

Module Descriptions

Computer Science (M.Sc.)

© Prof. Dr. Stephan Huber

March 11, 2025

Table of contents

1	Applications of Algorithms in Data Science and Artificial Intelligence	3
2	Robotics	5
3	Entrepreneurship for Computer Scientists	7
4	Cloud Computing and Remote Work Tools	10
5	Software Project, Process and Quality Management	13
6	Learning Outcomes	14
7	Software Verification and Validation	17
8	Decision-Making with Artificial Intelligence & Big Data	20
9	Deep Learning	21

1 Applications of Algorithms in Data Science and Artificial Intelligence

1.1 Learning Outcomes

After successful completion of the module, students are able to:

- Describe the fundamental concepts of algorithms, their significance, and their role in data science and artificial intelligence.
- Identify various types of algorithms and their specific applications within data science and AI.
- Judge the effectiveness of different algorithms based on their performance and suitability for a given problem.
- Investigate and critique statistical and machine learning algorithms, as well as their impact on data analysis and decision-making.
- Develop an understanding of deep learning techniques and neural networks and their applications in real-world scenarios.
- Investigate natural language processing algorithms, including their mechanisms and practical uses in analyzing text data.
- Structure and prepare independently algorithms for data analysis and problem-solving in the context of big data.
- Solve problems by applying appropriate optimization algorithms to enhance the performance of data-driven models.
- Reflect and discuss ethical considerations, including fairness and accountability, when designing and implementing algorithms in data science and AI.

1.2 Module Content

Throughout the course, students will engage in practical implementations and case studies to apply algorithms to real-world data science and AI problems. The course will emphasize both theoretical foundations and practical applications, preparing students for advanced research and development in the field.

- Introduction to algorithms
 - Basics of algorithms: Definition and importance
 - Types of algorithms and their applications
 - The role of algorithms in problem-solving

- Introduction to algorithms in data science and AI
- Data science algorithms
 - Statistical and machine learning algorithms
 - Graph algorithms for data science
 - Application of optimization algorithms in data science
 - Evaluation metrics for data science algorithms
- Deep learning techniques and neural networks
- Natural language processing algorithms
- Ethical considerations in data science and AI
 - Fairness in algorithms
 - Ethical practices in data science and AI
 - Transparency and accountability in algorithmic decision-making

Literature

Foster et al. (2020), Blum et al. (2020), Marsland (2015), Cormen et al. (2022), Norvig & Russell (2021), Hastie et al. (2009), Louridas (2020)

References

- Blum, A., Hopcroft, J., & Kannan, R. (2020). *Foundations of data science*. Cambridge University Press.
- Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2022). *Introduction to algorithms* (4th ed.). MIT Press.
- Foster, I., Ghani, R., Jarmin, R. S., Kreuter, F., & Lane, J. (2020). *Big data and social science: Data science methods and tools for research and practice* (2nd ed.). CRC Press. <https://textbook.coleridgeinitiative.org/>
- Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: Data mining, inference, and prediction* (2nd ed.). Springer.
- Louridas, P. (2020). *Algorithms*. MIT Press.
- Marsland, S. (2015). *Machine learning: An algorithmic perspective* (2nd ed.). CRC Press.
- Norvig, P., & Russell, S. (2021). *Artificial intelligence: A modern approach: Global edition* (4th ed.). Pearson Education.

2 Robotics

2.1 Learning Outcomes

After successful completion of the module, students are able to:

- Contextualize the historical development of robotics and analyze emerging trends within the field.
- Differentiate between various classifications and types of robotic systems.
- Explain image processing techniques and their role in enabling computer vision for robotic applications.
- Compare and categorize different machine learning and AI techniques used in robotics and evaluate their applications across various fields and industries.
- Present hands-on projects that illustrate the successful integration of robotics and computer vision technologies.
- Construct and evaluate robotic applications through the effective use of electronics, sensors, and software components.
- Contextualize the ethical and legal considerations related to the development and implementation of robotic systems in society.

2.2 Module Content

This course explores robotic systems, emphasizing the integration of hardware and software components that enable robots to perform tasks autonomously or semi-autonomously. Students will learn about the design, operation, and programming of robotic systems, gaining practical skills in robotics and understanding the challenges of real-world applications.

- Overview and history of robotics
- Emerging trends in robotics
- Classification of robots
- Components of robotic systems
 - Electronics and actuators
 - Sensors, displays, controllers, and I/O
 - Operating systems and software
- Computer vision
 - Image processing techniques
 - Feature detection and object recognition

- Machine learning and AI in robotics
- Tools and software for robotics development
 - Programming languages (e.g., Python, C++)
 - Robot operating system (ROS)
- Hands-on laboratory projects
 - Building and programming robotic systems
 - Integrating computer vision with robotics
- Applications of robotic systems in various fields
- Ethical and legal considerations in robotics

Literature

Corke (2023), Ben-Ari & Mondada (2018), Szeliski (2022), Niku (2020), Brombach (2021)

References

- Ben-Ari, M., & Mondada, F. (2018). *Elements of robotics*.
- Brombach, L. (2021). *Practical robotics in C++: Build and program real autonomous robots using Raspberry Pi*. BPB Publications.
- Corke, P. (2023). *Robotics, vision and control: Fundamental algorithms in python*. Springer.
- Niku, S. B. (2020). *Introduction to robotics: Analysis, control, applications* (3rd ed.). Wiley.
- Szeliski, R. (2022). *Computer vision: Algorithms and applications* (2nd ed.). Springer.

3 Entrepreneurship for Computer Scientists

3.1 Learning Outcomes

After successful completion of the module, students are able to:

- Differentiate between traditional and digital business models, evaluating their strengths and weaknesses.
- Explain, compare, and apply strategies for identifying opportunities in technology-driven markets.
- Categorize different types of digital business models and their implications for entrepreneurs.
- Create and present a comprehensive business plan of a startup business, including market analysis, financial planning, and value proposition.
- Evaluate the governance and organizational structures that support tech startups and their scalability.
- Analyze marketing strategies and customer acquisition plans tailored specifically for tech startups.
- Assess legal considerations, including contracts, intellectual property law, and the ethical implications of technology in entrepreneurship.

3.2 Module Content

This course introduces students to the field of entrepreneurship, with an emphasis on entrepreneurial activities in a digital context. They will become familiar with relevant markets and gain the skills to identify opportunities within those markets, enabling them to develop products that deliver value to customers and offer a competitive advantage. By the end of the semester, students present a comprehensive business plan they created in small groups during the course.

Introduction to entrepreneurship

- Contribution of entrepreneurship to the economy and society
- Digital business ecosystems
- The unique challenges and opportunities faced by tech startups
- Technology trends shaping the future of entrepreneurship
- Case studies

Classic business models

- Definition, purpose and key components of a business
- Phases of starting, growing, and exiting a company
- The Business Model Canvas
- Challenges in business modelling

Digital business models

- Evolution of digital business models
- Types of digital business models (free, on-demand, e-commerce, marketplace, access-over-ownership, ecosystem, subscription, open-source, freemium, digital advertising, etc.)
- Identifying and exploiting opportunities in digital entrepreneurship
- Strategies for creating competitive advantage in digital entrepreneurship

Strategic business planning and execution

- Product development and management
- Value proposition development and theory
- Financial planning
- Fundraising and crowdfunding processes and strategies
- Pitching and communicating with technology-oriented stakeholders
- Governance and organizational structures of start-ups and early phase ventures
- Evaluating scalability and scaling strategies
- Business venture paradigms: Classic vs. lean start-up models
- Innovation methodologies: design thinking and rapid prototyping
- Networking and support system strategies

Business plan for tech startups: elements, role, and purpose

Marketing and sales strategy

- Digital marketing basics
- Customer acquisition plan and pricing strategy
- Building a brand identity for tech startups
- Customer acquisition and retention strategies

Legal considerations

- Understanding business structures
- Contracts and agreements
- Information technology law and intellectual property law

Ethical entrepreneurship and social impact

- Corporate social responsibility in tech
- Addressing ethical challenges in technology development
- Data privacy, security, and ethical AI considerations

Literature

Chaffey et al. (2019), Chaffey & Ellis-Chadwick (2022), Soltanifar et al. (2021), Allen (2019), Osterwalder & Pigneur (2010), Chishti et al. (2020), Kennard (2021), Fossen et al. (2024), Tidd & Bessant (2024)

References

- Allen, J. P. (2019). *Digital entrepreneurship*. Routledge.
- Chaffey, D., Edmundson-Bird, D., & Hemphill, T. (2019). *Digital business and e-commerce management* (7th ed.). Pearson.
- Chaffey, D., & Ellis-Chadwick, F. (2022). *Digital marketing: Strategy, implementation and practice* (8th ed.). Pearson.
- Chishti, S., Bartoletti, I., Leslie, A., & Millie, S. M. (Eds.). (2020). *The AI book: The artificial intelligence handbook for investors, entrepreneurs and FinTech visionaries*. Wiley.
- Fossen, F. M., McLemore, T., & Sorgner, A. (2024). *Artificial intelligence and entrepreneurship* (IZA Discussion Paper Series 17055). Institute of Labor Economics (IZA).
- Kennard, M. (2021). *Innovation and entrepreneurship*. Routledge.
- Osterwalder, A., & Pigneur, Y. (2010). *Business model generation: A handbook for visionaries, game changers, and challengers*.
- Soltanifar, M., Hughes, M., & Göcke, L. (Eds.). (2021). *Digital entrepreneurship: Impact on business and society*. Springer.
- Tidd, J., & Bessant, J. R. (2024). *Managing innovation: Integrating technological, market and organizational change* (8th ed.). Wiley.

4 Cloud Computing and Remote Work Tools

4.1 Learning Outcomes

After successful completion of the module, students are able to:

- Contextualize the key features and benefits of cloud computing in the context of modern workplace dynamics.
- Outline the evolution of cloud technologies and their significance for remote work.
- Differentiate between various remote work tools and their specific applications in project management, collaboration, and productivity.
- Evaluate the advantages and challenges of cloud service models (IaaS, PaaS, SaaS) and deployment models (public, private, hybrid, community).
- Explain core cloud technologies, including virtualization, containers, and serverless computing.
- Compare different programming models and performance metrics used in high-performance cloud computing.
- Categorize remote work tools based on their functionalities and integrations with cloud services.
- Present case studies that showcase successful cloud adoption and the design of cloud solutions tailored for remote work.

4.2 Module Content

Introduction and general topics

- Key features of cloud computing (scalability, cost-efficiency, accessibility, collaboration, security)
- Evolution of cloud technologies
- Remote work in the computer science labor market
- Benefits and challenges of remote work
- Benefits and challenges of cloud adoption
- The role and future of AI in remote work
- Security and privacy in cloud computing
- Future trends in cloud computing and remote work

Remote work tools

- Project management tools (e.g., Trello, Asana, Jira, monday.com)
- File sharing solutions (e.g., Google Drive, Dropbox, OneDrive, Box, Nextcloud)
- Communication and collaboration platforms (e.g., Slack, Microsoft Teams, Zoom, Github, Gitlab, Miro, Blink)
- Time tracking and task management (e.g., Toggl, Motion, Trello, Harvest, Clockify, Todoist)
- Monitoring tools (e.g., Kickidler, Hubstaff, Toggl, Time Doctor)
- Security solutions for remote work (e.g., NordVPN, Cisco AnyConnect, Okta)
- Integration and automation tools (e.g., Zapier, Make, Workato)
- Productivity tools (e.g., Notion, Trello, Basecamp)

Cloud service models

- Infrastructure as a service (IaaS)
- Platform as a service (PaaS)
- Software as a service (SaaS)
- Comparison of service models and providers

Cloud deployment models

- Public, private, hybrid, and community clouds
- Key considerations for choosing a deployment model

Core cloud technologies

- Virtualization
- Containers and orchestration (e.g., Docker, Kubernetes)
- Serverless computing
- Storage solutions

High-performance cloud computing

- Introduction to parallel and distributed programming models
- Concepts of high-performance computing (e.g., cluster, grid, middleware)
- Programming models (e.g., shared memory, message passing)
- Performance metrics and profiling

Cluster computing

- Definition and role
- Architecture of cluster systems
- Design considerations and challenges

Designing cloud solutions for remote work

- Architecting scalable solutions
- Integrating remote work tools with cloud services
- Enhancing user experience in remote environments
- Case studies of successful implementations

Hands-on projects and case studies

- Cloud service configuration
- Remote tool suite development
- Real-world cloud adoption case studies

Literature

Song (2023) Erl & Monroy (2023) Steen & Tanenbaum (2023) Ruparelia (2023) Sterling et al. (2017) Zamora & Robey (2021) Greene & Sanderson (2023) Gregg (2020)

References

- Erl, T., & Monroy, E. B. (2023). *Cloud computing: Concepts, technology, security, and architecture* (2nd ed.). Prentice Hall.
- Greene, A., & Sanderson, T. (2023). *Remote works: Managing for freedom, flexibility, and focus*. Berrett-Koehler Publishers.
- Gregg, B. (2020). *Systems performance: Enterprise and the cloud* (2nd ed.). Pearson.
- Ruparelia, N. B. (2023). *Cloud computing. Revised and updated edition*. MIT Press.
- Song, L. (2023). *The self-taught cloud computing engineer: A comprehensive professional study guide to AWS, Azure, and GCP*.
- Steen, M. van, & Tanenbaum, A. S. (2023). *Distributed systems* (4th ed.). Maarten van Steen.
- Sterling, T., Brodowicz, M., & Anderson, M. (2017). *High performance computing*. Morgan Kaufmann.
- Zamora, Y., & Robey, R. (2021). *Parallel and high performance computing*. Manning Publications.

5 Software Project, Process and Quality Management

6 Learning Outcomes

After successful completion of the module, students are able to:

- identify and critically evaluate various software development process models,
- apply the principles of software configuration management to effectively manage version control, track changes, ensure consistency across software components, and maintain the integrity of the software throughout its development lifecycle,
- examine process and quality management standards, including SEI CMMI, quality functions, metrics, statistical control parameters, and quality assurance tools,
- manage and oversee the full lifecycle of a software project,
- develop and present comprehensive software project management plans that address key elements such as scope, time, cost, quality, risk, resources, stakeholder engagement, communication strategies, and team management activities,
- present methodologies for software process improvement to systematically evaluate, enhance, and optimize software development processes, -explain quality management practices for effective testing and validation procedures.

Software development process models

- Introduction to software development methodologies (Waterfall, Agile-V, Spiral, Incremental) with a focus on appropriate application contexts
- Features and selection criteria for software development models
- Agile methodologies: Scrum, Kanban, and Extreme Programming (XP)
- Process optimization and quality assurance, including integration with industry standards (CMMI, ISO 9001)
- Case Studies: Implementation of software development models

Software Configuration Management (SCM)

- Principles and practices of SCM, including version control, change management, build automation, and configuration identification
- SCM tools (e.g., Git, Subversion, Jenkins) and their practical applications

Software Project Management (SPM)

- Overview of project management frameworks (PMI, Scrum, Kanban, PRINCE2)
- Defining project scope and controlling scope changes
- Schedule development and time management techniques
- Budgeting, estimating, and controlling costs in software projects
- Risk management, resource allocation, stakeholder engagement, communication, and team management

Software Quality and Process Management

- Fundamentals of software quality management, including quality assurance and quality control
- Overview of software quality models and standards (ISO 9001, Six Sigma, CMMI, Capability Maturity Models)
- Metrics and measurements for software quality
- Quality management activities throughout the software development lifecycle

6.1 Assessment

Project work with short presentation The students prepare a project documentation (10 to 15 pages) and present it in a 5 – 10 minute presentation (e.g. PowerPoint or self-designed posters). Weighting of the assessment method for the module grade: 70 % written, 30 % oral.

6.2 Literature

Richards & Ford (2021) Orosz (2023) Pressman & Maxim (2019) Lewis (2009) Farley (2022) Martin (2008) Taulli (2024)

References

- Farley, D. (2022). *Modern software engineering: Doing what works to build better software faster*. Addison-Wesley Professional.
- Lewis, W. E. (2009). *Software testing and continuous quality improvement* (3rd ed.). CRC Press/Auerbach Publications.
- Martin, R. C. (2008). *Clean code: A handbook of agile software craftsmanship*. Prentice Hall.
- Orosz, G. (2023). *The software engineer's guidebook: Navigating senior, tech lead, and staff engineer positions at tech companies and startups*. Pragmatic Engineer BV.

- Pressman, R., & Maxim, B. (2019). *ISE software engineering: A practitioner's approach* (9th ed.). McGraw-Hill Education.
- Richards, M., & Ford, N. (2021). *Fundamentals of software architecture: An engineering approach* (4th ed.). O'Reilly.
- Taulli, T. (2024). *AI-assisted programming: Better planning, coding, testing, and deployment*. O'Reilly Media.

7 Software Verification and Validation

7.1 Learning Outcomes

After successful completion of the module, students are able to:

- explain the concepts of software testing, verification and validation (V&V),
- distinguish between software testing and V&V, and explain their significance in software development projects,
- describe various testing and V&V techniques, including static analysis and dynamic testing,
- develop and implement effective test cases based on best practices in software testing,
- evaluate software quality metrics and tools for verification and validation
- explain test plans and strategies, ensuring comprehensive coverage of functional and non-functional aspects of software,
- interpret static analysis methods to detect code irregularities.
- describe and apply industry testing standards,
- identify and evaluate activities aimed at enhancing testing procedures.

7.2 Module Content

Introduction

- Overview of V&V techniques during the software development lifecycle
- Distinction between V&V activities
- Role and key goals of software V&V in the context of software security, quality, and reliability

Verification and validation techniques

- Introduction to specification-based and structural-based testing techniques
- Overview of static analysis techniques (e.g., code reviews, linting, formal methods)
- Code inspection and model-based verification
- Overview of validation testing types (dynamic, unit, integration, system, acceptance testing)
- Test-driven development and behavior-driven development
- Black-box and white-box testing techniques

- Functional and non-functional testing
- Limitations and challenges of verification and validation techniques

Designing and implementing test strategies

- Test case design techniques to enhance software testing:
 - Equivalence partitioning
 - Boundary value analysis
 - Decision tables
 - State transition testing
 - Risk-based testing
 - Regression testing
 - Pairwise testing
 - Exploratory testing
- Test case prioritization based on business criticality and frequency of use
- Differences between positive and negative test scenarios

Quality metrics, verification, and validation tools

- Overview of key quality metrics and software quality characteristics: Defect density, test coverage, fault tolerance, mean time to failure
- Tools for verification and validation activities:
 - Static analysis tools (e.g., PMD, Checkstyle)
 - Test automation frameworks (e.g., Selenium, JUnit, TestNG)
 - Continuous integration and testing tools (e.g., Jenkins, Travis CI)
- Best practices for using verification and validation tools

Industry standards and best practices

- Overview of software testing standards (e.g., IEEE 829, ISO/IEC 29119)
- Regulatory compliance and safety in software testing
- Ethical considerations in V&V
- Reporting and communicating test results

7.2.1 Emerging trends in software validation and verification

- AI and machine learning in software testing
- Testing in DevOps and Agile environments

7.3 Literature

Richards & Ford (2021) Orosz (2023) Pressman & Maxim (2019) Lewis (2009) Farley (2022)
Martin (2008) Jorgensen (2014) Taulli (2024)

References

- Farley, D. (2022). *Modern software engineering: Doing what works to build better software faster*. Addison-Wesley Professional.
- Jorgensen, P. C. (2014). *Software testing: A craftsman's approach* (4th ed.). CRC Press/Auerbach Publications.
- Lewis, W. E. (2009). *Software testing and continuous quality improvement* (3rd ed.). CRC Press/Auerbach Publications.
- Martin, R. C. (2008). *Clean code: A handbook of agile software craftsmanship*. Prentice Hall.
- Orosz, G. (2023). *The software engineer's guidebook: Navigating senior, tech lead, and staff engineer positions at tech companies and startups*. Pragmatic Engineer BV.
- Pressman, R., & Maxim, B. (2019). *ISE software engineering: A practitioner's approach* (9th ed.). McGraw-Hill Education.
- Richards, M., & Ford, N. (2021). *Fundamentals of software architecture: An engineering approach* (4th ed.). O'Reilly.
- Taulli, T. (2024). *AI-assisted programming: Better planning, coding, testing, and deployment*. O'Reilly Media.

8 Decision-Making with Artificial Intelligence & Big Data

siehe M-DSIBM 8 Decision-Making with Artificial Intelligence & Big Data

9 Deep Learning

tba