and tutorials: A practical guide using R*. Ed. by M. Saqr and S. Lopez-Pernas. In Press. Springer, 2024. URL: https://lamethods.github.io/ch10-seq.html.

[170] M. Studer and G. Ritschard. ‘What Matters in Differences Between Life Trajectories: A Comparative Review of Sequence Dissimilarity Measures’. In: *Journal of the Royal Statistical Society Series A: Statistics in Society* 179.2 (Feb. 2016), pp. 481–511. DOI: 10.1111/rssa.12125.

[171] M. Bannert, P. Reimann and C. Sonnenberg. 'Process mining techniques for analysing patterns and strategies in students' self-regulated learning'. en. In: *Metacognition and Learning* 9.2 (Aug. 2014), pp. 161–185. DOI: 10.1007/s11409-013-9107-6.

[172] J. Saint, Y. Fan, D. Gašević and A. Pardo. 'Temporally-focused analytics of self-regulated learning: A systematic review of literature'. In: *Computers and Education: Artificial Intelligence* 3 (Jan. 2022), p. 100060. DOI: 10.1016/j.caeai.2022.100060.

[173] J. Helske, S. Helske, M. Saqr, S. López-Pernas and K. Murphy. 'A Modern Approach to Transition Analysis and Process Mining with Markov Models in Education'. en. In: *Learning analytics methods and tutorials: A practical guide using R*. Ed. by M. Saqr and S. López-Pernas. In Press. Springer, 2024. URL: https://lamethods.github.io/ch12-markov.html.

[174] C. Yan and Y. He. 'Auto-Suggest: Learning-to-Recommend Data Preparation Steps Using Data Science Notebooks'. In: *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*. SIGMOD '20. New York, NY, USA: Association for Computing Machinery, May 2020, pp. 1539–1554. DOI: 10.1145/3318464.3389738.

[175] M. T. H. Chi, P. J. Feltovich and R. Glaser. 'Categorization and Representation of Physics Problems by Experts and Novices'. en. In: *Cognitive Science* 5.2 (1981), pp. 121–152. DOI: 10.1207/s15516709cog0502_2.

[176] O. Hazzan and K. Mike. *Ten Challenges of Data Science Education*. en. July 2020. URL: https://cacm.acm.org/blogs/blog-cacm/246219-ten-challenges-of-data-science-education/fulltext.

[177] M. Valle Torre, T. van der Velden, M. Specht and C. Oertel. 'JELAI: Integrating AI and Learning Analytics in Jupyter Notebooks'. en. In: *Artificial Intelligence in Education*. Ed. by A. I. Cristea, E. Walker, Y. Lu, O. C. Santos and S. Isotani. Cham: Springer Nature Switzerland, 2025, pp. 68–75. DOI: 10.1007/978-3-031-98465-5_9.

[178] Z. Cai, R. L. Davis, R. Mariétan, R. Tormey and P. Dillenbourg. 'Jupyter Analytics: A Toolkit for Collecting, Analyzing, and Visualizing Distributed Student Activity in Jupyter Notebooks'. In: *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 1*. SIGCSETS 2025. New York, NY, USA: Association for Computing Machinery, Feb. 2025, pp. 172–178. DOI: 10.1145/3641554.3701971.

[179] M. de Laat, S. Joksimovic and D. Ifenthaler. 'Artificial intelligence, real-time feedback and workplace learning analytics to support in situ complex problem-solving: a commentary'. In: *The International Journal of Information and Learning Technology* 37.5 (Jan. 2020), pp. 267–277. DOI: 10.1108/IJILT-03-2020-0026.

[180] Y. Shen, X. Ai, A. G. Soosai Raj, R. J. Leo John and M. Syamkumar. 'Implications of ChatGPT for Data Science Education'. In: *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1*. SIGCSE 2024. New York, NY, USA: Association for Computing Machinery, Mar. 2024, pp. 1230–1236. DOI: 10.1145/3626252.3630874.

[181] M. Giannakos, R. Azevedo, P. Brusilovsky, M. Cukurova, Y. Dimitriadis, D. Hernandez-Leo, S. Järvelä, M. Mavrikis and B. Rienties. 'The promise and challenges of generative AI in education'. In: *Behaviour & Information Technology* 0.0 (2024), pp. 1–27. DOI: 10.1080/0144929X.2024.2394886.

[182] M. Kazemitabaar, X. Hou, A. Henley, B. J. Ericson, D. Weintrop and T. Grossman. 'How Novices Use LLM-based Code Generators to Solve CS1 Coding Tasks in a Self-Paced Learning Environment'. In: *Proceedings of the 23rd Koli Calling International Conference on Computing Education Research*. Koli Calling '23. New York, NY, USA: Association for Computing Machinery, Feb. 2024, pp. 1–12. DOI: 10.1145/3631802.3631806.

[183] S. Pal Chowdhury, V. Zouhar and M. Sachan. 'AutoTutor meets Large Language Models: A Language Model Tutor with Rich Pedagogy and Guardrails'. In: *Proceedings of the Eleventh ACM Conference on Learning @ Scale*. L@S '24. New York, NY, USA: Association for Computing Machinery, July 2024, pp. 5–15. DOI: 10.1145/3657604.3662041.

[184] D. Adams, K.-M. Chuah, E. Devadason and M. S. A. Azzis. 'From novice to navigator: Students' academic help-seeking behaviour, readiness, and perceived usefulness of ChatGPT in learning'. en. In: *Education and Information Technologies* 29.11 (Aug. 2024), pp. 13617–13634. DOI: 10.1007/s10639-023-12427-8.

[185] X. Zhang, P. Zhang, Y. Shen, M. Liu, Q. Wang, D. Gašević and Y. Fan. 'A Systematic Literature Review of Empirical Research on Applying Generative Artificial Intelligence in Education'. en. In: *Frontiers of Digital Education* 1.3 (Sept. 2024), pp. 223–245. DOI: 10.1007/s44366-024-0028-5.

[186] H. Su, Y. Tong, X. Zhang and Y. Fan. 'Uncovering Students' Processing Tactics Towards ChatGPT's Feedback in EFL Education Using Learning Analytics'. en. In: *Blended Learning. Intelligent Computing in Education*. Ed. by W. W. K. Ma, C. Li, C. W. Fan, L. H. U and A. Lu. Singapore: Springer Nature, 2024, pp. 238–250. DOI: 10.1007/978-981-97-4442-8_18.

[187] K. Kitto, J. Whitmer, A. Silvers and M. Webb. *Creating data for learning analytics ecosystems [position paper]*. en. Report. Society for Learning Analytics Research, Sept. 2020. URL: https://opus.lib.uts.edu.au/handle/10453/152209.

[188] J. Liu, S. Li and Q. Dong. 'Collaboration with Generative Artificial Intelligence: An Exploratory Study Based on Learning Analytics'. en. In: *Journal of Educational Computing Research* 62.5 (Sept. 2024), pp. 1234–1266. DOI: 10.1177/07356331241242441.

[189] S.-H. Ko and K. Stephens-Martinez. ‘The Trees in the Forest: Characterizing Computing Students’ Individual Help-Seeking Approaches’. In: *Proceedings of the 2024 ACM Conference on International Computing Education Research - Volume 1*. Vol. 1. ICER ’24. New York, NY, USA: Association for Computing Machinery, Aug. 2024, pp. 343–358. DOI: 10.1145/3632620.3671099.

[190] A. Cardoso, J. Leitão and C. Teixeira. ‘Using the Jupyter Notebook as a Tool to Support the Teaching and Learning Processes in Engineering Courses’. en. In: *The Challenges of the Digital Transformation in Education*. Ed. by M. E. Auer and T. Tsiatsos. Cham: Springer International Publishing, 2019, pp. 227–236. DOI: 10.1007/978-3-030-11935-5_22.

[191] E. Chen, R. Huang, H.-S. Chen, Y.-H. Tseng and L.-Y. Li. ‘GPTutor: A ChatGPT-Powered Programming Tool for Code Explanation’. en. In: *Artificial Intelligence in Education. Posters and Late Breaking Results, Workshops and Tutorials, Industry and Innovation Tracks, Practitioners, Doctoral Consortium and Blue Sky*. Springer, Cham, 2023, pp. 321–327. DOI: 10.1007/978-3-031-36336-8_50.

[192] W. Lyu, Y. Wang, T. (Chung, Y. Sun and Y. Zhang. ‘Evaluating the Effectiveness of LLMs in Introductory Computer Science Education: A Semester-Long Field Study’. In: *Proceedings of the Eleventh ACM Conference on Learning @ Scale*. L@S ’24. New York, NY, USA: Association for Computing Machinery, July 2024, pp. 63–74. DOI: 10.1145/3657604.3662036.

[193] M. Kazemitabaar, R. Ye, X. Wang, A. Z. Henley, P. Denny, M. Craig and T. Grossman. ‘CodeAid: Evaluating a Classroom Deployment of an LLM-based Programming Assistant that Balances Student and Educator Needs’. In: *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. CHI ’24. New York, NY, USA: Association for Computing Machinery, May 2024, pp. 1–20. DOI: 10.1145/3613904.3642773.

[194] B. Sheese, M. Liffiton, J. Savelka and P. Denny. ‘Patterns of Student Help-Seeking When Using a Large Language Model-Powered Programming Assistant’. In: *Proceedings of the 26th Australasian Computing Education Conference*. ACE ’24. New York, NY, USA: Association for Computing Machinery, Jan. 2024, pp. 49–57. DOI: 10.1145/3636243.3636249.

[195] V. Aleven, I. Roll, B. M. McLaren and K. R. Koedinger. ‘Help Helps, But Only So Much: Research on Help Seeking with Intelligent Tutoring Systems’. en. In: *International Journal of Artificial Intelligence in Education* 26.1 (Mar. 2016), pp. 205–223. DOI: 10.1007/s40593-015-0089-1.

[196] S. Nelson-Le Gall and S. Glor-Scheib. ‘Help seeking in elementary classrooms: An observational study’. In: *Contemporary Educational Psychology* 10.1 (Jan. 1985), pp. 58–71. DOI: 10.1016/0361-476X(85)90006-2.

[197] A. Chen, M. Xiang, J. Zhou, J. Jia, J. Shang, X. Li, D. Gašević and Y. Fan. ‘Unpacking help-seeking process through multimodal learning analytics: A comparative study of ChatGPT vs Human expert’. In: *Computers & Education* 226 (Mar. 2025), p. 105198. DOI: 10.1016/j.compedu.2024.105198.

[198] N. Kerimbayev, K. Adamova, R. Shadiev and Z. Altinay. 'Intelligent educational technologies in individual learning: a systematic literature review'. In: *Smart Learning Environments* 12.1 (Jan. 2025), p. 1. DOI: 10.1186/s40561-024-00360-3.

[199] M. Valle Torre, R. Weijers, M. Specht and C. Oertel. *Process Collapse or Compensatory Strategy? Profiling and Scaffolding Help-Seeking with an AI Tutor*. en. SSRN Scholarly Paper. Rochester, NY, Feb. 2026. DOI: 10.2139/ssrn.6297817.

[200] S. Jaffe, N. P. Shah, J. Butler, A. Farach, A. Cambon, B. Hecht, M. Schwarz and J. Teevan. *Generative AI in Real-World Workplaces*. en-US. Tech. rep. Microsoft, July 2024. URL: https://www.microsoft.com/en-us/research/publication/generative-ai-in-real-world-workplaces/.

[201] A. Chatterji, T. Cunningham, D. J. Deming, Z. Hitzig, C. Ong, C. Y. Shan and K. Wadman. *How People Use ChatGPT*. Working Paper. Sept. 2025. DOI: 10.3386/w34255.

[202] R. Wu and Z. Yu. 'Do AI chatbots improve students learning outcomes? Evidence from a meta-analysis'. en. In: *British Journal of Educational Technology* 55.1 (2024), pp. 10–33. DOI: 10.1111/bjet.13334.

[203] S. Wang, T. Xu, H. Li, C. Zhang, J. Liang, J. Tang, P. S. Yu and Q. Wen. *Large Language Models for Education: A Survey and Outlook*. Apr. 2024. DOI: 10.48550/arXiv.2403.18105.

[204] S. Gupta, R. R. Dharamshi and V. Kakde. 'An Impactful and Revolutionized Educational Ecosystem using Generative AI to Assist and Assess the Teaching and Learning benefits, Fostering the Post-Pandemic Requirements'. In: *2024 Second International Conference on Emerging Trends in Information Technology and Engineering (ICETITE)*. Feb. 2024, pp. 1–4. DOI: 10.1109/ic-ETITE58242.2024.10493370.

[205] W. Sung and C. L. Thomas. 'The mediating role of academic help-seeking in the relationship between achievement goals and help-seeking from ChatGPT'. en. In: *Social Psychology of Education* 28.1 (Aug. 2025), p. 155. DOI: 10.1007/s11218-025-10108-7.

[206] V. Aleven, E. Stahl, S. Schworm, F. Fischer and R. Wallace. 'Help Seeking and Help Design in Interactive Learning Environments'. en. In: *Review of Educational Research* 73.3 (Sept. 2003), pp. 277–320. DOI: 10.3102/00346543073003277.

[207] Z. Gao, C. Lynch, S. Heckman and T. Barnes. 'Automatically Classifying Student Help Requests: A Multi-Year Analysis'. en. In: *Proceedings of The 14th International Conference on Educational Data Mining (EDM21)*. Paris, France: International Educational Data Mining Society, July 2021. URL: https://eric.ed.gov/?id=ED615496.

[208] M. Puustinen, J. Bernicot and A. Bert-Erboul. 'Written computer-mediated requests for help by French-speaking students: An analysis of their forms and functions'. In: *Learning and Instruction*. Special Section I: Solving information-based problems: Evaluating sources and information 21.2 (Apr. 2011), pp. 281–289. DOI: 10.1016/j.learninstruc.2010.07.005.

[209] S. A. Karabenick and J. R. Knapp. 'Relationship of academic help seeking to the use of learning strategies and other instrumental achievement behavior in college students'. In: *Journal of Educational Psychology* 83.2 (1991), pp. 221–230. DOI: 10.1037/0022-0663.83.2.221.

[210] S. Schworm and H. Gruber. 'e-Learning in universities: Supporting help-seeking processes by instructional prompts'. In: *British Journal of Educational Technology* 43.2 (Mar. 2012), pp. 272–281. DOI: 10.1111/j.1467-8535.2011.01176.x.

[211] W. Ye, M. Ou, T. Li, Y. chen, X. Ma, Y. Yanggong, S. Wu, J. Fu, G. Chen, H. Wang and J. Zhao. *Assessing Hidden Risks of LLMs: An Empirical Study on Robustness, Consistency, and Credibility*. Aug. 2023. DOI: 10.48550/arXiv.2305.10235.

[212] A. Schulz and J. Voermanek. 'How do help-seeking and help-abuse affect learning achievement in an interactive learning environment?' In: *Computers and Education Open* 8 (June 2025), p. 100247. DOI: 10.1016/j.caeo.2025.100247.

[213] M. Lan and X. Zhou. 'A qualitative systematic review on AI empowered self-regulated learning in higher education'. en. In: *npj Science of Learning* 10.1 (May 2025), pp. 1–16. DOI: 10.1038/s41539-025-00319-0.

[214] I. Hou, S. Mettille, O. Man, Z. Li, C. Zastudil and S. MacNeil. 'The Effects of Generative AI on Computing Students' Help-Seeking Preferences'. In: *Proceedings of the 26th Australasian Computing Education Conference*. ACE '24. New York, NY, USA: Association for Computing Machinery, Jan. 2024, pp. 39–48. DOI: 10.1145/3636243.3636248.

[215] M. Abolnejadian, S. Alipour and K. Taeb. 'Leveraging ChatGPT for Adaptive Learning through Personalized Prompt-based Instruction: A CS1 Education Case Study'. In: *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*. CHI EA '24. New York, NY, USA: Association for Computing Machinery, May 2024, pp. 1–8. DOI: 10.1145/3613905.3637148.

[216] G. Sawalha, I. Taj and A. Shoufan. 'Analyzing student prompts and their effect on ChatGPT's performance'. In: *Cogent Education* 11.1 (Dec. 2024), p. 2397200. DOI: 10.1080/2331186X.2024.2397200.

[217] K. A. Makara and C. M. Kuusinen. 'Academic Help Seeking as a Process of Seeking Formative Feedback on Learning'. en. In: *Advances in Motivation and Achievement*. Ed. by T. Urdan and E. N. Gonida. Emerald Publishing Limited, Feb. 2023, pp. 69–90. DOI: 10.1108/S0749-742320230000022006.

[218] J. J. G. v. M. Kirschner Paul A. '4C/ID in the Context of Instructional Design and the Learning Sciences'. In: *International Handbook of the Learning Sciences*. Routledge, 2018.

[219] H. Nguyen and A. Nguyen. 'Reflective Practices and Self-Regulated Learning in Designing with Generative Artificial Intelligence: An Ordered Network Analysis'. en. In: *Journal of Science Education and Technology* (Nov. 2024). DOI: 10.1007/s10956-024-10175-z.

[220] S. Nelson-Le Gall. 'Help-seeking: An understudied problem-solving skill in children'. In: *Developmental Review* 1.3 (Sept. 1981), pp. 224–246. DOI: 10.1016/0273-2297(81)90019-8.

[221] S. A. Karabenick and M. H. Dembo. 'Understanding and facilitating self-regulated help seeking'. en. In: *New Directions for Teaching and Learning* 2011.126 (2011), pp. 33–43. DOI: 10.1002/tl.442.

[222] Q. Liu and J. C. Nesbit. 'The Relation Between Need for Cognition and Academic Achievement: A Meta-Analysis'. EN. In: *Review of Educational Research* 94.2 (Apr. 2024), pp. 155–192. DOI: 10.3102/00346543231160474.

[223] A. Dong, M. S.-Y. Jong and R. B. King. 'How Does Prior Knowledge Influence Learning Engagement? The Mediating Roles of Cognitive Load and Help-Seeking'. English. In: *Frontiers in Psychology* 11 (Oct. 2020). DOI: 10.3389/fpsyg.2020.591203.

[224] J. Grass, F. Krieger, P. Paulus, S. Greiff, A. Strobel and A. Strobel. 'Thinking in action: Need for Cognition predicts Self-Control together with Action Orientation'. en. In: *PLOS ONE* 14.8 (Aug. 2019), e0220282. DOI: 10.1371/journal.pone.0220282.

[225] G. Lins de Holanda Coelho, P. H. P. Hanel and L. J. Wolf. 'The Very Efficient Assessment of Need for Cognition: Developing a Six-Item Version*'. EN. In: *Assessment* 27.8 (Dec. 2020), pp. 1870–1885. DOI: 10.1177/1073191118793208.

[226] A. M. Ryan, P. R. Pintrich and C. Midgley. 'Avoiding Seeking Help in the Classroom: Who and Why?' en. In: *Educational Psychology Review* 13.2 (June 2001), pp. 93–114. DOI: 10.1023/A:1009013420053.

[227] K. Schenke, A. C. Lam, A. M. Conley and S. A. Karabenick. 'Adolescents' help seeking in mathematics classrooms: Relations between achievement and perceived classroom environmental influences over one school year'. In: *Contemporary Educational Psychology* 41 (Apr. 2015), pp. 133–146. DOI: 10.1016/j.cedpsych.2015.01.003.

[228] A. Kozanitis, J.-F. Desbiens and R. Chouinard. 'Perception of Teacher Support and Reaction towards Questioning: Its Relation to Instrumental Help-Seeking and Motivation to Learn'. en. In: *International Journal of Teaching and Learning in Higher Education* 19.3 (2007), pp. 238–250. URL: https://eric.ed.gov/?id=EJ901297.

[229] P. R. Pintrich and B. Schragben. 'Students' Motivational Beliefs and Their Cognitive Engagement in Classroom Academic Tasks'. In: *Student Perceptions in the Classroom*. Routledge, 1992.

[230] A. D. Samala, S. Rawas, T. Wang, J. M. Reed, J. Kim, N.-J. Howard and M. Ertz. 'Unveiling the landscape of generative artificial intelligence in education: a comprehensive taxonomy of applications, challenges, and future prospects'. en. In: *Education and Information Technologies* 30.3 (Feb. 2025), pp. 3239–3278. DOI: 10.1007/s10639-024-12936-0.

[231] J. Wittwer and A. Renkl. 'Why Instructional Explanations Often Do Not Work: A Framework for Understanding the Effectiveness of Instructional Explanations'. In: *Educational Psychologist* 43.1 (Jan. 2008), pp. 49–64. DOI: 10.1080/00461520701756420.

[232] J. J. G. Van Merriënboer and D. M. A. Sluijsmans. 'Toward a Synthesis of Cognitive Load Theory, Four-Component Instructional Design, and Self-Directed Learning'. en. In: *Educational Psychology Review* 21.1 (Mar. 2009), pp. 55–66. DOI: 10.1007/s10648-008-9092-5.

[233] I. Roll, R. S. J. d. Baker, V. Aleven and K. R. Koedinger. 'On the Benefits of Seeking (and Avoiding) Help in Online Problem-Solving Environments'. In: *Journal of the Learning Sciences* 23.4 (Oct. 2014), pp. 537–560. DOI: 10.1080/10508406.2014.883977.

[234] G. Biswas, K. Leelawong, D. Schwartz, N. Vye and The Teachable Agents Group at Vanderbilt. 'Learning by Teaching: A New Agent Paradigm for Educational Software'. In: *Applied Artificial Intelligence* 19.3-4 (Mar. 2005), pp. 363–392. DOI: 10.1080/08839510590910200.

[235] N. P. Bakas, M. Papadaki, E. Vagianou, I. Christou and S. A. Chatzichristofis. 'Integrating LLMs in Higher Education, Through Interactive Problem Solving and Tutoring: Algorithmic Approach and Use Cases'. en. In: *Information Systems*. Springer, Cham, 2024, pp. 291–307. DOI: 10.1007/978-3-031-56478-9_21.

[236] M. Valle Torre, M. Specht and C. Oertel. *LLM Chatbots in High School Programming: Exploring Behaviors and Interventions*. Nov. 2025. DOI: 10.48550/arXiv.2511.18985.

[237] L. Y. Tan, S. Hu, D. J. Yeo and K. H. Cheong. 'Artificial intelligence-enabled adaptive learning platforms: A review'. In: *Computers and Education: Artificial Intelligence* 9 (Dec. 2025), p. 100429. DOI: 10.1016/j.caeai.2025.100429.

[238] D. T. K. Ng, C. W. Tan and J. K. L. Leung. 'Empowering student self-regulated learning and science education through ChatGPT: A pioneering pilot study'. en. In: *British Journal of Educational Technology* 55.4 (2024), pp. 1328–1353. DOI: 10.1111/bjet.13454.

[239] H. Li and B. Ma. 'CodeRunner Agent: Integrating AI Feedback and Self-Regulated Learning to Support Programming Education'. en. In: *International Conference on Computers in Education* (Dec. 2025). URL: https://library.apsce.net/index.php/ICCE/article/view/5568.

[240] J. Saint, A. Whitelock-Wainwright, D. Gasevic and A. Pardo. 'Trace-SRL: A Framework for Analysis of Microlevel Processes of Self-Regulated Learning From Trace Data'. en. In: *IEEE Transactions on Learning Technologies* 13.4 (Oct. 2020), pp. 861–877. DOI: 10.1109/TLT.2020.3027496.

[241] H. Jeong, A. Gupta, R. Roscoe, J. Wagster, G. Biswas and D. Schwartz. 'Using Hidden Markov Models to Characterize Student Behaviors in Learning-by-Teaching Environments'. en. In: *Intelligent Tutoring Systems*. Ed. by B. P. Woolf, E. Aïmeur, R. Nkambou and S. Lajoie. Berlin, Heidelberg: Springer, 2008, pp. 614–625. DOI: 10.1007/978-3-540-69132-7_64.

[242] P. Sharma, A. E. B. Stewart, Q. Li, K. Ravichander and E. Walker. 'Building Learner Activity Models from Log Data Using Sequence Mapping and Hidden Markov Models'. en. In: *Proceedings of the 17th International Conference on Educational Data Mining*. Atlanta, GA: International Educational Data Mining Society, 2024. URL: https://eric.ed.gov/?id=ED675539.

[243] Anonymous. 'Blinded For Review'. In: 2000.

[244] J. Wang, J. Wang, B. Athiwaratkun, C. Zhang and J. Zou. *Mixture-of-Agents Enhances Large Language Model Capabilities*. June 2024. DOI: 10.48550/arXiv.2406.04692.

[245] H.-Y. Lee, P.-H. Chen, W.-S. Wang, Y.-M. Huang and T.-T. Wu. 'Empowering ChatGPT with guidance mechanism in blended learning: effect of self-regulated learning, higher-order thinking skills, and knowledge construction'. In: *International Journal of Educational Technology in Higher Education* 21.1 (Mar. 2024), p. 16. DOI: 10.1186/s41239-024-00447-4.

[246] M. Liu, L. J. Zhang and C. Biebricher. 'Investigating students' cognitive processes in generative AI-assisted digital multimodal composing and traditional writing'. In: *Computers & Education* 211 (Apr. 2024), p. 104977. DOI: 10.1016/j.compedu.2023.104977.

[247] V. Kotu and B. Deshpande. 'Chapter 3 - Data Exploration'. In: *Data Science: Concepts and Practice*. Ed. by V. Kotu and B. Deshpande. 2nd ed. Cambridge, USA: Morgan Kaufmann, Jan. 2019, pp. 39–64. DOI: 10.1016/B978-0-12-814761-0.00003-4.

[248] D. P. Kroese, Z. Botev, T. Taimre and R. Vaisman. *Data Science and Machine Learning: Mathematical and Statistical Methods*. New York: Chapman and Hall/CRC, Nov. 2019. DOI: 10.1201/9780367816971.

[249] M. Amoozadeh, D. Nam, D. Prol, A. Alfageeh, J. Prather, M. Hilton, S. Srinivasa Ragavan and A. Alipour. 'Student-AI Interaction: A Case Study of CS1 students'. In: *Proceedings of the 24th Koli Calling International Conference on Computing Education Research*. Koli Calling '24. New York, NY, USA: Association for Computing Machinery, Nov. 2024, pp. 1–13. DOI: 10.1145/3699538.3699567.

[250] S. A. Karabenick and E. N. Gonida. 'Academic help seeking as a self-regulated learning strategy: Current issues, future directions'. In: *Handbook of self-regulation of learning and performance, 2nd ed*. Educational psychology handbook series. New York, NY, US: Routledge/Taylor & Francis Group, 2018, pp. 421–433. DOI: 10.4324/9781315697048-27.

[251] A. Hunt and D. Thomas. *The pragmatic programmer: from journeyman to master*. en. 26. print. Boston: Addison-Wesley, 2011.

[252] V. Tinto. 'Reflections on student persistence'. In: *Student Success* 8.2 (Mar. 2021), pp. 1–8. DOI: 10.3316/informit.593199291602507.

[253] M. Valle Torre, M. Specht and C. Oertel. 'LLM Chatbots in High School Programming: Exploring Behaviors and Interventions'. In: *Proceedings of the 41st ACM/SIGAPP Symposium on Applied Computing*. SAC '26. New York, NY, USA: Association for Computing Machinery, June 2026, pp. 79–86. DOI: 10.1145/3748522.3779870.

[254] M. Valle Torre, M. Specht and C. Oertel. *What makes an Expert? Comparing Problem-solving Practices in Data Science Notebooks*. Feb. 2026. DOI: 10.48550/arXiv.2602.15428.

[255] M. Valle Torre. 'Learning Sequence Analytics for Support in Learning Tasks'. In: *EC-TEL (Doctoral Consortium)*. Aveiro, Portugal, 2023, pp. 57–62. URL: https://ceur-ws.org/Vol-3539/paper9.pdf.

[256] A. Soleymani, L. Itard, M. d. Laat, M. Valle Torre and M. Specht. 'Using Social Network Analysis to explore Learning networks in MOOCs discussion forums'. en. In: *CLIMA 2022 conference* (May 2022). DOI: 10.34641/clima.2022.300.

[257] X. Zhang, M. Specht and M. Valle Torre. 'An investigation on integration of computational thinking into engineering curriculum at Delft University of Technology'. eng. In: Universitat Politècnica de Catalunya, Sept. 2022, pp. 890–901. DOI: 10.5821/conference-9788412322262.1230.

[258] N. Roy, M. Valle Torre, U. Gadiraju, D. Maxwell and C. Hauff. 'Note the Highlight: Incorporating Active Reading Tools in a Search as Learning Environment'. en. In: *Proceedings of the 2021 Conference on Human Information Interaction and Retrieval*. Canberra ACT Australia: ACM, Mar. 2021, pp. 229–238. DOI: 10.1145/3406522.3446025.

[259] N. Roy, M. Valle Torre, U. Gadiraju, D. Maxwell and C. Hauff. 'How Do Active Reading Strategies Affect Learning Outcomes in Web Search?' en. In: *Advances in Information Retrieval*. Ed. by D. Hiemstra, M.-F. Moens, J. Mothe, R. Perego, M. Potthast and F. Sebastiani. Vol. 12657. Cham: Springer International Publishing, 2021, pp. 368–375. DOI: 10.1007/978-3-030-72240-1_37.

[260] I. Jivet, J. Wong, M. Scheffel, M. Valle Torre, M. Specht and H. Drachsler. 'Quantum of Choice: How learners' feedback monitoring decisions, goals and self-regulated learning skills are related'. en. In: *LAK21: 11th International Learning Analytics and Knowledge Conference*. Irvine CA USA: ACM, Apr. 2021, pp. 416–427. DOI: 10.1145/3448139.3448179.

[261] M. Valle Torre, C. Lofi and M. Ye. 'Perceptual Relational Attributes: Navigating and Discovering Shared Perspectives from User-Generated Reviews'. en. In: Gesellschaft für Informatik, Bonn, 2019, pp. 169–189. URL: https://dl.gi.de/handle/20.500.12116/21695.

[262] S. Mesbah, J. Yang, R.-J. Sips, M. Valle Torre, C. Lofi, A. Bozzon and G.-J. Houben. 'Training Data Augmentation for Detecting Adverse Drug Reactions in User-Generated Content'. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Ed. by K. Inui, J. Jiang, V. Ng and X. Wan. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 2349–2359. DOI: 10.18653/v1/D19-1239.

[263] S. Mesbah, G. Chen, M. Valle Torre, A. Bozzon, C. Lofi and G.-J. Houben. 'Concept Focus: Semantic Meta-Data for Describing MOOC Content'. en. In: *Lifelong Technology-Enhanced Learning*. Ed. by V. Pammer-Schindler, M. Pérez-Sanagustín, H. Drachsler, R. Elferink and M. Scheffel. Cham: Springer International Publishing, 2018, pp. 467–481. DOI: 10.1007/978-3-319-98572-5_36.

[264] S. Mesbah, C. Lofi, M. Valle Torre, A. Bozzon and G.-J. Houben. 'TSE-NER: An Iterative Approach for Long-Tail Entity Extraction in Scientific Publications'. en. In: *The Semantic Web – ISWC 2018*. Ed. by D. Vrandečić, K. Bontcheva, M. C. Suárez-Figueroa, V. Presutti, I. Celino, M. Sabou, L.-A. Kaffee and E. Simperl. Cham: Springer International Publishing, 2018, pp. 127–143. DOI: 10.1007/978-3-030-00671-6_8.

List of Publications

16. **M. Valle Torre**, R. Weijers, M. Specht and C. Oertel. *Process Collapse or Compensatory Strategy? Profiling and Scaffolding Help-Seeking with an AI Tutor*. en. SSRN Scholarly Paper. Rochester, NY, Feb. 2026. DOI: 10.2139/ssrn.6297817

15. **M. Valle Torre**, M. Specht and C. Oertel. 'LLM Chatbots in High School Programming: Exploring Behaviors and Interventions'. In: *Proceedings of the 41st ACM/SIGAPP Symposium on Applied Computing*. SAC '26. New York, NY, USA: Association for Computing Machinery, June 2026, pp. 79–86. DOI: 10.1145/3748522.3779870

14. **M. Valle Torre**, T. van der Velden, M. Specht and C. Oertel. 'JELAI: Integrating AI and Learning Analytics in Jupyter Notebooks'. en. In: *Artificial Intelligence in Education*. Ed. by A. I. Cristea, E. Walker, Y. Lu, O. C. Santos and S. Isotani. Cham: Springer Nature Switzerland, 2025, pp. 68–75. DOI: 10.1007/978-3-031-98465-5_9

13. **M. Valle Torre**, M. Specht and C. Oertel. *What makes an Expert? Comparing Problem-solving Practices in Data Science Notebooks*. arXiv:2602.15428 [cs]. Feb. 2026. DOI: 10.48550/arXiv.2602.15428

12. **M. Valle Torre**, C. Oertel and M. Specht. 'The Sequence Matters in Learning - A Systematic Literature Review'. In: *Proceedings of the 14th Learning Analytics and Knowledge Conference*. LAK '24. New York, NY, USA: Association for Computing Machinery, Mar. 2024, pp. 263–272. DOI: 10.1145/3636555.3636880

11. **M. Valle Torre**. 'Learning Sequence Analytics for Support in Learning Tasks'. In: *EC-TEL (Doctoral Consortium)*. Aveiro, Portugal, 2023, pp. 57–62. URL: https://ceur-ws.org/Vol-3539/paper9.pdf

10. A. Soleymani, L. Itard, M. d. Laat, **M. Valle Torre** and M. Specht. 'Using Social Network Analysis to explore Learning networks in MOOCs discussion forums'. en. In: *CLIMA 2022 conference* (May 2022). DOI: 10.34641/clima.2022.300

9. X. Zhang, M. Specht and **M. Valle Torre**. 'An investigation on integration of computational thinking into engineering curriculum at Delft University of Technology'. eng. In: Universitat Politècnica de Catalunya, Sept. 2022, pp. 890–901. DOI: 10.5821/conference-9788412322262.1230

8. N. Roy, **M. Valle Torre**, U. Gadiraju, D. Maxwell and C. Hauff. 'Note the Highlight: Incorporating Active Reading Tools in a Search as Learning Environment'. en. In: *Proceedings of the 2021 Conference on Human Information Interaction and Retrieval*. Canberra ACT Australia: ACM, Mar. 2021, pp. 229–238. DOI: 10.1145/3406522.3446025

7. N. Roy, **M. Valle Torre**, U. Gadiraju, D. Maxwell and C. Hauff. 'How Do Active Reading Strategies Affect Learning Outcomes in Web Search?' en. In: *Advances in Information*

Retrieval. Ed. by D. Hiemstra, M.-F. Moens, J. Mothe, R. Perego, M. Potthast and F. Sebastiani. Vol. 12657. Series Title: Lecture Notes in Computer Science. Cham: Springer International Publishing, 2021, pp. 368–375. DOI: 10.1007/978-3-030-72240-1_37

6. I. Jivet, J. Wong, M. Scheffel, **M. Valle Torre**, M. Specht and H. Drachsler. 'Quantum of Choice: How learners' feedback monitoring decisions, goals and self-regulated learning skills are related'. en. In: *LAK21: 11th International Learning Analytics and Knowledge Conference*. Irvine CA USA: ACM, Apr. 2021, pp. 416–427. DOI: 10.1145/3448139.3448179

5. **M. Valle Torre**, E. Tan and C. Hauff. 'edX log data analysis made easy: introducing ELAT: An open-source, privacy-aware and browser-based edX log data analysis tool'. In: *Proceedings of the Tenth International Conference on Learning Analytics & Knowledge*. LAK '20. New York, NY, USA: Association for Computing Machinery, Mar. 2020, pp. 502–511. DOI: 10.1145/3375462.3375510

4. **M. Valle Torre**, C. Lofi and M. Ye. 'Perceptual Relational Attributes: Navigating and Discovering Shared Perspectives from User-Generated Reviews'. en. In: Gesellschaft für Informatik, Bonn, 2019, pp. 169–189. URL: https://dl.gi.de/handle/20.500.12116/21695

3. S. Mesbah, J. Yang, R.-J. Sips, **M. Valle Torre**, C. Lofi, A. Bozzon and G.-J. Houben. 'Training Data Augmentation for Detecting Adverse Drug Reactions in User-Generated Content'. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Ed. by K. Inui, J. Jiang, V. Ng and X. Wan. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 2349–2359. DOI: 10.18653/v1/D19-1239

2. S. Mesbah, G. Chen, **M. Valle Torre**, A. Bozzon, C. Lofi and G.-J. Houben. 'Concept Focus: Semantic Meta-Data for Describing MOOC Content'. en. In: *Lifelong Technology-Enhanced Learning*. Ed. by V. Pammer-Schindler, M. Pérez-Sanagustín, H. Drachsler, R. Elferink and M. Scheffel. Cham: Springer International Publishing, 2018, pp. 467–481. DOI: 10.1007/978-3-319-98572-5_36

1. S. Mesbah, C. Lofi, **M. Valle Torre**, A. Bozzon and G.-J. Houben. 'TSE-NER: An Iterative Approach for Long-Tail Entity Extraction in Scientific Publications'. en. In: *The Semantic Web – ISWC 2018*. Ed. by D. Vrandečić, K. Bontcheva, M. C. Suárez-Figueroa, V. Presutti, I. Celino, M. Sabou, L.-A. Kaffee and E. Simperl. Cham: Springer International Publishing, 2018, pp. 127–143. DOI: 10.1007/978-3-030-00671-6_8

Curriculum Vitæ

Manuel Valle Torre

13-12-191	Born in San Luis Potosi, Mexico.

Education and Experience

2010–2015	Undergraduate in Business Informatics Tecnológico de Monterrey, Campus S.L.P, Mexico
2012	Telecommunications and Systems Management (Minor) Universitat Politècnica de Catalunya, Spain
2015–2018	MSc. Computer Science, Data Science and Technology Delft University of Technology, The Netherlands
2024–2025	Learning Analytics Specialist Npuls, The Netherlands
2026	PhD. Computer Science Delft University of Technology, The Netherlands *Thesis:* Sequences in Data to Represent Learning Processes *Promotor:* Prof. dr. M. Specht & Dr. C. Oertel

Pictures

Pictures taken by me.

Epigraph	Mt. Etna, Italy 2025
Chapter 1	Mt. Cotopaxi, Ecuador 2022
Chapter 2	Ikuchi Island, Japan 2024
Chapter 3	Perito Moreno Glacier, Argentina 2023
Chapter 4	El Angel, Ecuador 2022
Chapter 5	Salento, Colombia 2022
Chapter 6	Huacachina, Peru 2022
Chapter 7	Cerro Castillo, Chile 2023
Chapter 8	Ruta 40, Estancia Librun, Argentina 2023 (by Lieke)