Where do I want to be by the end of this period/year? What do I want to be doing? (Include as many learning needs as required to achieve agreed objectives)

What do I want/need to learn? Provide a specific description of the desired changes (e.g. skills to gain, knowledge to acquire, topics/themes/content to cover)	**What do I have to do to achieve this?** Some examples, a new/ongoing course, conference, self-development (like wider research or reading), coaching/mentoring, job shadowing	**What resources or support will I need?** Some examples, teaching staff support, library support, student advisor support, line manager, etc.	**How will I measure success?** Some examples, appraisals, course assessments, team feedback, tutor feedback	**Target dates for review and completion** Note that these need to be realistic/achievable
Advanced Computational Research (Algorithm design, Big Data analysis)	Complete Harvard University CS50 courses; implement parallel algorithms (e.g., using MPI) for research project	HPC cluster access, CS50 materials, Python libraries (Dask, NumPy)	Project submission with performance benchmarks; course certificates	Detailed plan added bellow the Action Plan template table
Experimental Research Methods (Hypothesis testing, reproducible research)	Complete: - Experimental Design for CS (Duke/Coursera) - Research Data Management (MIT/edX) Benchmark ML models	JupyterLab, LaTeX templates, statistical tools (SciPy, R)	GitHub repo with reproducible experiment; published results	Courses: Aug - Oct 2025 Repo: Nov 2025
AI/ML Systems (Model training, MLOps)	Deploy scalable ML pipeline (e.g., TensorFlow Serving + Kubernetes); contribute to PyTorch OSS	Cloud credits (AWS/GCP), ML engineering mentors	Functional pipeline with monitoring; accepted OSS contributions	Aug 2025
Master's Thesis/Project (Original research contribution)	1. Define thesis topic 2. Submit proposal 3. Implement prototype 4. Write/publish results	Thesis advisor, university HPC resources, ACM/IEEE paper templates	Accepted thesis proposal; prototype implementation; submission to peer-reviewed venue	Proposal: May 2025 Draft: Aug 2025 Final: Nov 2025
Academic Writing (Technical papers, peer review)	Write workshop paper using ACM template; complete Writing in the Sciences (Stanford/Coursera)	Overleaf (LaTeX), writing center feedback	Paper submission to student conference (e.g., ACM SIGCSE)	End of 2025/beginning of 2026

Professional Development Plan

Date: 2025.04.21

Student name: Andrius Busilas

Planned training

Training	Provider	Link to course	Development objectives	Status	Timeline
CS50's Introduction to Programming with Python (CS50 Python)	Harvard University	https://cs50.harvard.edu/python/2022/	1. Acquire foundational programming skills through an introductory course focusing on Python, including reading, writing, testing, and debugging code effectively. 2. Cater to individuals with varying levels of programming experience to foster learning specifically in Python. 3. Master core programming concepts such as functions, arguments, return values, variables, types, conditionals, Boolean expressions, and loops. 4. Develop proficiency in handling exceptions, debugging code, writing unit tests, utilizing third-party libraries, validating data with regular expressions, and manipulating files. 5. Gain practical experience in modeling real-world entities using classes, objects, methods, and properties. 6. Emphasize hands-on practice through exercises inspired by real-world programming challenges to reinforce learning. 7. Provide flexibility in coding environments, enabling participants to utilize a web browser or individual PC/Mac for coding purposes.	Completed	Dec 2023 - Apr 2024
100 Days of Code: The Complete Python Pro Bootcamp by Dr. Angela Yu	Udemy	https://www.udemy.com/course/100-days-of-code/	1. Explore a diverse range of tools and technologies within the Python ecosystem, including Python 3, PyCharm, Jupyter Notebook, Python Scripting, Web Development, Data Science, GUI Desktop App Development, Version Control, Backend Web Development, and deployment strategies. 2. Develop competence in key Python libraries and frameworks such as Pandas, NumPy, Matplotlib, Flask, REST APIs, SQL databases, authentication mechanisms, and web design principles, fostering a well-rounded skill set as a Python developer.	Completed	Nov 2023 - Sep 2024
CS50's Introduction to Computer Science (CS50x)	Harvard University	https://cs50.harvard.edu/x/2024/	1. Develop problem-solving skills with an emphasis on correctness, design, and coding style, encompassing both code-based and non-code-based solutions. 2. Delve into computational thinking, abstraction, algorithms, data structures, and broader computer science concepts to enhance overall understanding. 3. Work on problem sets inspired by various disciplines to foster creativity and interdisciplinary learning. 4. Learn fundamental programming principles that transcend specific languages, enabling the ability to adapt and learn new languages independently. 5. Begin with the foundational language C to grasp core concepts such as functions, variables, conditionals, loops, and gain insights into computer architecture and memory management. 6. Progress to Python, building on the understanding gained from C, and later explore SQL for database management, along with HTML, CSS, and JavaScript for web and mobile app development. 7. Conclude the course with a final project that integrates learned skills and knowledge into a practical application, showcasing proficiency in programming and problem solving.	Completed	Feb 2024 - Oct 2024

CS50’s Introduction to Cybersecurity (CS50 Cybersecurity)	Harvard University	https://cs50.harvard.edu/cybersecurity/2023/	1. Gain an introductory understanding of cybersecurity suitable for both technical and non-technical individuals. 2. Learn essential strategies to protect accounts, data, systems, and software from current threats and develop skills to anticipate and assess future threats in personal and professional settings. 3. Acquire knowledge on safeguarding personal privacy in the digital realm. 4. Develop a nuanced perspective on cybersecurity, viewing it as a balance between risks and rewards for adversaries and costs and benefits for oneself. 5. Recognize the trade-off between cybersecurity measures and usability, emphasizing the importance of finding a balance between security and user experience. 6. Comprehend cybersecurity threats through a mix of high-level and low-level examples to ensure a comprehensive technical understanding. 7. Engage in assignments inspired by real-world cybersecurity incidents to enhance practical		start from Apr 2025
CS50’s Introduction to Databases with SQL (CS50 SQL)	Harvard University	https://cs50.harvard.edu/sql/2024/	1. Utilizing SQL as the primary language to build foundational knowledge on data management. 2. Master the essential operations of creating, reading, updating, and deleting data within relational databases, structured in rows and columns. 3. Develop skills in modeling real-world entities and establishing relationships among them through tables equipped with appropriate data types, triggers, and constraints. 5. Learn techniques for normalizing data to enhance efficiency, eliminate redundancies, and minimize the risk of errors in database operations. 6. Acquire proficiency in joining tables using primary and foreign keys, facilitating efficient data retrieval and manipulation. 7. Explore advanced concepts such as using views to automate searches and indexes to optimize search performance. 8. Understand the integration of SQL with other programming languages like Python and Java to enhance versatility and interconnectedness in data management. 9. Begin with SQLite for portability and progress towards PostgreSQL and MySQL to delve into scalable database solutions, ensuring adaptability across different environments. Engage in assignments inspired by real-world datasets to apply learned concepts in practical scenarios and enhance problem-solving skills in data management.	Ongoing	Feb 2025 - Now
CS50’s Web Programming with Python and JavaScript (CS50 Web)	Harvard University	https://cs50.harvard.edu/web/2020/	1. Enhance proficiency in designing and developing web applications using Python, JavaScript, and SQL, with a focus on frameworks such as Django, React, and Bootstrap. 2. Gain in-depth knowledge and skills in database design, scalability, security, and enhancing user experience in web applications. 3. Develop practical expertise through hands-on projects involving writing and utilizing APIs, creating interactive user interfaces, and utilizing cloud services like GitHub and Heroku. 4. Acquire the ability to independently design and deploy web applications by the end of the personal development plan, demonstrating mastery of key principles, languages, and tools essential for internet application development.		Start from Aug 2025

CS50’s Introduction to Artificial Intelligence with Python (CS50 AI)	Harvard University	https://cs50.harvard.edu/ai/2024/	1. Enroll in a course focusing on the fundamental concepts and algorithms underpinning modern artificial intelligence, targeting a deeper understanding of technologies like game-playing engines, handwriting recognition, and machine translation within a defined timeline. 2. Engage in hands-on projects to apply theoretical concepts in practice, exploring graph search algorithms, classification, optimization, machine learning, large language models, and other key topics in artificial intelligence through the creation of Python programs. 3. Utilize project-based learning to enhance knowledge and skills in machine learning libraries, enabling practical application and implementation of artificial intelligence principles in personalized programming projects. 4. Cultivate expertise in utilizing libraries for machine learning and integrating artificial intelligence principles to develop intelligent systems, fostering the capability to design and construct innovative solutions informed by AI technologies. 5. Develop a comprehensive understanding of the core principles and methodologies governing artificial intelligence through practical application and experimentation, ensuring proficiency in designing intelligent systems and leveraging AI techniques effectively. 6. Focus on incorporating artificial intelligence concepts and algorithms into Python programs, reinforcing the connection between theory and practical implementation in order to strengthen overall proficiency in AI development. 7. Complete the course with a robust portfolio of hands-on projects showcasing expertise in machine learning libraries and practical application of artificial intelligence concepts, demonstrating readiness to tackle complex AI challenges and design intelligent systems of their		Start from Nov 2025