

LZSCC.300 3rd Year Projects 2025 – 2026

Choosing a 3rd Year Project

This booklet lists 62 project suggestions and topics organised by the LZSCC academic staff. Projects range from theoretical to experimental, and from design to implementation-led. Many projects are starting points for negotiation - they allow you to choose a project area and define this jointly with your supervisor over the summer and early next academic year.

Note that projects often have constraints, or particular skills that supervisors recommend. You should consider these when you make your choice, but remember that a project is always a chance to learn new skills - so aptitude and willingness to learn is at least as important as already having the specific skills in question. By all means discuss your suitability for the project with the supervisor who set the project.

On the information day, we will be providing an opportunity to hear more, and meet with the academic staff who suggested the projects to discuss the various projects further.

The Allocation Process

At the information day – Wednesday 7 May from 10:00 to 13:00 – you will receive instructions for entering your project choices. Using an online form on the Moodle page “[23/24: BSc Hons Computer Science \(Leipzig\) \(Full Time\) \[1\]](#)” under the section “[Third Year Projects](#)”, you will need to enter a number of project choices that you are willing to engage with. To guarantee a good division of projects, you are required to indicate at least one project per academic staff member. We will do our best to allocate you a project on your list, but it is impossible to give all students one of their top choices, while still balancing the overall supervision fairly across the academics. On the form, we will also ask you to write some text about the general topic areas that interest you, what you are hoping to learn in your final year project, as well as supervisors that you would potentially like to work with. This information is used if we cannot give you one of your choices.

Finally, the form will have the optional field to propose your own project. It is important that this kind of project is co-designed with a potential supervisor to ensure that the project is of the appropriate scope and difficulty. The form will ask you who the contacted academic staff member is and for 150–200 words about the project. This should contain information about the problem that is being investigated, how you expect to approach developing a solution, and what methods you will use to implement, analyse and evaluate the solution you develop.

From the information day, you will have until Wednesday 28 May at 15:00 German time to enter your choices. Within a week (i.e., by Wednesday 4 June), we will release the project allocations. You should get in touch with your supervisor before the end of Summer Term to get advice on what kind of things you can work on over the summer break.

Contents

1 Aakash Ahmad	4
1.1 Quantum Algorithms as a Service (QAaaS)	4
1.2 Generative Artificial Intelligence as Mobile App Developer (GenAI for MobileDev)	5
1.3 Generative Artificial Intelligence to Teach Computer Programming (GenAI Teach)	5
1.4 Mining GitHub Repositories to Extract Software Engineering Knowledge (GitMiner)	6
2 Amna Asif	8
2.1 Developing a Model to Estimate Disaster Damage Intensity Based on Environmental Factors.	8
2.2 Development of an Open-Source API Platform for AI-Powered Waste Sorting in a Circular Economy	9
2.3 Towards generating a Treatment plan for Protein-Losing Enteropathy (PLE) in Dogs using Large Language Models (LLMs).	9
2.4 Integrating Blockchain with IoT: Developing Open-Source APIs for Enhanced Security and Scalability	10
3 Marco Caminati	11
3.1 Formalising and Computing Variations of the Mutilated Chessboard	11
3.2 Realistic Models for Multimorbidity Problems	11
3.3 Interlocking Floor Tiling	12
3.4 Mining the Online Encyclopedia of Integer Sequences	12
3.5 Grammars and Machines Correspondences Beyond Context-free	13
4 Tianxiang Dai	14
4.1 Assessing the Current State of DNSSEC Live-signing	14
4.2 Assessing the Current State of Oblivious DNS	14
4.3 Automated Penetration Testing against 5G Core Network	15
4.4 Automated Penetration Testing against Content Management System	15
4.5 On-device Confidential Inference	16
5 Jiejun Hu-Bolz	17
5.1 Emergent Communication Strategies in Multi-Agent Reinforcement Learning	17
5.2 Simulating Social Dynamics in Multi-Agent Virtual Ecosystems	17
5.3 Agent-Based Resource Management in Blockchain	18
5.4 A Blockchain-Based Voting System	18
5.5 Designing Efficient Transaction Fee Mechanisms for Blockchain Networks	19
6 Christoph Jansen	20
6.1 Non-Parametric Statistical Tests for Non-Standard Data	20
6.2 Interval Probability: From Theory to Practice	20
6.3 Decision Making with Imprecise Probabilities	21
6.4 Rule-Based Dynamic Preference Elicitation	21
6.5 Social Choice in Multicriteria Benchmarking	22
7 Tobias Meggendorfer	23
7.1 The Next Level of DOMtutor	23
7.2 Designing an Interactive Educational Framework for LZSCC.111	23
7.3 Performance Evaluation of JBDD	24
7.4 Augmenting PET with Linear Equation Solving	24
7.5 Randomizing Probabilistic Models	25

8	Fabio Papacchini	26
8.1	Implementing an AI Player/Solver	26
8.2	“Simple” Explainable AI	26
8.3	Visualisation of Algorithms	27
8.4	Playing with Transitivity	27
8.5	Tableaux Prover for Modal Logic(s)	28
9	David Georg Reichelt	29
9.1	Benchmarking Document Reading Systems on Small-Scale Hardware	29
9.2	Examination of the Performance Monitoring Overhead of OpenTelemetry Using Python	29
9.3	Observability Energy Consumption Measurement	30
9.4	Source Code Node Change Detection for Scala / Kotlin /	30
9.5	Examination of Performance Change Detection Efficiency Using eBPF Probes	31
9.6	Performance Analysis of the Evolution of Jackson / Guava	32
10	Thomas Schmid	33
10.1	Parallelization of Genetic Algorithms	33
10.2	Development of a GUI Editor for the <i>Leipzig iLabSheet</i>	33
11	Guido Schmitz	34
11.1	Modern authorization for mobile and web applications	34
11.2	A Simple-to-Use Certification Authority	35
11.3	(Social) Web Single Sign-On in the Wild	36
12	Pascal Welke	38
12.1	How Many Graphs in Benchmark Datasets are “Amenable”?	38
12.2	Implement a Fast Convolution Layer in a Graph Neural Network Library	38
13	James Stovold	40
13.1	Extending DGCA-Grown Reservoirs to Liquid State Machines	40
13.2	Building Movement into Neural Cellular Automata	41
13.3	Cellular Mitosis in Neural Cellular Automata	42
13.4	Social Signalling in Neural Cellular Automata	43
13.5	Comparing Gliders in Different Game of Life Tilings	44
13.6	Developing the Lecturer Interface for Digital Systems App	45
13.7	Imperfections in Reaction–Diffusion Chemistry Simulations	46
14	James G. Wright	47
14.1	Swarm Identify Friend or Foe behaviour	47
14.2	Port Automation	47
14.3	Computer Network Description Language	48
14.4	‘Gadget bahn’ freight testbed	48

1 Aakash Ahmad

In the following, the listed projects provide a synopsis of potential research and development from a software development and engineering perspective. More specifically, the outlined project ideas aim to apply software engineering principles and practices to design and develop:

- Quantum software (theme: quantum software engineering),
- Software development bots/agents (theme: AI for SE), and
- Mining GitHub data (theme: software repository mining).

The proposed projects intend to produce publishable and applicable results in terms of prototype development and research reports that can advance state-of-the-art on software engineering. The general expectation is that interested students have knowledge about software design (preferably with UML or any design language) and skills in software development (source coding with Java, Python etc.). Knowledge of integrated development environments such as Microsoft Visual Studio. Each of the projects is summarised below. I would be happy to provide further elaborations or any clarifications that may be required.

I also welcome the ideas proposed by students in the broader or specific context of software engineering/software development and happy to supervise a student-proposed, jointly agreed project.

1.1 Quantum Algorithms as a Service (QAaaS)

Project type	implementation, research
Maximum instances running	1
Constraints/requirements	Preferred knowledge and skills: Unified Modeling Language, Integrated Development Environment (preferably Eclipse, VS-Code), Programming in Python or Java or C#
Industry linked	No

Description

Context: Quantum Computers (QCs) aim to disrupt the status-quo of computing - replacing traditional systems and platforms that are driven by digital circuits and modular applications - with hardware and software that operates on the principle of quantum mechanics. Currently, the main issues that impede mass-scale adoption of quantum systems are rooted in the fact that building, maintaining, and/or programming QCs is a complex and radically distinct engineering paradigm when compared to the challenges of classical computing and software engineering.

Quantum Software Engineering (QSE) has emerged as a quantum-specific genre of Software Engineering (SE) that enables engineers and developers who can leverage principles and practices of SE to design, develop, test, and maintain software-intensive systems for QCs. The aim of this project is to design and develop QC algorithm (e.g., Shor's algorithm or Groover's search) that can be executed as a service on a remotely located quantum computer.

Objectives and outcomes: The aim of the proposed project is to 'synergise the principles of QSE and practices of service-oriented computing to (i) develop a model that acts as a blue-print and (ii) implement a prototype that represents a proof-of-the-concept for Quantum Algorithm as a Service (QAaaS)'.

- Developing an architecture and prototype that can enable automation and empirical evaluation for potential collaboration with industry collaborators.
- Published results would help to develop research competencies in an emerging and strategically important research area on quantum software and service engineering.

Reading

- Ali, Shaukat, Tao Yue, and Rui Abreu. *When software engineering meets quantum computing*. Communications of the ACM, 65(4), 2022, pp. 84–88.
- Ahmad, A., Altamimi, A.B., and Aqib, J. *A reference architecture for quantum computing as a service*. Journal of King Saud University–Computer and Information Sciences, 36(6), p.102094, 2024.

1.2 Generative Artificial Intelligence as Mobile App Developer (GenAI for MobileDev)

Project type	implementation, research
Maximum instances running	1
Constraints/requirements	Preferred knowledge and skills: Unified Modeling Language, Basic knowledge of AI and Generative AI, Integrated Development Environment (Android Studio, VSCode), Programming in Python or Java or C#
Industry linked	No

Description

Context: Software engineering Bots (SEBots) represent conversational agents, recommender systems, or activity assistants that are driven by artificial intelligence - empowering the role of software engineers - to provide a certain degree of automation and/or inducing intelligence in SE process. One of the recently introduced examples of such SEBot is GitHub Copilot which is also referred to an “AI pair programmer” with capabilities to process natural language descriptions provided by software programmers to generate corresponding source code in several programming languages. In recent times, bots and AI systems are powered by Generative Artificial Intelligence (GenAI) as a type of AI driven by deep learning methods and algorithms to recognise, summarise, and generate content based on knowledge gained from massive datasets.

Objectives and outcomes: The aim of the proposed project is to 'understand the role of GenAI-based SEBots in software engineering to investigate the automation and artificial intelligence support in development of mobile applications'.

- Developing a prototype that can enable automation and empirical evaluation of SEBot in the software engineering process for mobile app development.
- Published results would help to develop research competencies in an emerging and strategically important research area on artificial intelligence for software engineering (AI for SE).

Reading

- Erlenov, L., de Oliveira Neto, F.G., Scandariato, R., and Leitner, P. *Current and future bots in software development*. In *Proceedings of the 2019 IEEE/ACM 1st International Workshop on Bots in Software Engineering (BotSE)*, pp. 7–11. IEEE, 2019.
- Peng, S., Kalliamvakou, E., Cihon, P., and Demirer, M. *The impact of AI on developer productivity: Evidence from GitHub Copilot*. arXiv preprint arXiv:2302.06590, 2023.

1.3 Generative Artificial Intelligence to Teach Computer Programming (GenAI Teach)

Project type	implementation, research
Maximum instances running	1
Constraints/requirements	Preferred knowledge and skills: Unified Modeling Language, Basic knowledge of AI and Generative AI, Integrated Development Environment (Android Studio, VSCode), Programming in Python or Java or C#
Industry linked	No

Description

Context: Generative Artificial Intelligence (GenAI) has demonstrated its potential as a data-driven recommender system that can generate content, facilitate academic research, and support learning activities etc. by interacting with humans through natural language prompts. Education 4.0, a recent concept, supports teaching and learning support activities that focus on essential skills including IT proficiency, digital literacy, and professional values — key competencies identified by the World Economic Forum in its ‘Future of Jobs Report 2023’. There is a growing interest both from research as well as from an industrial perspective to develop solutions that utilise GenAI tools such as ChatGPT and Google Gemini to assist teaching and learning support. GenAI can act as a virtual tutor to enable learners to understand the fundamentals of computer programming.

Objectives and outcomes: The aim of the proposed project is to utilise existing GenAI tools that can act as virtual tutors (learning recommender systems) to help learners to gain knowledge and skills of introductory level programming.

- Research and design a solution that can enable the implementation of GenAI tool to act as a teaching assistant
- Develop a prototype, as a proof-of-the-concepts to evaluate the designed solution

Reading

- Dickey, E., Bejarano, A., and Garg, C. *AI-Lab: A Framework for Introducing Generative Artificial Intelligence Tools in Computer Programming Courses*. SN Computer Science, 5(6), 2024, Article 720.
- Waseem, M., Das, T., Ahmad, A., Liang, P., Fehmideh, M., and Mikkonen, T. *ChatGPT as a software development bot: A project-based study*. In *Proceedings of Evaluation of Novel Approaches to Software Engineering (ENASE)*, 2024.

1.4 Mining GitHub Repositories to Extract Software Engineering Knowledge (GitMiner)

Project type	implementation, research
Maximum instances running	1
Constraints/requirements	Basic knowledge of Data Mining, GitHub, Integrated Development Environment (Android Studio, VSCode), Programming in Python or Java or C#
Industry linked	No

Description

Context: GitHub as a social coding platform enables software developers or practitioners to deploy their software development projects, documents and other artifacts as public or privately available repositories. GitHub as a project repository contains invaluable software engineering knowledge in terms of source code, design documents, testcases etc. that can be investigated to understand and predict future course of actions. A systematic extraction and semi-automated analysis of GitHub data can be challenging, however; it can provide useful knowledge to developers and project managers. There is a need for solutions that can allow non-technical stakeholders of a software project to extract and analyse GitHub data in a seamless way.

Objectives and outcomes: The aim of the proposed project is to develop a solution that supports an automated extraction of software engineering knowledge such as project issues, their causes and software implementation strategies. The main objectives of this project are:

Reading

- Gousios, G., and Spinellis, D. *Mining software engineering data from GitHub*. In *Proceedings of the 2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C)*, pp. 501–502. IEEE, 2017.

- Cosentino, V., Luis, J., and Cabot, J. *Findings from GitHub: methods, datasets and limitations*. In *Proceedings of the 13th International Conference on Mining Software Repositories*, pp. 137–141, 2016.

2 Amna Asif

I am offering a variety of projects in applied machine learning, deep learning, and transformer-based approaches. However, topics are not limited to these areas; if you have your own ideas, particularly in human-computer interaction, machine learning, deep learning, transformers, and other areas beneficial for the community, you are welcome to propose them. Feel free to approach me to discuss potential project ideas.

2.1 Developing a Model to Estimate Disaster Damage Intensity Based on Environmental Factors.

Project type	Research (with Implementation for Model Development)
Maximum instances running	X, where $X \geq 1$, depending on data processing/model deployment scale.
Constraints/requirements	Colab, Mid-to-high-performance CPU/GPU (for training ML models), Python or R, Python, Pandas/NumPy (data analysis), Scikit-learn/TensorFlow (modeling), GIS tools (e.g., QGIS for geospatial factors), this project requires understanding of Machine Learning, ...
Industry linked	No

Description

This project covers all phases of disaster management: the calm before the disaster, during the disaster, the peak, the plateau, the decline, and the return to normal. One of the key factors in determining the cause and intensity of damage caused by a disaster is the intensity of the disaster itself. However, there are many other factors that contribute to the damage caused by disasters, such as climate change. In this project, we aim to explore environmental causes that may increase disaster damage intensity. The findings of this project will help disaster management agencies prepare to cope with such events more effectively.

Objectives of this project include:

1. Exploring disaster damage datasets that can be effectively utilized for damage assessments.
2. Conducting research and literature review to identify environmental factors that hinder disaster damage prevention.
3. Developing a model to predict disaster damage along with environmental factors to explore the causes of disaster damage.

Reading

- Khatoon, S., Asif, A., Hasan, M. M., & Alshamari, M. (2022). Social media-based intelligence for disaster response and management in smart cities. In *Artificial Intelligence, Machine Learning, and Optimization Tools for Smart Cities: Designing for Sustainability* (pp. 211-235). Cham: Springer International Publishing.
- Wu, Xianhua, et al. "A New Economic Loss Assessment System for Urban Severe Rainfall and Flooding Disasters Based on Big Data Fusion." In *Economic Impacts and Emergency Management of Disasters in China* (2021): 259-287.
- Mazhin, Sadegh Ahmadi, et al. "Worldwide Disaster Loss and Damage Databases: A Systematic Review." *Journal of Education and Health Promotion* 10.1 (2021).
- Cappelli, Federica, Valeria Costantini, and Davide Consoli. "The Trap of Climate Change-Induced 'Natural' Disasters and Inequality." *Global Environmental Change* 70 (2021): 102329.

2.2 Development of an Open-Source API Platform for AI-Powered Waste Sorting in a Circular Economy

Project type	Implementation (with elements of Research)
Maximum instances running	$X > 1$ (scalable based on server/cloud deployment)
Constraints/requirements	Python, TensorFlow/PyTorch, Rust, Machine Learning (waste classification), API development, GitHub/GitLab repository.
Industry linked	No

Description

Solid waste management is a critical area that requires innovative solutions, especially within the framework of the circular economy, a trending and essential topic in sustainability. Integrating artificial intelligence (AI) can significantly enhance waste management processes by enabling efficient sorting and recycling. While numerous public datasets exist for training machine learning models in waste classification, there is a lack of open-source, AI-powered APIs that can be readily utilized for developing robotic or hardware-based applications. This project aims to bridge this gap by creating an open-source API platform that provides accessible waste-sorting solutions. The platform will enable students, hobbyists, and academic researchers to integrate AI-driven waste classification into their projects simply by calling predefined APIs. By offering modular and scalable APIs, this project will support the rapid development of end-to-end waste management solutions, fostering innovation in sustainable technology. The proposed platform will empower developers to build hardware and robotic applications without extensive AI expertise, accelerating the adoption of intelligent waste management systems in line with circular economy principles.

Reading

- [API4AI: How AI-Powered APIs Improve Waste Management with Image Recognition](#)
- Yi, Chan Jia, and Chong Fong Kim. "AI-Powered Waste Classification Using Convolutional Neural Networks (CNNs)." *International Journal of Advanced Computer Science & Applications* 15.10 (2024).
- Aberger, Julian, et al. "Prototype of AI-powered assistance system for digitalisation of manual waste sorting." *Waste Management* 194 (2025): 366-378.
- Olawade, David B., et al. "Smart waste management: A paradigm shift enabled by artificial intelligence." *Waste Management Bulletin* (2024).
- [AgaMiko: Waste Datasets Review \(GitHub\)](#)

2.3 Towards generating a Treatment plan for Protein-Losing Enteropathy (PLE) in Dogs using Large Language Models (LLMs).

Project type	Research, implementation, data preparation, Collaborative (Hospital and Veterinary Medicine partnership)
Maximum instances running	$X 1$ (Can scale based on API/web app usage)
Constraints/requirements	Python, TensorFlow, Huggingface Transformers
Industry linked	Yes

Description

Protein-losing enteropathy (PLE) involves conditions that cause plasma proteins to leak from the bloodstream into the gastrointestinal (GI) tract due to GI diseases, cardiac issues, or lymphatic system problems. Normally, these proteins are broken down and reabsorbed, but disease can cause excessive protein loss, leading to hypoproteinemia and potentially severe fluid leakage into the abdomen, chest, or limbs. The treatment for PLE typically relies on lab results and involves a special diet low in fat with highly digestible carbohydrates and either hydrolyzed or novel

proteins. Historical data from treated dogs will be leveraged to develop diet planners using large language models (LLMs). The first step will be to prepare this data and train the model. The end product will be an application designed to generate specific diet plans for affected dogs. This project is in collaboration with a hospital and the Faculty of Veterinary Medicine in Leipzig.

Reading

- Protein-Losing Enteropathy (PLE) in Dogs: <https://vcahospitals.com/know-your-pet/proteinlosing-enteropathy-ple-in-dogs>
- Nagata, Noriyuki, et al. "Clinical characteristics of dogs with food-responsive protein-losing enteropathy." *Journal of Veterinary Internal Medicine* 34.2 (2020): 659-668.
- Rudinsky, A. J., et al. "Dietary management of presumptive protein-losing enteropathy in Yorkshire terriers." *Journal of Small Animal Practice* 58.2 (2017): 103-108.

2.4 Integrating Blockchain with IoT: Developing Open-Source APIs for Enhanced Security and Scalability

Project type	The project involves building APIs (implementation) and testing their effectiveness in real-world IoT scenarios (experimental).
Maximum instances running	Use-case simulations (e.g., smart home, supply chain).
Constraints/requirements	IoT devices such as Raspberry Pi, GitHub/GitLab repository, Blockchain framework.
Industry linked	No

Description

The Internet of Things (IoT) is increasingly essential in our daily lives, improving our interactions with each other and our environment due to its versatility. It enables small businesses, initiatives, and hobbyists to create IoT-based solutions for everyday challenges swiftly. Yet, as IoT networks expand in complexity and scale with more connected devices, issues like security, authenticity, reliability, and scalability become prevalent. Blockchain technology offers potential solutions to these problems with its secure, transparent, reliable, and traceable features, ideally suited to enhance IoT systems and support their growth. Despite the suitability of IoT devices for blockchain-enhanced security, readily available blockchain-based APIs for IoT applications are scarce. This project aims to develop such APIs and release them as open-source for developers to utilize in their projects.

Reading

- Al Sadawi, Alia, Mohamed S. Hassan, and Malick Ndiaye. "A Survey on the Integration of Blockchain with IoT to Enhance Performance and Eliminate Challenges." *IEEE Access* 9 (2021): 54478-54497.
- Atlam, Hany F., et al. "Blockchain with Internet of Things: Benefits, Challenges, and Future Directions." *International Journal of Intelligent Systems and Applications* 10.6 (2018): 40-48.

3 Marco Caminati

3.1 Formalising and Computing Variations of the Mutilated Chessboard

Project type	Implementation; Verification
Maximum instances running	1
Constraints/requirements	logic; programming skills
Industry linked	No

Description

A chessboard being an 8x8 array of squares, it is obviously possible to cover it with several 2x1 tiles. The mutilated chess problem asks whether this is still possible upon removing two cells from opposite corners of the chessboard:

The answer to the riddle is no: it is not possible, and the argument is quite short and neat.

This project will consider variations of this classical problems, and implement the corresponding proofs (either of existence of a solution or of impossibility) using a proof assistant such as Isabelle/HOL (which has been used to provide a proof in the classical case, see references). In case of possibility, it will also implement the algorithm producing a solution. Possible variations are:

- what happens if we remove other pairs of squares?
- What happens if we remove larger amounts of squares?
- What happens if we wish to cover with tiles of other shapes than 2x1?

The ones above are suggestions: the project is open to constraints and variations making the problem interesting.

Reading

- https://en.wikipedia.org/wiki/Mutilated_chessboard_problem
- <https://www.cl.cam.ac.uk/~lp15/papers/Deduction/lecture4.pdf>

3.2 Realistic Models for Multimorbidity Problems

Project type	Implementation
Maximum instances running	1
Constraints/requirements	programming skills; modelling skills
Industry linked	No

Description

In the UK, the National Institute for Health and Care Excellence (NICE) publishes guidelines for the treatment of many diseases including common chronic conditions (diabetes, cancer, cardiovascular diseases, chronic respiratory diseases, etc), as well as guidelines for health protection and lifestyle recommendations. In 2015, 54.0% of British people aged over 65 had two or more conditions (multi-morbidity). By 2035 this is predicted to have risen to 67.8%. One of the problems for patients with multimorbidity is that they are subject to multiple treatment pathways and the risk of undetected adverse reactions to prescribed medications. The number of possible combinations and subsequent adverse reactions is too big to be handled manually. Therefore, a number of computational approaches have been devised to help clinicians deciding on the execution of the guidelines for multimorbid patients.

However, most of these approaches dramatically simplify the problem through several assumptions.

This project will tackle the challenge of designing more realistic computational models of the problem, by taking into account, for example:

- the temporal separation between potentially conflicting treatments;

- the existence of many-to-many conflicts: for example, two medications could have severe interaction, but when combined with a third one, the interaction is mitigated.

Reading

- <https://www.therapeutics.scot.nhs.uk/wp-content/uploads/2018/04/Polypharmacy-Guidance-2018.pdf>
- https://doi.org/10.1007/978-3-319-33693-0_24
- <https://doi.org/10.1016/j.scico.2019.07.002>

3.3 Interlocking Floor Tiling

Project type	implementation
Maximum instances running	1
Constraints/requirements	programming skills; visualisation
Industry linked	No

Description

Modern vinyl click and laminate floor tiles and planks are prevalent today, in that they offer durability and ease of installation, and are accessible to diy-ers, as well as to professionals. When laying them, it is important to know in advance how to cut the tiles to cover the room at hand in order to minimise the costs, the tiles used or the amount of cuts. This is complicated by the fact that the cuts must preserve the coupling between the joining edges: only some pairs of edges can be adjacent to exploit the click joint mechanism. Additionally, there are constraints on the minimum length and height of a cut tile, given by the location of pipes traversing the floor, the fact that planks in neighbouring rows should be staggered for structural and aesthetic reasons, etc. Existing floor planning software available to the industry and to the traders, such as ProtoTILER and Callidus, provide minimisation algorithms not taking into account all the constraints above. This project will produce a tool to allow the user to specify the shape and size of the room and of the tiles, obtaining a tiling which satisfies the wanted constraints (e.g., minimal number of used tiles or of cuts). It will require to familiarise with the existing literature about the corresponding theoretical problems: rectilinear polygon partitioning and convex regular euclidean tiling. Using this information, a suitable constraint programming tool will be chosen, which can then be used to model the problem.

Reading

- <https://people.csail.mit.edu/rivest/pubs/LPRS82.pdf>
- <https://doi.org/10.1016%2F978-044482537-7%2F50012-7>
- https://doi.org/10.1007%2F978-3-642-11409-0_1

3.4 Mining the Online Encyclopedia of Integer Sequences

Project type	implementation, experimental
Maximum instances running	1
Constraints/requirements	math skills, database skills
Industry linked	No

Description

The Online Encyclopedia of Integer Sequences (OEIS) is an important resource in many mathematical and computer science branches, including combinatorics, computational algebra, discrete mathematics, and experimental

mathematics. Given its big size, it is inherently interesting to mine, both internally and with respect to external sources (such as Google Scholar), to find new connections between different mathematical objects. However, recent attempts to do that are stifled by the limitations on how the OEIS can be accessed. There is a somehow intelligent program, called Superseeker, to find matches between a given sequence and OEIS modulo transformations. However, Superseeker is run remotely, imposing one query per hour, clearly making it unsuitable for mining. This project will start by familiarising with the source code of Superseeker, and with the combination of languages and tools it uses in the background, so to be able to run it locally. The second step will be to run mining experiments to find interesting matches within the OEIS and across different sources. Since there will be too many possible potential matches to explore, some heuristics will be needed to prune their number. For external matches, a way of dealing with Google Scholar's inability to perform arithmetic manipulation on its queries will need a work-around. Finally, some found matches could be trivial or not interesting (e.g., between sequences which are already known to be mathematically or otherwise linked), which will require a post-processing. Along the way, the student will have a chance to understand Superseeker and how it can be extended to increase the number of interesting matches found.

Reading

- <https://github.com/sidneycadot/oeis>
- <https://github.com/pjt33/oeis-superseeker>
- https://oeis.org/ol_source.txt

3.5 Grammars and Machines Correspondences Beyond Context-free

Project type	implementation, experimental
Maximum instances running	1
Constraints/requirements	math skills, implementation skills
Industry linked	No

Description

The Chomsky Hierarchy classifies formal grammars according to their generative power and is therefore a fundamental notion to study language design. Through one-to-one correspondences between formal grammars and other computational paradigms such as state machines, its influence is passed on and generalised to other domains, such as that of computability. These correspondences provide implementable constructions to pass from grammars to machines and vice-versa, offering both a theoretical and practical tool for the cross fertilisation of these diverse domains. Over the years, the Chomsky Hierarchy has been enriched with grammars of intermediate power, such as mildly context-free grammars, indexed grammars, context-free tree grammars, etc... Some of them have been shown to be linked by correspondences, between each other and with machines, like previous ones. However, the correspondence remains often on paper, with no actual implementation. This project aims at creating some of these implementations, so to provide researchers with practical tools to explore and test. For example, by providing algorithms to pass between indexed grammars and nested stack automata (Aho), or between macro grammars and indexed grammars (Fischer).

Reading

- <https://doi.org/10.1007/978-3-642-14846-0>
- <https://doi.org/10.1145/321526.321529>
- <https://doi.org/10.1109/SWAT.1968.12>

4 Tianxiang Dai

All my projects are **AI-friendly**, if and only if, you explain it clearly why and how you use AIs.

4.1 Assessing the Current State of DNSSEC Live-signing

Project type	Research and Implementation
Maximum instances running	1
Constraints/requirements	Linux, DNS
Industry linked	No

Description

DNSSEC is a security extension for DNS that adds digital signatures to DNS responses, ensuring their authenticity and integrity. Typically, these signatures are generated offline and then served by DNS servers. However, some DNS hosting services (such as deSEC and Cloudflare) and DNS software (such as PowerDNS and CoreDNS) also support live-signing, where DNS records are signed on the fly. This project aims to assess the current state of DNSSEC live-signing by investigating how it can be identified on the Internet, estimating how many domains employ it, evaluating its impact on server performance, and exploring related aspects.

Reading

1. Hoffman, P., 2023. RFC 9364: DNS Security Extensions (DNSSEC).
2. <https://desec.io>
3. <https://blog.cloudflare.com/dnssec-complexities-and-considerations/>
4. <https://doc.powerdns.com/authoritative/dnssec/modes-of-operation.html>
5. <https://coredns.io/plugins/dnssec/>

4.2 Assessing the Current State of Oblivious DNS

Project type	Research and Implementation
Maximum instances running	1
Constraints/requirements	Linux, DNS
Industry linked	No

Description

Oblivious DNS (ODNS) is a privacy-enhancing technology designed to decouple client identities from their DNS queries by encrypting requests and routing them through proxies, ensuring that no single server can link a user's IP address with the contents of their DNS queries. This project aims to assess the current state of Oblivious DNS by evaluating its deployment, adoption, and effectiveness in protecting user privacy across the Internet, as well as examining its impact on DNS performance and compatibility with existing infrastructure.

Reading

1. Schmitt, P., Edmundson, A., Mankin, A. and Feamster, N., 2019. Oblivious DNS: Practical Privacy for DNS Queries. Proceedings on Privacy Enhancing Technologies, 2, pp.228-244.
2. Singanamalla, S., Chunhaphanya, S., Hoyland, J., Vavruša, M., Verma, T., Wu, P., Fayed, M., Heimerl, K., Sullivan, N. and Wood, C., 2021. Oblivious DNS over HTTPS (ODoH): A Practical Privacy Enhancement to DNS. Proceedings on Privacy Enhancing Technologies, 4, pp.575-592.

3. Kinnear, E., McManus, P., Pauly, T., Verma, T. and Wood, C.A., 2022. RFC 9230: Oblivious DNS over HTTPS.
4. Trevisan, M., Drago, I., Schmitt, P. and Bronzino, F., 2023, March. Measuring the performance of icloud private relay. In International Conference on Passive and Active Network Measurement (pp. 3-17). Cham: Springer Nature Switzerland.

4.3 Automated Penetration Testing against 5G Core Network

Project type	Research and Implementation
Maximum instances running	1
Constraints/requirements	Linux, Docker, C/C++, Python
Industry linked	No

Description

The 5G Core Network serves as the central infrastructure for modern 5G cellular systems. Through its integration of Software-Defined Networking (SDN) and Network Functions Virtualization (NFV), it becomes inherently dependent on distributed network services. Emerging trends toward public cloud deployment of 5G Core Networks have introduced new security challenges, particularly regarding vulnerability exposure and attack surface expansion. This project involves developing an automated penetration testing framework specifically for open-source 5G Core implementations, with OpenAirInterface as the primary target platform.

Reading

1. GIAMBARTOLOMEI, F., Penetration testing applied to 5G Core Network.
2. F. Giambartolomei, M. Barceló, A. Brighente, A. Urbeta and M. Conti, "Penetration Testing of 5G Core Network Web Technologies," ICC 2024 - IEEE International Conference on Communications, Denver, CO, USA, 2024, pp. 702-707.
3. Da Canal, S., Mancini, F. and Bianchi, G., 2024, September. 5GRecon: Automating 5G Network Function Recognition and Misconfiguration Troubleshooting. In 2024 13th IFIP/IEEE International Conference on Performance Evaluation and Modeling in Wired and Wireless Networks (PEMWN) (pp. 1-6). IEEE.

4.4 Automated Penetration Testing against Content Management System

Project type	Research and Implementation
Maximum instances running	1
Constraints/requirements	Linux, Playwright, Javascript, Python
Industry linked	No

Description

Content Management Systems (CMS) like WordPress are widely used but increasingly targeted by web-based attacks due to their complexity and reliance on third-party plugins. This project aims to develop an automated penetration testing framework for WordPress, focusing on identifying vulnerabilities in core features and plugins, assessing modern attack vectors, and evaluating the effectiveness of existing security measures to provide a comprehensive overview of its security posture.

Reading

1. Jagamogan, R.S., Ismail, S.A., Hafizah, N. and Abas, H.H., 2021, October. A review: Penetration testing approaches on content management system (cms). In 2021 7th International Conference on Research and Innovation in Information Systems (ICRIIS) (pp. 1-6). IEEE.

2. Auricchio, N., Cappuccio, A., Caturano, F., Perrone, G. and Romano, S.P., 2022. An automated approach to web offensive security. *Computer Communications*, 195, pp.248-261.
3. Altulaihan, E.A., Alismail, A. and Frikha, M., 2023. A survey on web application penetration testing. *Electronics*, 12(5), p.1229.
4. Gupta, D., 2023. A Critical Review of WordPress Security Scanning Tools and the Development of a Next-Generation Solution.

4.5 On-device Confidential Inference

Project type	Implementation
Maximum instances running	1
Constraints/requirements	Android, C/C++, Kotlin
Industry linked	No

Description

This project develops a confidential inference framework for Android smartphones using Trusty TEE, leveraging hardware-enforced isolation to securely run machine learning models and protect user data directly on the device. By executing all inference computations within Trusty's secure environment, the framework ensures both data and model confidentiality without relying on external servers or accelerators, addressing privacy and security needs for sensitive mobile applications while considering the performance and memory constraints typical of TEE-based deployments.

Reading

1. <https://source.android.com/docs/security/features/trusty>
2. Islam, M.S., Zamani, M., Kim, C.H., Khan, L. and Hamlen, K.W., 2023, April. Confidential execution of deep learning inference at the untrusted edge with ARM TrustZone. In Proceedings of the Thirteenth ACM Conference on Data and Application Security and Privacy (pp. 153-164).
3. Moon, M., Kim, M., Jung, J. and Song, D., 2025. ASGARD: Protecting On-Device Deep Neural Networks with Virtualization-Based Trusted Execution Environments. Proceedings 2025 Network and Distributed System Security Symposium.

5 Jiejun Hu-Bolz

5.1 Emergent Communication Strategies in Multi-Agent Reinforcement Learning

Project type	Research, Implementation
Maximum instances running	2
Constraints/requirements	Game theory; Programming; Modelling
Industry linked	No

Description

Communication plays a crucial role in enabling effective coordination and collaboration in multi-agent systems, particularly in dynamic or partially observable environments. This project focuses on designing and evaluating communication mechanisms within Multi-Agent Reinforcement Learning (MARL) frameworks. Students will investigate how agents can learn to share information to improve group performance. One promising approach to explore is Intention Sharing (IS), where agents simulate future action trajectories based on learned models of the environment and other agents. These simulated trajectories can then be compressed into compact intention messages using techniques such as attention mechanisms. These messages allow agents to coordinate by sharing their plans rather than just immediate observations. Students will customize one or more PettingZoo environments to create communication-critical scenarios where effective information exchange is essential for success. Game-theoretic analysis (e.g., Nash equilibrium, signaling games) will be used to study the impact of different communication strategies on coordination, competition, and cooperation among agents.

Reading

- Kim, Woojun, Jongeui Park, and Youngchul Sung. "Communication in multi-agent reinforcement learning: Intention sharing." International conference on learning representations. 2020.
- Lowe, Ryan, et al. "On the Pitfalls of Measuring Emergent Communication." Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems. 2019.
- Terry, Jordan, et al. "Pettingzoo: Gym for multi-agent reinforcement learning." Advances in Neural Information Processing Systems 34 (2021): 15032-15043.

5.2 Simulating Social Dynamics in Multi-Agent Virtual Ecosystems

Project type	Implementation, research
Maximum instances running	2
Constraints/requirements	Programming; modelling
Industry linked	No

Description

This project investigates how simple individual rules can lead to complex group behaviours in a virtual ecosystem of digital creatures. Each creature is autonomous, driven by basic needs such as food, social interaction, and shelter. Over time, agents can exhibit emergent behaviors such as cooperation, competition, migration, reproduction, and social organization. Students will design and implement a multi-agent environment where agents interact with each other and with the environment. Each agent will learn using reinforcement learning methods, potentially enhanced with communication mechanisms and hierarchical decision making. The ecosystem will naturally evolve as agents adapt their strategies to survive and thrive in a dynamic world. The environment can be developed using Pygame or PettingZoo.

Reading

- Ndousse, Kamal K., et al. "Emergent social learning via multi-agent reinforcement learning." International conference on machine learning. PMLR, 2021.
- Kölle, Michael, et al. "Aquarium: A Comprehensive Framework for Exploring Predator-Prey Dynamics through Multi-Agent Reinforcement Learning Algorithms." arXiv preprint arXiv:2401.07056 (2024).

5.3 Agent-Based Resource Management in Blockchain

Project type	Research, simulation
Maximum instances running	1
Constraints/requirements	Agent-based modelling, dynamic systems
Industry linked	No

Description

This project explores resource allocation within a blockchain network using an agent-based model (ABM). We will design software agents representing blockchain participants (miners, validators) competing for computational resources. Agents will require resources to perform tasks like mining and validation. The network will employ a resource management protocol where agents negotiate access and share information. By analysing agent interactions, we aim to understand how different resource allocation strategies impact:

- Network efficiency: How effectively are resources utilised for transaction processing?
- Security: Does the strategy maintain a secure and decentralized network?
- Fairness: Do all participants have a fair chance to access resources?

This project will contribute to the development of efficient and fair resource management mechanisms for blockchain-based systems.

Reading

- Hu, Jiejun, et al. "A dynamic service trading in a dlt-assisted industrial iot marketplace." IEEE Transactions on Network and Service Management 19.4 (2022): 4691-4705.

5.4 A Blockchain-Based Voting System

Project type	Implementation
Maximum instances running	1
Constraints/requirements	Programming
Industry linked	No

Description

Blockchain is known for its decentralisation, transparency, security, immutability, and traceability in a network. It may offer new opportunities for the future of democracy. In this project, a blockchain-assisted voting system will be designed and implemented for conducting secure and transparent elections.

Specifically, a voting DApp (decentralised application) should be designed in Hyperledger Fabric. DApp should have several key functions, such as

- allowing voters to create an account and vote;
- rejecting duplicated votes from the same voter;
- protecting the poll from third-party breaching;
- finalising the poll and providing the statistics.

It is ideal to familiarise yourself with Hyperledger Fabric before the designing process. There is no specific requirement of language when implementing DApp. Preferably, a user interface is implemented.

Reading

- <https://www.hyperledger.org>
- Huang, Jun, et al. "The application of the blockchain technology in voting systems: A review." ACM Computing Surveys (CSUR) 54.3 (2021): 1-28.

5.5 Designing Efficient Transaction Fee Mechanisms for Blockchain Networks

Project type	Research, implementation
Maximum instances running	1
Constraints/requirements	Modelling; Programming
Industry linked	No

Description

Transaction fees are an integral component of blockchain systems, playing a crucial role in maintaining the network's functionality, security, and economic sustainability. In this project, you are expected to design novel fee structures and pricing algorithms to balance network security and transaction throughput, incentivise miners' participation while ensuring fair access for users, and enhance fee predictability and transparency.

This project aims to

- Explore the basics of transaction fees in blockchain networks and their importance in network security and efficiency.
- Design incentive mechanisms to ensure fair participation of miners and users in the transaction validation process.
- Evaluate the proposed transaction fee mechanism using simulated blockchain environments.

Reading

- Li, Juanjuan, et al. "Transaction queuing game in bitcoin blockchain." 2018 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2018.
- Bahrani, Maryam, Pranav Garimidi, and Tim Roughgarden. "Transaction fee mechanism design with active block producers." arXiv preprint arXiv:2307.01686 (2023).

6 Christoph Jansen

In general, I am interested in topics related to statistics, machine learning, decision theory and optimization theory. The projects listed below are sought of as starting point that can be adapted by the student based on initial discussions.

6.1 Non-Parametric Statistical Tests for Non-Standard Data

Project type	theoretical, research
Maximum instances running	1
Constraints/requirements	Basic order theory, basic statistics, linear programming, R
Industry linked	No

Description

Non-standard data, like multidimensional structures with differently scaled dimensions, are pretty common in statistics and machine learning. However, despite their immense practical significance, there are often no suitable parametric statistical models available for such data types. The alternative here is often non-parametric methods, such as permutation tests. The aim of the project is to implement a non-parametric statistical test (known from the literature) for a specific type of non-standard data, so-called weakly structured variables. Subsequently, the power of this test will be systematically investigated through appropriate simulation studies. If progress allows, it can also be investigated whether the power can be improved through appropriate regularization strategies.

Reading

- John Pratt and Jean Gibbons (1981): *Concepts of Nonparametric Theory*. Springer.
- Christoph Jansen, Georg Schollmeyer, Hannah Blocher, Julian Rodemann and Thomas Augustin (2023): Robust statistical comparison of random variables with locally varying scale of measurement. In: *Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence (UAI 2023)*.

6.2 Interval Probability: From Theory to Practice

Project type	theoretical, implementation
Maximum instances running	1
Constraints/requirements	basic probability theory, visualization, linear programming
Industry linked	No

Description

The theory of interval probabilities is a fundamental generalization of Kolmogorov's axiomatic concept of probability to interval-valued uncertainty assessments. While the theory was very popular in the early 1990s, for example in connection with expert systems, it received relatively little attention in the 2000s. Only in recent years, with the meteoric rise of machine learning, has it experienced a renaissance as a natural language for describing uncertainty quantification. The initial aim of the project is to provide a modern and compact presentation of the theory of interval probabilities. Subsequently, algorithms and visualization techniques that have only been described theoretically in the literature will be implemented in order to improve the accessibility of this interesting theory.

Reading

- Kurt Weichselberger (2000): The theory of interval-probability as a unifying concept for uncertainty. *International Journal of Approximate Reasoning*, vol. 24, pp. 149–170.
- Thomas Augustin, Frank Coolen, Gert De Cooman and Matthias Troffaes, editors (2014): *Introduction to imprecise probabilities*. John Wiley & Sons.

6.3 Decision Making with Imprecise Probabilities

Project type	theoretical, research, implementation
Maximum instances running	1
Constraints/requirements	basic probability theory, linear programming, order theory
Industry linked	No

Description

Decision theory is the study of rational decision-making under uncertainty. For many decades, decision theory was dominated by the paradigm of expected utility maximization, as manifested in the groundbreaking work of de Finetti, von Neumann and Morgenstern, and Savage (among others). However, since the advent of the first expert systems, this paradigm has been increasingly questioned: What if the uncertainty of the decision-making situation cannot be modeled using classical probability? Here, imprecise probabilities have increasingly established themselves as a more reliable alternative, which can capture not only stochastic uncertainty but also the so-called ambiguity of the situation. After a brief introduction to classical decision theory and credal set theory, the project aims to compile the most important decision criteria under generalized uncertainty. Subsequently, linear programs for optimal decision-making based on a selection of these criteria will be formulated and implemented. If the progress of the project allows, a relatively new criterion and its connection to uncertainty quantification in robust statistics will be investigated.

Reading

- Matthias Troffaes (2007): Decision making under uncertainty using imprecise probabilities. *International Journal of Approximate Reasoning*, vol. 45(1), pp. 17-29.
- Christoph Jansen, Georg Schollmeyer and Thomas Augustin (2022): Quantifying Degrees of E-Admissibility in Decision Making with Imprecise Probabilities. In: *Reflections on the Foundations of Probability and Statistics: Essays in Honor of Teddy Seidenfeld*. Theory and Decision Library A, vol 54. Springer.

6.4 Rule-Based Dynamic Preference Elicitation

Project type	research, implementation
Maximum instances running	1
Constraints/requirements	order theory, optimization theory, basic probability theory
Industry linked	No

Description

The term *preference elicitation* refers to the systematic retrieval of initially unknown (human) preferences, often achieved by successively (and adaptively) presenting queries to the agent under consideration. In the context of decision-making under uncertainty, this raises the question of when the elicitation process can be stopped, since the information elicited so far is already sufficient to make an informed decision based on a pre-formulated optimality criterion. Furthermore, the question arises as to how to select the most efficient query sequences in order to arrive at an optimal decision with regard to a predetermined criterion with as few queries as possible. The literature distinguishes between two different types of efficiency: (I) avoiding redundant information requests and (II) separating the decision boundaries between the various alternative courses of action as quickly as possible. While (I) is often formalized using structural rules based on the agent's preferences (for example, in the case of transitive preferences, after querying $a > b$ and $b > c$, the query between a and c is redundant), (II) often relies on heuristics. The aim of this project is to first review common techniques for preference elicitation from literature and systematize them in a meaningful way. Afterwards, dynamic preference elicitation approaches with rules different than transitivity should be developed, mathematically formulated and implemented. If progress allows, simulation-based comparison studies of the different approaches should be performed.

Reading

- Christoph Jansen, Georg Schollmeyer and Thomas Augustin (2018): Concepts for decision making under severe uncertainty with partial ordinal and partial cardinal preferences. *International Journal of Approximate Reasoning*, 98: 112 - 131.
- Christoph Jansen, Hannah Blocher, Thomas Augustin and Georg Schollmeyer (2022): Information efficient learning of complexly structured preferences: Elicitation procedures and their application to decision making under uncertainty. *International Journal of Approximate Reasoning*, 144 : 69 - 91.

6.5 Social Choice in Multicriteria Benchmarking

Project type	theory, research, implementation
Maximum instances running	1
Constraints/requirements	order theory, basic probability theory, basic machine learning
Industry linked	No

Description

Stanford Encyclopedia of Philosophy (SEP) states: *Social choice theory is the study of collective decision procedures and mechanisms. It is not a single theory, but a cluster of models and results concerning the aggregation of individual inputs (e.g., votes, preferences, judgments, welfare) into collective outputs (e.g., collective decisions, preferences, judgments, welfare).* According to this very general definition, certain areas of machine learning, such as ensemble methods or multi-criteria benchmarking, fit very naturally into this theory. This also allows results from social choice theory to be directly transferred to problems in machine learning. Very recently this formal connection has been used to adapt Arrow's famous impossibility result to multicriteria benchmarking studies, thereby revealing theoretical limitations which have been undiscovered before. On the other hand, the 'empirical relevance' of these limitations was questioned, arguing that these impossibilities would rather be theoretical artefacts than problems of practical relevance. The aim of this project is to first review the state-of-the-art of this young yet rapidly growing field. Afterwards, meaningful simulation studies should be designed to contribute some new insights to the 'artefact vs. relevant' debate mentioned above. If progress allows, the candidate can think of possibilities to transfer other (ideally positive) social choice results to the multicriteria benchmarking setting.

Reading

- Guanhua Zhang and Moritz Hardt (2024): Inherent Trade-Offs between Diversity and Stability in Multi-Task Benchmarks. In: *Proceeding of the Forty-first International Conference on Machine Learning*. Online: <https://openreview.net/forum?id=fwxnHViGNj>
- Moritz Hardt's keynote *The emerging science of benchmarks* at ICLR 2024: <https://iclr.cc/virtual/2024/invited-talk/21799>
- SEP entry on Social Choice: <https://plato.stanford.edu/entries/social-choice/>

7 Tobias Meggendorfer

My projects can be categorized into two kinds:

First come several teaching-related topics, mainly focussed on improving DOMjudge (and my underlying script-library DOMtutor) as well as directly designing didactic exercises. These are more implementation oriented; there is less theory to read before getting started. Nevertheless, in many cases a discussion of the usefulness in teaching (e.g. by reviewing literature on effective teaching methods) or a small scale survey (e.g. with other members of the cohort) to evaluate the positive effects is expected.

Second come projects related to current state-of-the-art research tools. These require more reading up-front and understanding of algorithms & data structures, but are clearer to evaluate and contrast with existing methods.

In either case, you will come into contact with many practically relevant tools and frameworks (e.g. CI, ORM, polyglot development, dependency management, etc.), as nearly all of the projects integrate into established tools.

7.1 The Next Level of DOMtutor

Project type	implementation
Maximum instances running	3
Constraints/requirements	Familiarity with Python, basic understanding of SQL and Docker
Industry linked	No

Description

DOMtutor is the set of automation scripts used to automate interactions with DOMjudge. Currently, DOMtutor targets version 7 of DOMjudge, while version 9 is currently in development. This project aims to migrate DOMtutor to the new version of DOMjudge and upgrade other technical dependencies, too. Concretely, the sub-components are:

- Migrate DOMtutor's SQL queries and docker images to the new version of DOMjudge and ensure feature parity.
- Migrate DOMtutor's ad-hoc data schema to a SQLAlchemy ORM model.
- Integrate judgehost docker images in the problem validation pipeline.

Reading

- Documentation of the respective tools

7.2 Designing an Interactive Educational Framework for LZSCC.111

Project type	didactics, implementation
Maximum instances running	2
Constraints/requirements	Familiarity with Python
Industry linked	No

Description

LZSCC.111 Software Development introduces students to the very basics of programming. As a beginner, the lack of visual, "interactive" feedback often is prohibitive to learning, especially in low-level languages such as C. The aim of this project is to design a series of interactive tasks with increasing difficulty, where students can directly see the effects of their code. Each task should come with scaffolding in C together with a visualizer written in Python, such that students only have to implement central logic, without needing to focus on input / output behaviour.

Tasks could, for example, include finding the maximal element of an array, navigating a robot in a labyrinth (with increasingly complicated tasks), etc.

Alternatively, instructive tasks for DOMjudge that fit the curriculum and can be solved during the workshop can also be designed.

Reading

7.3 Performance Evaluation of JBDD

Project type	implementation
Maximum instances running	2
Constraints/requirements	Familiarity with Java and C
Industry linked	No

Description

JBDD is one of the state-of-the-art implementation of binary decision diagrams in Java. Other well-known BDD libraries are CUDD, BuDDy, and Sylvan (all in C / C++). So far, a comprehensive performance evaluation is missing. The goal of this project is to compare the performance of these tools by implementing standard performance benchmarks in these libraries (at least JBDD and CUDD) and perform a comprehensive evaluation.

A second, connected issue is implementing (parts of) the DDDMP format for reading and writing BDDs to simplify large scale performance regression testing.

Reading

- Documentation of the respective tools
- Knuth, D.E. "Fascicle 1: Bitwise tricks & techniques; Binary Decision Diagrams". The Art of Computer Programming. Vol. 4.
- Bryant, Randal E. "Graph-Based Algorithms for Boolean Function Manipulation".

7.4 Augmenting PET with Linear Equation Solving

Project type	experimental
Maximum instances running	1
Constraints/requirements	Familiarity with Java, basic understanding of equation systems
Industry linked	No

Description

The de-facto standard solution approach to solving large probabilistic systems are variants of *value iteration* (VI), a dynamic programming approach. However, VI struggles with solving so called *stiff* models. Here, considering the induced system of equations can provide notable speed-ups.

The main goal of this project is to heuristically detect when the value iteration seems to be “stuck” and then switch to an efficient equation solving paradigm (i.e. performing a few steps of *strategy iteration* or even solving the linear program). The actual solving should be implemented using high-performance libraries such as `objAlgo`.

Reading

- Wikipedia article on linear equation systems https://en.wikipedia.org/wiki/System_of_linear_equations
- My thesis, Section 2.3 <https://nbn-resolving.org/urn:nbn:de:bvb:91-diss-20210226-155025>
- Related work on combining VI and SI https://doi.org/10.1007/978-3-319-68167-2_25

7.5 Randomizing Probabilistic Models

Project type	implementation, evaluation
Maximum instances running	2
Constraints/requirements	Familiarity with Programming
Industry linked	No

Description

The probabilistic verification community has a severe lack of proper models to test and benchmark various algorithms on. This was already attempted to be addressed in previous work, however the models their approach produced were highly inefficient, in the sense that it took more time to parse the model than actually solving it. As such, the generated models mostly were a benchmark of the parsing speed.

This project tries to rectify this issue by overhauling the model generation process in two different ways. Firstly, by generating *explicit* model files (effectively, writing down every single transition) or by more efficiently generating modularized PRISM models. The random generator should be configurable with various structural properties of the generated graph (e.g. the number of states, whether the graph is connected, the average branching factor, etc.). On top, in the spirit of generative adversarial networks, the random generator could be augmented to create difficult or breaking examples for a given tool.

The generator would ideally be written in Python for usability and maintainability.

Reading

- The PRISM explicit format <https://prismmodelchecker.org/manual/Appendices/ExplicitModelFormat>
- The PRISM Language <https://prismmodelchecker.org/manual/ThePRISMLanguage/Introduction>
- Previous work on generating models https://doi.org/10.1007/978-3-031-19992-9_18
- Generating graphs randomly <https://arxiv.org/abs/2201.04888>

8 Fabio Papacchini

8.1 Implementing an AI Player/Solver

Project type	Implementation
Maximum instances running	2
Constraints/requirements	Programming knowledge
Industry linked	No

Description

This project requires interested students to choose and implement a (relatively simple) game. Once all game rules are implemented, then you are expected to implement an AI player/solver.

Cluedo, Risk, and Sudoku are examples of game implemented by former students.

The game needs to be agreed with me and AI techniques should mainly be rule/search/logic based. You can extend the game in several ways to make it better (e.g., online multiplayer, difficulty selection with different AI players, ...).

There is no constraint on the programming language, libraries, game engine, and so on. This is, however, an implementation-based project, and I am expecting to see your code, not a set of third-party assets!

Reading

- Russell, S.J. and Norvig, P., 2021. Artificial Intelligence: A Modern Approach. Fourth Edition. Pearson.

8.2 “Simple” Explainable AI

Project type	Implementation
Maximum instances running	2
Constraints/requirements	Programming Knowledge
Industry linked	No

Description

This project requires interested students to choose and implement a (relatively simple) software with an AI component. It could potentially be anything as long as there is an AI providing suggestions or making moves.

Once the topic has been agreed with me, your task is to allow the user to query the AI and ask why a particular decision/suggestion was made. The explanation should be understandable by the user.

Explainable AI is a hot and huge topic nowadays. For this project the AI techniques should mainly be rule/logic based. This is because the project does not aim to solve an open and incredibly difficult task, but rather to provide interested students with an understanding of how complex the problem is even within a much simpler and restricted domain.

Reading

- Russell, S.J. and Norvig, P., 2021. Artificial Intelligence: A Modern Approach. Fourth Edition. Pearson.

8.3 Visualisation of Algorithms

Project type	Implementation, Educational
Maximum instances running	2
Constraints/requirements	Programming knowledge
Industry linked	No

Description

Creation of examples and animating algorithms can be an effective way to explain and teach them. Visualising algorithms, however, is a very time consuming task and it is not always possible to carry it out.

The grandiose aim of the project is to create a framework making the visualisation of algorithms simpler. The scope is too wide and general, so the interested student can decide how to narrow it down to a more manageable size (e.g., focusing on the visualisation of a few graph algorithms).

As a suggestion and as an example, the project could be about the visualisation of the main algorithm, DPLL, for satisfiability checking of propositional logic. The visualisation should show the user the different internal steps happening in the algorithms.

Reading

- Possible animation engine library: Manim – <https://docs.manim.community/en/stable/index.html>
- Robert Nieuwenhuis, Albert Oliveras, and Cesare Tinelli. 2006. “Solving SAT and SAT Modulo Theories: From an abstract Davis–Putnam–Logemann–Loveland procedure to DPLL(T)”. J. ACM 53, 6 (November 2006), 937–977. <https://doi.org/10.1145/1217856.1217859>

8.4 Playing with Transitivity

Project type	Experimental, Research
Maximum instances running	1
Constraints/requirements	Programming knowledge, willingness to study logical languages and frameworks
Industry linked	No

Description

As soon as transitivity is involved within a formal/logical context, it does not take much to have to deal with infinite graph structures. An example of this is when checking satisfiability of the basic modal logic extended with the transitivity axiom.

The common approach to avoid infinite graphs is via loop-checking techniques (i.e., recognise “repeated” states and loop instead of creating new successors). Another approach is to create finite trees with a high branching factor, and show that their transitive closure is still a witness of the wanted structure. A further approach could be obtained by taking ideas from the former two. Other approaches exist as well, but there is no unanimous consent on which approach performs better.

This project requires the interested student to implement a tableaux calculus for the basic modal logic extended with transitivity, where at least two approaches to deal with transitivity are possible, to compare their performance and to draw conclusions based on experimental results.

Reading

- van Benthem, J. and Blackburn, P., 2006, “Modal logic: A semantic perspective”, in Blackburn, P., van Benthem, J.F. and Wolter, F. (eds.) “Handbook of modal logic”. Elsevier, pp. 1–84.

- Goré, R., 1999. “Tableau Methods for Modal and Temporal Logics”, in: D’Agostino, M., Gabbay, D.M., Hähnle, R., Posegga, J. (eds) *Handbook of Tableau Methods*. Springer, Dordrecht. https://doi.org/10.1007/978-94-017-1754-0_6
- <https://plato.stanford.edu/entries/logic-modal/>

8.5 Tableaux Prover for Modal Logic(s)

Project type	Implementation, Research
Maximum instances running	1
Constraints/requirements	Programming knowledge, willingness to study logical languages and frameworks
Industry linked	No

Description

Modal logic is a simple extension of propositional logic able to express some graphs and their properties.

This project aims at implementing a tableaux algorithm to check satisfiability of modal formulae.

The project can take several directions depending on the students’ interests. For example, the prover could cover a range of modal logics, the focus could be on improving performance with strategies such as backjumping, or on visualisation of models extracted from open branches.

Reading

- van Benthem, J. and Blackburn, P., 2006, “Modal logic: A semantic perspective”, in Blackburn, P., van Benthem, J.F. and Wolter, F. (eds.) “Handbook of modal logic”. Elsevier, pp. 1–84.
- Goré, R., 1999. “Tableau Methods for Modal and Temporal Logics”, in: D’Agostino, M., Gabbay, D.M., Hähnle, R., Posegga, J. (eds) *Handbook of Tableau Methods*. Springer, Dordrecht. https://doi.org/10.1007/978-94-017-1754-0_6
- <https://plato.stanford.edu/entries/logic-modal/>

9 David Georg Reichelt

David is Assistant Professor in Computer Science and Software Engineering at Lancaster University Leipzig. He joined from the computing center of Leipzig University, where he served as scientific staff. His main research interest is in the field of software performance engineering, where it examines how to identify and understand performance changes that happen during the software development lifecycle.

9.1 Benchmarking Document Reading Systems on Small-Scale Hardware

Project type	Experimental
Maximum instances running	1
Constraints/requirements	Basic knowledge of LLMs, Python, and Bash (or the urge to get it)
Industry linked	No

Description

Document reading systems powered by Large Language Models (LLMs) have become increasingly popular for extracting and interacting with information from text-based documents. However, their deployment on small-scale hardware, such as standard desktop computers or computing hardware available at small-scale computing centers, poses significant challenges in terms of efficiency and performance. This project focuses on benchmarking the performance of document reading systems, such as LLaMA 3.2 or DeepSeek in their smaller-parameter-versions, on small-scale hardware.

The goal of this thesis is to conduct a benchmarking study of document reading systems on different hardware. After setting up appropriate tooling, like LlamaIndex, on the different hardware systems, an established benchmark like DocBench should be executed and evaluated on the different hardware. By the benchmark evaluation, it should become possible to make an informed decision about the trade-off between response quality, hardware resource consumption and response times across the different systems. The candidate will be given access to computing resources at Leipzig University for this thesis.

Reading

Zou, A., Yu, W., Zhang, H., Ma, K., Cai, D., Zhang, Z., ... & Yu, D. (2024). DocBench: A Benchmark for Evaluating LLM-based Document Reading Systems. arXiv preprint arXiv:2407.10701.

Afsoon, S., Naseem, U., Beheshti, A., & Jamali, Z. (2024). PersoBench: Benchmarking Personalized Response Generation in Large Language Models. arXiv preprint arXiv:2410.03198.

LLamaIndex Repository: https://github.com/run-llama/llama_index

9.2 Examination of the Performance Monitoring Overhead of OpenTelemetry Using Python

Project type	Implementation
Maximum instances running	1
Constraints/requirements	Basic knowledge of Python and Java
Industry linked	No

Description

To have optimal performance of software, monitoring its behaviour is crucial. Unfortunately, monitoring itself introduces overhead that needs to be estimated. This can be done by measuring the monitoring overhead of various monitoring frameworks and configurations using the MooBench benchmark.

Currently, MooBench supports benchmarking monitoring using the Kieker framework with Java and Python and benchmarking monitoring using OpenTelemetry with Java. Therefore, the main goal of this project is the extension of MooBench to also support OpenTelemetry with Python overhead benchmarking. This capability should be demonstrated through the execution of benchmark runs.

The project requires reading into the details of application performance monitoring.

Reading

J. Waller, N. C. Ehmke und W. Hasselbring, "Including Performance Benchmarks into Continuous Integration to Enable DevOps," ACM SIGSOFT Software Engineering Notes, Jg. 40, Nr. 2, S. 1–4, März 2015. doi: 10.1145/2735399.2735416.

H. Knoche und H. Eichelberger, "The Raspberry Pi: A Platform for Replicable Performance Benchmarks?" Softwaretechnik-Trends, Jg. 37, Nr. 3, S. 14–16, 2017.

D. G. Reichelt, S. Kühne und W. Hasselbring, "Overhead Comparison of OpenTelemetry, inspectIT and Kieker," Symposium on Software Performance, 2021.

D. G. Reichelt, L. Bulej, R. Jung, and A. van Hoorn, "Overhead Comparison of Instrumentation Frameworks.", In Companion of the 15th ACM/SPEC International Conference on Performance Engineering (ICPE '24 Companion), May 7–11, 2024, ACM, New York, NY, USA.

MooBench Repository: <https://github.com/kieker-monitoring/moobench>

9.3 Observability Energy Consumption Measurement

Project type	Implementation
Maximum instances running	1
Constraints/requirements	Deeper knowledge of Java and the JVM (or the urge to get it)
Industry linked	No

Description

Understanding how software behaves—observability—is crucial for effective debugging and resolving performance issues. Gaining observability is only possible by introducing additional operations into the software system, which causes overhead. This overhead has been well-researched in terms of time consumption using the MooBench benchmark. However, due to the increasing need for saving energy, it becomes necessary to also measure the overhead in terms of energy consumption.

To measure the energy consumption of observability agents, the MooBench benchmark should be extended by energy consumption measurements. Using appropriate hardware counters, the energy consumption with and without tracing an application should be determined. By comparing the energy consumption of different observability frameworks, it becomes possible to choose the framework with the lowest energy consumption and thereby save costs and reduce CO₂ emissions of software operations.

Reading

Reichelt, David Georg, Stefan Kühne, and Wilhelm Hasselbring. "Overhead Comparison of OpenTelemetry, inspectIT and Kieker." (2021). <https://oceanrep.geomar.de/id/eprint/55367/1/SSP2021overhead.pdf>

Brunnert, Andreas, and Ferdinand Gutzy. "Extending the OpenTelemetry Java Auto-Instrumentation Agent to Publish Green Software Metrics." Softwaretechnik-Trends Band 44, Heft 4. Gesellschaft für Informatik eV, 2024. <https://dl.gi.de/bitstreams/1a1bf841-d91b-4832-9c6b-1e2788251bf0/download>

Yang, S., Reichelt, D. G., Jung, R., Hansson, M., & Hasselbring, W. (2025). The Kieker Observability Framework Version 2. ICPE 2025 Companion [in press]. <https://arxiv.org/pdf/2503.09189>.

MooBench Repository: <https://github.com/kieker-monitoring/moobench>

9.4 Source Code Node Change Detection for Scala / Kotlin / ...

Project type	implementation
Maximum instances running	2+
Constraints/requirements	Basic knowledge of Java and Scala / Kotlin / ...
Industry linked	No

Description

The performance of a software is crucial for its everyday use. Therefore, developers strive for optimal performance of their implementations. Changes to the source code of a software may result in varied performance. In order to prevent the occurrence of performance regressions and check the effect of source changes, which are expected to result in performance improvements, both the measurement of the impact of source code changes and a deep understanding of the runtime behaviour of the used source code elements are necessary.

Due to the non-deterministic nature of the execution of software on modern infrastructure like the JVM, the identification of performance changes is time-consuming and error-prone. To overcome this, the performance measurement effort can be reduced by prior static code analysis. This code analysis should derive where potential performance changes can happen and support the developer by presenting source code diffs.

The tool *Peass* uses the source code change detection in Java to reduce the performance measurement overhead. It therefore relies on the *nodeDiffDetector* that parses two commits of a software for the existing performance changes. In this project, the existing implementation for Java should be extended by a change detection for **Scala / Kotlin / ...**.

This topic can be varied by changing the language. Scala and Kotlin are natural choices, since they both compile to the JVM and interoperability is high. If you have special interest in another language, we can also discuss this. Note that while I can help you with technical problems for Scala and Kotlin, I will not be able to help you with exotic languages like FORTRAN.

Reading

D. G. Reichelt, S. Kühne und W. Hasselbring, “Peass: A Tool for Identifying Performance Changes at Code Level,” in Proceedings of the 33rd ACM/IEEE ASE, ACM, 2019. doi: 10.1109/ASE.2019.00123.

Repository of Peass: <https://github.com/DaGeRe/Peass>

Repository of nodeDiffDetector: <https://github.com/DaGeRe/nodeDiffDetector>

9.5 Examination of Performance Change Detection Efficiency Using eBPF Probes

Project type	Implementation
Maximum instances running	1
Constraints/requirements	Deeper knowledge of Java and the JVM (or the urge to get it)
Industry linked	No

Description

Understanding how software behaves—observability—is crucial for effective debugging and resolving performance issues. To achieve this, we can use various technical interfaces that help us observe and analyze software in action. These interfaces can be categorized into two main types: userspace techniques, like the JVM Technical Interface, which interact closely with the application, and kernelspace techniques, such as Performance Counters for Linux (PCL), which often operate with lower overhead. One advancement in this area is the use of extended Berkeley Packet Filters (eBPF). This powerful tool allows us to gather insights with minimal performance impact and without needing root access to the machines.

The Kieker observability tool currently contains several agents that operate at the userspace level. In this project, an innovative eBPF-based agent will be implemented to gather method execution times and call tree information using eBPF. Based on this, the existing benchmark *MooBench* will be utilized to compare the overhead of classical JVMTI observability with eBPF observability. This comparison will not only highlight the benefits of eBPF but also provide a hands-on opportunity to explore cutting-edge technology in software observability.

Reading

Gbadamosi, Bolaji, et al. The eBPF Runtime in the Linux Kernel. arXiv:2410.00026, arXiv, 3 Oct. 2024. arXiv.org, <https://doi.org/10.48550/arXiv.2410.00026>.

Reichelt, David Georg, Stefan Kühne, and Wilhelm Hasselbring. "Overhead Comparison of OpenTelemetry, inspectIT and Kieker." (2021). <https://oceanrep.geomar.de/id/eprint/55367/1/SSP2021overhead.pdf>

Reichelt, D. G., Bulej, L., Jung, R., & van Hoorn, A. (2024, May). Overhead Comparison of Instrumentation Frameworks. In Companion of the 15th ACM/SPEC International Conference on Performance Engineering (pp. 249-256). <https://doi.org/10.1145/3629527.3652269>

Yang, S., Reichelt, D. G., Jung, R., Hansson, M., & Hasselbring, W. (2025). The Kieker Observability Framework Version 2. ICPE 2025 Companion [in press]. <https://arxiv.org/pdf/2503.09189>.

MooBench Repository: <https://github.com/kieker-monitoring/moobench>

9.6 Performance Analysis of the Evolution of Jackson / Guava

Project type	Experimental
Maximum instances running	2
Constraints/requirements	Knowledge of git; deeper knowledge of Java and the JVM (or the urge to get it)
Industry linked	No

Description

Performance is a crucial non-functional property of a software. To fully understand the performance of a software, it is necessary to measure even small performance changes. The measurement of these changes can be done using the tool Peass. Peass transforms the unit test into executable performance unit test definitions. To compare the performance of a software between its commit, Peass first analyses the commits for potential performance changes and afterwards measures the potentially changed performance unit tests.

In this thesis, the tool Peass should be used to analyse the evolution of the performance of the library Jackson / Guava. The candidate should obtain performance changes, understand their root causes and, as far as possible, estimate their effect on productive usages of Jackson / Guava.

Reading

D. G. Reichelt, S. Kühne und W. Hasselbring, "Peass: A Tool for Identifying Performance Changes at Code Level," in Proceedings of the 33rd ACM/IEEE ASE, ACM, 2019. doi: 10.1109/ASE.2019.00123.

J. Chen und W. Shang, "An Exploratory Study of Performance Regression Introducing Code Changes," in Proceedings of the 2017 IEEE ICSME, IEEE, 2017, S. 341–352. doi: 10.1109/ICSME.2017.13.

G. Jin, L. Song, X. Shi, J. Scherpelz und S. Lu, "Understanding and Detecting Real-world Performance Bugs," in Proceedings of the 33rd ACM SIGPLAN PLDI, Ser. PLDI '12, Beijing, China: ACM, 2012, S. 77–88, doi: 10.1145/2254064.2254075.

Repository of Peass: <https://github.com/DaGeRe/Peass>

Repository of Jackson: <https://github.com/FasterXML/jackson-core>

Repository of Guava: <https://github.com/google/guava>

10 Thomas Schmid

10.1 Parallelization of Genetic Algorithms

Project type	Research/Implementation
Maximum instances running	1
Constraints/requirements	Basic understanding of artificial intelligence concepts is helpful
Industry linked	No

Description

Genetic algorithms (GA) are a subfield of evolutionary computing and artificial intelligence. First introduced in 1975 by John H. Holland, GAs mimic the Darwinian theory of survival of the fittest in nature in a simplified way. In contrast to biology, however, not organisms but potential solutions for a given problem are created and selected for fitness. In the standard GA, each new generation of solutions consists of the previous generation's offspring. A large variety of GA variants exists, such as Whitley's rank-based GA and others.

While GAs have the theoretical advantage to yield an optimal solution for a given problem, they possess at the same time the practical disadvantage that they are rather slow and take longer to achieve convergence than other approaches (which in turn are not guaranteed to produce optimal solutions). While a common approach to improve algorithmic performance is to parallelize calculations, this has not for all known GAs been implemented so far. In order to improve the practical relevance of genetic algorithms, you will acquire an overview of existing GA variants with the aim of analyzing their potential for parallelization of the GA. On this basis, you will propose, implement and evaluate a parallelized version of an existing GA variant (that has not been parallelized so far).

Reading

If you have never heard of genetic algorithms before, this general introduction published by its inventor in the popular scientific journal *Scientific American* may be a good starting point:

<https://www.isislab.it/wp-content/uploads/2021/01/Holland-Genetic-Algorithms.pdf>

10.2 Development of a GUI Editor for the Leipzig iLabSheet

Project type	Implementation
Maximum instances running	1
Constraints/requirements	Programming experience with Java and Python
Industry linked	No

Description

The *Leipzig Interactive Laboratory Worksheet* (iLabSheet) is an open educational resource (OER) for creating introductory tutorials with Jupyter Notebook. Originally developed at Leipzig University, it allows to create interactive worksheets, which will ask the worksheet user to answer to pre-defined questions of various types. Most importantly, the *Leipzig iLabSheet* provides its users direct feedback on whether his or her solution is correct.

So far, any contents for the *Leipzig iLabSheet* need to be specified directly inside a Python source code file. In particular, this requires to get familiarized with the underlying syntax instead of focussing directly on the tasks of the worksheet that is to be created. For many users, this may be an inefficient way of using this OER.

Using either Java or Python, you will develop an editor that will allow to create new or edit existing source code files for the *Leipzig iLabSheet* via a graphical user interface (GUI). As a first step, you will define a conceptional representation of questions and worksheets as XML file. Based on this, you will implement a way to transform the contents of the XML file into the actual Python source files for *Leipzig iLabSheet*.

Reading

For code and further details about *Leipzig iLabSheet*, please refer to its open source repository:

<https://git.informatik.uni-leipzig.de/ml-group/tools/Leipzig-iLabSheet>

11 Guido Schmitz

11.1 Modern authorization for mobile and web applications

Project type	(implementation, research)
Maximum instances running	2
Constraints/requirements	This is a challenging project that will require good programming skills with proficiency in multiple programming languages.
Industry linked	No

Description

Modern applications often consist of multiple parts that are connected using Web-based APIs (often following REST principles). A (cloud-based) backend provides persistence and implements business logic, while frontends, such as mobile and web apps, provide user interfaces that interact with the backend services. Often, 3rd-party providers can integrate their services as well, sometimes using the same APIs as the first-party application components.

In such a diverse and distributed setting, authentication and authorization are key to protect resources from unauthorized access. Protocols like OAuth 2.0 and OpenID Connect aim to solve this issue and have found widespread adoption in industry. However, their application is often not trivial and can be easily implemented in flawed ways. To illustrate how such mechanisms can be applied in a good way, we want to build a mock application that consists of multiple components (such as a backend, a mobile app, and a web interface) and that allows for 3rd-party integration. This application should make use of common authentication and authorization frameworks (using appropriate libraries). Due to the nature of such applications, you will need to develop several components in different programming languages, potentially using different libraries.

Objectives

- Explore how single sign-on authentication and authorization is performed in modern mobile and web applications as well as for 3rd-party integrations
- Provide an overview on common mechanisms for authentication and authorization
- Build a multi-user (mock) application that employs such mechanisms. The application should include a backend, a RESTful API (for 3rd-party integrations and optionally frontend), and at least one frontend (web interface or mobile app)

Reading

- <https://tools.ietf.org/html/rfc6749> (OAuth 2.0)
- <https://tools.ietf.org/html/rfc6750> (Bearer Token Usage)
- <https://tools.ietf.org/html/rfc6819> (OAuth 2.0 Threat Model and Security Considerations)
- <https://tools.ietf.org/html/rfc9700> (OAuth Security BCP)
- <https://tools.ietf.org/html/draft-ietf-oauth-browser-based-apps-24> (Draft Best Practices for OAuth Browser-Based Apps)
- https://openid.net/specs/openid-connect-core-1_0.html (OpenID Connect Core 1.0)
- <https://www.oauth.com/> (OAuth 2.0 Simplified)
- <https://www.oauth.net/> (Community page for OAuth 2.0)

- Daniel Fett, Ralf Küsters, Guido Schmitz. 2016. A Comprehensive Formal Security Analysis of OAuth 2.0. In: 23rd ACM SIGSAC Conference on Computer and Communications Security (CCS 2016). <https://dl.acm.org/doi/pdf/10.1145/2976749.2978385>
- Daniel Fett, Ralf Küsters, Guido Schmitz. 2017. The Web SSO Standard OpenID Connect: In-Depth Formal Security Analysis and Security Guidelines. In: IEEE 30th Computer Security Foundations Symposium (CSF 2017). <https://ieeexplore.ieee.org/iel7/8048777/8049639/08049720.pdf>

11.2 A Simple-to-Use Certification Authority

Project type	(implementation, research)
Maximum instances running	1
Constraints/requirements	No constraints
Industry linked	No

Description

To establish secure channels, e.g., using the TLS protocol, public keys are often provided ad-hoc by the remote communication endpoint. Trust in such keys is then established by relying on a Public Key Infrastructure (PKI). Such a PKI provides a chain of trust on certificates (and intermediate certificates) to so-called trust anchors. The trust anchor is a public key belonging to a trusted party, a certification authority (CA), which attests relations between identities and public keys.

Managing a CA is a critical task, which involves the generation, distribution, storing, and revoking of certificates as well as validating the authenticity of certification requests. There are many commercial entities offering such services as well as community-sponsored organisations (such as Let's Encrypt). However, these services aim at use cases for publicly-recognised certificates. For internal IT infrastructures, however, such recognition is typically not necessary as trust can be easily established to an internal trust anchor and obtaining certificates for internal-only services from external entities can also be complicated and laborious.

Tools to manage an internal PKI are often hard to set up or only follow deprecated recommendations for creating certificates. Hence, the goal in this topic is to build a simple-to-use tool which follows current best practices for certificate management.

Objectives

- Research on current best practices for certificates, such as cipher suites and extensions
- Survey the landscape of tools for managing an internal PKI
- Identify shortcomings of existing tools with respect to current best practices in particular.
- Implement a simple-to-use tool to manage certificates for small-scale IT infrastructures

Reading

- <https://sourceforge.net/projects/gnomint/>
- <https://opsec.eu/src/tinyca/>
- <https://www.hohnstaedt.de/xca/>
- <https://www.dogtagpki.org/>
- <https://github.com/OpenVPN/easy-rsa>
- <https://pki-tutorial.readthedocs.io/>
- <https://www.openssl.org>

- EJBCA <https://www.ejbcna.org>
- Rescorla, E. (2018). The transport layer security (TLS) protocol version 1.3, RFC-8446
- Vacca, J. R. (2004). Public Key Infrastructure. Boca Raton: C R C Press LLC.
- Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R. Polk, W. (2008) Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile (RFC-5280)
- Eastlake, D., (2011) Transport Layer Security (TLS) Extensions: Extension Definitions (RFC-6066)

11.3 (Social) Web Single Sign-On in the Wild

Project type	(implementation, research)
Maximum instances running	1
Constraints/requirements	
Industry linked	No

Description

(Social) web single sign-on systems, such as Facebook Login or Google Sign-in, have found a widespread adoption in the web to authenticate users. These systems are typically based on the standards OpenID Connect 1.0 and OAuth 2.0. These protocols, however, can be used incorrectly, making their usage insecure. While many of these problems are addressed by guidance and protocol improvements, web sites might not follow this advice.

In this project, we want to (semi-)automatically scan the web for web sites that use social web single sign-on and check if they are using these authentication systems in insecure ways.

Objectives

- Provide an overview on the current state of social single sign-on.
- Develop a web crawler that detects insecure usage of (social) web single sign-on based on OpenID Connect 1.0 or OAuth 2.0
- Use the crawler to conduct an analysis of a reasonable set of web sites (e.g., Tranco list)

Reading

- <https://tools.ietf.org/html/rfc6749> (OAuth 2.0)
- <https://tools.ietf.org/html/rfc6750> (Bearer Token Usage)
- <https://tools.ietf.org/html/rfc6819> (OAuth 2.0 Threat Model and Security Considerations)
- <https://tools.ietf.org/html/rfc9700> (OAuth 2.0 Best Current Practices)
- <https://tools.ietf.org/html/draft-ietf-oauth-browser-based-apps-24> (Draft Best Practices for OAuth Browser-Based Apps)
- https://openid.net/specs/openid-connect-core-1_0.html (OpenID Connect Core 1.0)
- Ronghai Yang, Guanchen Li, Wing Cheong Lau, Kehuan Zhang, and Pili Hu. 2016. Model-based Security Testing: An Empirical Study on OAuth 2.0 Implementations. ASIA CCS '16. ACM, New York, 651–662. DOI:<https://doi.org/10.1145/2897845.2897874>
- Shernan E., Carter H., Tian D., Traynor P., Butler K. 2015. More Guidelines Than Rules: CSRF Vulnerabilities from Noncompliant OAuth 2.0 Implementations. In: Almgren M., Gulisano V., Maggi F. (eds) Detection of Intrusions and Malware, and Vulnerability Assessment. DIMVA 2015. Lecture Notes in Computer Science, vol 9148. Springer, Cham. https://doi.org/10.1007/978-3-319-20550-2_13

- San-Tsai Sun and Konstantin Beznosov. 2012. The devil is in the (implementation) details: an empirical analysis of OAuth SSO systems. CCS '12. ACM, New York, 378–390. DOI:<https://doi.org/10.1145/2382196.2382238>
- Yuchen Zhou and David Evans. 2014. SSOScan: Automated Testing of Web Applications for Single Sign-On Vulnerabilities. USENIX Security 2014. <https://www.usenix.org/conference/usenixsecurity14/technical-sessions/presentation/zhou>
- Daniel Fett, Ralf Küsters, Guido Schmitz. 2016. A Comprehensive Formal Security Analysis of OAuth 2.0. In: 23rd ACM SIGSAC Conference on Computer and Communications Security (CCS 2016). <https://dl.acm.org/doi/pdf/10.1145/2976749.2978385>
- Daniel Fett, Ralf Küsters, Guido Schmitz. 2017. The Web SSO Standard OpenID Connect: In-Depth Formal Security Analysis and Security Guidelines. In: IEEE 30th Computer Security Foundations Symposium (CSF 2017). <https://ieeexplore.ieee.org/iel7/8048777/8049639/08049720.pdf>

12 Pascal Welke

I do not know if you want to add a couple of sentences here. Up to you.

12.1 How Many Graphs in Benchmark Datasets are “Amenable”?

Project type	Implementation, Experimental
Maximum instances running	3
Constraints/requirements	Candidate should have knowledge about graphs and interest in algorithms. Python programming skills are required.
Industry linked	No

Description

Most current graph neural networks are based on message passing. Such message passing graph neural networks (MPNNs) are closely connected [1] to the Weisfeiler-Leman color refinement algorithm [2,3], which is an inexact algorithm for graph isomorphism testing. Graphs where the Weisfeiler-Leman algorithm always works correctly are called amenable and there is a near linear time algorithm to decide whether any given graph is amenable, or not [4]. Your task in this project is to understand and implement this algorithm efficiently in python and to compute the number of amenable graphs in a large number of graph datasets that are commonly used for benchmarking MPNN performance.

Reading

1. Xu et al. (2019): How Powerful are Graph Neural Networks? ICLR
2. Weisfeiler and Leman (1968): A Reduction of a Graph to a Canonical Form and an Algebra Arising during This Reduction. Nauchno-Technicheskaya Informatsia
3. for further reference, see [https://en.wikipedia.org/wiki/Weisfeiler_Leman_graph_isomorphism test](https://en.wikipedia.org/wiki/Weisfeiler_Leman_graph_isomorphism_test)
4. Arvind et al. (2017): Graph Isomorphism, Color Refinement, and Compactness. Computational Complexity

12.2 Implement a Fast Convolution Layer in a Graph Neural Network Library

Project type	Implementation, Experimental
Maximum instances running	2
Constraints/requirements	Candidate should have knowledge about graphs and interest in algorithms. Python coding skills required. Experience with pytorch beneficial,
Industry linked	No

Description

Pytorch Geometric [1] is the de-facto-standard library for deep learning on graphs. It implements several message passing graph neural network layers, among other things and makes it (comparatively) easy to experiment with new ideas or to develop, run, and deploy new deep learning systems.

Among other things, it contains an implementation [2] of the Weisfeiler Leman algorithm [3,4]. However, it uses a very basic Python implementation that does not benefit from parallelization that is possible within pytorch.

Your task is to implement a faster variant [5, 6], based on prime numbers and sparse matrix multiplication and to compare its speed against the existing implementation.

Reading

1. <https://pytorch-geometric.readthedocs.io>
2. https://pytorch-geometric.readthedocs.io/en/latest/generated/torch_geometric.nn.conv.WLConv.html
3. Weisfeiler and Leman (1968): A Reduction of a Graph to a Canonical Form and an Algebra Arising during This Reduction. Nauchno-Technicheskaya Informatsia
4. for further reference, see https://en.wikipedia.org/wiki/Weisfeiler_Leman_graph_isomorphism_test
5. Kersting, Mladenov, Garnett, Grohe (2014): Power Iterated Color Refinement. AAAI
6. Pascal Welke (2025): Fast Weisfeiler Lehman <https://pwelke.de/ramblings/FastWLImplementation.pdf>

13 James Stovold

In general, I am interested in projects relating to artificial life, unconventional computing, and distributed cognition. I assume that all projects listed below would be adapted to the interests of the student through initial communication.

I actively encourage interested students to reach out in order to learn more about the projects.

13.1 Extending DGCA-Grown Reservoirs to Liquid State Machines

Project type	Research, experimental
Maximum instances running	1
Constraints/requirements	An interest in spiking neural networks.
Industry linked	No

Description

Recent research has shown that the DGCA (Developmental Graph Cellular Automata) can grow effective Echo State Networks (ESNs) as reservoir computers. This project will extend this work to Liquid State Machines (a spiking version of the ESN) and compare their performance.

Objectives:

- Apply the DGCA to grow liquid state machines.
- Compare the performance to ‘normal’ LSMs and to DGCA-grown ESNs on benchmark tests.
- Compare the performance of metric-grown LSMs to task-grown LSMs.

Reading

- Maass and Markram (2004) “On the computational power of circuits of spiking neurons”, JCSS 69(4):593–616, <https://doi.org/10.1016/j.jcss.2004.04.001>¹
- Jaeger (2007) “Echo State Networks”, Scholarpedia 2(9):2330 http://www.scholarpedia.org/article/Echo_state_network
- Griffin and Stepney (2024) “Entropy Transformation Measures for Computational Capacity”, Proc. UCNC 2024, https://doi.org/10.1007/978-3-031-63742-1_9
- Waldegrave et al. (2023) “Developmental Graph Cellular Automata”, Proc. ALIFE 2023, https://doi.org/10.1162/isal_a_00658

¹N.B. no need to understand the formal theory here, but worth taking a look to understand where the model came from

13.2 Building Movement into Neural Cellular Automata

Project type	Research, experimental
Maximum instances running	1
Constraints/requirements	An interest in neural networks and basic biological processes.
Industry linked	No

Description

Neural Cellular Automata (NCA) are an artificial model of morphogenesis (how shapes grow from a single cell without centralised control). Recent advances have shown that NCAs can respond to external signals, so a natural question would be “what can we get the NCA to do in response to a signal?”. This project will look at this question, focussing on getting the NCA to exhibit basic movements in response to a signal.

Objectives:

- Build an extension to the NCA that trains basic movement into the model.
- Test the robustness of the response through systematic variation of the signal and repeated signals.
- Ask whether the extension can interpret different signals and move accordingly (e.g. a ‘left’ signal and a ‘right’ signal).

Reading

- Mordvintsev, et al. (2020) “Growing Neural Cellular Automata”, Distill, <https://dx.doi.org/10.23915/distill.00023>
- Stovold (2023) “Neural Cellular Automata Can Respond to Signals”, Proc. ALIFE 2023, https://doi.org/10.1162/isal_a_00567

13.3 Cellular Mitosis in Neural Cellular Automata

Project type	Research, experimental
Maximum instances running	1
Constraints/requirements	An interest in neural networks and basic biological processes.
Industry linked	No

Description

Neural Cellular Automata (NCA) are an artificial model of morphogenesis (how shapes grow from a single cell without centralised control). Recent advances have shown that NCAs can respond to external signals, so a natural question would be “what can we get the NCA to do in response to a signal?”. This project will look at this question, focussing on getting the NCA to split into two child cells in response to a signal.

Objectives:

- Build an extension to the NCA that exhibits mitosis in response to a signal.
- Test the robustness of the response through systematic variation of the signal and repeated signals.

Reading

- Mordvintsev, et al. (2020) “Growing Neural Cellular Automata”, Distill, <https://dx.doi.org/10.23915/distill.00023>
- Stovold (2023) “Neural Cellular Automata Can Respond to Signals”, Proc. ALIFE 2023, https://doi.org/10.1162/isal_a_00567

13.4 Social Signalling in Neural Cellular Automata

Project type	Research, experimental
Maximum instances running	1
Constraints/requirements	An interest in neural networks and basic biological processes.
Industry linked	No

Description

Neural Cellular Automata (NCA) are an artificial model of morphogenesis (how shapes grow from a single cell without centralised control). Recent advances have shown that NCAs can respond to external signals, so a natural question would be “how can we build NCAs that can send signals to each other?”. This project will look at this question, focussing on training the NCA to produce a signal that could be sent to another NCA.

Objectives:

- Build an extension to the NCA that produces its own signal.
- Test the robustness of the response through systematic variation of the signal and repeated signals.
- (If time permits) Test the ability for one NCA to signal directly to another NCA.

Reading

- Mordvintsev, et al. (2020) “Growing Neural Cellular Automata”, Distill, <https://dx.doi.org/10.23915/distill.00023>
- Stovold (2023) “Neural Cellular Automata Can Respond to Signals”, Proc. ALIFE 2023, https://doi.org/10.1162/isal_a_00567

13.5 Comparing Gliders in Different Game of Life Tilings

Project type	Research, experimental
Maximum instances running	1
Constraints/requirements	
Industry linked	No

Description

Conway's Game of Life is characterised by emergent behaviour (especially in the form of gliders). This project will look at how consistently the glider emerges when the tiles change shape. In particular, we will focus on the difference between behaviour on standard tiles (square grid) compared with either periodic, but non-square tiles (e.g. hexagonal) and with aperiodic tiles (e.g. Penrose tiling).

Objectives:

- Build a simulation of Conway's Game of Life
- Adapt the simulation to permit different tiling regimes.
- Characterise how the behaviour changes and quantify the difference between them.

Reading

- https://en.wikipedia.org/wiki/Conway%27s_Game_of_Life
- https://en.wikipedia.org/wiki/Penrose_tiling
- Owens, Stepney (2010) "The Game of Life Rules on Penrose Tilings: Still Life and Oscillators" In: Adamatzky, A. (eds) Game of Life Cellular Automata. Springer, London. https://doi.org/10.1007/978-1-84996-217-1_8

13.6 Developing the Lecturer Interface for Digital Systems App

Project type	Implementation
Maximum instances running	1
Constraints/requirements	
Industry linked	No

Description

The Digital Systems app is a revision and learning support app designed for the first-year Digital Systems module. This project will help to expand the app to work for other modules by developing a lecturer interface, so that a user can implement their own puzzles and tasks without needing to code. The app is developed using Flutter (<https://flutter.dev/>) with a backend in Dart (<https://dart.dev/>).

Objectives:

- Develop basic code for producing custom widgets in the app at runtime.
- Produce user-friendly interface for lecturer to create interactive widgets.
- (Time permitting) Perform user testing of interface.

Reading

- Flutter language: <https://flutter.dev>
- Dart language: <https://dart.dev>
- Remote Flutter widgets <https://pub.dev/packages/rfw>

13.7 Imperfections in Reaction–Diffusion Chemistry Simulations

Project type	Research, experimental
Maximum instances running	1
Constraints/requirements	Comfortable working with differential equations and an interest in non-standard approaches to computation
Industry linked	No

Description

Reaction–Diffusion Chemistry (RDC) is an unconventional computing paradigm which exploits the interaction of chemical wavefronts to perform computation. Most work in RDC is performed using computer simulations, which do not take into account real-world imperfections such as dust. A simple experiment can show that a speck of dust in the path of an RDC wave is sufficient to disrupt that wave, or even prevent it from progressing entirely.

This project will look at how susceptible different types of reaction are to imperfections in the reactor.

Objectives:

- Build a simulation of RDC waves
- Experiment with different types of imperfections
- Characterise the parameter space for different RDC reactions and imperfections

Reading

- Adamatzky, Holley, Bull, De Lacy Costello, “On computing in fine-grained compartmentalised Belousov–Zhabotinsky medium”, <https://doi.org/10.1016/j.chaos.2011.03.010>
- Stovold, O’Keefe, “Reaction–diffusion chemistry implementation of associative memory neural networks”, <https://doi.org/10.1080/17445760.2016.1155579>

14 James G. Wright

I do not know if you want to add a couple of sentences here. Up to you.

14.1 Swarm Identify Friend or Foe behaviour

Project type	Implementation
Maximum instances running	1
Constraints/requirements	Programming, Distributed Systems, State Machines
Industry linked	No

Description

In this project the candidate will simulate an identify friend or foe behaviour for a swarm of agents. The swarm will manoeuvre together through space, and every so often a member of the swarm will arbitrarily separate and eventually return. After the initial swam simulation has been accomplished, the goal of this project is to implement the behaviour of the detached agent it separates, performs its task, and eventually returns. Along side the candidate will have to develop a mechanism where a propionate subset of the swarm verifies the identity of the returning agent, before allowing them to rejoin.

Reading

- C. Reynolds, Flocks, Herds and Schools: A Distributed Behavioral Model, *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, pp. 25–34, 1987.
- N. A. Lynch, *Distributed Algorithms*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1996.

14.2 Port Automation

Project type	Implementation,Experimental
Maximum instances running	1
Constraints/requirements	Programming, Distributed Systems, Graph Theory, Real-Time Systems
Industry linked	No

Description

This project will simulate autonomous gantries in a port environment. The candidate will design and implement how these autonomous agent's tasks are scheduled and how they plan their routes in an evolving environment, all the while obeying strict real time safety requirements. It is up to the candidate to decide how centralised or distributed their solutions will be. This project can be extended into a mixed modal environment, where there are uncontrolled agents, e.g. trucks, that must be accounted for by the system.

Reading

- H. Kopetz and W. Steiner, *Real-Time Systems: Design Principles for Distributed Embedded Applications*, Springer Cham, Switzerland, 2022.
- N. A. Lynch, *Distributed Algorithms*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1996.
- R. Diestel, *Graph Theory*, Springer Berlin, Heidelberg, Germany, 2017.

14.3 Computer Network Description Language

Project type	Research, Theoretical
Maximum instances running	1
Constraints/requirements	Programming, Computer Linguistics, Computer Networking
Industry linked	Yes. In collaboration with the Norwegian Cyber Range

Description

The goal of this project is to develop a language that can encapsulate the topology of a computer network, as well as the basic parameters of the network services (such as firewalls, DHCP, etc) of the devices on the network. The developed language will be used to create reproducible networks for cyber exercises at the Norwegian Cyber Range. An additional objective is for the candidate to prove that the language they have developed is context-free.

Reading

- J. E. Hopcroft and R. Motwani and J. D. Ullman, *Introduction to Automata Theory, Languages, and Computation*, Addison-Wesley Longman Publishing Co., Inc., USA, 2006.
- D. Grune and C. J. H. Jacobs, *Parsing Techniques: A Practical Guide*, Springer-Verlag, New York, USA, 2008.
- T. Lammle, *CCNA Certification: Exam 200-301*, Sybex, CA, USA, 2024.

14.4 ‘Gadget bahn’ freight testbed

Project type	Implementation
Maximum instances running	1
Constraints/requirements	Programming, Queuing Theory, Graph Theory
Industry linked	No

Description

Various alternatives to rail freight have been proposed in recent years, for example hyperloop, that claim that speed increase will revolutionise the shipment of containers. Professional rail engineers have doubted the veracity of these claims and dubbed these proposals ‘gadget bahns’. The purpose of this project is to develop a simulation of a simple rail network to evaluate the efficacy of these claims of increased freight capacity.

Reading

- R. Borndörfer and T. Klug and L. Lamorgese and C. Mannino and M. Reuther and T. Schlechte, *Handbook of Optimization in the Railway Industry*, Springer Berlin, Heidelberg, Germany, 2018.
- A. Landex and B. Schittenhelm and A. Kaas and J. Schneider-Tilli, Capacity measurement with the UIC 406 capacity method, *Eleventh International Conference on Computer System Design and Operation in the Railway and Other Transit Systems*, pp. 55–64, 2008.
- M. Harchol-Balter, *Performance Modeling and Design of Computer Systems: Queueing Theory in Action*, Cambridge University Press, UK, 2013.
- R. C. Larson and A. R. Odoni, *Urban Operations Research*, Dynamic Ideas, Charlestown, MA, USA, 2007.