

INTERDISCIPLINARY COMPUTATIONAL AND DATA LITERACY IN ACTION: INSIGHTS FROM THE FIRST COMPUTATIONAL & DATA SCIENCE HACKATHON

Jasmin Hörter

Karlsruhe Institute of Technology (KIT)
jasmin.hoerter@kit.edu

How to predict floods, how to evaluate radio signals, how to navigate robots for search and rescue, or how to design a learning agent for data extraction? Undergraduate students examine these and other topics in a newly developed teaching concept, the CDS Hackathon at Karlsruhe Institute of Technology. It aims to connect students with real-world problems from science, engineering and economics, to establish collaborative and interdisciplinary work, to promote data literacy and solution-oriented thinking, and to teach the implementation of solutions using modern data science technologies within a dedicated Jupyter environment. We introduce the course structure and present initial findings from the first CDS Hackathon.

INTRODUCTION

As the influence of artificial intelligence and computer-aided modeling in almost all areas of life grows, competencies in data literacy and scientific computing as well as their application to complex problems are increasingly important. Moreover, the rise of generative AI opens a vast range of possibilities and blurs the lines between formerly distinct research fields. To take advantage of this development, experts at the intersection of scientific domains are needed. In 2025, the Karlsruhe Institute of Technology (KIT) introduced two new study programs, Computational & Data Science (CDS) B.Sc. and M.Sc., which provide graduates with a multidisciplinary education in both mathematics and computer science, as well as an application subject from science, engineering or economics (SEE disciplines). Students are taught to effectively combine data-driven and model-based methods and to create solutions to complex challenges in diverse application fields. To facilitate this, new and innovative teaching concepts have been developed, including the CDS Hackathon. During the weeklong course, students work in teams on research questions from the SEE disciplines, using state-of-the-art High-Performance Computing (HPC) and AI infrastructures.

OBJECTIVES

The CDS Hackathon aims to familiarize students with open-ended research questions and to promote modeling and problem-solving skills in a real-world setting. Students learn to examine a problem from all angles and investigate different solution paths. Critical thinking and reflection on why certain attempts do not lead to the expected outcome are equally valued.

Compared to many standardized exercises in the curriculum, the hackathon aims to transfer theoretical knowledge to a practical application. This includes real-world data and software whose specifications may differ from those of previously studied synthetic examples. In their professional lives, students will often encounter similar situations, and the hackathon is an opportunity to develop advanced techniques for dealing with large amounts of data and highly complex, interdisciplinary problems.

In addition, students learn to present results in a coherent way and follow good scientific practice. The computational essay, usually a Jupyter Notebook, should not only include the final solution but also trace the group's progress throughout and mention failed attempts or rejected ideas.

Moreover, students learn to work as a team, manage group dynamics and cooperate in the most effective way possible. They work autonomously and take ownership of their project progression. On an individual level, each member may learn about their own skills that distinguish them and that complement skills of other members. These may include technical skills such as subject matter expertise or programming skills, or soft skills such as management or leadership skills.

COURSE STRUCTURE

The CDS Hackathon is jointly organized by the Scientific Computing Center of KIT as well as researchers from the SEE departments who act as supervisors and design topics from their respective field. It runs for one week twice a year before the start of the regular lecture period. Students can register

their preferences for topics beforehand, and groups of four to six people are assigned by an optimization algorithm. This leads to more diverse teams, simulating a real workplace setting. On the first day of the hackathon, each SEE supervisor introduces their topic and open problem in more detail, including any supporting material like data sets or software packages. Afterwards the group works independently. Throughout the week, on average the supervisor checks in once a day for about an hour to answer questions or offer hints. Preliminary work by the students is not required, although an introductory self-study course is provided to familiarize students with basic programming skills and the Jupyter environment. On the last day of the hackathon, each group introduces their findings in a ten-minute pitch. Two weeks after the hackathon, each group must hand in their computational essay for review. Students pass the course if they successfully participate in all three parts: group work, presentation of results, and computational essay.

The hackathon combines aspects of the student centered learning pedagogies challenge-based learning (CBL), problem-based learning (PBL) and project-based learning (PjBL). All three concepts are closely related, with many overlapping characteristics (See Doulougeri et al. (2024) and Membrillo-Hernández et al. (2019) for classifications). In CBL, students address a global challenge by identifying and focusing on a subproblem. An applicable solution is often reached in an interdisciplinary context and with the help of external stakeholders. (Dikilitaş et al., 2025) PBL on the other hand focuses on a specific problem which is predefined by the teacher and solved through literature research. The main focus is on knowledge acquisition and skill development. (Hmelo-Silver, 2004) PjBL centers around a specific question or task that needs to be solved within a project framework. The outcome is a concrete product or implementation. (Kolmos, 1996)

For the moment, the hackathon topic description follows the PBL and PjBL approach as it outlines a specific, real-world scenario that includes a clearly defined open question or task, ideally without a unique or obvious textbook solution. As students advance through the study program, topic instructions might become sparser and more challenge-based.

For the first one or two days, the CDS Hackathon follows the PBL concept and trains the students' ability to clearly understand the context and identify the problem. They are encouraged to familiarize themselves with their topic by conducting their own research and learn to abstract the problem by formulating it mathematically. Then they critically evaluate advantages and disadvantages of appropriate CDS methods.

As the week progresses, the hackathon moves to a PjBL setting, where the students realize defined strategies by applying CDS specific knowledge and competencies. Compared to the CBL approach, this does not require advanced expertise in the individual SEE disciplines. To implement solutions, students receive access to computing clusters, and they are free to use any digital resources, AI-powered tools or software packages to implement the proposed solution. However, each group is expected to follow good scientific practice, cite sources and clearly describe the application scenario. The critical evaluation of model outputs is especially important and might lead to reevaluations and readjustments of the solution process. This implementation cycle should take two to three days.

As the hackathon is student centered, the groups decide for themselves how they work, whether and how tasks are assigned, and which solutions are pursued. Because of the limited time and defined end point, students need to structure their work process and define project goals. Exercising self-management follows the PjBL approach, where the supervisor's role is more restrained and limited to advice or feedback.

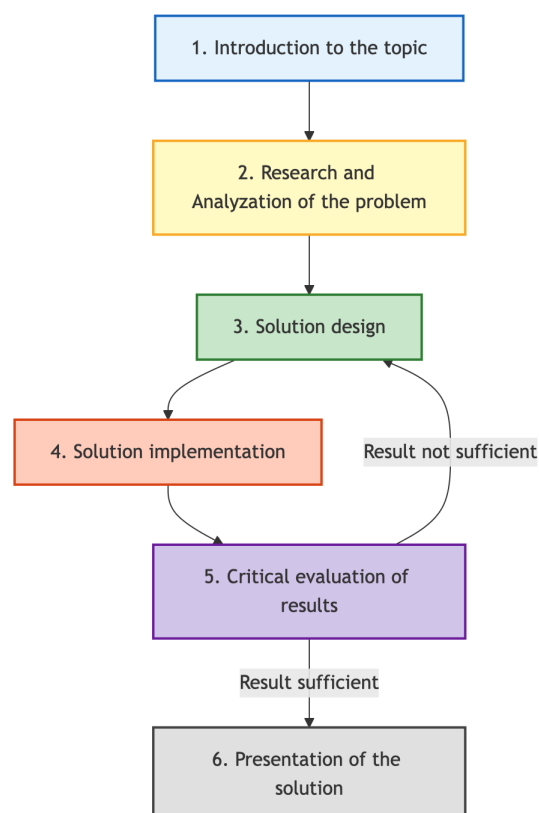


Figure 1: Progression through the week

To conclude, all groups present their results as well as their progression through the week. The hackathon is non-competitive and solutions are not ranked. While each group should aim for the best possible outcome, equal emphasis is on the process and the knowledge gained, a central component of all three learning pedagogies.

TECHNICAL SETUP

To simulate research conditions, students work with infrastructures that scientists use routinely. In the CDS context, these infrastructures are typically HPC clusters with both CPU partitions for classical simulations as well as GPU partitions for advanced modelling and AI applications. A low-threshold way to access these resources is provided by the Jupyter environment, which enables interactive and visual representation of research processes, allowing students to engage with scientific methods and data analysis in a practical way. By combining code, text and graphics in a Jupyter Notebook, complex concepts can be communicated in an understandable way, and the flexible integration of programming languages such as Python, Julia, Matlab or R further promotes the development of programming skills. (See Fig. 2 and 3)

KIT operates *bwJupyter*, a JupyterHub environment dedicated to university teaching and accessible to students throughout the state of Baden-Württemberg. Access is browser based, and thus independent of devices and locations. Each supervisor creates a bwJupyter profile for their individual topic and uploads relevant data sets and supplementary material. A real-time collaboration feature allows for simultaneous work on the same Notebook by multiple team members and the AI supported programming assistant supports novice programmers.

FIRST ITERATION

The first CDS Hackathon took place at the beginning of the 2026 summer term. In total 50 students registered, of which 38 had just finished the first term of the CDS program consisting of introductory lectures in calculus, linear algebra and programming, and 12 were undergraduates from other study programs (computer science, economics, mathematics).

#	TOPIC	SEE DISCIPLINE
1	Analysis of Annual Reports	Business and Economics
2	Learning Agents for Scientific Data Extraction and Structuring	Business and Economics
3	Flood Forecast for Germany	Computational Earth System Sciences
4	Development of a Temperature Model for the Atmosphere	Computational Earth System Sciences
5	Learning Constitutive Material Laws from Synthetic Data: A Comparison of Neural Networks and physics-augmented Neural Networks	Computational Mechanics and Thermodynamics
6	Protect your Asset – Avalanche Safety	Computational Mechanics and Thermodynamics
7	Interaction between Flame Intensity and Flow Characteristics	Computational Mechanics and Thermodynamics
8	Trick or Tweet? – Recognizing Bird Calls	Computational Physics
9	Analysis of Radio Signals	Electrical Engineering and Information Technology
10	Swinging and Stabilization of an Inverted Pendulum on a Cart	Process Engineering
11	Autonomous Robots for Search and Rescue following a Reactor Accident at Campus North	Robotics and Autonomous Systems

All topics involved the preparation, processing, and analysis of data, although the methods differed. Topics 1, 4, 6, 7, 9, 10 and 11 were solved using classical modeling of (physical) quantities or statistical analysis, while topics 2, 3, 5 and 8 utilized neural networks.

Taking a closer look at selected topics, the first two topics investigated different approaches to data extraction and analysis. Topic 1 required the automated analysis of annual reports, which could

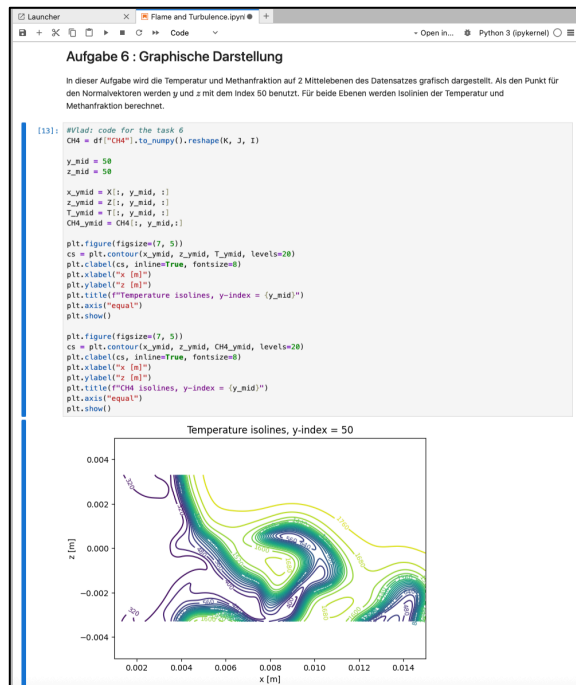


Figure 2: Comp. Essay Topic 7, Temperature isolines

was to simulate the fundamental processes of the earth's radiation balance and to highlight the differences between pre-industrial and modern climate conditions. The challenge lay in implementing the various physical laws and quantities that influence the climate within a classical model. Topic 3, in contrast, used neural networks to simulate flooding. The group worked with a set of real measurement data and developed its own Long Short-Term Memory (LSTM) model, which recognized recurring patterns. To ensure physical consistency, the LSTM model was further expanded to include a conceptual Simple Hydrological model.

Topic 7 analyzed a comprehensive three-dimensional dataset from a numerical simulation of a premixed turbulent flame composed of methane and oxygen. (Fig. 2) The dataset described the flow and combustion within a Cartesian computational domain and included the spatial coordinates, the temperature, the velocity components in all three spatial directions, and the mass fractions of methane and oxygen. The goal was to process this raw data, determine characteristic and physical quantities, and investigate the relationships between flame intensity, flow properties, and turbulent effects using classical computational tools.

Topic 8 addressed a classification issue in environmental science, namely the recording and identification of birds to monitor biodiversity and track migration patterns. The group developed a tool to detect bird calls in a sound file using a machine learning based approach. They build their own neural network with three convolutional layers from scratch using Pytorch. To train the model, they had to extract data from the internet and preprocess it to suit their setup.

Topic 9 was structured as a treasure hunt: the group received a radio signal and had to decode its message. (Fig. 3) In doing so, they explored fundamental methods of signal processing and spectral analysis, illustrating how communication systems are structured and how information is physically transmitted. The initial challenges were to understand the format

then be used for trading recommendations. The group was tasked with developing a tool that automatically downloads reports from the internet, extracts and standardizes the different formats, and consolidates them into an Excel spreadsheet. For data preparation, the group utilized various Python libraries. In a second step, the obtained data was statistically evaluated and compared to data from previous years to generate recommendations.

In topic 2 on the other hand, the group built an AI-powered agent system capable of ingesting PDF formats and automatically extracting structured, machine-readable data, regardless of how the information was presented. The group used a combination of pre-trained models (large language and vision language models) to design a system that could handle prose descriptions, data tables and figures, and output a unified, structured dataset.

Topics 3 and 4 focused on problems related to the earth system. In topic 4, a simplified climate model was developed to illustrate the global distribution of radiation and temperature. The task



Figure 3: Comp. Essay Topic 9, Spectrogram

of the provided data and analyze and visualize it using Fast Fourier Transform and spectrograms. Subsequently, they programmed a decoder that decrypted the message.

Finally, topic 11 outlined a disaster scenario, where the KIT campus was destroyed by a nuclear accident. The students had to navigate a robot to rescue survivors as fast as possible by optimizing partially complete navigation code. They were provided with a simulation that maps out the path to the finish line, various robot designs with different characteristics and a scanner that renders the environment around the robot as a point cloud. Using this the group was able to navigate the robot and, after a few attempts, reduce the travel time.

Challenges and Results

As time was limited, constraints such as the duration of a simulation run or the cumbersome (pre-)processing of raw data posed obstacles and the groups had to prioritize which approach to pursue. Nevertheless, all groups were able to present results and reflected on what skills they had practiced. These can be grouped into four larger segments: applying theory to practice, finding and processing of data sets, working with established research concepts and tools, and establishing and structuring projects. At the heart of many topics is an optimization problem, which is mathematically well understood. The difficulty here is to apply the theoretical machinery to the given circumstances, for example to compare and choose the most suitable loss-function, or to choose the best hyperparameters in each learning algorithm. The first approach was often through trial and error, but the longer the groups were working on the problems and the more results they obtained, the more they began to develop a sense of how to improve their solution.

While all groups had to work with data sets in one way or another, topics 1, 2 and 8 had to find data sets online and convert different formats into a standardized form before they could process them further. This highlighted the complexity of data-driven work, which is often not reflected in standardized exercises. Moreover, groups using machine learning approaches discovered how much data is necessary during training to receive reliable results. Instead of building everything from scratch, some groups also tried to incorporate existing codes or pre-trained networks into their solutions.

In addition, most students were introduced to concepts such as software modules and libraries, workflows, version control and Git for the first time. Although these are well-established tools in CDS, students are rarely exposed to them this early in the curriculum.

Apart from these technical skills, students also learned to structure and realize projects in a short amount of time. Once they understood the problem and developed an approach, they split the workload in different sub-tasks and worked in parallel. Finally, all groups developed ideas on how to further improve their presented results, and which next steps could be implemented.

STUDENT FEEDBACK

After the essay submission deadline, each team was asked to participate in a dedicated questionnaire on technology, topics, group work, organization and overall impressions. Only ten groups participated in the survey and not every group answered every question.

Even though most participants had just finished the first term of their Bachelor's degree, all of them indicated at least basic programming experience in either Python (eight groups) or Java (eight groups). To solve their tasks, all groups used the provided bwJupyter environment but only three groups had previous experience with Jupyter Notebooks, with seven groups stating to have no experience with any development environment. bwJupyter got mixed results with most groups indicating a good user experience, however some groups struggled with resource utilization and run times. Seven groups mentioned the use of AI-supported tools like programming assistants to solve their tasks. Introductions to Git and Jupyter as well as more in-depth programming courses in the relevant language were among the technology suggestions to improve the course.

The topics were well received, with eight groups rating them either very good or good. The students enjoyed the insights into practical applications and the realistic topics. They liked the open-endedness of the problems and the freedom to formulate their own goals and solutions without the immediate intervention of the supervisors. Moreover, they were surprised that they were able to produce results in a relatively short amount of time. Working with large data sets and AI models as well as visualizing simulation results were named as highlights. The level of difficulty was mostly rated neutral (four groups) or hard (four groups) and the allocated time was mainly perceived as appropriate (neutral)

or rather short (seven groups) with two groups rating it as too short. This is not surprising given the timeframe and structure of the problems which encourages students to explore beyond the given setting. Nevertheless, the absence of a predetermined target lead to frustration that some students perceived negatively.

The group work was mainly well perceived, and eight groups indicated that they enjoyed the experience. Motivation and equal participation of all members were mentioned as areas for improvement. The supervision was also rated highly with four groups rating their supervisors as very good and four groups rating them very good to good. The only negative aspect mentioned was the limited availability throughout the week.

Eight groups were either very satisfied or satisfied with the overall organization of the course and a majority (four very much, three much, two neutral) enjoyed the event overall. Among the positive aspects mentioned are the diverse and relevant topics and real-world applications compared to many classical lectures and courses, the collaboration in teams, the freedom to try different things and being allowed to make mistakes, and the opportunity to learn new skills. Negative aspects mentioned are the short timeframe and high workload, the (at least perceived) varying difficulty levels of the topics as well as the overlong closing event. The four-hour presentation of results was also the most frequently mentioned topic of improvement for the next hackathon.

SUPERVISOR FEEDBACK

The supervisors were overall satisfied with the students' participation during the week as well as the presented results. Before the hackathon, many supervisors worried about the difficulty level as well as the lack of domain specific knowledge by the students. This concern did not turn out to be valid, as all groups were able to get up to speed on their topics and produce results. Nevertheless, some tasks were only partially fulfilled, which raised the idea to build on these achievements and continue with the same topic in the next iteration of the hackathon. One area identified for improvement is the structure and content of the computational essay, as some essays contained 20+ pages of code without adequate explanation or indication of GenAI usage.

CONCLUSION AND OUTLOOK

All in all, the first iteration of the hackathon was well received. The topics covered a wide range of subjects and highlighted various aspects of computer aided modeling and simulation. At the same time, all topics were clearly linked to a real-world application, a fact that was highly praised by the students. All groups were able to present results, even though some were unable to complete their tasks entirely due to time constraints.

The documentation and final reporting have room for improvement and students will receive more in-depth instructions on good scientific practice as well as current policies regarding AI usage before the start of the next hackathon.

As the number of students in the CDS program and thus the number of topics and groups grows, the final presentation will become increasingly lengthy. Thus, it will be replaced by an interactive poster session starting with a one-minute pitch of each topic. This will allow for more in-depth discussions and individual questions.

To highlight CDS applications not only in research but also in business and industry, the hackathon could be extended to include topics from companies as well. This concept has proven successful within the CAMMP modeling week. (Schönbrodt et al., 2022 and Engelhardt et al., 2025) It would expose students to various fields of work, while companies got the chance to play an active role in the education of future graduates.

REFERENCES

- Engelhardt, F., Wrede, S., Schönbrodt, S., Büsing, C., & Stamm, B. (2025). The Computational and Mathematical Modeling Program (CAMMP) – Lessons Learned on Interdisciplinary Challenge-Based Learning. *SEFI 53rd Annual Conference* (SEFI 2025). <https://doi.org/10.5281/zenodo.17631434>
- Dikilitaş K., Marshall T., & Shahverdi M. (2025) *A Practical Guide to Understanding and Implementing Challenge-Based Learning* (1). Palgrave Macmillan Cham. <https://doi.org/10.1007/978-3-031-67011-4>

- Doulougeri, K., Vermunt, J.D., Bombaerts, G., & Bots, M. (2024). Challenge-based learning implementation in engineering education: A systematic literature review. *Journal of Engineering Education*, 113(4), 1076–1106. <https://doi.org/10.1002/jee.20588>
- Hmelo-Silver, C.E. (2004). Problem-Based Learning: What and How Do Students Learn? *Educational Psychology Review* **16**, 235–266. <https://doi.org/10.1023/B:EDPR.0000034022.16470.f3>
- Kolmos, A. (1996). Reflections on Project Work and Problem-based Learning. *European Journal of Engineering Education*, 21(2), 141–148. <https://doi.org/10.1080/03043799608923397>
- Membrillo-Hernández, J., J. Ramírez-Cadena, M., Martínez-Acosta, M. *et al.* Challenge based learning: the importance of world-leading companies as training partners. *Int J Interact Des Manuf* **13**, 1103–1113 (2019). <https://doi.org/10.1007/s12008-019-00569-4>
- Schönbrodt, S., Wohak, K., & Frank, M. (2022). Digital tools to enable collaborative mathematical modeling online. *Modelling in Science Education and Learning*, 15(1), 151–174, <https://doi.org/10.4995/msel.2022.16269>