Modulhandbuch

Master Studiengang Informatik (M.Sc.)

Hochschule Landshut
gültig ab dem Sommersemester 2025

beschlossen am 28. Januar 2025

Inhaltsverzeichnis

IM100 Methodik Angewandter Wissenschaften . . . 3
IM110 Computer Systems . . . 4
IM120 Advanced Software Engineering . . . 6
IM130 Mobile Computing . . . 8
IM260 IoT Projektarbeit in der Praxis . . . 9
IM270 Industrial Internet of Things . . . 11
IM280 Hardware-Software-Codesign . . . 13
IM410 Operational Security . . . 15
IM430 Computer Algebra . . . 16
IM450 Mixed Reality . . . 17
IM470 Autonomes Fahren . . . 18
IM480 Semantic Web . . . 20
IM510 Advanced Machine Learning . . . 22
IM520 Computer Vision . . . 24
IM530 Deep Reinforcement Learning . . . 26
IM540 Causal Machine Learning . . . 28
IM550 Machine Learning Systems from Scratch . . . 30
IM560 Robotics and Manipulation . . . 32
IM600 Smart Grids . . . 34
IM720 Computational Geometry . . . 35
IM730 Kryptographie . . . 36
IM740 Inverse Problems . . . 37
IM750 Decision Making and Optimal Control . . . 38
IM820 Wissenschaftliches Seminar . . . 39
IM830 Masterarbeit . . . 40
BW210 Digital Entrepreneurship . . . 41
WM140 Prozess-Simulation . . . 42
WM230 Advanced Process Mining . . . 44

Methodik Angewandter Wissenschaften IM100

Modulverantwortlicher:	**Prof. Dr. Johannes Busse**
Dozent:	Prof. Dr. Johannes Busse
Studiengang:	Master
Modultyp:	Wahlpflichtmodul
Sprache:	Deutsch
Angebot:	Sommersemester
Dauer:	Ein Semester
Vorkenntnisse:	-
Voraussetzungen:	Bachelor in einem MINT-Fach
Leistungspunkte:	5
Arbeitsaufwand:	60 Stunden Präsenzzeit im Unterricht 90 Stunden Selbststudium
Lehrformen:	2 SWS seminaristischer Unterricht 2 SWS Praktikum in kleinen Gruppen (14tägig 4 Stunden)
Leistungsnachweise und Prüfung:	Leistungsnachweise im Praktikum, Studienarbeit 6 Wochen.

Qualifikationsziele und Inhalte:

Qualifikationsziele:

Kompetenzen zu selbstständigem wissenschaftlichem Arbeiten auf Niveau 7 (Master) des Europäischen Qualifikationsrahmens; akademische Kompetenzen nach Shaper (HRK-Gutachten 2012, S.29)

Lehrinhalte:

- Wissenschaftliches Schreiben: Texttypen, Schreibdenken, formalisierte Sprachen.
- Wissenschaftsverständnis von Informatik und Wirtschaftsinformatik (WI): WI als Design-Science; Memorandum Wirtschaftsinformatik.
- Methodologie von Informatik und WI; Kritik der Modellbildung in den angewandten Wissenschaften.

Literatur:

Ulrike Scheuermann: Schreibdenken. Schreiben als Denk- und Lernwerkzeug nutzen und vermitteln. Barbara Budrich / UTB, 2. Auflage 2013
Otto Kruse: Keine Angst vor dem leeren Blatt. Ohne Schreibblockaden durchs Studium. Campus, 12. Auflage 2008
Österle et al: Memorandum Wirtschaftsinformatik. Zeitschrift für betriebswirtschaftliche Forschung, 6, 2010, Nr. 62, S. 664 – 672. http://memo.iwi.unisg.ch/fileadmin/docs/zfbf.pdf
Hevner et al: Design Science in Information Systems Research. MIS Quarterly Vol. 28 No. 1, pp. 75-105 / March 2004
http://www.brian-fitzgerald.com/wp-content/uploads/2014/05/Hevner-et-al-2004-misq-des-sci.pdf
Sybille Krämer: Symbolische Maschinen: die Idee der Formalisierung im geschichtlichen Abriß, Wissenschaftliche Buchgesellschaft, Darmstadt 1988
Philipp Mayring: Einführung in die qualitative Sozialforschung. Beltz 2002

Weitere Literatur in der Veranstaltung

Computer Systems IM110

Responsible for the course:	**Prof. Dr. Christian Osendorfer**
Lecturer:	Prof. Dr. Christian Osendorfer
Study program:	Master
Course type:	Required module
Language:	English
Term:	Summer Semester
Duration:	One Semester
Prior knowledge:	Programming 1, Basic Computer Science, Programming 2
Requirements:	
Credit points:	5
Workload:	30 hours of attendance time in class 30 hours of attendance time in tutorials 90 hours self-study
Teaching methods:	2 semester hours per week seminar-based instruction 2 semester hours per week accompanying tutorials
Performance records and examinations:	Examination according to the current study and examination regulations.

Qualification goals and content:

Qualification goals:

Knowledge: By the end of this course, students will have a comprehensive understanding of the fundamental principles and concepts underlying computer systems. They will gain in-depth knowledge of operating systems, concurrency, networking, and distributed systems. Students will understand the intricate workings of processes, threads, virtual memory, file systems, and network protocols. They will also grasp the complexities of synchronization, deadlock prevention, and distributed consensus algorithms.

Skills: Students will develop practical skills in systems programming and design. They will learn to:

- Implement multi-threaded programs and manage concurrency issues.
- Design and develop system-level software using C and C++.
- Utilize operating system APIs effectively.
- Implement basic network protocols and distributed systems.
- Debug complex system-level issues.
- Optimize code for performance in a systems context.

Competencies: Students will be able to analyze, design, and implement complex computer systems. They will develop the competency to tackle real-world systems problems and contribute to the development of robust, efficient software systems. The focus of this course is on building a deep understanding of computer systems principles while considering both theoretical foundations and practical applications. Students will gain the competency to:

- Critically evaluate the design choices in existing computer systems.
- Develop efficient and scalable solutions to systems-level problems.
- Apply systems thinking to software design and implementation.
- Effectively communicate technical concepts related to computer systems.
- Analyze trade-offs in system design and make informed decisions.

Content:

The course covers core principles of computer systems, including operating systems, concurrent programming, and distributed systems. Starting from fundamental concepts in process management and virtual memory, the course progresses through various aspects of modern computer systems, including file systems, networking, and distributed computing. Students will explore both the theoretical underpinnings and practical implementations of these systems while developing an understanding of their design principles, performance characteristics, and security implications. Topics include:

- File systems
- Multiprocessing
- Virtual Memory
- Multithreading
- Networking

Literature:

Saltzer, J. H., Kaashoek, M. F. Principles of computer system design: an introduction. Morgan Kaufmann., 2009.

Matthews, Suzanne J., Tia Newhall, and Kevin C. Webb. Dive Into Systems: A Gentle Introduction to Computer Systems. No Starch Press, 2022.

Advanced Software Engineering IM120

Responsible for the course:	**Prof. Dr. Peter Scholz**
Lecturer:	Prof. Dr. Peter Scholz
Study program:	Master
Course type:	Required module
Language:	English
Term:	Winter Semester
Duration:	One Semester
Prior knowledge:	Software Engineering I
Requirements:	Knowledge acquired in the module Software Engineering 1 (IB061) or an equivalent module.
Credit points:	5
Workload:	45 hours of attendance time in class 15 hours of attendance time in lab sessions 90 hours self-study
Teaching methods:	2 semester hours per week seminar-based instruction 2 semester hours per week accompanying lab sessions
Performance records and examinations:	Examination according to the current study and examination regulations.

Qualification goals and content:

Qualification goals:

Upon successful completion of this module, students will:

- have an in-depth understanding of advanced concepts, methodologies, and techniques in software engineering.
- be able to analyze and optimize complex software development processes.
- apply state-of-the-art design patterns, architectural principles, and agile methodologies to solve real-world software challenges.
- critically evaluate current research topics in software engineering.
- collaborate effectively in interdisciplinary teams to address advanced software development problems.

Content:

- Advanced Software Architectures (Domain-driven design, Software product lines)
- Modern Software Development Practices (DevOps: Continuous integration/development pipelines, Automation tools, test-driven development, and behavior-driven development, Quality assurance in large-scale software projects)
- Agile and Hybrid Approaches (Scaled agile frameworks (e.g. SAFe, LeSS), Agile workflows for distributed and remote teams)
- Formal Methods and Modeling (Model-driven engineering, Formal methods for critical software systems)
- Teams (Team roles, developer strengths and growth, meetings, collaboration tools)
- Requirements (documentation, traceability, elicitation, scenarios, use cases, code navigation)
- Non-functional Properties (Safety, Security etc.)
- Software Quality (Models, Metrics, Reviews, Testing, Model Checking, Theorem Proofing)
- Research and Trends (Artificial intelligence in software engineering, Ethics and sustainability in software development, Emerging tools and technologies)

Literature:

Selected research papers and articles (to be provided during the lecture).
Davide Tosi: Software Engineering and Data Science, Special Issue in Future Internet, MDPI, 2023.
Julian Michael Bass: Agile Software Engineering Skills, Springer International Publishing, Cham, 2023.
Gerard O'Regan: Concise Guide to Software Engineering – From Fundamentals to Application Methods, Springer, 2022.
Ian Sommerville: Software Engineering, Pearson Studies, 2018.

Mobile Computing IM130

Responsible for the course:	**Prof. Dr. Markus Mock**
Lecturer:	Prof. Dr. Markus Mock
Study program:	Master
Course type:	Required module
Language:	English
Term:	Winter Semester
Duration:	One Semester
Prior knowledge:	
Requirements:	
Credit points:	5
Workload:	30 hours of attendance time in class 30 hours attendance time in lab sessions 90 hours self-study
Teaching methods:	2 semester hours per week seminar-based instruction 2 semester hours per week accompanying lab sessions
Performance records and examinations:	Unless otherwise specified in the current study and examination regulations: portfolio exam, i.e., written exam 90 minutes (weight factor: 65%) und study project (= implementation of a mobile app) over the course of the entire instructional semester (weight factor: 35%)

Qualification goals and content:

Qualification goals:

Students learn to develop applications for mobile devices efficiently. They master the entire development cycle from design to implementation and debugging of applications for mobile devices. They will also be able to use the necessary development tools and libraries and transfer software development methods and processes familiar to them from the Bachelor's degree program to mobile applications. In addition to the development of mobile applications, students are familiar with technologies, device classes, and design patterns of mobile computing and can evaluate them.

Content:

- Design patterns for applications based on mobile devices
- Mobile technologies
- Development of applications for mobile devices
- Design, implementation, and debugging of mobile applications
- Basics of the Android and iOS operating systems
- Memory management and security model in Android and iOS
- Design and testing of mobile applications

Literature:

Android und iOS online documentation
Selected papers
Brian Goetz: Java Concurrency in Practice
Schmidt et al: Pattern-Oriented Software Architecture Vol. 2: Patterns for Concurrent and Networked Systems.

IoT Projektarbeit in der Praxis IM260

Responsible for the course:	**Prof. Dr. Abdelmajid Khelil**
Lecturer:	Prof. Dr. Abdelmajid Khelil, Prof. Dr. Johann Uhrmann, Prof. Dr. Markus Mock, Prof. Dr. Eduard Kromer
Study program:	Master
Course type:	Wahlpflichtmodul
Language:	Deutsch / Englisch
Term:	Jedes Semester
Duration:	Ein Semester
Prior knowledge:	Kenntnisse der IoT Grundlagen
Requirements:	
Credit points:	5
Workload:	150 Stunden nicht ständig betreute Projektarbeit im Labor
Teaching methods:	4 SWS nicht ständig betreute Projektarbeit. Eigenverantwortliches Arbeiten der Studierenden in Teams von einer kritischen Größe, so dass das Auftreten typischer Schnittstellenprobleme gewährleistet ist, regelmäßige Projekttreffen mit dem Betreuer. Präsentation des Projektergebnisses zum Semesterende in einem Seminar.
Performance records and examinations:	Benotete individuelle schriftliche Ausarbeitung jedes Teammitglieds zum eigenen Beitrag im Projekt, im Team erstellte Gesamtdokumentation, im Team durchgeführte mündliche Präsentation des Projekts. Das Gesamtprojekt wird benotet. Die Note der Teammitglieder wird als Mittelwert aus der individuellen Note und der Projektnote gebildet.

Qualification goals and content:

Qualification goals:

Die Studierenden sind in der Lage wissenschaftliche, technische und soziale Kompetenzen einzusetzen, um komplexe IoT Projekte zu organisieren und durchzuführen. Sie haben Teamarbeit, Management und Kontrolle von IoT Projekten, selbstständige wissenschaftliche und technische Arbeit im Team trainiert. Sie können fachübergreifende Kenntnisse anwenden und Projektergebnisse professionell präsentieren. Durch die gezielte Anwendung von geeigneten Methoden aus Design Thinking und agilem Projektmanagement sowie durch eigenverantwortliche Durchführung von Projekten agieren die Studierenden ziel- und kundenorientiert.

Content:

Die kooperierenden Unternehmen bieten den Studierenden reale Problemstellungen aus den Domänen der IoT Architekturen, IoT Plattformen und deren Interoperabilität, IoT Protokollen, IoT Betriebssystemen, Semantic Web of Things, Zuverlässigkeit und Sicherheit in IoT, Fog und Edge Computing sowie Digital Twin. Die Problemstellung wird anhand definierter Anwendungsfälle beschrieben und während des Projektes als Product-Backlog vom Product-Owner des jeweiligen Unternehmens detailliert. Die Studierenden werden vom Dozenten und dem Coach des Innovationslabors fachlich betreut.

Literature:

Siehe Projektbeschreibung. Weitere Anregungen:

Ervin Varga, Drasko Draskovic, Dejan Mijic, "Scalable Architecture for the Internet of Things – An Introduction to Data-Driven Computing Platforms", O'Reilly, ISBN: 978-1-492-02412-5, 2018.

Boris Adryan, Dominik Obermaier, Paul Fremantlez, The Technical Foundations of IOT", Artech House, ISBN 978-1630812515, 2017.

Jean-Philippe Vasseur, Adam Dunkels, Ïnterconnecting Smart Objects with IP: The next Internet", Morgan Kaufmann, 2010.

Charalampos Doukas, "Building Internet of Things with the Arduino", CreateSpace Independent Publishing Platform, 2012.

J. R. Vic Winkler, "Securing the Cloud", Syngress, 2011.

Industrial Internet of Things IM270

Responsible for the course:	**Prof. Dr. Abdelmajid Khelil**
Lecturer:	Ahmed Chebaane, Aziz Abdennebi, Makki Ben Salem, Asma Guibene
Study program:	Master
Course type:	Required-elective module in the Edge Computing, Smart Manufacturing majors
Language:	English
Term:	Winter Semester
Duration:	One Semester
Prior knowledge:	
Requirements:	Knowledge acquired in the module Internet of Things (IB764) or an equivalent module; programming skills.
Credit points:	5
Workload:	45 hours of attendance time in class 15 hours of attendance time in lab sessions 90 hours self-study
Teaching methods:	2 semester hours per week seminar-based instruction 2 semester hours per week accompanying lab sessions
Performance records and examinations:	Examination according to the current study and examination regulations.

Qualification goals and content:

Qualification goals:

- Technical Knowledge: Understand the fundamentals of Edge Computing (EC), the publish-subscribe model, value stream analysis, and Catena-X.
- Practical Skills: Design and implement scalable, efficient systems for IIoT applications.
- Problem-Solving: Optimize processes using value stream analysis and interoperable data ecosystems with Catena-X.
- Collaboration: Work effectively in teams to solve real-world challenges bridging IT and OT in industrial environments.

Content:

Lecture:

- EC in IIoT: Fundamentals of EC; Device-edge-cloud continuum; Virtualization and containerization; Resource management such as reservation and task offloading; and time-sensitive EC.
- Event-driven architectures using Publish-Subscribe models: Established Pub-Sub protocols (MQTT/Sparkplug, DDS, OPC-UA); Topic naming; Data-Centric Networking (DCN); AI-supported flexible subscription and routing.
- Value Stream Analysis in IIoT: Advanced waste detection techniques using Value Stream Mapping (VSM) and KPIs to evaluate value-added and non-value-added activities.
- Catena-X: Interoperable data ecosystems; Data sovereignty and smart contracts; Data catalogs; Data lineage.

Lab:

- EC resource orchestration using Docker and Kubernetes
- AI-supported flexible subscription and forwarding
- Value stream analysis for Fischertechnik learning factory
- Use Eclipse Tractus-X for implementing data sharing, data lineage, and a data catalog.

Literature:

T. Qiu et al. ''Edge Computing in Industrial Internet of Things: Architecture, Advances and Challenges", IEEE Communications Surveys & Tutorials, 2020.
R. Buyya and S. N. Srirama ''Fog and Edge Computing: Principles and Paradigms", ISBN: 9781119525080, 2019.
R. Queiroz et al. ''Container-based Virtualization for Real-time Industrial Systems – A Systematic Review", ACM Computing Surveys, 2023.
Wu, Yulei. ''Cloud-edge orchestration for the Internet of Things: Architecture and AI-powered data processing", IEEE Internet of Things Journal, 2020.
Michelson, Brenda M. ''Event-driven architecture overview", Patricia Seybold Group 2.12, 2006.
Shi, R. et al. ''Towards Scalable Subscription Aggregation and Real-Time Event Matching in a Large-Scale Content-Based Network", ArXiv:1811.07088, 2018.
M. Rother, and J. Shook ''Learning to See: Value Stream Mapping to Create Value and Eliminate Muda", 2003.
S. Vinodh ''Lean Manufacturing: Fundamentals, Tools, Approaches, and Industry 4.0 Integration", 2024.
Catena-X Consortium ''Data Sovereignty and Interoperability in Automotive Supply Chains", retrieved from: https://catena-x.net, 2024.
Eclipse Foundation ''Eclipse Tractus-X Documentation", Retrieved from: https://eclipse-tractusx.github.io, 2024.

Hardware-Software-Codesign IM280

Modulverantwortlicher:	**Prof. Dr. Martin Pellkofer**
Dozent:	Prof. Dr. Martin Pellkofer
Studiengang:	Master
Modultyp:	Wahlpflichtmodul
Sprache:	Deutsch
Angebot:	Wintersemester
Dauer:	Ein Semester
Vorkenntnisse:	
Voraussetzungen:	Kenntnisse in der Programmiersprache C und in Digitaltechnik
Leistungspunkte:	5
Arbeitsaufwand:	30 Stunden Präsenzzeit im Unterricht 30 Stunden Präsenzzeit im Praktikum 90 Stunden Selbststudium
Lehrformen:	2 SWS seminaristischer Unterricht 2 SWS begleitendes Praktikum
Leistungsnachweise und Prüfung:	Leistungsnachweis im Praktikum, schriftliche Prüfung 60 Min.

Qualifikationsziele und Inhalte:

Qualifikationsziele:

Die Studierenden können das Design eines nebenläufigen Systems durch Datenflussgraphen modellieren und durch Graphen-Transformationen hinsichtlich Echtzeiteigenschaften optimieren. Die Studierenden sind in der Lage, werkzeuggestützt ein Design von Gleitkomma- in Festkommaarithmetik umzustellen und das Verhalten durch Simulation zu verifizieren. Sie können ein Design geeignet in Hardware und Software partitionieren, daraus automatisch VHDL- bzw. C-Code generieren und es auf einem System on Chip (SoC), bestehend aus Prozessor und FPGA, ablaufen lassen. Die Studierenden sind in der Lage, durch Processor-in-the-Loop- und FPGA-in-the-Loop-Simulation die Funktionalität und Qualität der Partitionierung auf dem SoC zu verifizieren.

Lehrinhalte:

- Motivation: Dualismus von Hardware- und Software-Design, Vorteile des gemeinsamen Designs
- Einführung in Matlab/Simulink und in die verwendeten Toolboxen (insbes. Fixed-Point Designer)
- Programmierbare Logikschaltkreise: Kenngrößen, Architekturen, IP-Cores, System on Chip
- Entwurf digitaler Schaltungen: Abstraktionsebenen, Verhaltens- und Struktursicht, Doppeldachmodell, Einsparpotenziale
- Syntheseschritte: Architektur-Synthese, RTL-Synthese, Layout-Synthese, Software-Synthese
- VHDL: Beschreibung des Verhaltens durch Struktur, sequentielle Prozesse oder Datenfluss; Signal- und Variablenzuweisungen, Simulationsablauf
- Hybride Sprachen: Einsatzmöglichkeiten, SystemC
- Datenfluss-Modellierung: Eigenschaften, Semantik, SDF, Leistungsanalyse, Transformationen an Datenflussmodellen; alternative Implementierungen durch Software und Hardware
- Analyse des Kontroll- und Datenflusses von C-Programmen: Bedeutung von Datenkanten und Kontrollkanten, Konstruktion von Kontrollflussgraphen und Datenflussgraphen
- Systemsynthese: HW-SW-Partitionierung, Entwurfsraumexploration, Zielkonflikt, Optimierungsproblem, Strategien zur Überdeckung und Beschneidung des Entwurfsraums
- Einführung in das Xilinx Zynq-7000 All programmable System-on-Chip (SoC) (ZedBoard)
- C- und HDL-Codegenerierung aus Simulink-Modellen und Matlab-Funktionen für eingebettete Systeme und Applizieren des Autocodes im External Mode
- Optimierung des HDL-Codes durch Transformationen an Datenflussmodellen (in Simulink)
- Verifikation des HDL-Codes mittels Cosimulation (mit QuestaSim)
- IP-Core-Generierung und Applizieren im External Mode (mit Xilinx Vivado)
- Verifikation des Hardware-Software-Systems durch Processor-in-the-Loop- und FPGA-in-the-Loop-Simulation und Profiling (mit Simulink)
- Beispiele: Verarbeitung von digitalen Audio- und Bilddatenströmen mit dem ZedBoard.

Literatur:

P. Schaumont: A Practical Introduction to Hardware/Software Codesign, 2. Auflage, 2013
J. Teich, C. Haubelt: Digitale Hardware/Software-Systeme-Synthese und Optimierung, Springer-Verlag, 2007
R. Gessler, T. Mahr: Hardware-Software-Codesign, Springer-Verlag, 2007
A. Sikora, R. Drechsler: Software-Engineering und Hardware-Design, Hanser, 2002
Handbücher der benutzten Hardware und Software

Operational Security IM410

Modulverantwortlicher:	**Prof. Dr. Johann Uhrmann**
Dozent:	Prof. Dr. Johann Uhrmann
Studiengang:	Master
Modultyp:	Wahlpflichtmodul
Sprache:	Deutsch
Angebot:	Sommersemester
Dauer:	Ein Semester
Vorkenntnisse:	
Voraussetzungen:	Kenntnisse die dem Inhalt des Moduls IT-Sicherheit (IB360) aus dem Bachelor Informatik entsprechen, Programmierkenntnisse.
Leistungspunkte:	5
Arbeitsaufwand:	45 Stunden Präsenzzeit im Unterricht 15 Stunden Präsenzzeit im Praktikum 90 Stunden Selbststudium
Lehrformen:	2 SWS seminaristischer Unterricht 2 SWS begleitendes Praktikum
Leistungsnachweise und Prüfung:	Prüfung gemäß der aktuellen Studien- und Prüfungsordnung. Prüfungsform wird hochschulweit im Studien- und Prüfungsplan bekanntgegeben.

Qualifikationsziele und Inhalte:

Qualifikationsziele:

- tiefgehendes Wissen über operative Sicherheitskonzepte und -prozesse, um diese in IT-Infrastrukturen zu implementieren und zu überwachen
- Kenntnis über Inhalt und Hintergründe aktueller Sicherheitsvorfälle und -entwicklungen
- Sicherheitsvorfälle erkennen, analysieren und effektiv darauf reagieren können
- Fähigkeit, operative Sicherheit in Organisationen strategisch zu planen, umuzusetzen und kontinuierlich zu verbessern

Lehrinhalte:

In der Vorlesung: Operational Security vs. IT-Sicherheit, Rollen und Verantwortlichkeiten der operativen Sicherheit, Bedrohungsmodelle und Angriffsvektoren, Sicherheitsarchitekturen, Planung und Durchführung von Incident Response, Schwachstellenmanagement, SOC-Konzepte
Im Praktikum: Endpoint Protection Systeme, Netzwerksicherheit, Schwachstellenscans, Log-Analyse

Literatur:

Michael Messner, Hacking mit Metasploit: Das umfassende Handbuch zu Penetration Testing und Metasploit, dpunkt.verlag, 2017
Security Monitoring: Proven Methods for Incident Detection on Enterprise Networks; ISBN: 0596518161
Malware Analyst's Cookbook: Tools and Techniques for Fighting Malicious Code; ISBN: 0470613033
Practical Malware Analysis: The Hands-On Guide to Dissecting Malicious Software; ISBN: 1593272901
Digital Forensics with Open Source Tools: Using Open Source Platform Tools for Performing Computer Forensics on Target Systems: Windows, Mac, Linux, Unix, ...; ISBN: 1597495867

Computer Algebra IM430

Modulverantwortlicher:	**Prof. Dr. Michael Sagraloff**
Dozent:	Prof. Dr. Michael Sagraloff
Studiengang:	Master
Modultyp:	Wahlpflichtmodul
Sprache:	Deutsch
Angebot:	Wintersemester
Dauer:	Ein Semester
Vorkenntnisse:	
Voraussetzungen:	
Leistungspunkte:	5
Arbeitsaufwand:	30 Stunden Präsenzzeit im Unterricht 30 Stunden Präsenzzeit im Praktikum 90 Stunden Selbststudium
Lehrformen:	2 SWS seminaristischer Unterricht 2 SWS Praktikum in kleinen Gruppen (14tägig 4 Stunden)
Leistungsnachweise und Prüfung:	Leistungsnachweise im Praktikum, 20 Min. mündliche Prüfung

Qualifikationsziele und Inhalte:

Qualifikationsziele:

Die Studierenden lernen schnelle Methoden im Umgang mit ganzen Zahlen und Polynomen kennen. Dies umfasst sowohl deren Studium in Bezug auf ihre theoretische Laufzeit als auch deren Implementierung und praktische Anwendbarkeit. Innerhalb des Praktikums erlernen die Studierenden den Umgang mit entsprechender Computer Algebra Software.

Lehrinhalte:

- Schnelle Multiplikation von Polynomen und ganzen Zahlen (Toom-Cook und FFT Multiplikation)
- Approximative Verfahren im Umgang mit Polynomen (Floating Point Arithmetik, Intervallarithmetik)
- Polynomdivision und Euklidischer Algorithmus
- Modulare Berechnungen und Anwendung: Chinesischer Restsatz, Primzahltests, Verschlüsselung
- Nullstellenbestimmung von univariaten Polynomen
- Lösen von polynomiellen Gleichungssystemen
- LLL Algorithmus und Anwendung

Literatur:

Joachim von zur Gathen, Jürgen Gerhard: Modern Computer Algebra, Cambridge University Press, 2013.
Wolfram Koepf: Computeralgebra: Eine algorithmisch orientierte Einführung, Springer, 2006.
Weitere Literatur in der Veranstaltung.

Mixed Reality IM450

Modulverantwortlicher:	**Prof. Dr. Christopher Auer**
Dozent:	Prof. Dr. Christopher Auer
Studiengang:	Master
Modultyp:	Wahlpflichtmodul
Sprache:	Deutsch
Angebot:	Wintersemester
Dauer:	Ein Semester
Vorkenntnisse:	
Voraussetzungen:	Programmierkenntnisse
Leistungspunkte:	5
Arbeitsaufwand:	30 Stunden Präsenzzeit im Unterricht 30 Stunden Präsenzzeit im Praktikum 90 Stunden Selbststudium
Lehrformen:	2 SWS seminaristischer Unterricht 2 SWS Praktikum (14-tägig)
Leistungsnachweise und Prüfung:	Portfolioprüfung: Studienarbeit (VR/AR-Anwendung, 6 Wochen, 35% der Prüfungsleistung), mündliche Prüfung (20 Minuten, 65% der Prüfungsleistung)

Qualifikationsziele und Inhalte:

Qualifikationsziele:

Die Studierenden kennen Methoden der Virtuellen und Augmentierten Realität (VR/AR) und wissen die dabei verwendeten Technologien einzuschätzen und praktisch bei der Implementierung von VR/AR-Anwendungen einzusetzen. Sie sind mit den möglichen Benutzerschnittstellen („User Interaction“) in virtuellen Realitäten auf Hardware- und Softwarebene vertraut und berücksichtigen dabei die damit verbundenen Aspekte und auftretenden Probleme bezüglich der menschlichen Wahrnehmung. Die Studierenden können eine VR/AR-Anwendung in einer 3D-Game Engine umsetzen (Unity3D) und kennen die dabei zu verwendenen Primitive (3D-Meshes, Licht, Materialien, Collider) und VR/AR-Entwicklungsframeworks.

Lehrinhalte:

- Grundlagen: Vektorräume, affine Räume, homogene Koordinaten, Koordinatentransformationen
- Grundlagen VR/AR: Wahrnehmungsaspekte, virtuelle dreidimensionale Welten
- VR/AR-Eingabe und Ausgabe: Trackingverfahren, kamerabasiertes/markerbasiertes Tracking, HMDs („head-mounted displays“)
- Benutzerinteraktion: Selektion, Navigation, Objektmanipulation
- Echtzeitaspekte: Latenz, Rendering, Kollisionserkennung
- Weitere Aspekte der augmentierten Realität

Literatur:

R. Dörner, W. Broll, P. Grimm, B. Jung (Hrsg.): Virtual und Augmented Reality (VR/AR) — Grundlagen und Methoden der Virtuellen und Augmentierten Realität, 2. Auflage, Springer 2013
J. Glover, J. Linowes: Complete Virtual Reality and Augmented Reality Development with Unity, Packt Publishing 2019
Weitere Literatur und Online-Resourcen werden in der Vorlesung bekanntgegeben

Autonomes Fahren IM470

Modulverantwortlicher:	**Prof. Dr. Martin Pellkofer**
Dozent:	Prof. Dr. Martin Pellkofer
Studiengang:	Master
Modultyp:	Wahlpflichtmodul
Sprache:	Deutsch
Angebot:	Sommersemester
Dauer:	ein Semester
Vorkenntnisse:	Programmierkenntnisse in C/C++, Matlab/Simulink oder Python
Voraussetzungen:	
Leistungspunkte:	5
Arbeitsaufwand:	30 Stunden Präsenzzeit im Unterricht 30 Stunden Präsenzzeit im Praktikum 90 Stunden Selbststudium
Lehrformen:	2 SWS seminaristischer Unterricht 2 SWS begleitendes Praktikum
Leistungsnachweise und Prüfung:	Prüfung gemäß der aktuellen Studien- und Prüfungsordnung. Prüfungsform wird hochschulweit im Studien- und Prüfungsplan bekanntgegeben.

Qualifikationsziele und Inhalte:

Qualifikationsziele:

Die Studierenden kennen den Stand der Technik hinsichtlich autonom fahrender Landfahrzeuge. Dies beinhaltet die eingesetzte Sensorik, Sensordatenverarbeitung, Sensordatenfusion, Fahrzeugregelung, Navigation und Planung, sowie Systemdesign und Test. Die Studierenden kennen ferner die Vorteile aktiv sehender autonomer Systeme sowie den 4D-Ansatz. In individuell ausgewählte Themenbereiche mit Bezug zum Autonomen Fahren haben sich die Studierenden tiefer eingearbeitet, praktische Entwicklungsarbeit geleistet und ihre Ergebnisse präsentiert.

Lehrinhalte:

- Historischer und thematischer Überblick
- Sensoren für autonomes Fahren
- Paradigmen beim Autonomen Fahren: Modulare Pipelines, End-to-End-Learning, Direct Perception
- MLP, CNN, Imitation Learning, Direct Perception
- Fahrdynamik und Fahrzeugregelung
- Odometrie, SLAM, Lokalisierung
- Straßen- und Fahrspurerkennung
- Stereoskopisches Sehen, Freiraumerkennung, Optischer Fluss, Szenenfluss
- Objektdetektion: Detektionsleistung, Region-Based-CNN, Fast(er) R-CNN, 3D-Objektdetektion
- Objektverfolgung: Single & Multi Object Tracking, Assoziationsproblem, Track-Level-Fusion
- Sensordatenfusion und Zustandsschätzung: Bayes-Filter, (Erweiterter) Kalman-Filter
- Routenplanung, Verhaltensplanung, Trajektorienplanung
- Aktives Sehen, Blickrichtungssteuerung
- 4D-Ansatz
- Projektarbeit und -präsentation zu einem selbst ausgewählten Thema

Literatur:

H. Winner, S. Hakuli, F. Lotz, C. Singer: Handbuch Fahrerassistenzsysteme, 3. Auflage, Springer, 2015
E. D. Dickmanns: Dynamic Vision for Perception and Control of Motion, Springer, 2007
M. Maurer, J. Ch. Gerdes, B. Lenz, H. Winner (Hrsg.): Autonomes Fahren: Technische, rechtliche und gesellschaftliche Aspekte, Springer, 2015
M. Botsch, W. Utschick: Fahrzeugsicherheit und autonomes Fahren, Hanser, 2020
Q. Zhou, Z. Shen, B. Yong, R. Zhao, P. Zhi: Theories and Practices of Self-Driving Vehicles, Elsevier, 2022
M. Du: Autonomous Vehicle Technology, Springer, 2023

Semantic Web IM480

Modulverantwortlicher:	**Prof. Dr. Johannes Busse**
Dozent:	Prof. Dr. Johannes Busse
Studiengang:	Master
Modultyp:	Wahlpflichtmodul
Sprache:	Deutsch
Angebot:	Sommersemester
Dauer:	ein Semester
Vorkenntnisse:	Grundlagen Prädikatenlogik
Voraussetzungen:	-
Leistungspunkte:	5
Arbeitsaufwand:	30h seminaristisch 30h Übung 90h Selbststudium
Lehrformen:	2 SWS seminaristischer Unterricht 2 SWS begleitendes Praktikum
Leistungsnachweise und Prüfung:	siehe SPO

Qualifikationsziele und Inhalte:

Qualifikationsziele:

Die Studierenden sind in der Lage, sog. ''Knowlegde Graphs" so auf eine semantische Ebene zu heben, dass mit Semantic Web Ontologien in den Graphen enthaltendes implizites Wissen explizit gemacht werden kann. Die TN können mit geeigneten Tools Wissen als RDF(S), SKOS und OWL2-Ontologien modellieren, sowie verschiedene Linked Open Data (LOD) -Quellen und Knowlede Graphs abfragen, zusammenführen, RDF(S) oder OWL 2 Inferencing ausführen, und das Ergebnis wieder als Knowledge Graph z.B. nachfolgenden Machine-Learning Schritten zur Verfügung stellen.

Lehrinhalte:

Knowledge Graphs

- Linked Open Data (LOD), Linked Open Goverment Data (LOGD)
- Abfragesprachen für Semantische Knowledge Graphs wie insbes. SPARQL u.a.

Ontologien

- Grundlagen der Description Logic und OWL2
- Grundlagen des Knowledge Engineerings
- RDF(S) inferencing, Inferencing über SKOS, OWL2

Literatur:

- Heiner Stuckenschmidt: Ontologien.Konzepte, Technologien und Anwendungen. Springer 2011
- Allemang, Dean: Semantic web for the working ontologist. Effective Modeling for Linked Data, RDFS, and OWL. Third Edition 2020. https://flatp20.bib-bvb.de/search?bvnr=BV047056521 .
- Bernhard Humm: Grundlagen der Wissensrepräsentation, in Hoppe, Thomas: Semantische Suche. Grundlagen und Methoden semantischer Suche von Textdokumenten. Springer Vieweg 2020, Kap. 4. https://flatp20.bib-bvb.de/search?bvnr=BV044934607
- Tassilo Pellegrini, Harald Sack, Sören Auer: Linked Enterprise Data. Management und Bewirtschaftung vernetzter Unternehmensdaten mit Semantic Web Technologien. Springer Vieweg 2014. https://flatp20.bib-bvb.de/search?bvnr=BV042176858

Advanced Machine Learning IM510

Responsible for the course:	**Prof. Dr. Sandra Eisenreich**
Lecturer:	Prof. Dr. Sandra Eisenreich, Prof. Dr. Eduard Kromer, Prof. Dr. Christian Osendorfer, Prof. Dr. Veronika Wanner-Seidl
Study program:	Master
Course type:	Required-elective module
Language:	English
Term:	Summer Semester
Duration:	One Semester
Prior knowledge:	Machine Learning I-III, Statistics, Programming 1, Artificial Intelligence 2
Requirements:	
Credit points:	5
Workload:	30 hours of attendance time in class 30 hours of attendance time in tutorials 90 hours self-study
Teaching methods:	2 semester hours per week seminar-based instruction 2 semester hours per week accompanying tutorials
Performance records and examinations:	Examination according to the current study and examination regulations.

Qualification goals and content:

Qualification goals:

Knowledge: By the end of this course, students will understand the fundamental principles and theoretical frameworks of advanced generative modeling and probabilistic approaches in machine learning. They will gain comprehensive knowledge of various generative model architectures, including autoregressive models, flow models, GANs, energy-based models, and diffusion models. Students will understand the mathematical foundations of maximum likelihood learning and probabilistic modeling, distinguishing between discriminative and generative approaches.

Skills: Students will develop practical skills in implementing and training different types of generative models in a project-based setting. They will learn to:

- Design and implement various generative model architectures using modern deep learning frameworks
- Apply probabilistic reasoning techniques to solve complex machine learning problems
- Evaluate and compare different generative modeling approaches
- Debug and optimize advanced machine learning models
- Analyze and interpret results from different types of generative models

Competencies: Students will be able to independently assess and select appropriate generative modeling approaches for specific applications. They will develop the competency to conduct research in the field of generative AI and contribute to the development of novel approaches. The focus of this advanced course is on building the ability to understand, implement, and advance state-of-the-art generative modeling techniques while considering both theoretical foundations and practical applications. Students will gain the competency to:

- Critically evaluate the strengths and limitations of different generative modeling approaches
- Develop novel solutions to complex machine learning problems using generative models
- Apply probabilistic reasoning to real-world problems
- Effectively communicate technical concepts and results in the field of generative modeling

Content:

The course covers advanced topics in machine learning with a focus on generative modeling and probabilistic approaches. Starting from fundamental concepts in probabilistic modeling, the course progresses through various types of generative models, from classical approaches to state-of-the-art architectures. Students will explore the theoretical foundations and practical implementations of these models while developing an understanding of their applications and limitations.

- Probabilistic Modeling (discriminative vs. generative models and Bayesian Networks)
- Autoregressive Models
- Maximum Likelihood Learning
- Latent Variable Models
- Flow Models
- GANs (Advanced Topics)
- Energy-based Models
- Score-based Models
- Diffusion Models
- Probabilistic Reasoning and Discrete Deep Generative Models

Literature:

Current research on generative deep learning and self-supervised / unsupervised learning; selected papers from major conferences (NeurIPS, ICML, ICLR) covering latest developments in generative modeling
Kevin P. Murphy, Probabilistic Machine Learning: An Introduction, MIT Press, 2022
Kevin P. Murphy, Probabilistic Machine Learning: Advanced Topics, MIT Press, 2023
Christopher M. Bishop, Deep Learning: Foundations and Concepts, Springer, 2024

Computer Vision IM520

Responsible for the course:	**Prof. Dr. Veronika Wanner-Seidl**
Lecturer:	Prof. Dr. Veronika Wanner-Seidl, Prof. Dr. Christian Osendorfer, Prof. Dr. Eduard Kromer, Prof. Dr. Sandra Eisenreich
Study program:	Master
Course type:	Required-elective module
Language:	Englisch
Term:	Summer Semester
Duration:	One Semester
Prior knowledge:	Linear Algebra (Matrices, Vectors, Norms, Scalar/Vector Products, Orthogonality), Machine Learning (Gradient Descent, Loss Functions, Optimization, Neural Networks)
Requirements:	
Credit points:	5
Workload:	30 hours of attendance time in class 30 hours of attendance time in tutorials 90 hours self-study
Teaching methods:	2 semester hours per week seminar-based instruction 2 semester hours per week accompanying tutorials
Performance records and examinations:	Examination according to the current study and examination regulations.

Qualification goals and content:

Qualification goals:

Knowledge: By the end of this course, students will understand the fundamental principles and advanced concepts of computer vision. They will gain comprehensive knowledge of object detection and scene understanding using various complex deep learning architectures. Students will understand neural rendering, 3D representations, 3D reconstruction, and deep learning for 3D data.

Skills: Students will develop practical skills in:

- Implementing and training various computer vision learning algorithms for object detection and scene understanding
- Utilizing vision-language models for multimodal tasks involving images and text
- Understanding different 3D representations
- Understanding 3D reconstruction with NeRF and 3D Gaussian Splatting
- Self-supervised learning of scene representations via 3D-aware auto-encoding
- Implementing and training 3D generative models to synthesize new 3D structures.

Competencies: Students will be able to independently select, implement, and optimize deep learning-based methods for complex computer vision problems. The focus of this advanced course is on building the ability to understand and implement current deep learning-based computer vision techniques while considering both theoretical foundations and practical applications. Students will gain the competency to:

- Select and adapt appropriate computer vision learning algorithms for specific applications
- Assess the strengths and limitations of CNNs, vision transformers, and vision-language models in various applications
- Implement CNNs, vision transformers, and vision-language models in real-world scenarios
- Apply neural rendering and generative 3D models for synthesizing and manipulating 3D structures and scenes
- Evaluate and compare different 3D representations for deep learning
- Effectively communicate technical concepts and results in the field of computer vision
- Critically assess the latest developments in computer vision research and conduct their own research in this field
- Apply computer vision concepts to real-world applications in robotics, augmented reality, autonomous systems, and healthcare.

Content:

The course covers advanced topics in computer vision, combining theoretical knowledge with practical applications. The content includes:

- Fundamentals of Computer Vision
- Vision language models
- Various 3D representations for Deep Learning
- Neural rendering and deep learning-based 3D reconstruction
- Generative 3D models
- Fundamentals of geometric deep learning
- Other current research topics and applications in computer vision.

Literature:

Current research on computer vision; selected papers from major conferences covering latest developments in computer vision.

Deep Reinforcement Learning IM530

Responsible for the course:	**Prof. Dr. Eduard Kromer**
Lecturer:	Prof. Dr. Eduard Kromer, Prof. Dr. Christian Osendorfer, Prof. Dr. Sandra Eisenreich, Prof. Dr. Veronika Wanner-Seidl
Study program:	Master
Course type:	Required-elective module
Language:	English
Term:	Summer Semester
Duration:	One Semester
Prior knowledge:	Machine Learning I-III, Reinforcement Learning, Programming 1, Artificial Intelligence 2
Requirements:	
Credit points:	5
Workload:	30 hours of attendance time in class 30 hours of attendance time in tutorials 90 hours self-study
Teaching methods:	2 semester hours per week seminar-based instruction 2 semester hours per week accompanying tutorials
Performance records and examinations:	Examination according to the current study and examination regulations.

Qualification goals and content:

Qualification goals:

Knowledge: By the end of this course, students will understand the fundamental principles and advanced concepts of deep reinforcement learning. They will gain comprehensive knowledge of policy gradient methods, actor-critic architectures, value function approaches, and Q-learning with deep neural networks. Students will understand the theoretical foundations of optimal control, planning, and model-based reinforcement learning methods. Additionally, they will grasp advanced concepts such as exploration strategies, offline reinforcement learning, variational inference in control, and inverse reinforcement learning.

Skills: Students will develop practical skills in:

- Implementing and training various deep reinforcement learning algorithms
- Designing and optimizing policy gradient and actor-critic architectures
- Developing and evaluating model-based reinforcement learning systems
- Implementing advanced exploration strategies
- Applying offline reinforcement learning techniques to real-world problems
- Creating and training sequence learning models in reinforcement learning contexts
- Debugging and analyzing complex reinforcement learning systems
- Using modern deep learning frameworks for reinforcement learning applications

Competencies: Students will be able to independently design, implement, and optimize deep reinforcement learning solutions for complex problems. The focus of this advanced course is on building the ability to understand and implement current reinforcement learning techniques while considering both theoretical foundations and practical applications. Students will gain the competency to:

- Select and adapt appropriate reinforcement learning algorithms for specific applications
- Design and implement novel solutions to complex control and decision-making problems
- Evaluate and compare different reinforcement learning approaches
- Integrate sequence learning with reinforcement learning methods
- Develop robust and efficient exploration strategies
- Apply inverse reinforcement learning to learn from expert demonstrations
- Effectively communicate technical concepts and results in the field of reinforcement learning

Content:

The course covers advanced topics in deep reinforcement learning, progressing from fundamental concepts to state-of-the-art approaches. Students will explore both theoretical foundations and practical implementations of various reinforcement learning methods while developing an understanding of their applications and limitations. The content includes:

- Policy Gradients
- Actor-Critic Algorithms
- Value Function Methods
- Deep RL with Q-Functions
- Advanced Policy Gradients
- Optimal Control & Planning
- Model-based Reinforcement Learning
- Model-based Policy Learning
- Exploration
- Offline Reinforcement Learning
- Variational Inference & Control
- Inverse Reinforcement Learning
- Reinforcement Learning with Sequence Learning

Literature:

Current research on deep reinforcement learning; selected papers from major conferences (NeurIPS, ICML, ICLR) covering latest developments in deep reinforcement learning and multi-agent reinforcement learning.
Reinforcement Learning: An Introduction (2nd Edition); Sutton and Barto; MIT Press, 2018.
Algorithms for Reinforcement Learning; Csaba Szepesvari; Morgan & Claypool, 2010.
Reinforcement Learning and Optimal Control; Bertsekas; Athena Scientific, 2019.
Algorithms for Decision Making; Kochenderfer, Wheeler, Wray; MIT Press, 2022.
Multi-Agent Reinforcement Learning - Foundations and Modern Approaches; Albrecht, Christianos, Schäfer; MIT Press, 2024.

Causal Machine Learning IM540

Responsible for the course:	**Prof. Dr. Sandra Eisenreich**
Lecturer:	Prof. Dr. Sandra Eisenreich, Prof. Dr. Veronika Wanner-Seidl, Prof. Dr. Eduard Kromer, Prof. Dr. Christian Osendorfer
Study program:	Master
Course type:	Required-elective module
Language:	English
Term:	Summer Semester
Duration:	One Semester
Prior knowledge:	Machine Learning I-III, Statistics, Artificial Intelligence 1 & 2, Programming 1
Requirements:	
Credit points:	5
Workload:	30 hours of attendance time in class 30 hours of attendance time in tutorials 90 hours self-study
Teaching methods:	2 semester hours per week seminar-based instruction 2 semester hours per week accompanying tutorials
Performance records and examinations:	Examination according to the current study and examination regulations.

Qualification goals and content:

Qualification goals:

Knowledge: By the end of this course, students will understand the fundamental principles of causal inference and its integration with machine learning. They will gain comprehensive knowledge of probabilistic graphical models, structural causal models, and the key concepts of confounding, mediation, and collider bias. Students will understand the theoretical foundations of randomized controlled trials, methods for causal inference from observational data, and approaches to handling distributional shift in machine learning applications.

Skills: Students will develop practical skills in:

- Implementing and analyzing causal inference methods using modern machine learning frameworks
- Designing and evaluating randomized controlled trials
- Identifying and addressing confounding variables in observational studies
- Applying techniques for handling distributional shift in machine learning systems
- Critically analyzing and presenting current research papers in causal machine learning
- Developing solutions that combine causal inference with machine learning approaches

Competencies: Students will be able to independently assess and select appropriate causal inference methods for specific applications. They will develop the competency to conduct studies at the intersection of causality and machine learning. The focus of this advanced course is on building the ability to understand and implement causal machine learning techniques while considering both theoretical foundations and practical applications. Students will gain the competency to:

- Design and implement causal inference studies in real-world settings
- Evaluate the causal implications of machine learning systems
- Develop robust machine learning solutions that account for distributional shift
- Critically assess the latest developments in causal machine learning research
- Effectively communicate complex causal concepts and findings to both technical and non-technical audiences

Content:

The course covers advanced topics in causal machine learning, combining theoretical foundations with practical applications. Starting from fundamental concepts in probabilistic graphical models and structural causal models, the course progresses through various aspects of causal inference and its integration with modern machine learning approaches. Students will explore both theoretical frameworks and practical implementations while developing an understanding of their applications and limitations. The content includes:

- Probabilistic Graphical Models and Structural Causal Models
- Confounders, Colliders & Mediators
- Randomized Controlled Trials
- Causal Inference from Observ. Data
- Distributional Shift in ML
  - Invariance: Stable Correlations
  - Meta-Learning
  - Reinforcement Learning with Binary Preference
- Paper Presentations on Current Research on Causal Machine Learning

Literature:

Current research on causal machine learning; selected papers from major conferences (NeurIPS, ICML, ICLR) covering latest developments in generative modeling.

Machine Learning Systems from Scratch IM550

Responsible for the course:	**Prof. Dr. Eduard Kromer**
Lecturer:	Prof. Dr. Eduard Kromer, Prof. Dr. Christian Osendorfer, Prof. Dr. Sandra Eisenreich, Prof. Dr. Veronika Wanner-Seidl
Study program:	Master
Course type:	Required-elective module
Language:	English
Term:	Summer Semester
Duration:	One Semester
Prior knowledge:	Machine Learning I-III, Natural Language Processing, Advanced Machine Learning, Programming 1
Requirements:	
Credit points:	5
Workload:	30 hours of attendance time in class 30 hours of attendance time in tutorials 90 hours self-study
Teaching methods:	2 semester hours per week seminar-based instruction 2 semester hours per week accompanying tutorials
Performance records and examinations:	Examination according to the current study and examination regulations.

Qualification goals and content:

Qualification goals:

Knowledge: By the end of this course, students will understand the fundamental principles and implementation details of large-scale machine learning systems. They will gain comprehensive knowledge of transformer architectures, optimization techniques, and the infrastructure required for LLM inference. Students will understand hardware architectures including GPUs and TPUs, compiler design for machine learning, distributed training approaches, and techniques for handling long context lengths. They will also gain deep knowledge of memory management, sparsity optimization, and the design of efficient inference systems.

Skills: Students will develop practical skills in:

- Implementing transformer architectures from scratch
- Building and optimizing LLM inference systems
- Programming for specialized hardware architectures (GPU/TPU)
- Developing and debugging compiler optimizations for ML
- Implementing distributed training systems
- Managing memory efficiently in large-scale ML systems
- Designing and optimizing sparse computation techniques
- Creating efficient inference pipelines
- Handling and optimizing long context length scenarios
- Performance profiling and optimization of ML systems

Competencies: Students will be able to independently design, implement, and optimize machine learning systems from fundamental principles. The focus of this advanced course is on building the ability to understand and implement the entire ML system stack, from hardware interfaces to high-level model architectures. Students will gain the competency to:

- Design and implement efficient transformer-based architectures
- Make informed decisions about hardware acceleration strategies
- Develop and optimize custom ML compilers
- Create scalable distributed training solutions
- Implement memory-efficient ML systems
- Build robust inference systems
- Analyze and optimize system performance
- Balance trade-offs between computational efficiency and model capability
- Effectively communicate complex system design decisions

Content:

The course covers the full stack of machine learning systems, focusing on practical implementation and optimization. Starting from fundamental architectural components, the course progresses through various aspects of building efficient and scalable ML systems, with particular emphasis on transformers and large language models. The content includes:

- Transformer Model Architecture and Optimizations
- LLM Inference Systems
- Hardware Architectures: GPU, TPU
- Languages and Compilers
- Distributed Training
- Long Context Lengths
- Memory, Sparsity
- Inference Systems

Literature:

Current research and technical documentation on machine learning systems; selected papers from major conferences (NeurIPS, ICML, ICLR) covering latest developments in ML systems.

Robotics and Manipulation IM560

Responsible for the course:	**Prof. Dr. Christian Osendorfer**
Lecturer:	Prof. Dr. Christian Osendorfer, Prof. Dr. Sandra Eisenreich, Prof. Dr. Eduard Kromer, Prof. Dr. Veronika Wanner-Seidl
Study program:	Master
Course type:	Required-elective module
Language:	English
Term:	Summer Semester
Duration:	One Semester
Prior knowledge:	Linear Algebra, Calculus, Programming 1
Requirements:	
Credit points:	5
Workload:	30 hours of attendance time in class 30 hours of attendance time in tutorials 90 hours self-study
Teaching methods:	2 semester hours per week seminar-based instruction 2 semester hours per week accompanying tutorials
Performance records and examinations:	Examination according to the current study and examination regulations.

Qualification goals and content:

Qualification goals:

Knowledge: By the end of this course, students will have a comprehensive understanding of the fundamental algorithmic approaches and mathematical principles underlying robot manipulation in unstructured environments. They will gain in-depth knowledge of:

- Robot kinematics, including forward, inverse, and velocity kinematics.
- Trajectory generation and motion planning, including collision-free motion, task-and-motion planning and planning under uncertainty.
- Dynamics and control of robotic systems, focusing on nonlinear dynamics, model-based control.
- Grasping and manipulation techniques for handling physical objects effectively.

Skills: Students will develop the ability to implement and apply computational methods for solving robotics problems. They will learn to:

- Analyze and model the dynamics of robotic systems.
- Design and implement algorithms for trajectory generation, motion planning, and control.
- Develop and evaluate solutions for grasping and manipulating objects in complex, real-world settings.
- Use state-of-the-art tools and frameworks for robotics development.

Competencies: Students will be equipped to design, analyze, and implement robotic manipulation systems for applications in environments like homes, restaurants, and factories. They will develop the competency to:

- Independently assess and select suitable planning and control methods for specific robotic tasks.
- Formulate and solve problems involving robot dynamics, motion planning, and control.
- Critically evaluate and improve existing robotics algorithms.
- Communicate complex technical concepts and results effectively in the field of robotics.

Content:

This course provides a thorough introduction to the algorithmic and computational methods for enabling robots to autonomously manipulate objects in unstructured environments. Topics include:

- Configuration Space: Understanding and representing robot joint and workspace configurations.
- Rigid-Body Motions: Mathematical modeling of rigid-body transformations.
- Forward and Inverse Kinematics: Analysis and computation of robot poses and joint motions.
- Velocity Kinematics and Statics: Deriving and applying relationships between joint velocities, end-effector velocities, and forces.
- Kinematics of Closed Chains: Solving kinematic constraints in complex mechanisms.
- Dynamics of Open Chains: Modeling and analyzing the nonlinear dynamics of robotic manipulators.
- Trajectory Generation and Motion Planning: Generating feasible and efficient trajectories, including collision avoidance and planning under uncertainty.
- Robot Control: Techniques for robust, model-based control of robots.
- Grasping and Manipulation: Approaches for securely grasping and manipulating physical objects in real-world scenarios.

The course emphasizes a balance between theoretical foundations and practical applications, preparing students for advanced work in robotics and manipulation.

Literature:

Lynch K. M. and Park, F.C., Modern Robotics: Mechanics, Planning, and Control, Cambridge University Press, 2017.

Smart Grids IM600

Responsible for the course:	**Prof. Dr. Alexander Wallis**
Lecturer:	Prof. Dr. Alexander Wallis
Study program:	Master
Course type:	Required-elective module in the Machine Learning, Smart Manufactoring, and Edge Computing majors
Language:	English
Term:	Summer Semester
Duration:	One Semester
Prior knowledge:	
Requirements:	
Credit points:	5
Workload:	30 hours of attendance time in class 30 hours attendance time in lab sessions 90 hours self-study
Teaching methods:	2 semester hours per week seminar-based instruction 2 semester hours per week accompanying tutorials
Performance records and examinations:	Portfolio exam, i.e., written exam 60 minutes (weight factor: 65%) and study project over the course of the entire instructional semester (weight factor: 35%)

Qualification goals and content:

Qualification goals:

Students understand the fundamental concepts of Smart Grids and their significance for the energy transition. They are able to analyze the structure and architecture of distributed systems and apply their communication protocols to intelligent energy systems. Additionally, students can assess security concepts and challenges in the operation and optimization of modern power grids.

Content:

- Introduction to Smart Grids: Definitions, objectives, and challenges
- Energy generation, storage, and consumption in intelligent networks
- Communication infrastructures: Protocols and standards
- Energy management systems for households with agent-based modeling
- Grid optimization through AI methods: Forecasting, classification, and optimization
- Learning methods in distributed systems, particularly Federated Learning

Literature:

K. Siozios, D. Anagnostos, D. Soudris: IoT for Smart Grids: Design Challenges and Paradigms, MIT Press 2005
B. Buchholz , Z. Styczynski: Smart Grids – Fundamentals and Technologies in Electric Power Systems of the future, Springer 2020
V. C. Gungor et al., Smart Grid Technologies: Communication Technologies and Standards, 2011
X. Fang, S. Misra, G. Xue and D. Yang, Smart Grid — The New and Improved Power Grid: A Survey, 2012
Current research on Smart Grid and AI development; selected papers from major conferences (ACM e-Energy, IEEE SmartGridComm, AAMAS)

Computational Geometry IM720

Modulverantwortlicher:	**Prof. Dr. Michael Sagraloff**
Dozent:	Prof. Dr. Michael Sagraloff
Studiengang:	Master
Modultyp:	Wahlpflichtmodul
Sprache:	Deutsch
Angebot:	Sommersemester
Dauer:	Ein Semester
Vorkenntnisse:	
Voraussetzungen:	Kenntnisse, die dem Inhalt des Moduls Algorithmen und Datenstrukturen (IB330) aus dem Bachelor Informatik entsprechen.
Leistungspunkte:	5
Arbeitsaufwand:	45 Stunden Präsenzzeit im Unterricht 15 Stunden Präsenzzeit im Praktikum 90 Stunden Selbststudium
Lehrformen:	2 SWS seminaristischer Unterricht 2 SWS begleitendes Praktikum
Leistungsnachweise und Prüfung:	Prüfung gemäß der aktuellen Studien- und Prüfungsordnung. Prüfungsform wird hochschulweit im Studien- und Prüfungsplan bekanntgegeben.

Qualifikationsziele und Inhalte:

Qualifikationsziele:

- Die Studierenden erlangen das Verständnis grundlegender Konzepte und Probleme der algorithmischen Geometrie.
- Sie beherrschen Lösungstechniken zur Behandlung einer Vielzahl von Problemen geometrischer Natur und können die Komplexität der Verfahren abschätzen.
- Die Studierenden erhalten zudem Einblick in die Anwendbarkeit der behandelten Methoden auf Problemstellung aus der industriellen Praxis.

Lehrinhalte:

- Geometrische Probleme und Datenstrukturen
- Konvexe Hüllen
- Triangulierungen (Polygon, Delaunay)
- Punktlokalisierung
- Methoden: plane sweep, divide and conquer, (randomized) incremental algorithms

Literatur:

Mark de Berg, Marc van Kreveld, Mark Overmars and Otfried Schwarzkopf: Computational Geometry: Algorithms and Applications, Springer-Verlag, 2000.
Rolf Klein: Algorithmische Geometrie: Grundlagen, Methoden, Anwendungen, Springer Verlag, 2005.

Kryptographie IM730

Modulverantwortlicher:	**Prof. Dr. Konstantin Ziegler**
Dozent:	Prof. Ludwig Griebl, Prof. Dr. Konstantin Ziegler
Studiengang:	Master
Modultyp:	Wahlpflichtmodul
Sprache:	Deutsch
Angebot:	Sommersemester
Dauer:	Ein Semester
Vorkenntnisse:	
Voraussetzungen:	
Leistungspunkte:	5
Arbeitsaufwand:	30 Stunden Präsenzzeit im Unterricht 30 Stunden Präsenzzeit im Praktikum 90 Stunden Selbststudium
Lehrformen:	2 SWS seminaristischer Unterricht 2 SWS begleitendes Praktikum
Leistungsnachweise und Prüfung:	Prüfung gemäß der aktuellen Studien- und Prüfungsordnung. Prüfungsform wird hochschulweit im Studien- und Prüfungsplan bekanntgegeben.

Qualifikationsziele und Inhalte:

Qualifikationsziele:

- Verständnis für Sicherheitsanforderungen und -maßnahmen sowie für das Zusammenspiel von Rechenleistung und Sicherheitsanforderungen.
- Beherrschung der grundlegenden Techniken für Kryptosysteme und Kryptoanalyse.

Lehrinhalte:

- Private-key und public-key Kryptosysteme: AES, RSA, gruppen-basiert
- Sicherheits-Reduktionen, Schlüsseltausch, kryptographische Hashfunktionen, Signaturen
- Faktorisieren und diskrete Logarithmen, untere Schranken

Literatur:

Joachim von zur Gathen (2015), CryptoSchool, Springer-Verlag Berlin Heidelberg.
Jonathan Katz and Yehuda Lindell (2020), Introduction to Modern Cryptography, 3rd Edition, CRC Press.
Douglas R. Stinson (2005), Cryptography: Theory and Practice, 3rd edition, Chapmann and Hall/CRC.

Inverse Problems IM740

Responsible for the course:	**Prof. Dr. Konstantin Ziegler**
Lecturer:	Prof. Dr. Hannah Jörg, Prof. Dr. Konstantin Ziegler
Study program:	Master
Course type:	Required-elective module in the Scientific Computing major
Language:	English
Term:	Summer Semester
Duration:	One Semester
Prior knowledge:	
Requirements:	
Credit points:	5
Workload:	30 hours of attendance time in class 30 hours of attendance time in lab sessions 90 hours self-study
Teaching methods:	2 semester hours per week seminar-based instruction 2 semester hours per week lab sessions
Performance records and examinations:	Examination according to the current study and examination regulations.

Qualification goals and content:

Qualification goals:

At the end of the module students know important theoretical and analytical properties of inverse problems. They know how to use analytical tools to describe the degree of ill-posedness of an inverse problem. They can estimate which accuracy can be obtained by the numerical reconstruction in the optimum case and how to utilize special properties of the problem under consideration to come as close as possible to that limit by numerical methods. They have a detailed knowledge of important types of numerical algorithms for the solution of (mainly linear) inverse problems: Direct approaches e.g. based on the truncated SVD or Tikhonov regularization as well as iterative numerical techniques (e.g. Landweber or conjugate gradient methods). An important learning outcome is the ability to select the proper strategy for the choice of parameters in regularization schemes and to implement stopping criteria for the algorithms. The students are able to select and apply the numerical algorithms for the treatment of typical application problems e.g. in medical image analysis.

Content:

Based on the theory of inverse problems different algorithms for the numerical solution of mainly linear inverse problems are analyzed.

- Applications of relevance for inverse problems.
- Analytical Tools: Ill-posedness and regularization, filters, generalized inverse, SVD, generalized Tikhonov regularization.
- Numerical Tools: Iterative regularization techniques: Landweber, steepest descent, conjugate gradient;projection methods, mollifier, inexact Newton methods; stability, convergence and stopping criteria.

Literature:

C.R. Vogel (2002), Computational Methods for Inverse Problems, SIAM.
M. Hanke (2017), A taste of inverse problems, SIAM.

Decision Making and Optimal Control IM750

Responsible for the course:	**Prof. Dr. Konstantin Ziegler**
Lecturer:	Prof. Dr. Hannah Jörg, Prof. Dr. Konstantin Ziegler
Study program:	Master
Course type:	Required-elective module in the Scientific Computing major
Language:	English
Term:	Winter Semester
Duration:	One Semester
Prior knowledge:	
Requirements:	
Credit points:	5
Workload:	30 hours of attendance time in class 30 hours of attendance time in lab sessions 90 hours self-study
Teaching methods:	2 semester hours per week seminar-based instruction 2 semester hours per week lab sessions
Performance records and examinations:	Examination according to the current study and examination regulations.

Qualification goals and content:

Qualification goals:

After successfully completing the module, students will be able to analyse an optimal control problem and, depending on the requirements, formalize it as a static optimization problem in the framework of model predictive control, dynamic programming or LQ control, using a learning procedure if necessary. Students are able to select, apply and further develop a numerical method for solving the static optimization problem.

Content:

- Model Predictive Control (MPC): Principle of MPC; Stability of MPC; Linear Model Predictive Control (DMC, GPC); Robust and Stochastic Model Predictive Control; Learning-Based MPC;
- Dynamic Programming (DP): Bellman Recursion; Reinforcement Learning; Policy Iteration; Value Iteration; Control of Markov Decision Processes;
- QL control: QL controllers; Stability of QL control; Iterative QL control;
- Numerics of optimal control (non-linear optimization, KKT conditions, QP methods, SQP methods, IP methods);
- Applications in robotics and power engineering.

Literature:

Camacho, Bordons (2007), Model Predictive Control, Springer.
Grüne, Pannek (2017), Nonlinear Model Predictive Control, Springer.

Wissenschaftliches Seminar IM820

Modulverantwortlicher:	**Prof. Andreas Siebert, Ph.D.**
Dozent:	Prof. Andreas Siebert, Ph.D
Studiengang:	Master
Modultyp:	Wahlpflichtmodul
Sprache:	Deutsch oder Englisch
Angebot:	Wintersemester
Dauer:	Ein Semester
Vorkenntnisse:	
Voraussetzungen:	
Leistungspunkte:	5
Arbeitsaufwand:	60 Stunden Präsenzzeit 90 Stunden Selbststudium
Lehrformen:	Zwei Präsentationen der Studierenden, mit Diskussion; Moderation durch den Dozenten
Leistungsnachweise und Prüfung:	Die Studierenden halten zwei Präsentationen, es besteht Präsenzpflicht. In der ersten, benoteten Präsentation von 60 Min. Länge stellt der Student / die Studentin den Inhalt eines aktuellen wissenschaftlichen Artikels vor. In der zweiten Präsentation von 30 Min. Länge führt der Student / die Studentin in ein Thema aus der Forschungswelt ein. Diese Präsentation wird nur mit bestanden/nicht bestanden bewertet. Das Modul ist nur dann bestanden, wenn beide Präsentationen erfolgreich gehalten wurden. Die Gesamtnote ergibt sich aus der Note der ersten Präsentation.

Qualifikationsziele und Inhalte:

Qualifikationsziele:

Die Studierenden können sich ein komplexes technisches oder wissenschaftliches Thema aus der forschungsnahen, insbesondere auch aus der englischsprachigen Literatur selbstständig erarbeiten. Sie können das Thema in einem fachlichen Vortrag unter Einbezug moderner Medien präsentieren und mit einem technisch versierten Publikum eine Diskussion über die Präsentationsinhalte führen.
Den Studierenden gelingt die Abgrenzung von Wissenschaft und Nicht-Wissenschaft. Sie verstehen die wichtigsten Abläufe in Forschung und Forschungsbetrieben. Sie kennen die aktuellen Trends in der Veränderung der Forschungswelt.

Lehrinhalte:
Aktuelle Themen der Informatik.

Literatur:
Abhängig von den behandelten Themen.

Masterarbeit IM830

Modulverantwortlicher:	**Prof. Andreas Siebert, Ph.D.**
Dozent:	Dozenten der Informatikstudiengänge. Mindestens einer der Prüfer ist hauptamtlicher Professor der Fakultät Informatik.
Studiengang:	Master
Modultyp:	Pflichtmodul
Sprache:	Deutsch oder Englisch
Angebot:	Im Wintersemester oder Sommersemester. Die Bearbeitungsdauer der Masterarbeit beträgt sechs Monate.
Dauer:	Sechs Monate
Vorkenntnisse:	
Voraussetzungen:	Voraussetzung zur Ausgabe des Themas ist, dass der/die Studierende mindestens 30 ECTS-Punkte erworben hat.
Leistungspunkte:	30
Arbeitsaufwand:	900 Stunden selbstständige Arbeit
Lehrformen:	Selbstständiges Arbeiten
Leistungsnachweise und Prüfung:	Schriftliche Masterarbeit. Die Masterarbeit schließt mit einem Kolloquium ab, in dem die Eigenständigkeit der Leistung der/des Studierenden überprüft wird.

Qualifikationsziele und Inhalte:

Qualifikationsziele:

Die Studierenden haben die Fähigkeit ein komplexes praxisbezogenes Informatik-Thema selbstständig und auf wissenschaftlicher Grundlage methodisch zu bearbeiten und schriftlich die Problemstellung und deren Lösung darzustellen.

Lehrinhalte:
Abhängig vom Thema der Masterarbeit.

Literatur:
Abhängig vom Thema der Masterarbeit.

Digital Entrepreneurship BW210

Modulverantwortlicher:	**Prof. Dagmar Schuller**
Dozent:	Prof. Dagmar Schuller
Studiengang:	Master
Modultyp:	Wahlpflichtfach
Sprache:	Deutsch/Englisch
Angebot:	im Sommersemester
Dauer:	ein Semester
Vorkenntnisse:	-
Voraussetzungen:	-
Leistungspunkte:	5
Arbeitsaufwand:	30 Stunden Präsenzzeit im Unterricht 30 Stunden praktische Übungszeit 90 Stunden Selbststudium
Lehrformen:	2 SWS Unterricht 2 SWS praktische Übungszeit
Leistungsnachweise und Prüfung:	Portfolioprüfung (Pitch Days und schriftliche Ausarbeitung)

Qualifikationsziele und Inhalte:

Qualifikationsziele:

Den Studierenden werden die Grundlagen des Digital Entrepreneurship vermittelt. Dies beinhaltet den Fokus auf den Einfluss digitaler Technologien auf unternehmerische Prozesse (bspw. Prototyping, Skalierbarkeit, etc.), auf neue Produkte und Services (bspw. neue Geschäftsmodelle) und Plattformen. Die Studierenden entwickeln die Fähigkeit, anhand von realen Problemstellungen Methoden innerhalb des Digital Entrepreneurship wie Rapid Prototyping, Design Thinking und Business Case Development zu verstehen und anzuwenden. Die Studierenden lernen, digitale Ansätze mit wirtschaftlichen und gesellschaftlichen Problemstellungen zu verknüpfen und in Geschäftsideen auszuformulieren. Weiter erarbeiten sie die Fähigkeit, konkrete Geschäftspläne zu erstellen und zu professionell zu präsentieren.

Lehrinhalte:

- Grundlage der Unternehmensgründung mit Fokus auf digitalen Technologien
- Einfluss digitaler Technologien auf Unternehmensprozesse, Geschäftsmodelle, Ökosysteme
- Methoden digital Entrepreneurship: Design Thinking, Rapid Prototyping, Geschäftsplanung
- Erstellen von Geschäftsplänen, Pitch Decks, grober Finanzplan
- Präsentationstechniken für Geschäftspläne, Elevator Pitch

Literatur:

- Tobias Kollmann, Digital Entrepreneurship, Springer Gabler 2022
- Alex Genadinik: How to Write A Business Plan, CreateSpeach, 2015
- Jim Horan: The One Page Business Plan, Independently Published, 2020
- Alberto Felicetti, Vincenzo Corvello, Salvatore Ammirato: Digital Innovation in Entrepreneurial Firms: A Systematic Literature review, Review of Managerial Science, 2023
- Frederik Briel et al.: Researching Digital Entrepreneurship, Communications of the Association for Information Systems, 2021

Prozess-Simulation WM140

Process simulation

Modulverantwortlicher:	**Prof. Dr. Jürgen Wunderlich**
Dozent:	Prof. Dr. Jürgen Wunderlich
Studiengang:	Master
Modultyp:	Wahlpflichtfach
Sprache:	Deutsch
Angebot:	Im zweiten Semester
Dauer:	Ein Semester
Vorkenntnisse:	-
Voraussetzungen:	-
Leistungspunkte:	5
Arbeitsaufwand:	30 Stunden Präsenzzeit im seminaristischen Unterricht 30 Stunden Präsenzzeit im Praktikum 90 Stunden Selbststudium
Lehrformen:	2 SWS seminaristischer Unterricht mit Teamübungen 2 SWS begleitendes Praktikum am Rechner
Leistungsnachweise und Prüfung:	Benoteter Leistungsnachweis

Qualifikationsziele und Inhalte:

Qualifikationsziele:

Die Studierenden haben Kenntnisse zu den Einsatzfeldern, Vorgehensweisen und Nutzeffekten der Prozesssimulation erworben. Damit sind sie in der Lage, die Simulationstechnik auf Prozesse aus der industriellen Praxis zielgerichtet anzuwenden und auch auf andere Einsatzfelder wie die Simulation von Geschäftsprozessen, Verkehrsströmen oder Kommunikationsnetzen zu übertragen. Weiterhin können sie ein gängiges Simulationswerkzeug sicher bedienen sowie Abläufe aus Produktion, Logistik und Administration selbstständig aufnehmen, modellieren, bewerten und optimieren.

Lehrinhalte:

- Grundbegriffe und Einführung in die Prozesssimulation
- Aufbau und Funktionsweise eines Simulationswerkzeugs
- Vorgehensweise und Phasen bei der Simulation eines Systems
- Erhebung und Analyse der simulationsrelevanten Daten
- Aufbau und Erstellung von experimentierfähigen Simulationsmodellen
- Planung, Durchführung und Auswertung von Simulationsexperimenten
- Vorstellung von Anwendungsbeispielen aus unterschiedlichen Branchen
- Praktische Übungen mit dem Simulationswerkzeug Plant Simulation

Im Rahmen des studienbegleitenden Leistungsnachweises ist ein vorgegebener Prozess zu modellieren und zu simulieren. Die Simulationsergebnisse sind geeignet aufzubereiten, zu analysieren und zu interpretieren.

Literatur:

S. Bangsow, "Tecnomatix Plant Simulation", Springer, Berlin, 2020
M. Elay, "Simulation in der Logistik", Springer, Berlin, 2012
A. Law, W. Kelton, "Simulation Modeling and Analysis", McGraw-Hill, 1991
J. Wunderlich, Studienheft „Simulation", Klett, Stuttgart, 2020
VDI Richtlinie 3633 Blatt 1, „Simulation von Logistik-, Materialfluss- und Produktionssystemen" (Gründruck), Beuth-Verlag, Berlin, 2017

Advanced Process Mining WM230

Modulverantwortlicher:	**Prof. Dr. Markus Böhm**
Dozent:	Prof. Dr. Markus Böhm
Studiengang:	Master
Modultyp:	Wahlpflichtfach
Sprache:	Deutsch
Angebot:	Sommersemester
Dauer:	ein Semester
Vorkenntnisse:	gutes Textverständnis beim Lesen englischsprachiger Texte
Voraussetzungen:	• Geschäftsprozessmodellierung mit BPMN • Grundlagen des Process Mining
Leistungspunkte:	5
Arbeitsaufwand:	30 Stunden Präsenzzeit im Unterricht 30 Stunden Präsenzzeit im Praktikum 90 Stunden Selbststudium
Lehrformen:	2 SWS seminaristischer Unterricht 2 SWS Praktikum in kleinen Gruppen (14-tägig 4 Stunden)
Leistungsnachweise und Prüfung:	Benoteter Leistungsnachweis

Qualifikationsziele und Inhalte:

Qualifikationsziele:

Process Mining bietet Verfahren um automatisiert Geschäftsprozesse aus Ereignisdaten (i.e. Event-Logs aus Transaktionsystemen wie bspw. dem SAP ERP System) zu rekonstruieren. Dies schafft die Grundlage für eine fortschrittliche Prozessanalyse und Simulation sowie eine kontinuierliche evidenzbasierten Prozessoptimierung. Eine reibungslose Einführung von Process Mining erhebt jedoch einen hohen Anspruch an die Datenqualität, welche in der Praxis nur bedingt gegeben ist. Ziel dieses Moduls ist es, Lösungen zu erarbeiten wie mit unvollständigen, heterogenen und verteilten Daten umgegangen werden kann.
Nach Abschluss des Moduls kennen die Teilnehmer Methoden, Verfahren und Konzepte um Process Mining in Unternehmen einführen zu können. Sie können diese, unterstützt durch Softwarewerkzeuge, auf praktische Aufgabenstellungen anwenden und verschiedene Datenquellen einbinden. Zudem lernen die Studierenden Ansätze kennen, wie bislang nicht über EventLogs verfügbare Informationen eingebunden und somit für das Process Mining nutzbar gemacht werden können.
Darüber hinaus können die Studierenden Anwendungsfälle für unternehmensübergreifendes Prozess Mining identifizieren und deren besonderen Herausforderungen einschätzen.
Im Praktikum führen die Studierenden in Teams jeweils eine Fallstudie mit einem Unternehmen durch, um wertschaffende Anwendungsfälle zu identifizieren und diese prototypisch zu implementieren. Die Ergebnisse des Praktikums fassen sie in einer Machbarkeitsstudie zusammen.

Lehrinhalte:

1) Anwendungsfälle für das Process Mining

- Traditionelle Anwendungsfälle
- Innovative Anwendungsfälle
- Bewertung von Anwendungsfällen

(2) Datenquellen und Datenanbindung

- Identifizierung relevanter Datenquellen
- Aufbereitung von Daten für das Process Mining
- Integration verschiedener Datenquellen
- Umgang mit unvollständigen Daten

(3) Unternehmensübergreifendes Process Mining

- Anwendungsfälle für unternehmensübergreifendes Process Mining
- Herausforderungen im unternehmensübergreifenden Process Mining

Literatur:

van der Aalst, W. (2016). Process Mining: Data Science in Action (2. Auflage). Springer.
Reinkemeyer, L. (Hrsg.) (2020). Process Mining in Action: Principles, Use Cases and Outlook. Springer.
Aktuelle wissenschaftliche Aufsätze.