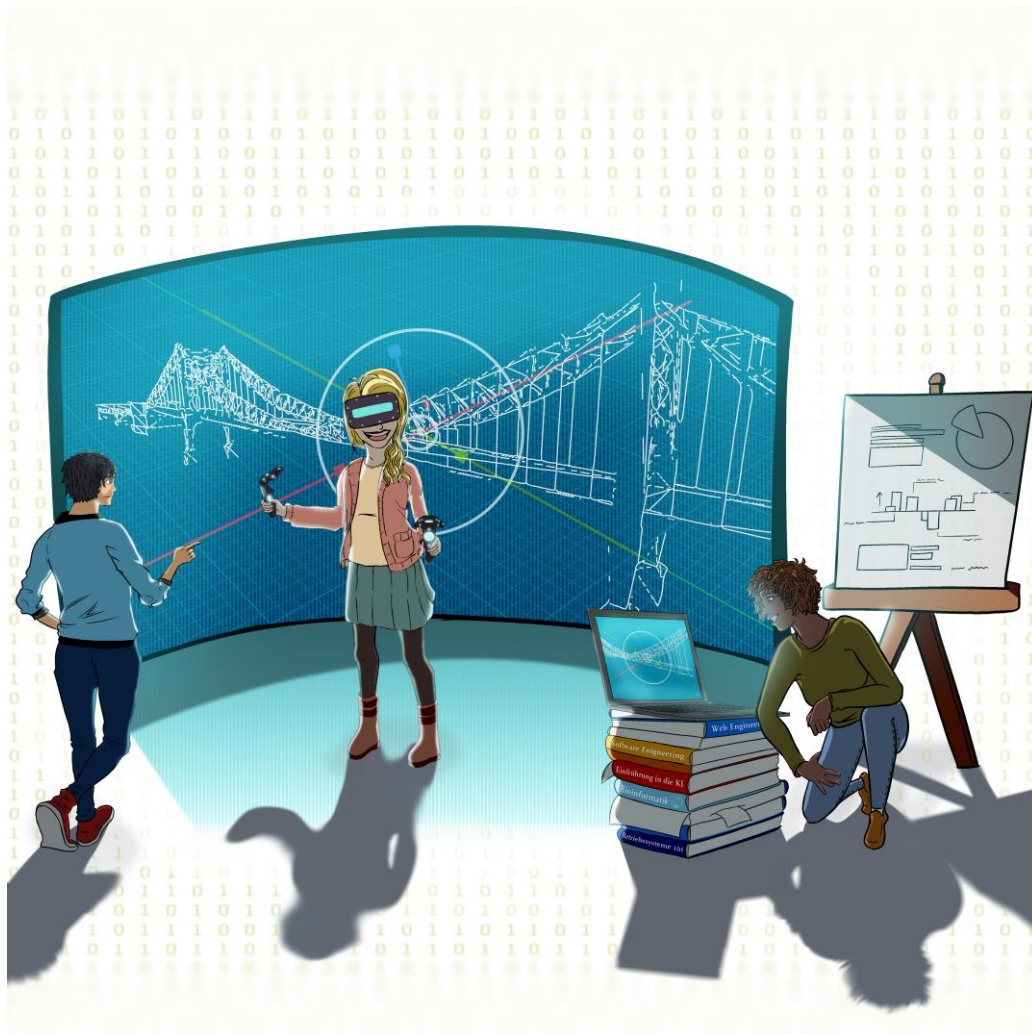


Modulhandbuch Master of Science (M.Sc.)

Angewandte Informatik [P020]

Stand: Wintersemester 2023/24

<https://informatik.rub.de/studium/studiengaenge/ai/msc/>



Studienplan Master Angewandte Informatik PO 20

Nr	Modul	Lehrveranstaltung	Umfang bzw. Mind. Umfang (CP)	Empfohlenes Semester	Bewertung
Wahlpflichtbereich					
1	Wahlpflichtmodule*	Wahlpflichtmodule **	20*	1-3	benotet
Anwendungsbereich					
2	Anwendungsmodule	Anwendungsmodule**	35*	1-3	benotet
3	Fachwissenschaftliche Vertiefung	Vertiefungsseminare**	6	1-3	benotet
Freier Wahlbereich					
4	Freie Wahlmodule	Freie Wahlmodule***	15	1-3	unbenotet
Wahlpflichtbereich					
5	Master-Studienprojekt	Master-Studienprojekt	10	3	benotet
Masterarbeit					
6	Masterarbeit und Kolloquium	Masterarbeit und Kolloquium	30	4	benotet
Summe:			120		

#

* Der Gesamtumfang der Wahlpflichtmodule und Anwendungsmodule darf 59 CP nicht unterschreiten.

** Informationen zu den wählbaren Wahlpflicht- und Anwendungsmodulen befinden sich im jeweils aktuellen Modulhandbuch. Informationen zu den angebotenen Seminaren finden Sie im Vorlesungsverzeichnis der RUB.

*** Hier können (nahezu) alle Veranstaltungen des Vorlesungsverzeichnisses der RUB, sowie Veranstaltungen im Rahmen der Universitätsallianz Ruhr gewählt werden.

#

#

Lehrveranstaltung	Einheit	Umfang Modul (LP)	Semester	Bewertung
Wahlpflichtmodule				
Advanced Algorithms	Informatik	9	WS	benotet
Effiziente Algorithmen	Mathe	9	SS	benotet
Geometrische Algorithmen (ab SS 24 Vertiefungsmodul)	Informatik	5	SS	benotet
Fundamentals of GPU Programming	ETIT	5	WS	benotet
Parallel Computing	Bauing	6	Letztmalig SS 21	benotet
Machine Learning: Evolutionary Algorithms	Informatik	6	WS	benotet
Machine Learning: Supervised Methods	Informatik	6	SS	benotet
Machine Learning: Unsupervised Methods	Informatik	9	WS (kein Angebot im WS 23/24)	benotet
Kryptographie	Informatik	8	WS	benotet
Komplexitätstheorie	Informatik	9	Letztmalig SS 22	benotet
Deterministic Network Calculus	Informatik	5	SS	benotet
Theorie des maschinellen Lernens	Informatik	9	SS	benotet
Information Theory (ab SS 24 Wahlpflichtmodul)	Informatik	5	SS	benotet
Anwendungsmodule				
Ingenieurinformatik				
Design Optimization	Bauing	6	WS (kein Angebot im WS 23/24)	benotet
Design soziotechnischer Informationssysteme	AW	5	Letztmalig WS 21/22	benotet
IT im Engineering	MB	6	Letztmalig WS 21/22	benotet
Verkehrstechnik	Bauing	6	SS	benotet
Grundlagen der automatischen Spracherkennung	ETIT	6	Letztmalig SS 21	benotet
Bildverarbeitung in der Medizin	ETIT	5	SS	benotet
Advanced System Engineering	MB	5	WS	benotet
Energy-Aware Computing Systems	Informatik	6	WS	benotet
Grundlagen der FEM	Bauing	5	WS	benotet
Performance Engineering Praktikum	Informatik	5	WS	benotet
Programmier- und Simulationstechnik				
Geometrische Algorithmen (ab SS 24 Anwendungsmodul)	Informatik	5	SS	benotet
Simulationstechnik	Bauing	5	WS	benotet
3D-Simulation in der Automatisierungstechnik	MB	5	WS	benotet
Complexity Economics and Agent-based Modeling	WiWi	10	Unregelmäßig (findet im WS 23/24 statt)	benotet
High-Performance Computing on Clusters	Bauing	6	WS	benotet
High-Performance Computing on Multi- and Manycore Processors	Bauing	6	SS	benotet
Knowledge Graphs	Informatik	5	Letztmalig SS 23	benotet
Inklusives IT-Design	AW	6	SS	benotet
Neuroinformatik				
Autonomous Robotics: Action, Perception and Cognition	Informatik	6	SS	benotet
Computational Neuroscience: Neural Dynamics	Informatik	6	WS	benotet
Computational Neuroscience: Vision and Memory	Informatik	5	SS	benotet
Computational Neuroscience: Single Neuron Models	Informatik	5	SS	benotet
Master-Praktikum: Autonomous Robotics	Informatik	3	WS/SS	benotet
Deep Learning	Informatik	5	WS	benotet
Autonomous Vehicles and Artificial Intelligence	Informatik	5	SS	benotet
Autonomous Vehicles and Artificial Intelligence lab	Informatik	5	WS	benotet
Master-Praktikum: Deep Learning and Natural Language Processing	Informatik	5	Unregelmäßig (nicht im WS 23/24)	benotet
Kryptologie und theoretische Informatik				
Kryptanalyse 1 (Einführung in die asymmetrische Kryptanalyse)	Informatik	5	Letztmalig SS 20	benotet

Symmetrische Kryptanalyse	Informatik	5	Letztmalig WS 22/23	benotet
Kryptographische Protokolle	Informatik	5	Letztmalig SS 23	benotet
Quantenalgorithmen	Informatik	5	Letztmalig SS 20 (letztmalig im WS 21/22)	benotet
Systemsicherheit	Informatik	5	SS	benotet
Model Checking	Informatik	5	SS	benotet
Quantum Information and Computation	Informatik	5	SS	benotet
Netzsicherheit 1	Informatik	5	WS	benotet
Netzsicherheit 2	Informatik	5	SS	benotet
Software Security	Informatik	9	SS	benotet
Information Theory (ab SS 24 Wahlpflichtmodul)	Informatik	5	SS	benotet
IT im Management				
Groupware und Wissensmanagement	AW	5	Letztmalig SS 21	benotet
Industrielles Kundenmanagement	MB	5	Letztmalig SS 21	benotet
Product Lifecycle Management	MB	6	SS	benotet
Managing Digital Platform Ecosystems	TUDo	7	Letztmalig SS 21	benotet
KIbox: Lösung von Nachhaltigkeits-Challenges durch den Einsatz künstlicher Intelligenz	AW + MB	5	SS/WS (nicht im SS 23)	benotet
Computerlinguistik				
Anwendungen der Computerlinguistik (Im WS 23/24: 050042 Semantischer Wandel oder 050043 NLP für Bildungsanwendungen)	Philologie	10 bzw. 12	SS/WS	benotet
Introduction to CL in Python	Philologie	5	WS	benotet
Introduction to Linguistic Data Science with R	Philologie	5	WS	benotet
Computational Linguistics and AI	Philologie	5	SS	benotet
Linguistic Data Science	Philologie	5	SS	benotet
Schwerpunktseminar Computational Linguistics	Philologie	5	WS/SS (ab SS 23)	benotet
Schwerpunktseminar Linguistic Data Science	Philologie	5	WS/SS (ab SS 23)	benotet
Bioinformatik				
Bioinformatik in der Proteomik I	MPC	5	Letztmalig WS 22/23	benotet
Bioinformatik in der Proteomik II	MPC	5	Letztmalig SS 22	benotet
Bioinformatics for Proteomics	MPC	5	WS	benotet
Bioimage Informatics	Biologie	5	WS	benotet
Master-Praktikum: Big Data in der Bioinformatik	Biologie	5	WS/SS	benotet
Master-Praktikum: Computational Proteomics	MPC	5	SS	benotet

Angebotene Vertiefungsseminare im Modul „Fachwissenschaftliche Vertiefung“

Lehrveranstaltung	Einheit	Umfang Modul (LP)	Semester	Bewertung
Vertiefungsseminare				
Seminar Ingenieurinformatik	Baulng	3	SS und WS	benotet
Seminar Bioinformatik	Biologie / MPC	3	SS und WS	benotet
Security Engineering	Informatik	3	SS und WS	benotet
Seminar Ressourceneffiziente Systemsoftware	Informatik	3	SS und WS	benotet
Algorithms for Decision Making	Informatik	3	WS	benotet
Seminar Geometrische Algorithmen	Informatik	3	WS	benotet
Security and Privacy for Mobile Systems	Informatik	3	WS	benotet
Seminar on Current Topics for Systems Security and Privacy	Informatik	3	WS	benotet
Seminar Distributed and Networked Systems	Informatik	3	WS	benotet
Seminar Quantum Cryptography	Informatik	3	WS	benotet

Seminar Approximationsalgorithmen	Informatik	3	SS	benotet
Seminar Satisfiability	Informatik	3	SS	benotet
Seminar Computational Neuroscience	Informatik	3	SS	benotet
Seminar From Biological to Artificial Neural Networks	Informatik	3	SS	benotet
Seminar Quantum Algorithms	Informatik	3	SS	benotet
Seminar Advanced topics in Deep Learning	Informatik	3	SS	benotet
Seminar Fortgeschrittene Themen des Model Checking	Informatik	3	SS (nicht SS 23)	benotet
Seminar über Grenzen in der theoretischen Informatik	Informatik	3	unregelmäßig	benotet
Seminar Implementation Security	Informatik	3	Letztmalig SS 23	benotet
Seminar on Knowledge Graphs	Informatik	3	Letztmalig WS 22/23	benotet
Seminar Topics in Deep Learning for Sequence Processing	Informatik	3	Letztmalig WS 22/23	benotet
Seminar Symmetrische Kryptanalyse	Informatik	3	Letztmalig WS 22/23	benotet

Abkürzungen:

AW: Institut für Arbeitswissenschaften
 Baulng: Fakultät für Bau- und Umweltingenieurwissenschaften
 ETIT: Fakultät für Elektrotechnik und Informationstechnik
 MB: Fakultät für Maschinenbau
 MPC: Medizinisches Proteom Center
 WIWI: Fakultät für Wirtschaftswissenschaft

SS: Sommersemester
 WS: Wintersemester

CP: Creditpoints

MODULHANDBUCH

Übersicht der Module

Angewandte Informatik - Master (1-Fach, PO 2020)

Wahlpflichtbereich

3D-Simulation in der Automatisierungstechnik
Advanced System Engineering
Anwendungen der Computerlinguistik [Master]
Autonomous Robotics: Action, Perception, Cognition
Autonomous Vehicles and Artificial Intelligence
Autonomous Vehicles and Artificial Intelligence Lab
Bildverarbeitung in der Medizin
Bioimage Informatics
Bioinformatics for Proteomics
Complexity Economics and Agent-based Modeling
Computational Linguistics and AI
Computational Neuroscience: Neural Dynamics
Computational Neuroscience: Single-Neuron Models
Computational Neuroscience: Vision and Memory
Deep Learning
Design Optimization (kein Angebot im WS 23/24)
Energy-Aware Computing Systems
Geometrische Algorithmen
Grundlagen der FEM
High-Performance Computing on Clusters
High-Performance Computing on Multi- and Manycore Processors
Inklusives IT-Design
Introduction to Computational Linguistics in Python
Introduction to Linguistic Data Science with R
Klbox: Lösung von Nachhaltigkeits-Challenges durch den Einsatz künstlicher Intelligenz
Linguistic Data Science
Master-Praktikum Deep Learning and Natural Language Processing
Master-Praktikum: Autonomous Robotics
Master-Praktikum: Big Data in der Bioinformatik
Master-Praktikum: Computational Proteomics
Model Checking
Netzsicherheit 1
Netzsicherheit 2
Performance Engineering Praktikum

Product Lifecycle Management
Quantum Information and Computation
Schwerpunktseminar Computational Linguistics
Schwerpunktseminar Linguistic Data Science
Simulationstechnik
Software Security
Systemsicherheit
Verkehrstechnik
Advanced Algorithms
Deterministic Network Calculus
Effiziente Algorithmen
Fundamentals of GPU Programming
Information Theory
Kryptographie
Machine Learning: Evolutionary Algorithms
Machine Learning: Supervised Methods
Machine Learning: Unsupervised Methods (kein Angebot im WS 23/24)
Parallel Computing
Theorie des maschinellen Lernens
Fachwissenschaftliche Vertiefung (Angewandte Informatik)
Studienprojekt (Master)

Wahlbereich

Freie Wahlmodule

Masterarbeit

Masterarbeit und Kolloquium (AI)

Titel des Moduls: 3D-Simulation in der Automatisierungstechnik

Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester siehe Prüfungsordnung	Turnus Wintersemester	Dauer 1 Semester
Lehrveranstaltungen 3D-Simulation in der Automatisierungstechnik (139050)			Kontaktzeit 60 h	Selbststudium 90 h	Gruppengröße Studierende
Unterrichtssprache Deutsch			Teilnahmevoraussetzungen		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr.-Ing. Bernd Kuhlenkötter Lehrende: Dr.-Ing. Alfred Hypki					
Verwendung des Moduls M.Sc. Angewandte Informatik M.Sc. Maschinenbau M.Sc. Sales Engineering and Product Management					
Vorkenntnisse <p>Empfohlen: Besuch der Vorlesung "Grundlagen der Automatisierungstechnik"</p>					
Lernziele (learning outcomes) Nach einer allgemeinen Einführung in die Anforderungen und Möglichkeiten der 3D Simulation in der Automatisierungstechnik erlernen die Studierenden die verschiedenen Schritte, die sich von der Idee über die Modellierung der Arbeitszelle und Programmierung der Automatisierungskomponenten bis zur Virtuellen Inbetriebnahme erstrecken. Ein besonderes Augenmerk wird in dieser Vorlesung auf die Industrielle Robotik gelegt, die in zahlreichen Beispielen und Anwendungen thematisiert wird. Nach dem erfolgreichen Abschluss des Moduls <ul style="list-style-type: none">• beherrschen die Studierenden die Grundlagen der 3D-Simulationstechnik.• haben die Studierenden Fähigkeit erworben, automatisierungstechnische Aufgabenstellungen vorab über eine 3D-Simulation abzubilden und abzusichern.• kennen die Studierenden die wichtigsten Methoden und Softwaresysteme zur Lösung simulationstechnischer Probleme.					
Inhalt Die Vorlesung deckt die folgenden Themenbereiche ab <ul style="list-style-type: none">• Simulation in der Automatisierungstechnik: Anforderungen und Möglichkeiten• Grafische 3D-Simulation• CAD-basierte Arbeitszellenmodellierung und 3D-Datenaustausch• Roboterprogrammierung• Offline-Programmierung und Virtuelle Inbetriebnahme• Grundlagen und Leistungsmerkmale von grafischen 3D-Simulationssystemen im industriellen Einsatz Im Bereich der Kompetenzen im Kontext Digitalisierung ist besonders die Übung zu nennen, diese besteht aus der praktischen Umsetzung der von Modellierungs-, Programmierungs- und Simulationsaufgaben mit einem kommerziell verfügbaren und industriell eingesetzten 3D- Robotersimulations- und Offline-Programmiersystem.					
Lehrformen Gemeinsame Vorlesung und Übung in kleineren Gruppen					

Prüfungsformen

Klausur '3D-Simulation in der Automatisierungstechnik' (90 Min., Anteil der Modulnote 100 %)
Studienbegleitende Aufgaben: Umsetzung einer Simulationsaufgabe (Sofern die Aufgabe vor der Modulabschlussprüfung absolviert wird, sind optional Bonuspunkte für die Klausur möglich) (Umfang und Abgabefristen wird am Anfang des Semesters bekanntgegeben)

Voraussetzungen für die Vergabe von Credits

Bestandene Modulabschlussprüfung: Klausur

Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)

5/105: M.Sc. Angewandte Informatik

Titel des Moduls: Advanced Algorithms Advanced Algorithms					
Modul-Nr./Code	Credits 9 CP	Workload 270 h	Semester siehe Prüfungsordnung / see Examination regulations	Turnus Wintersemester	Dauer 1 Semester
Lehrveranstaltungen Advanced Algorithms (212029)			Kontaktzeit 90 h	Selbststudium 180 h	Gruppengröße 40 Studierende
Unterrichtssprache Englisch			Teilnahmevoraussetzungen keine		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Maike Buchin Lehrende: Prof. Maike Buchin					
Verwendung des Moduls M.Sc. Computer Science M.Sc. Angewandte Informatik M.Sc. Mathematik M.Sc. IT-Sicherheit / Informationstechnik					
Vorkenntnisse Empfohlen: Erwartet werden grundlegende Kenntnisse zu Algorithmenentwurf und -analyse wie sie aus dem Bachelorstudium bekannt sind.					
Lernziele (learning outcomes) <ul style="list-style-type: none"> • Fortgeschrittene Entwurfsmethoden für Algorithmen • Fortgeschritten Analysemethoden für Algorithmen • Kenntnis weiterer Datenstrukturen und Methoden zum Entwurf von Datenstrukturen • Anwendung der gelernten Methoden auf neue Probleme 					
Inhalt In der Vorlesung betrachten wir fortgeschrittene Themen der Algorithmik. Nach einer kurzen Wiederholung bekannter Inhalte betrachten wir vor allem Graphalgorithmen, Approximationsalgorithmen und FPT-Algorithmen sowie exakte Algorithmen für NP-schwere Probleme. Ebenfalls betrachten wir einige neue und bekannte Datenstrukturen und deren Analyse. Die betrachteten Probleme dabei sind sowohl kombinatorisch, graphentheoretisch also auch geometrisch.					
Lehrformen Vorlesung (als Folien- und Tafelvortrag) und Übungen, in denen die vorgestellten Inhalte vertieft werden					
Prüfungsformen Mündliche (20-30 Minuten) oder schriftliche Modulabschlussprüfung (120 Minuten) (wird zu Semesterbeginn bekannt gegeben)					
Voraussetzungen für die Vergabe von Credits Bestandene Modulabschlussprüfung und erfolgreiche Teilnahme an Übungen					
Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS) 9/97: M.Sc. Computer Science 9/105: M.Sc. Angewandte Informatik 9/91: M.Sc. IT-Sicherheit / Informationstechnik [PO 22]					

Titel des Moduls: Advanced System Engineering
Advanced System Engineering

Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester siehe Prüfungsordnung	Turnus Wintersemester	Dauer 1 Semester
Lehrveranstaltungen Advanced System Engineering			Kontaktzeit 60 h	Selbststudium 90 h	Gruppengröße Studierende

Unterrichtssprache Deutsch	Teilnahmevoraussetzungen
--------------------------------------	---------------------------------

Modulbeauftragte/r und hauptamtlich Lehrende
 Modulbeauftragte/r: Prof. Dr.-Ing. Detlef Gerhard
 Lehrende: Prof. Dr.-Ing. Detlef Gerhard

Verwendung des Moduls
 M.Sc. Angewandte Informatik

 M.Sc. Sales Engineering and Product Management
 M.Sc. Maschinenbau

Vorkenntnisse

Lernziele (learning outcomes)

Nach dem erfolgreichen Abschluss des Moduls

- verfügen Studierende über ein breites, integriertes Wissen über die über Methoden und Werkzeuge des Advanced Systems Engineering
- haben Studierende ein umfassendes Verständnis vom Zusammenwirken der relevanten Softwaresysteme und von Integrationskonzepten aus Sicht des Advanced Systems Engineerings erworben und können dieses situativ angepasst anwenden, um z.B. differenziert die CAx und PDM/ PLM Prozessketten zu beurteilen.
- sind die Studierenden in der Lage Problemstellungen im Rahmen der Entwicklung komplexer technischer Systeme eigenständig zu analysieren, zu bewerten und eine Systemarchitektur auf Basis von Anforderungsmanagement zu erarbeiten.
- verfügen Studierende über exemplarische, interdisziplinäre Methodenkompetenz der domänenübergreifenden Modellierung und können komplexe technische Systeme selbstständig in einer objektorientierten Modellierungssprache (z.B. SysML) modellieren, verstehen und die erarbeiteten Modelle bewerten
- können kooperativ Aufgabenstellungen der Modellierung und Simulation im Umfeld von Advanced Systems Engineering in heterogenen Gruppen bearbeiten, Abläufe und Ergebnisse begründen und über Sachverhalte in Verbindung mit dem Aufbau von IoT-Lösungen und Digitalen Zwillingen kommunizieren. Indem sie praktische Fallbeispiele und Aufgaben bearbeiten, können sie die erlernten Fertigkeiten im Umgang mit entsprechenden Softwaresystemen auf konkrete Problemstellungen der des Modellbasierten Systems Engineering übertragen.

Inhalt

Die Digitalisierung eröffnet zahlreiche Nutzenpotenziale für neuartige Marktleistungen der Industrie auf Basis von technischen Systemen, die durch hohe Komplexität und Interdisziplinarität gekennzeichnet sind. Ein Ansatz, die Grenzen klassischer Entwicklungsmethoden zu überwinden, ist das Advanced Systems Engineering, das auf das modellbasierte und integrative Zusammenwirken der Disziplinen und Aspekte der Produktentstehung in einem Entwicklungsansatz fokussiert und somit eine ganzheitliche Produktentstehungsmethodik unter Einbeziehung systemtechnischer Methoden darstellt. Die Vorlesung behandelt aus einer übergeordneten Perspektive des Gesamtprozesses die Methoden und Werkzeuge, die bei der Entwicklung von komplexen technischen Systemen Anwendung finden. Dabei wird insbesondere das Modell Based Systems Engineering (MBSE) Konzept sowie die

Systemmodellierungssprache SysML angewendet und von den Studierenden selbst Projektaufgaben im Umfeld von IoT Anwendungen und dem Aufbau von Digitalen Zwillingen erarbeitet.

Folgende Themen sind Gegenstand des Moduls

- Einführung Grundlagen und Begriffe des Advanced Systems Engineering
 - Model Based Definition und Model Based Systems Engineering
 - Systemdefinition und formale Modellierung, Modellierungssprachen, z.B. Sys-ML
 - Anforderungsmanagement
 - Modellierung und Simulation zusätzlich zur mechanischen Struktur
 - Domänenübergreifende Aspekte von Steuerungstechnik/Elektrotechnik
 - Spezialisierte Modellierungsmethoden für Logik- und Verhalten
 - Lückenlose Werkzeugketten und Durchgängigkeit der Modelle
 - Integration aller Domänen in PLM (Mechanik + E/E + Software/Steuerungstechnik)
 - Simulation, Verifikation und Validierung
-
- Methoden für die Planung, Überwachung und Durchführung von Projekten im Hinblick auf Technologie, Zeit und Kosten

Lehrformen

Übung (2 SWS) / Vorlesung (2 SWS)

Prüfungsformen

Klausur 'Advanced Systems Engineering' (90 Min., Anteil der Modulnote 100 %)

Studienbegleitende Aufgaben: Hausarbeiten (Sofern die Hausarbeiten vor der Modulabschlussprüfung absolviert werden, sind optional Bonuspunkte für die Klausur möglich) (Umfang und Abgabefristen wird am Anfang des Semesters bekanntgegeben)

(Beide Prüfungsteile sind über Flexnow anzumelden)

Voraussetzungen für die Vergabe von Credits

- Bestandene Modulabschlussprüfung: Klausur
- Bestandene studienbegleitende Aufgaben: Hausarbeiten (Umfang und Abgabefristen wird am Anfang des Semesters bekanntgegeben)

Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)

5/150: M.Sc. Angewandte Informatik

Titel des Moduls: Anwendungen der Computerlinguistik [Master]

Modul-Nr./Code	Credits 10 CP	Workload 300 h bzw. 360 h	Semester siehe Prüfungsordnung	Turnus Jedes Semester	Dauer 1 Semester
Lehrveranstaltungen Wechselndes Seminarangebot (siehe unter Veranstaltungen)			Kontaktzeit 30 h	Selbststudium 270 h bzw. 330 h	Gruppengröße Studierende
Unterrichtssprache Deutsch oder Englisch			Teilnahmevoraussetzungen		

Modulbeauftragte/r und hauptamtlich Lehrende

Modulbeauftragte/r: Prof. Dr. Stefanie Dipper
Lehrende: Prof. Dr. Stefanie Dipper,
Dr. Laarmann-Quante,
u.a.

Verwendung des Moduls

M.Sc. Angewandte Informatik

Vorkenntnisse

<p>Inhalte aus den Bachelormodulen "Einführung in die Linguistik" sowie "Methoden der Computerlinguistik".</p>
<p>Alternativ oder ergänzend wird das Durcharbeiten von Standard-Lehrbüchern der Computerlinguistik empfohlen, z. B. von: Jurafsky, D., Martin, J. H. (2008). Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition (2nd ed.). Prentice-Hall.</p>

Lernziele (learning outcomes)

10 CP-Variante

- Nach dem erfolgreichen Abschluss des Moduls verfügen Sie über vertieftes Wissen und Fertigkeiten in weiteren von Ihnen ausgewählten Teilgebieten der Computerlinguistik.
- Sie haben einen Überblick über mehrere Bereiche computerlinguistischer Forschung. Sie kennen jeweils verschiedene Ansätze und Methoden, die in diesen Bereichen Anwendung finden, und können diese miteinander vergleichen und nach ihren Stärken und Schwächen bewerten.
- Sie können die passende Methode für eine Aufgabe auswählen und sie implementieren und evaluieren. Zudem können Sie zu Ihrer Implementation eine entsprechende wissenschaftliche Publikation verfassen.

12 CP-Variante:

- Nach dem erfolgreichen Abschluss des Moduls verfügen Sie über fundiertes Wissen zum aktuellen Forschungsstand in weiteren von Ihnen ausgewählten Teilgebieten der Computerlinguistik.
- Sie haben einen Überblick über mehrere Bereiche computerlinguistischer Forschung und kennen die jeweiligen Ansätze und Methoden. Basierend auf diesen Kenntnissen können Sie einen eigenen Ansatz entwickeln und diesen in einer Implementation umsetzen sowie evaluieren.
- Sie können zu ihrem eigenen Ansatz eine Publikation in Form eines wissenschaftlichen Aufsatzes verfassen.

Inhalt

10 CP-Variante:

In den im Modul angebotenen Seminaren werden beispielhaft Themen aus dem gesamten Spektrum der Computerlinguistik abgedeckt, wie z. B. automatische Textzusammenfassung, Koreferenzauflösung, Dialogmodellierung, Metaphernanalyse. In jedem Themenbereich werden Sie eigenständig mit computerlinguistischer Fachliteratur arbeiten, diese einordnen und bewerten und ausgewählte Ansätze nachimplementieren.

12 CP-Variante:

In den im Modul angebotenen Seminaren werden beispielhaft Themen aus dem gesamten Spektrum der Computerlinguistik abgedeckt, wie z. B. automatische Textzusammenfassung, Koreferenzauflösung, Dialogmodellierung, Metaphernanalyse. In jedem Teilgebiet werden Sie eigenständig mit computerlinguistischer

Fachliteratur arbeiten und eigene Ansätze entwickeln und nachimplementieren.

Lehrformen

Seminaristischer Unterricht mit Gruppenarbeiten, mündlichen Präsentationen von Forschungsarbeiten. Die Betreuung bei Selbststudiumanteilen von 150h+ pro Lehrveranstaltung erfolgt nach individueller Absprache. Im Regelfall erfolgen individuelle Sitzungen (Zoom oder Präsenz) im wöchentlichen oder 14-tägigen Wechsel.

Prüfungsformen

Implementation und Research Thesis (semesterbegleitend)

Voraussetzungen für die Vergabe von Credits

Erfolgreicher Abschluss der Modulprüfung (7,0 bzw. 9 CP) sowie bestandene Studienleistung (3,0 CP) im Seminar.

In den zugehörigen Lehrveranstaltungen steht die Diskurs-, Handlungs- oder Praxisorientierung im Vordergrund. Daher besteht in allen Grundkursen und Seminaren Anwesenheitspflicht und Sie sollten nach Möglichkeit nicht fehlen. Das bedeutet im Einzelnen:

Erlaubte Fehlzeiten in Seminaren: max. 3 Sitzungen

Über diese Sitzungen hinaus dürfen Sie nur mit Entschuldigung (z. B. mit einem ärztlichen Attest) fehlen. In der ersten Sitzung besteht grundsätzlich Anwesenheitspflicht, weil dann die grundlegenden Informationen zum Ablauf des Kurses besprochen werden. Darüber hinaus können die Lehrenden einzelne Sitzungen, die z. B. für gemeinsames Arbeiten wesentlich sind, als verpflichtend erklären.

Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)

10/105: M.Sc. Angewandte Informatik

12/105: M.Sc. Angewandte Informatik

Titel des Moduls: Autonomous Robotics: Action, Perception, Cognition
Autonomous Robotics: Action, Perception, Cognition

Modul-Nr./Code	Credits 6 CP	Workload 180 h	Semester siehe Prüfungsordnung / see examination regulations	Turnus Sommersemester	Dauer 1 Semester
Lehrveranstaltungen Autonomous Robotics: Action, Perception, Cognition (211048)			Kontaktzeit 45 h	Selbststudium 135 h	Gruppengröße 30 Studierende
Unterrichtssprache Englisch			Teilnahmevoraussetzungen		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Gregor Schöner Lehrende: Prof. Dr. Gregor Schöner					
Verwendung des Moduls M.Sc. Angewandte Informatik M.Sc. Computer Science					
Vorkenntnisse Der Schwerpunkt des Kurses liegt auf dem Erlernen von Konzepten und dem Erwerb von interdisziplinärer Wissenschaft, einschließlich Lesen und Schreiben auf wissenschaftlichem und technischem Niveau. Mathematische Konzepte werden durchgehend verwendet, daher ist das Verständnis dieser Konzepte wichtig. Mathematische Kenntnisse sind für die Beherrschung des Stoffes nicht entscheidend, aber hilfreich. Die Mathematik stammt hauptsächlich aus der qualitativen Theorie dynamischer Systeme, Attraktoren und deren Instabilitäten. Zu einigen dieser Konzepte werden kurze Tutorien angeboten.					
Lernziele (learning outcomes) Nach erfolgreichem Abschluss: <ul style="list-style-type: none"> • sind die Studierenden mit den Konzepten der Theorie dynamischer Systeme vertraut und können diese praktisch anwenden, • kennen sie die mathematischen Modelle der Bewegungserzeugung • haben sie Übung im Lesen und Schreiben von wissenschaftlichen Arbeiten. 					
Inhalt Die autonome Robotik ist ein interdisziplinäres Forschungsgebiet, in dem verkörperte Systeme, die mit eigenen Sensoren und Aktoren ausgestattet sind, ein Verhalten erzeugen, das nicht vollständig vorprogrammiert ist. Autonome Robotik umfasst somit Wahrnehmung, Bewegungsgenerierung sowie Kernelemente der Kognition wie Entscheidungsfindung, Planung und Integration mehrerer Randbedingungen. In diesem Kurs werden verschiedene Ansätze für dieses interdisziplinäre Problem behandelt. In der ersten Hälfte des Kurses liegt der Schwerpunkt auf Methoden der dynamischen Systeme zur Erzeugung von Bewegung in Fahrzeugen. Das Hauptaugenmerk des Kurses liegt jedoch auf Lösungen zur autonomen Bewegungserzeugung, die durch Analogien zur Bewegungserzeugung in Nervensystemen inspiriert sind. In der zweiten Hälfte des Kurses werden Kernprobleme der menschlichen Bewegungswissenschaft behandelt, darunter das Freiheitsgradproblem, die Koordination, die motorische Kontrolle und die Reflexsteuerung von Muskeln.					
Lehrformen Vorlesung mit Übung					
Prüfungsformen Mündliche Prüfung (30 Minuten); Übungsaufgaben für Bonuspunkte					
Voraussetzungen für die Vergabe von Credits Bestandene mündliche Modulabschlussprüfung					

Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)

6/105: M.Sc. Angewandte Informatik

6/97: M.Sc. Computer Science

Titel des Moduls: Autonomous Vehicles and Artificial Intelligence
Autonomous Vehicles and Artificial Intelligence

Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester siehe Prüfungsordnung / see examination regulations	Turnus Sommersemester	Dauer 1 Semester
Lehrveranstaltungen Autonomous Vehicles and Artificial Intelligence (211044)			Kontaktzeit 60 h	Selbststudium 90 h	Gruppengröße 25 Studierende
Unterrichtssprache Englisch			Teilnahmevoraussetzungen keine		

Modulbeauftragte/r und hauptamtlich Lehrende
 Modulbeauftragte/r: Prof. Thorsten Berger
 Lehrende: Prof. Dr. Thorsten Berger, Dr. Sven Peldszus

Verwendung des Moduls
 B.Sc. Informatik [bis SS 23]
 B.Sc. IT-Sicherheit [bis SS 23]
 B.Sc. Angewandte Informatik [bis SS 23]
 M.Sc. Computer Science
 M.Sc. Angewandte Informatik
 M.Sc. IT-Sicherheit/Informationstechnik
 M.Sc. IT-Sicherheit/Netze und Systeme [bis SS 23]

Vorkenntnisse
 Die Vorlesung Software Engineering oder eine vergleichbare Veranstaltung, Programmiererfahrungen z.B. im Rahmen anderer Lehrveranstaltungen.

- Lernziele (learning outcomes)**
- Verständnis der Anforderungen an autonome Fahrzeuge
 - Verständnis der Architektur von autonomen Fahrzeugen
 - Fähigkeit, ein selbstfahrendes Auto mit ROS2 zu bauen
 - Verstehen und Anwenden der Qualitätssicherung für autonome Fahrzeuge

Inhalt
 Autonomes Fahren ist die Zukunft der individuellen Mobilität und alle großen Hersteller arbeiten an vollautonomen Fahrzeugen. Während es für die einzelnen Probleme des autonomen Fahrens robuste und gute Lösungen gibt, liegt die größte Herausforderung in deren Integration. Insgesamt stellt die Software eines autonomen Fahrzeugs das größte Problem dar. Daher liegt der Schlüssel für selbstfahrende Fahrzeuge darin, die Software richtig zu machen. In diesem Kurs werden wir die verschiedenen Aspekte von selbstfahrenden Fahrzeugen sowie die Bedeutung und Anwendung von künstlicher Intelligenz in diesem Bereich untersuchen. Der Kurs wird sich hauptsächlich auf die folgenden Themen konzentrieren:

- Anforderungen an autonome Fahrzeuge
- Architektur von autonomen Fahrzeugen
- Betriebssysteme und Frameworks für Robotersysteme
- Spezifikation und Implementierung von autonomen Fahrzeugen auf Basis von ROS2
- Künstliche Intelligenz für autonome Fahrzeuge
- Simulation von autonomen Fahrzeugen Lokalisierung und Wahrnehmung

- Missionsplanung
- Qualitätssicherung für autonome Fahrzeuge In der Vorlesung werden die notwendigen theoretischen Grundlagen vermittelt und die Inhalte in Übungen durch den Bau eines selbstfahrenden Roboters praktisch angewendet.

Lehrformen

Vorlesung mit Übungen

Prüfungsformen

Mündliche Modulabschlussprüfung (30 Minuten)

Voraussetzungen für die Vergabe von Credits

Bestandene Modulabschlussprüfung und erfolgreiche Teilnahme an den Übungen

Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)

5/97: M.Sc. Computer Science

5/105: M.Sc. Angewandte Informatik

5/91: M.Sc. IT-Sicherheit/Informationstechnik [PO 22]

5/84: M.Sc. IT-Sicherheit/Informationstechnik [PO 20]

Titel des Moduls: Autonomous Vehicles and Artificial Intelligence Lab
Autonomous Vehicles and Artificial Intelligence Lab

Modul-Nr./Code	Credits 5 CP	Workload 150h	Semester siehe Prüfungsordnung / see examination regulations	Turnus Wintersemester	Dauer 1 Semester
Lehrveranstaltungen Autonomous Vehicles and Artificial Intelligence Lab (212035)			Kontaktzeit 60h	Selbststudium 90h	Gruppengröße Studierende
Unterrichtssprache Englisch			Teilnahmevoraussetzungen -		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr Thorsten Berger Lehrende: Prof. Dr. Thorsten Berger Dr. Sven Peldszus					
Verwendung des Moduls M.Sc. Computer Science M.Sc. Angewandte Informatik M.Sc. IT-Sicherheit/Informationstechnik					
Vorkenntnisse <p>Empfohlen:</p> Teilnahme an der Vorlesung Autonomous Vehicles and Artificial Intelligence (211044) Programmiererfahrung in C++ oder Python (z.B. als Teil von anderen Kursen) Teilnahme an der Vorlesung Software Engineering (212000) oder einer vergleichbaren Lehrveranstaltung 					
Lernziele (learning outcomes) Wissen - relevante theoretische Kenntnisse über künstliche Intelligenz und autonome Fahrzeuge erläutern können Fertigkeiten und Fähigkeiten - Anforderungen an autonome Fahrzeuge definieren und validieren - eine Architektur für autonome Fahrzeuge erstellen - ein selbstfahrendes Auto mit ROS2 bauen - Management und Integration von künstlicher Intelligenz in komplexe, softwareintensive Systeme - Organisation eines Teams und seines Entwicklungsprozesses für ein komplexes, softwareintensives System - Qualitätssicherung für autonome Fahrzeuge durchführen - Erstellen der Dokumentation des Entwicklungsprozess und der Artefakte, die für eine Zertifizierung nach den ISO-Normen für Straßenfahrzeuge benötigt werden - professionell mit Gruppenmitgliedern und Stakeholdern kommunizieren (in Wort und Schrift)					
Inhalt Autonomes Fahren ist die Zukunft der individuellen Mobilität, und alle großen Hersteller arbeiten an vollständig					

autonomen Fahrzeugen. Während es für die einzelnen Probleme des autonomen Fahrens robuste und gut erforschte Lösungen gibt, liegt die größte Herausforderung in deren Integration. Insgesamt stellt die Software eines autonomen Fahrzeugs das größte Problem dar. Daher liegt der Schlüssel für selbstfahrende Fahrzeuge darin, die Software richtig zu gestalten.

In diesem Kurs werden wir die verschiedenen Aspekte von selbstfahrenden Fahrzeugen sowie die Bedeutung und Anwendung von künstlicher Intelligenz in diesem Bereich anhand der Entwicklung eines selbstfahrenden Rennwagens praktisch studieren. Zu diesem Zweck werden die Teilnehmer mit ROS2-basierten Modellautos arbeiten. Ziel ist es, den Studierenden praktische Erfahrungen bei der Entwicklung eines autonomen Rennwagens und der Organisation des Entwicklungsprozesses zu vermitteln.

Lehrformen

Die wichtigste Lernsequenz des Kurses ist ein großes Praxisprojekt. Das Projekt wird in Gruppen durchgeführt, die iterativ einen autonomen Rennwagen entwickeln und dabei theoretisches Wissen über autonomes Fahren und Softwareentwicklung anwenden und festigen. Um das Lernen zu unterstützen, basiert das Autonomous Vehicles and Artificial Intelligence Lab auf seminarähnlichen Vorlesungen, die eine Plattform für Feedback und weitere Informationen bieten. Auf der Grundlage der gesammelten Informationen aktualisieren und verfeinern die Studierenden ihre Lösungen für ein autonomes Rennauto. Die kontinuierliche Reflexion über Praxis und Theorie wird durch die laufende Erstellung eines abschließenden Projektberichts parallel zur Entwicklung des Rennwagens unterstützt, in dem die Studierenden über ihr eigenes Lernen im Kurs, die Art und Weise, wie sie und ihr Team ihren Entwicklungsprozess angehen, und ihre technischen Lösungen reflektieren. Die Studenten erhalten regelmäßiges Feedback und Anleitung, um ihr Lernen zu unterstützen.

Prüfungsformen

Die Endnote wird auf der Grundlage der Teilnahme an der Entwicklung des selbstfahrenden Rennwagens, schriftlicher Projektberichte, des entwickelten autonomen Rennwagens und einer mündlichen Präsentation der Gruppenergebnisse ermittelt. Die Einzelnoten werden aus der Bewertung der individuellen und gruppenbezogenen Ergebnisse gebildet.

Voraussetzungen für die Vergabe von Credits

Eine aktive Beteiligung an der Entwicklung eines autonomen Rennwagens, regelmäßiger Besuch der seminarähnlichen Vorlesungen und erfolgreiche Erbringung aller Leistungen.

Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)

5/97: M.Sc. Computer Science

5/105: M.Sc. Angewandte Informatik

5/91: M.Sc. IT-Sicherheit/Informationstechnik [PO 22]

5/84: M.Sc. IT-Sicherheit/Informationstechnik [PO 20]

Titel des Moduls: Bildverarbeitung in der Medizin

Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester siehe Prüfungsordnung	Turnus Sommersemester	Dauer 1 Semester
Lehrveranstaltungen Bildverarbeitung in der Medizin (141220)			Kontaktzeit 60 h	Selbststudium 90 h	Gruppengröße 20 Studierende
Unterrichtssprache Deutsch			Teilnahmevoraussetzungen Eine Teilnahme ist ausgeschlossen, wenn im Bachelor bereits das Modul Computersehen: Einführung (Digitale Bildverarbeitung) absolviert wurde.		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr.-Ing. Georg Schmitz Lehrende: Dr.-Ing. Stefanie Dencks					
Verwendung des Moduls M.Sc. Angewandte Informatik					
Vorkenntnisse <p>Empfohlen: Kenntnisse der Systemtheorie, Fourier-Transformation und Signalverarbeitung, die denen entsprechen, die als Grundlagen in den Vorlesungen des Bachelorstudienganges Elektrotechnik und Informationstechnik vermittelt werden. Grundkenntnisse in der Matlab-Programmierung.</p>					
Lernziele (learning outcomes) <p>Nach erfolgreichem Abschluss des Moduls verfügen die Studierenden über Kenntnisse der mehrdimensionalen digitalen Signalverarbeitung. Sie kennen und verstehen die Aufnahme mehrdimensionaler Bilddaten der wichtigsten diagnostischen Abbildungsverfahren, können diese modellieren und hieraus Konsequenzen für ihre Verarbeitung ableiten. Die Studierenden können die verschiedenen Schritte der Bildverarbeitung in abstrakte Aufgabenkategorien einordnen (z.B. Filterung, Segmentierung, Klassifikation) und kennen ausgewählte Verfahren im Detail und können diese erläutern und anwenden. Die Studierenden sind in der Lage, eine gegebene Bildverarbeitungsaufgabe zu analysieren, sowie einen geeigneten Lösungsweg zu entwickeln und algorithmisch umzusetzen. Die Verfahren werden am Beispiel medizinischer Bilddaten vermittelt, die Studierenden können die Verfahren aber auch auf andere Anwendungsgebiete übertragen. Durch die Übungen in Kleingruppen, teilweise an Rechnern, sind die Studierenden befähigt, das Erlernte im Team praktisch umzusetzen, Lösungsansätze zu erläutern und argumentativ zu vertreten.</p>					
Inhalt <p>Es werden die Grundlagen und speziellen Verfahren der Bildverarbeitung vorgestellt, die insbesondere bei medizinischen Bilddaten Anwendung finden. Viele Verfahren werden jedoch auch in anderen Anwendungsfeldern wie z.B. der industriellen Bildverarbeitung eingesetzt.</p> <p>Im ersten Abschnitt werden sowohl die Rezeption durch das menschliche visuelle System behandelt, als auch Definitionen und Grundlagen für die Bildverarbeitung eingeführt (z.B. Diskretisierung, Abtasttheorem, globale Kenngrößen von Bildern). Der zweite Abschnitt vermittelt die wichtigsten Operationen im Ortsbereich (Histogrammmodulation, Filterung morphologische Operationen, geometrische Bildoperationen, distance transform, ...). Der dritte Abschnitt umfasst Methoden der Informationsextraktion (Segmentierung, Texturanalyse, Formbeschreibung). Im vierten Abschnitt liegt der Schwerpunkt auf der Klassifikation und verschiedenen Verfahren des Machine Learning (z.B. support vector machines, deep learning). Der fünfte Abschnitt beinhaltet die Bildrestauration. Zusätzlich wird ein Überblick über die Bildregistrierung und 3D-Visualisierung gegeben.</p>					

Lehrformen

Vorlesung mit begleitender Übung. Diese Lehrveranstaltung wird über Moodle organisiert. Die notwendigen Informationen werden in der ersten Vorlesung mitgeteilt.

Prüfungsformen

Mündliche Modulabschlussprüfung (30 Minuten)

Voraussetzungen für die Vergabe von Credits

Bestandene mündliche Modulabschlussprüfung

Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)

5/105: M.Sc. Angewandte Informatik

Titel des Moduls: Bioimage Informatics Bioimage Informatics					
Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester siehe Prüfungsordnung	Turnus Wintersemester	Dauer 1 Semester
Lehrveranstaltungen Bioimage Informatics (190800)			Kontaktzeit 60 h	Selbststudium 90 h	Gruppengröße Studierende
Unterrichtssprache Deutsch			Teilnahmevoraussetzungen		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Axel Mosig Lehrende: Prof. Dr. Axel Mosig					
Verwendung des Moduls M.Sc. Angewandte Informatik					
Vorkenntnisse Grundlegende Programmierkenntnisse					
Lernziele (learning outcomes) Die Studierenden sollen elementare und aktuelle Techniken der Bildverarbeitung kennenlernen, die zur Analyse mikroskopischer Bilddaten verwendet werden. Sie sollen die dahinterliegenden algorithmischen, mathematischen und statistischen Ideen verstehen, und lernen in der Praxis, wie diese Methoden auf reale Daten und Fragestellungen angewendet werden.					
Inhalt Die Analyse von mikroskopischen Bilddaten mit Methoden der Bildverarbeitung ist in den vergangenen Jahren ein wichtiges Thema in vielen Anwendungen in den Lebenswissenschaften geworden. In der Vorlesung werden grundlegende Konzepte der Verarbeitung mikroskopischer Bilddaten und deren Anwendungen behandelt. Die Vorlesung gliedert sich wie folgt: <ul style="list-style-type: none"> • Grundlagen der Mikroskopie: Durchlicht- und Konfokalmikroskopie; Fluoreszenzmikroskopie; Mikroskopie in der Histopathologie; Mikroskopie in der Zellbiologie; Mikroskopie in der Neurobiologie • Morphologische Bildanalyse • Texturelle Bildanalyse • Kolokalisations-Verfahren • Algorithmen zur Bildregistrierung • Algorithmen zum Verfolgen von Bewegungsmustern von Zellen ("Cell Tracking") • Verfahren zur Rekonstruktion von Neuronen ("Neuron Tracing") • Deep Learning Methoden zur mikroskopischen Bilddatenanalyse • Methoden der Analyse von markerfreien Mikroskopie-Daten • Methoden zur überwachten und unüberwachten Segmentierung von mikroskopischen Bildern 					
Lehrformen Vorlesung mit Übung					
Prüfungsformen Schriftliche Modulabschlussprüfung (120 Minuten)					
Voraussetzungen für die Vergabe von Credits Bestandene schriftliche Modulabschlussprüfung					
Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS) 5/105: M.Sc. Angewandte Informatik					

Titel des Moduls: Bioinformatics for Proteomics Bioinformatics for Proteomics					
Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester siehe Prüfungsordnung	Turnus Wintersemester	Dauer 1 Semester
Lehrveranstaltungen Bioinformatics for Proteomics (201911)			Kontaktzeit 45 h	Selbststudium 105 h	Gruppengröße Studierende
Unterrichtssprache English			Teilnahmevoraussetzungen		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Martin Eisenacher Lehrende: Prof. Dr. Martin Eisenacher					
Verwendung des Moduls M.Sc. Angewandte Informatik					
Vorkenntnisse					
Lernziele (learning outcomes)					
Inhalt					
Lehrformen					
Prüfungsformen					
Voraussetzungen für die Vergabe von Credits					
Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS) 5/105: M.Sc. Angewandte Informatik					

Titel des Moduls: Complexity Economics and Agent-based Modeling
Complexity Economics and Agent-based Modeling

Modul-Nr./Code	Credits 10 CP	Workload 300 h	Semester siehe Prüfungsordnung	Turnus unregelmäßig (in der Regel im WS)	Dauer 1 Semester
Lehrveranstaltungen Complexity economics and agent-based modeling (075240 + 075241)			Kontaktzeit 60 h	Selbststudium 240 h	Gruppengröße Studierende
Unterrichtssprache Englisch			Teilnahmevoraussetzungen		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Michael Roos Lehrende: Prof. Dr. Michael Roos					
Verwendung des Moduls M.Sc. Angewandte Informatik					
Vorkenntnisse Kenntnisse der Makroökonomie (z.B. durch Besuch der Veranstaltung Grundlagen der Makroökonomie als Freies Wahlfach)					
Lernziele (learning outcomes) Das Modul verfolgt das Ziel, Studierende der Angewandten Informatik sowie der Wirtschaftswissenschaft in die Methode der agentenbasierten Modellierung und den dazugehörigen Simulationstechniken einzuführen. Hierbei lernen Studierende wie volkswirtschaftliche und sozialwissenschaftliche Fragestellungen entworfen, implementiert und analysiert werden. Im Rahmen der Seminararbeit sollen die Studierenden ein eigenes Modell implementieren und analysieren. Im Rahmen des Moduls erwerben Studierende folgende Kenntnisse: <ul style="list-style-type: none"> • Grundlegende Kenntnisse der Komplexitätsökonomik und agentenbasierter Modellierung • Überführung von wissenschaftlichen Fragestellungen aus dem Bereich Volkswirtschaftslehre in agentenbasierte Computersimulationen • Praktische Arbeit mit agentenbasierten Modellen, Interpretation von Simulationsergebnissen und mögliche Grenzen des Ansatzes • Grundlegende Kenntnisse, um eigene Modelle zu implementieren und Simulationen selbständig durchzuführen • Wissenschaftliches Schreiben • Die Programmiersprache NetLogo 					
Inhalt Komplexitätsökonomik ist eine neue Denkschule im Bereich der Volkswirtschaftslehre, der zunehmend an Relevanz gewinnt. Hierbei wird eine Volkswirtschaft als ein komplexes, adaptives System betrachtet, das sich aus verschiedenen Agenten, z.B. Firmen, Haushalten und Banken, zusammensetzt. Im Vergleich zu bisherigen volkswirtschaftlichen Ansätzen sind die Agenten des Systems nicht (vollständig) rational und das System entwickelt sich aus den Interaktionen zwischen den (heterogenen) Agenten. Komplexitätsökonomik eignet sich sehr zur Untersuchung von sozialen Transformations- und Innovationsprozessen, wie der Digitalisierung. In der Vorlesung wird Komplexitätsökonomik und die Methode der agentenbasierten Computersimulation vorgestellt und gezeigt, wie sie zur Analyse komplexer ökonomischer Systeme angewendet werden können. Im Rahmen der Übung werden diese Kenntnisse vertieft. Zudem wird geübt, wie agentenbasierte Modelle mithilfe NetLogo implementiert und simuliert werden.					
Lehrformen Vorlesung und Übung					
Prüfungsformen Seminararbeit (70%) und Klausur (30%) (benotet)					

Studienleistung (unbenotet)

Voraussetzungen für die Vergabe von Credits

Bestehen der folgenden Leistungen:

- 1) Studienleistung (unbenotet)
- 2) Seminararbeit (70%) und Klausur (30%) (benotet)

Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)

10/105: M.Sc. Angewandte Informatik

Titel des Moduls: Computational Linguistics and AI					
Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester siehe Prüfungsordnung	Turnus Sommersemester	Dauer 1 Semester
Lehrveranstaltungen Computational Linguistics and AI (050825)			Kontaktzeit 60 h	Selbststudium 90 h	Gruppengröße 20 Studierende
Unterrichtssprache Englisch			Teilnahmevoraussetzungen Erfolgreich abgeschlossenes Modul "Introduction to Computational Linguistics in Python".		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Tibor Kiss Lehrende: Mirjam Koch M.Sc.					
Verwendung des Moduls M.Sc. Angewandte Informatik					
Vorkenntnisse Inhalte aus dem Modul "Introduction to Computational Linguistics in Python"					
Lernziele (learning outcomes) <ul style="list-style-type: none"> Die Studierenden können auf den Grundlagen der Programmierung und Nutzung von Modulen aufbauen, um Verfahren der „künstlichen Intelligenz“ bzw. des maschinellen Lernens auf linguistische Fragestellungen anzuwenden. Sie kennen einige Grundlagen der kollaborativen Softwareentwicklung in der Computerlinguistik. Sie kennen grundlegende Evaluationsmethoden für Programme und können sie anwenden. Sie können sich kritisch mit ihrer Rolle in der Verarbeitung von Daten auseinandersetzen und wissen, dass sie durch die Wahl von Datentypen/-codierung, Algorithmen und ihre Auswertung der Ergebnisse Bias einfügen. 					
Inhalt In diesem Kurs geht es um die Zusammenhänge zwischen Computerlinguistik und Verfahren des maschinellen Lernens bzw. der „KI“. Aufbauend auf den Grundkenntnissen aus „Introduction to Computational Linguistics (with Python“) im Sommersemester werden diese nachvollzogen und im Bezug auf linguistische Fragestellungen diskutiert. Die „Vorlesungen“ bestehen aus einer Vortrags- und Diskussionskomponente. Dabei enthalten die auch hochgeladenen Folien die Inhalte, die für die Klausur relevant sind, aber keine ausführlichen Erklärungen. Die Einordnung neuer Verfahren, Methoden und Ideen findet gemeinsam im Plenum statt. Die begleitenden Übungen zur Vorlesung sind nicht verpflichtend, aber sind zur Selbstkontrolle offen für alle Studierenden. In den Übungen und Übungsaufgaben liegt der Fokus eher auf der praktischen Anwendung und Implementierung.					
Lehrformen Die Veranstaltung setzt sich aus einer Präsenzkomponekte (Fragen und Antworten zu den Folien, Diskussionen, praktische Übungen und Überlegungen zum aktuellen Thema) und Übungen zum Kurs zusammen. Die Studierenden müssen sich eigenständig auf die Sitzungen vorbereiten und entscheiden, ob die Teilnahme ihr Verständnis fördert.					
Prüfungsformen Schriftliche Modulabschlussprüfung					
Voraussetzungen für die Vergabe von Credits Bestandene schriftliche Modulabschlussprüfung					
Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS) 5/105: M.Sc. Angewandte Informatik					

Titel des Moduls: Computational Neuroscience: Neural Dynamics
Computational Neuroscience: Neural Dynamics

Modul-Nr./Code	Credits 6 CP	Workload 180 h	Semester siehe Prüfungsordnung / see examination regulations	Turnus Wintersemester	Dauer 1 Semester
Lehrveranstaltungen Computational Neuroscience: Neural Dynamics (212005)			Kontaktzeit 45 h	Selbststudium 135 h	Gruppengröße Studierende

Unterrichtssprache Englisch	Teilnahmevoraussetzungen
---------------------------------------	---------------------------------

Modulbeauftragte/r und hauptamtlich Lehrende
 Modulbeauftragte/r: Prof. Dr. Gregor Schöner
 Lehrende: Prof. Dr. Gregor Schöner

Verwendung des Moduls
 M.Sc. Angewandte Informatik
 M.Sc. Computer Science

Vorkenntnisse
 <p>Dieser Kurs erfordert eine gewisse mathematische Grundvorbereitung, wie sie typischerweise in zwei Semestern höherer Mathematik vermittelt wird (Funktionen, Differenzierung, Integration, Differentialgleichungen, lineare Algebra). Der Kurs macht keinen umfassenden Gebrauch von den zugrunde liegenden mathematischen Techniken, sondern verwendet die mathematischen Konzepte, um wissenschaftliche Ideen auszudrücken. Studierende, die keine Vorkenntnisse in der entsprechenden Mathematik haben, können dem Kurs zwar folgen, müssen sich aber intensiver mit den Konzepten vertraut machen.</p>

- Lernziele (learning outcomes)**
- Erfahrungen in der Interdisziplinarität zwischen Informatik und Kognitionswissenschaft sammeln.
 - Konzepte und Methoden nichtlinearer dynamischer Systeme in einem konkreten Anwendungskontext erlernen.
 Verbesserung der Vertrautheit mit Methoden der quantitativen Naturwissenschaft, einschließlich Messung, grafische Darstellung von Beobachtungswerten in Abhängigkeit von experimentellen Kontrollparametern und Verwendung von Modellen zur Interpretation von Daten.
 - Lesen von wissenschaftlicher Literatur.

Inhalt
 Dieser Kurs bietet eine Einführung in die theoretischen kognitiven und funktionellen Neurowissenschaften aus einem besonderen theoretischen Blickwinkel, dem Ansatz der dynamischen Systeme. Dieser Ansatz betont die zeitliche Entwicklung von Verhaltens- und neuronalen Mustern als Grundlage für ihre Analyse und Synthese. Dynamische Stabilität, ein Konzept, das mit der klassischen biologischen Kybernetik geteilt wird, ist ein Eckpfeiler dieses Ansatzes. Instabilitäten (oder Bifurkationen) erweitern diesen Rahmen und bieten eine Grundlage für das Verständnis von Flexibilität, aufgabenspezifischer Anpassung, Adaptation und Lernen. Der Kurs umfasst Lernmodule, die mathematische Grundlagen vermitteln. Theoretische Konzepte werden in Bezug auf eine Reihe von experimentellen Modellsystemen erläutert, die die Bewegungskoordination, die Haltungsverstabilität, die Bewegungswahrnehmung und elementare Formen der verkörperten Kognition umfassen. Im Geiste von Braitenbergs "synthetischer Psychologie" werden autonome Roboter zur Veranschaulichung einiger der Ideen verwendet.
 In die Vorlesungen sind Übungen integriert. Sie bestehen aus elementaren mathematischen Übungen, dem Entwurf von (Gedanken-)Experimenten und deren Analyse sowie dem Entwurf einfacher künstlicher Systeme, alles auf der Grundlage des in den Hauptvorlesungen dargestellten theoretischen Rahmens.
 Eine Übung hat die Form eines Aufsatzes, für den die Teilnehmer eine wissenschaftliche Arbeit lesen und Fragen in einem längeren illustrierten Text beantworten.

Lehrformen
 Vorlesung mit Übung

Prüfungsformen

Mündliche Modulabschlussprüfung, Bonuspunkte für die Abschlussprüfung können erreicht werden durch das Einreichen von Hausaufgaben und eines Essays (10 Seiten).

Voraussetzungen für die Vergabe von Credits

Bestandene mündliche Modulabschlussprüfung.

Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)

6/105: M.Sc. Angewandte Informatik

6/97: M.Sc. Computer Science

Titel des Moduls: Computational Neuroscience: Single-Neuron Models
Computational Neuroscience: Single-Neuron Models

Modul-Nr./Code	Credits 6 CP	Workload 180 h	Semester siehe Prüfungsordnung / see examination regulations	Turnus Sommersemester	Dauer 1 Semester
Lehrveranstaltungen Computational Neuroscience: Single-Neuron Models (211039)			Kontaktzeit 60 h	Selbststudium 120 h	Gruppengröße Studierende
Unterrichtssprache Englisch			Teilnahmevoraussetzungen		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Robert Schmidt Lehrende: Prof. Dr. Robert Schmidt					
Verwendung des Moduls M.Sc. Computer Science M.Sc. Angewandte Informatik					
Vorkenntnisse Programmierung in Python, mathematische Kenntnisse (lineare Algebra und Infinitesimalrechnung) und Interesse an Neurobiologie.					
Lernziele (learning outcomes) Nach Abschluss des Moduls können Studierende <ul style="list-style-type: none"> • können Studierende Techniken der Computational Neuroscience zur Simulation neuronaler Aktivität anwenden • haben sich Studierende mit verschiedenen Arten von Modellen einzelner Neuronen, ihrer mathematischen Beschreibung und ihren verschiedenen biologischen Abstraktionsebenen vertraut gemacht • haben Studierende die Fähigkeiten zur Modellierung von Neuronen, Synapsen und Schaltkreisen erworben und können diese Modelle mit Biologie und Berechnung verknüpfen • besitzen Studierende ein Verständnis der biologischen Grundlage für Berechnungen in Neuronen 					
Inhalt Dieses Modul beginnt mit einer Einführung in die Neurowissenschaften und die Rolle der Computational Neuroscience. Der nächste Teil des Moduls befasst sich mit biologisch fundierten Modellen einzelner Neuronen, einschließlich Leaky-Integrate-and-Fire- und Leitwert-basierter Neuronen, aber auch mit abstrakteren Modellen neuronaler Aktivität und Spike Trains. Sie werden lernen, wie diese verschiedenen Berechnungsmodelle die zugrunde liegenden biologischen Prozesse in unterschiedlichem Maße beschreiben und vereinfachen. Wir werden im Detail untersuchen, wie diese verschiedenen Neuronenmodelle in numerischen Simulationen verwendet werden können, um Forschungsfragen zur Berechnung in einzelnen Neuronen und Schaltkreisen zu beantworten. In den vorlesungsbegleitenden Übungen werden Sie praktische Erfahrungen mit der Implementierung der verschiedenen Neuronenmodelle in Python, der Durchführung numerischer Simulationen und der Durchführung von Berechnungen im Zusammenhang mit analytischen Lösungen der Modellgleichungen und der Biophysik sammeln. Der Schwerpunkt liegt auf Einzelneuronenmodellen, aber wir werden auch verfügbare Software (z.B. NEST Desktop) nutzen, um zu untersuchen, wie Einzelneuronenmodelle in Simulationen von neuronalen Netzwerken integriert werden können. Während der Schwerpunkt des Moduls auf methodischen Fragen liegt und darauf, wie Modelle auf jeder Ebene aufgebaut, getestet und validiert werden können, werden wir auch Verbindungen zu bestimmten Gehirnregionen herstellen, um die Modelle zu motivieren und zu veranschaulichen.					
Lehrformen Vorlesung mit Übungen					
Prüfungsformen Schriftliche Modulabschlussprüfung (120 Minuten)					

Voraussetzungen für die Vergabe von Credits

Bestehen der schriftlichen Modulabschlussprüfung.

Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)

6/97: M.Sc. Computer Science

6/105: M.Sc. Angewandte Informatik

Titel des Moduls: Computational Neuroscience: Vision and Memory
Computational Neuroscience: Vision and Memory

Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester siehe Prüfungsordnung / see examination regulations	Turnus Sommersemester	Dauer 1 Semester
Lehrveranstaltungen Computational Neuroscience: Vision and Memory (211049)			Kontaktzeit 60 h	Selbststudium 90 h	Gruppengröße 20 Studierende
Unterrichtssprache Englisch			Teilnahmevoraussetzungen		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Laurenz Wiskott Lehrende: Prof. Dr. Laurenz Wiskott					
Verwendung des Moduls M.Sc. Computer Science M.Sc. Angewandte Informatik M.Sc. Cognitive Science					
Vorkenntnisse Das mathematische Niveau des Kurses ist unterschiedlich, aber im Allgemeinen hoch. Das Tutorium ist fast vollständig mathematisch. Zu den erforderlichen mathematischen Grundlagen gehören Infinitesimalrechnung (Funktionen, Ableitungen, Integrale, Differentialgleichungen, ...), lineare Algebra (Vektoren, Matrizen, inneres Produkt, orthogonale Vektoren, Basissysteme, ...) und ein wenig Wahrscheinlichkeitstheorie (Wahrscheinlichkeiten, Wahrscheinlichkeitsdichten, Satz von Bayes, ...).					
Lernziele (learning outcomes) Nach erfolgreichem Abschluss dieses Kurses <ul style="list-style-type: none"> • kennen Studierende grundlegende neurobiologische Fakten über das visuelle System und den Hippocampus, • kennen sie eine Reihe von verwandten Modellen und Methoden der Computergestützten Neurowissenschaften, • verstehen sie die Mathematik dieser Methoden, • können über all dies in englischer Sprache kommunizieren. 					
Inhalt Diese Vorlesung behandelt die Grundlagen der Neurobiologie und Modelle der Selbstorganisation in neuronalen Systemen, insbesondere die Themen					
Lernen und Selbstorganisation <ul style="list-style-type: none"> • Hebbsches Lernen • Neuronale Lerndynamik und eingeschränkte Optimierung • Dynamische Feldtheorie 					
Sehen <ul style="list-style-type: none"> • Wahrnehmungsfelder • Neuronale Karten • Hippocampus • Navigation 					

- Episodisches Gedächtnis
- Hopfield-Netzwerk

Lehrformen

Dieser Kurs wird nach dem Konzept des "flipped/inverted classroom" durchgeführt. Zunächst arbeiten die Studierenden das Online-Material selbständig durch. In der Vorlesungszeit diskutieren wir dann den Stoff, finden Verbindungen zu anderen Themen, stellen Fragen und versuchen, sie zu beantworten. Im Tutorium wird das neu erworbene Wissen in analytischen Übungen angewendet und dadurch vertieft. Ich ermutige alle Studierenden, sowohl in der Selbstlernzeit als auch im Tutorium in Teams zu arbeiten.

Prüfungsformen

Schriftliche (digitale) Modulabschlussprüfung (90 Minuten).

Voraussetzungen für die Vergabe von Credits

Bestandene schriftliche Modulabschlussprüfung.

Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)

5/97: M.Sc. Computer Science

5/105: M.Sc. Angewandte Informatik

Titel des Moduls: Deep Learning Deep Learning					
Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester siehe Prüfungsordnung / see examination regulations	Turnus Wintersemester	Dauer 1 Semester
Lehrveranstaltungen Deep Learning (212018)			Kontaktzeit 60 h	Selbststudium 90 h	Gruppengröße 50 Studierende
Unterrichtssprache Deutsch			Teilnahmevoraussetzungen		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Asja Fischer Lehrende: Prof. Dr. Asja Fischer					
Verwendung des Moduls M.Sc. IT-Sicherheit/ Informationstechnik M.Sc. IT-Sicherheit/ Netze und Systeme [Bis WS 22/23] M.Sc. Angewandte Informatik M.Sc. Computer Science					
Vorkenntnisse Grundkenntnisse der Linearen Algebra und Wahrscheinlichkeitstheorie sind von Vorteil.					
Lernziele (learning outcomes) Die Vorlesung hat das Ziel, einen Einblick in dieses Gebiet zu vermitteln. Zu Beginn werden die grundlegenden Begriffe und Konzepte des maschinellen Lernens eingeführt. Im weiteren Verlauf wird auf verschiedene neuronale Netze, Gradienten-basierte Optimierungsverfahren und generative Modelle eingegangen.					
Inhalt Deep Learning ist ein Untergebiet des maschinellen Lernens, welches in den letzten Jahren zu Durchbrüchen in zahlreichen Anwendungsgebieten (wie z.B. in der Objekt- und Spracherkennung und der maschinellen Übersetzung) geführt hat. Deep Learning Methoden finden unter anderem Anwendung im Bereich IT Security.					
Lehrformen Vorlesung und Übung					
Prüfungsformen Schriftliche Modulabschlussprüfung (120 Minuten)					
Voraussetzungen für die Vergabe von Credits Bestandene schriftliche Modulabschlussprüfung.					
Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS) 5/91: M.Sc. IT-Sicherheit/ Informationstechnik [PO 22] 5/84: M.Sc. IT-Sicherheit/ Informationstechnik [PO 20] 5/105: M.Sc. Angewandte Informatik 5/ 97: M.Sc. Computer Science					

Titel des Moduls: Design Optimization (kein Angebot im WS 23/24)
Design Optimization (No offer in WS 23/24)

Modul-Nr./Code	Credits 6 CP	Workload 180 h	Semester siehe Prüfungsordnung	Turnus Wintersemester	Dauer 1 Semester
Lehrveranstaltungen Design Optimization (129007)			Kontaktzeit 60 h	Selbststudium 120 h	Gruppengröße Studierende
Unterrichtssprache Englisch			Teilnahmevoraussetzungen		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Markus König Lehrende: Prof. Dr. Markus König					
Verwendung des Moduls M.Sc. Angewandte Informatik					
Vorkenntnisse					
Lernziele (learning outcomes)					
Inhalt					
Lehrformen					
Prüfungsformen					
Voraussetzungen für die Vergabe von Credits					
Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS) 6/105: M.Sc. Angewandte Informatik					

Titel des Moduls: Deterministic Network Calculus Deterministic Network Calculus					
Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester siehe Prüfungsordnung / see examination regulations	Turnus Sommersemester	Dauer 1 Semester
Lehrveranstaltungen Deterministic Network Calculus (211054)			Kontaktzeit 60 h	Selbststudium 90 h	Gruppengröße Studierende
Unterrichtssprache Englisch			Teilnahmevoraussetzungen		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Steffen Bondorf Lehrende: Prof. Dr. Steffen Bondorf					
Verwendung des Moduls M.Sc. Computer Science M.Sc. IT-Sicherheit/ Informationstechnik M.Sc. Angewandte Informatik					
Vorkenntnisse Mathematik (Funktionsanalyse), Computernetze / Verteilte Systeme					
Lernziele (learning outcomes) Nach erfolgreichem Abschluss des Moduls werden die Studierenden in der Lage sein, <ul style="list-style-type: none"> komplexe, vernetzte Systeme als deterministische Warteschlangensysteme zu modellieren, worst-case Leistungsanalysen von bestehenden Systemen bzw. Modellen durchzuführen, die Herausforderungen bei der Leistungsdimensionierung von geplanten Systemen zu verstehen, &#8729; dabei die Wirkungsweise zentraler Mechanismen in Computernetzen anhand des Network Calculus zu erklären, die vorgestellten Verfahren gegeneinander abzugrenzen und auf wissenschaftliche Fragestellungen anzuwenden. 					
Inhalt Verteilte Systeme sind heutzutage allgegenwärtig, und ihre Vernetzung ist von grundlegender Bedeutung für die kontinuierliche Verbreitung und damit Verfügbarkeit von Daten. Die Bereitstellung von Daten in Echtzeit ist einer der wichtigsten nichtfunktionalen Aspekte, den sicherheitskritische Netze gewährleisten müssen. Die formale Verifizierung der Datenkommunikation im Hinblick auf die worst-case Deadlines ist grundlegend für die Zertifizierung von neu entwickelten x-by-Wire-Systemen. Diese Verifizierung erlaubt den Start von Flugzeugen, das Lenken von Autos ohne mechanische Verbindung und den Betrieb sicherheitskritischer Industrieanlagen. Daher wurden verschiedene Methoden für die worst-case Modellierung und Analyse von Echtzeitsystemen entwickelt. Eine davon ist der Deterministische Network Calculus (DNC), eine vielseitige Technik, die in verschiedenen Bereichen wie Paketvermittlung, Task Scheduling, System on Chip, softwaredefinierte Netzwerke, Netzwerke in Rechenzentren und Netzwerkvirtualisierung eingesetzt werden kann. DNC ist eine Methode zur Ableitung deterministischer Schranken für zwei der vorrangigsten Leistungsmetriken in Kommunikationssystemen: <ul style="list-style-type: none"> die Ende-zu-Ende-Verzögerung von Datenflüssen und der Speicherplatz, den ein Server benötigt, um alle eingehenden Daten in einer Warteschlange zu puffern. 					
Lehrformen Vorlesung mit Übung					
Prüfungsformen Mündliche Modulabschlussprüfung (30 Minuten)					
Voraussetzungen für die Vergabe von Credits Bestandene mündliche Modulabschlussprüfung.					

Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)

5/97: M.Sc. Computer Science

5/91: M.Sc. IT-Sicherheit/ Informationstechnik [PO 22]

5/84: M.Sc. IT-Sicherheit/ Informationstechnik [PO 20]

5/105: M.Sc. Angewandte Informatik

Titel des Moduls: Effiziente Algorithmen					
Modul-Nr./Code	Credits 9 CP	Workload 270 h	Semester siehe Prüfungsordnung	Turnus Sommersemester	Dauer 1 Semester
Lehrveranstaltungen Effiziente Algorithmen (150320 + 150321)			Kontaktzeit 90 h	Selbststudium 180 h	Gruppengröße Studierende
Unterrichtssprache Deutsch			Teilnahmevoraussetzungen		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: PD Dr. Daniela Kacso Lehrende: PD Dr. Daniela Kacso					
Verwendung des Moduls M.Sc. Angewandte Informatik M.Sc. IT-Sicherheit/ Informationstechnik					
Vorkenntnisse Die Inhalte der Veranstaltung "Datenstrukturen" bzw. "Informatik 2".					
Lernziele (learning outcomes) Nach dem erfolgreichen Abschluss des Moduls: Die Studierenden <ul style="list-style-type: none"> • kennen, wählen aus und nutzen grundlegende Datenstrukturen und Graphenalgorithmen • sind in der Lage Analysetechniken (Korrektheitsbeweise und Laufzeitanalyse) zu erläutern und zu beurteilen • können auch bei praktischen Problemen entscheiden, welche der vermittelten Methoden/Algorithmen/Datenstrukturen anwendbar sind und diese nach Effizienz (insb. Laufzeit der Algorithmen) bewerten • können konkrete Anwendungsprobleme modellieren und bei Bedarf diese Algorithmen weiter entwickeln 					
Inhalt Die Lehrveranstaltung kann sowohl in das Gebiet der praktischen als auch in das Gebiet der theoretischen Informatik eingeordnet werden. Die zentralen Themen sind die Folgenden: <ul style="list-style-type: none"> • Berechnung kürzester Pfade in Digraphen • Berechnung eines maximalen Flusses in einem Transportnetzwerk • Berechnung einer optimalen Lösung bei einem Zuordnungsproblem (auch Matching-Problem genannt) Darüber hinaus beschäftigen wir uns mit Anwendungen dieser grundlegenden Probleme.					
Lehrformen Vortrag der Lehrenden in der Vorlesung, Gruppenarbeit in den Übungen					
Prüfungsformen Schriftliche Modulabschlussprüfung (120 Minuten)					
Voraussetzungen für die Vergabe von Credits Bestandene schriftliche Modulabschlussprüfung					
Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS) 9/105: M.Sc. Angewandte Informatik					

9/91: M.Sc. IT-Sicherheit/ Informationstechnik [PO 22]

9/84: M.Sc. IT-Sicherheit/ Informationstechnik [PO 22]

Titel des Moduls: Energy-Aware Computing Systems
Energy-Aware Computing Systems

Modul-Nr./Code	Credits 6 CP	Workload 180 h	Semester siehe Prüfungsordnung / see examination regulations	Turnus Wintersemester	Dauer 1 Semester
Lehrveranstaltungen Energy-Aware Computing Systems (212030)			Kontaktzeit 60 h	Selbststudium 120 h	Gruppengröße 20 Studierende
Unterrichtssprache Englisch			Teilnahmevoraussetzungen keine		

Modulbeauftragte/r und hauptamtlich Lehrende

Modulbeauftragte/r: Prof. Dr.-Ing. Timo Hönig
 Lehrende: Prof. Dr.-Ing. Timo Hönig

Verwendung des Moduls

M.Sc. Computer Science
 M.Sc. IT-Sicherheit/ Informationstechnik
 M.Sc. Angewandte Informatik

Vorkenntnisse

Lernziele (learning outcomes)

Studierende, die die Vorlesung und die Übungen erfolgreich besucht haben, haben die Lernziele verfolgt und die unten aufgeführten Kompetenzen erworben. Die Studierenden können

- die Bedeutung von elektrischer Energie als Betriebsmittel für Rechensysteme verstehen
- Trade-off-Entscheidungen im Hinblick auf ein effizientes Systemdesign (d.h. Energiebedarf vs. Leistung), insbesondere von Betriebssystemen, treffen
- modellieren den Energiebedarf für einzelne synchrone und asynchrone Operationen
- Strategien zur Reduzierung des Energiebedarfs für Software-Aktivitäten auf der Grundlage spezifischer Hardware-Eigenschaften (z. B. Ruhezustände) anwenden
- Software auf kritische Abschnitte, die einen hohen Energiebedarf verursachen, analysieren.

Inhalt

Elektrische Energie ist die wichtigste Betriebsressource für Computersysteme. Obwohl der Energiebedarf von Computern an sich eine unsichtbare Systemeigenschaft ist, sind die Auswirkungen des Energiebedarfs allgegenwärtig und in verschiedenen Erscheinungsformen offensichtlich. Als praktische Beispiele dienen plötzliche Systemausfälle (d.h. Systemzusammenbrüche) und wiederkehrende Standard-Systemoperationen (d.h. Energiemanagement). Die Vorlesung befasst sich mit dem Entwurf energiebewusster Computersysteme und konzentriert sich auf die folgenden Themen:

- Leistungs- und Energiemanagement
- Energiebuchhaltung
- Analyse des Energiebedarfs
- energiebewusste Betriebssystem-Architektur
- Hardware-Energiemanagement (z.B. DVFS, Drosselung, Ruhezustände)
- Wärmemanagement
- Speicher- und Dateisysteme
- Speicherverwaltung
- Netzwerk, drahtlose Kommunikation und Protokolle
- Energiebewusste Server/Cluster
- Compiler-Optimierungen und Code-Umwandlung
- Anzeigetechnik
- Stromnetz

Die Vorlesung ist mit den Übungen durch Forschungsarbeiten verbunden. Die Studierenden lesen die Papiere zur Vorbereitung auf die Vorlesung. Von dort aus bilden die Forschungspapiere die Grundlage für die Diskussion und den Ausgangspunkt für die Aufgabenstellungen der Übungen. Im Rahmen der Übungen wenden die Studierenden Konzepte und Strategien aus den Forschungsarbeiten auf Systeme an und bewerten die Auswirkungen auf die Energieeffizienz des Systems.

Lehrformen

Die Vorlesung wird in Form eines Seminars abgehalten. Forschungsarbeiten zum energiebewussten Rechnen und Systemdesign werden von den Studierenden vorbereitet und in den Sitzungen diskutiert und analysiert. Zusätzlich vermittelt die Vorlesung theoretisches Wissen über grundlegende Konzepte zu den einzelnen Themen.

Im Rahmen der Übungen wenden die Studierenden ihr erworbenes Wissen an, indem sie Systemsoftware und Systemkonfigurationen zur Verbesserung der Energieeffizienz anpassen. Die Ergebnisse analysieren sie durch Leistungs- und Energiebedarfsauswertungen.

Prüfungsformen

Mündliche Modulabschlussprüfung (30 Minuten)

Voraussetzungen für die Vergabe von Credits

Bestandene Modulabschlussprüfung und erfolgreiche Teilnahme an den Übungen.

Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)

6/97: M.Sc. Informatik

6/105: M.Sc. Angewandte Informatik

6/91: M.Sc. IT-Sicherheit/ Informationstechnik [PO 22]

6/84: M.Sc. IT-Sicherheit/ Informationstechnik [PO 20]

Titel des Moduls: Fachwissenschaftliche Vertiefung (Angewandte Informatik)					
Modul-Nr./Code	Credits 6 CP	Workload 180 h	Semester 1-3	Turnus Jedes Semester	Dauer 2 Semester
Lehrveranstaltungen Seminar zu Approximationsalgorithmen Seminar Computational Neuroscience Seminar From Biological to Artificial Neural Networks Seminar Satisfiability Quantum Algorithms Seminar über Grenzen in der theoretischen Informatik Fortgeschrittene Themen des Model Checking Seminar Implementation Security Seminar Ingenieurinformatik Seminar Bioinformatik Seminar Ressourceneffiziente Systemsoftware Advanced Topics in Deep Learning Seminar Security Engineering Algorithms for Decision Making Seminar Geometrische Algorithmen Seminar Security and Privacy for Mobile Systems Seminar on Current Topics for Systems Security and Privacy Seminar Distributed and Networked Systems Seminar Quantum Cryptography Seminar Knowledge Graphs (ausgelaufen) Seminar Topics in Deep Learning for Sequence Processing (ausgelaufen) Seminar zur symmetrischen Kryptographie (ausgelaufen)			Kontaktzeit 60 h	Selbststudium 120 h	Gruppengröße 20 Studierende
Unterrichtssprache Deutsch oder Englisch			Teilnahmevoraussetzungen		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Studiendekan Angewandte Informatik Lehrende: siehe jeweiliges Seminar					
Verwendung des Moduls M.Sc. Angewandte Informatik					
Vorkenntnisse Abhängig von der Seminarwahl.					
Lernziele (learning outcomes) Nach dem erfolgreichen Abschluss des Moduls <ul style="list-style-type: none"> • verfügen Studierende über aktuelle fachwissenschaftliche Kenntnisse aus 2 verschiedenen Vertiefungsrichtungen 					

- können die Teilnehmer Lehrinhalte und Forschungsergebnisse eigenständig in einem didaktisch wohl aufbereiteten Vortrag vermitteln
- können die Teilnehmer konstruktives Feedback formulieren und entgegennehmen
- sind die Studierenden in der Lage, die Ergebnisse der eigenen Arbeit schriftlich zu dokumentieren

Inhalt

Es werden Vertiefungsseminare zu mehreren relevanten Themen angeboten, wie beispielsweise zu Deep Learning, zu Bioinformatik oder zu Ingenieurinformatik. Von den angebotenen Themen wählen die Studierenden abhängig von den eigenen Interessen und den individuellen Vertiefungswünschen in der Regel 2 verschiedene Seminarethemen aus. Diese sollen die Studierenden selbstständig bearbeiten. Dazu gehören die Literaturrecherche, die Einarbeitung in das Thema und schließlich die Präsentation sowie eine schriftliche Ausarbeitung.

Nähere Informationen zu den jeweiligen Seminaren sind dem Vorlesungsverzeichnis zu entnehmen.

Anmerkung: Die meisten Seminare werden mit 3 CP kreditiert. Hier müssen 2 verschiedenen Seminare absolviert werden (z.B. Ingenieurinformatik und Knowledge Graphs). Sollte ein Seminar mit 6 CP angeboten werden, so ist dieses ausreichend, um das Modul abzuschließen.

Lehrformen

Seminar

Prüfungsformen

Semesterbegleitend; 1-2 Seminarvorträge zu unterschiedlichen Themenbereichen und evtl. jeweils eine schriftliche Ausarbeitung.

Voraussetzungen für die Vergabe von Credits

Die meisten Seminare werden mit 3 CP kreditiert. In diesem Fall errechnet sich die Modulnote als arithmetisches Mittel aus beiden Seminarnoten. Bei Seminaren mit 6 CP ist die jeweilige Seminarnote auch die Modulnote. Die Gesamtnote muss mindestens "ausreichend" sein.

Um die Lernziele zu erreichen, besteht im Seminaren Anwesenheitspflicht an mindestens 9 von 10 Terminen. Mehrfaches Fehlen muss durch ein ärztliches Attest entschuldigt werden. Die Anwesenheit beim ersten Termin ist obligatorisch, da zu diesem Termin die Themen verteilt werden. Ein Seminar gilt als nicht bestanden, wenn an mehr als einem Termin unentschuldigt gefehlt wurde.

Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)

6/105: M.Sc. Angewandte Informatik

Titel des Moduls: Fundamentals of GPU Programming Fundamentals of GPU Programming					
Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester siehe Prüfungsordnung	Turnus Wintersemester	Dauer 1 Semester
Lehrveranstaltungen Fundamentals of GPU Programming (141374)			Kontaktzeit 45 h	Selbststudium 105 h	Gruppengröße Studierende
Unterrichtssprache Englisch			Teilnahmevoraussetzungen		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Dr. Denis Eremin Lehrende: Dr. Denis Eremin					
Verwendung des Moduls M.Sc. Angewandte Informatik M.Sc. IT-Sicherheit/ Informationstechnik					
Vorkenntnisse C (Pro­gram­mier­spra­che)					
Lernziele (learning outcomes) Die Studierenden erlernen das Programmieren auf Grafikprozessoren (GPUs)					
Inhalt Zu einem bestimmten Zeitpunkt um 2003 stieg die Rechenleistung nicht auf Kosten der Taktfrequenz des Prozessors, sondern durch Erhöhung der Anzahl der auf dem Prozessorchip zugewiesenen Rechenkern. Grafikprozessoren (GPUs) sind die Meister dieser Computer-Hardware-Entwicklung und bieten bis zu Zehntausende einzelner Kerneinheiten. Gleichzeitig wird das GPU-Speichersystem nicht so sehr durch die Kompatibilitätsanforderungen mit älteren Generationen eingeschränkt wie CPU-Speichersysteme. Deswegen zeigen GPUs im Vergleich zu ihren älteren "Bruder" -Zentraleinheiten (CPUs) eine deutlich bessere Rohleistung der Recheneinheiten und des Speichersystems. Ursprünglich für Videobearbeitungsaufgaben entwickelt, wird die enorme Rechenleistung moderner GPUs üblicherweise zur Unterstützung von CPUs oder zur Lösung einer Vielzahl von Rechenproblemen mit (massiv) parallelisierbaren Teilen verwendet, wodurch Teraflops-hohe Rechenleistung kann schon auf Laptop- / Desktop-Computers erzielt werden. Der vorliegende Kurs zeigt, wie CUDA C (Erweiterung der C-Sprache für die GPU-Programmierung) und das entsprechende (sehr flexible!) CUDA-Laufzeit-API-Framework verwendet werden kann, um die Ausführung einiger typischer Programmiermuster um einen Faktor von 10 oder mehr zu beschleunigen das der CPU. Ausgehend vom CUDA-Programmiermodell geht man zum CUDA-Ausführungsmodell über und betrachtet grundlegende konzeptionelle, Software- und Hardwareprobleme, die zum Verständnis der Funktionsweise von GPUs beitragen. Fallstudien zu mehreren Problemen mit massiv parallelen Algorithmen, die in GPUs implementiert sind, werden ebenfalls weiter ausgeführt. Das theoretische Wissen, das in den Vorlesungen vermittelt wird, wird durch eine Vielzahl von praktischen Beispielen untermauert, an denen die SchülerInnen zu Hause arbeiten können.					
Lehrformen Vorlesung mit Übung					
Prüfungsformen Semesterbegleitend: Projektarbeit und schriftliche Prüfung					
Voraussetzungen für die Vergabe von Credits Bestehen der Projektarbeit und der schriftlichen Prüfung					
Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS) 5/105: M.Sc. Angewandte Informatik					

5/91: M.Sc. IT-Sicherheit/ Informationstechnik [PO 22]

5/84: M.Sc. IT-Sicherheit/ Informationstechnik [PO 20]

Titel des Moduls: Geometrische Algorithmen
Geometric Algorithms

Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester siehe Prüfungsordnung / see examination regulations	Turnus Sommersemester	Dauer 1 Semester
Lehrveranstaltungen Geometrische Algorithmen (211056)			Kontaktzeit 60 h	Selbststudium 90 h	Gruppengröße 20 Studierende
Unterrichtssprache Deutsch			Teilnahmevoraussetzungen		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Maike Buchin Lehrende: Prof. Dr. Maike Buchin					
Verwendung des Moduls M.Sc. Computer Science M.Sc. Angewandte Informatik					
Vorkenntnisse Zwingend erforderlich: Grundlegende Kenntnisse über Algorithmen und Datenstrukturen. Empfohlen: Grundlagen der Stochastik.					
Lernziele (learning outcomes) Nach dem erfolgreichen Abschluss des Moduls <ul style="list-style-type: none"> • kennen Studierende grundlegende geometrische Algorithmen und Datenstrukturen • können Studierende Algorithmen nach dem Sweep-Paradigma analysieren und entwerfen • können Studierende inkrementelle Algorithmen entwerfen und analysieren, insbesondere randomisiert inkrementelle Algorithmen • können Studierende geometrische Algorithmen nach dem Teile-und-Herrsche Prinzip analysieren und entwerfen • können Studierende für Bereichsanfragen geeignete Datenstrukturen aussuchen 					
Inhalt Die Algorithmische Geometrie beschäftigt sich mit dem Entwurf und der Analyse von Algorithmen und Datenstrukturen für geometrische Probleme. Dazu werden zunächst einige grundlegende Probleme betrachtet, wie das Berechnen der konvexen Hülle einer Punktmenge, der Schnittpunkte einer Menge von Strecken oder einer Triangulierung eines einfachen Polygons. Anschließend sehen wir Algorithmen zum Berechnen bekannter geometrische Strukturen, wie das Voronoi-Diagramm, die Delaunay-Triangulierung und Arrangements. Ebenfalls betrachten wir Datenstrukturen für effiziente Anfragen auf geometrischen Daten, wie Rangetrees, kd-Bäume und Quadrees. Dabei kommen vor allem drei Arten von Algorithmen zum Einsatz: inkrementell, teile-und-herrsche, und sweep. Manche von diesen treten als randomisierte Algorithmen auf.					
Lehrformen Vorlesung als kombinierter Folien- und Tafelvortrag und zugehörige Übungen.					
Prüfungsformen Mündliche Modulabschlussprüfung (30 Minuten)					
Voraussetzungen für die Vergabe von Credits Bestandene mündliche Modulabschlussprüfung.					
Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS) 5/105: M.Sc. Angewandte Informatik 5/97: M.Sc. Computer Science					

Titel des Moduls: Grundlagen der FEM					
Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester siehe Prüfungsordnung	Turnus Jedes Wintersemester	Dauer 1 Semester
Lehrveranstaltungen Grundlagen der FEM (129015)			Kontaktzeit 60 h	Selbststudium 90 h	Gruppengröße Studierende
Unterrichtssprache Deutsch			Teilnahmevoraussetzungen		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Daniel Balzani Lehrende: Prof. Dr. Daniel Balzani					
Verwendung des Moduls M.Sc. Angewandte Informatik B.Sc. Maschinenbau					
Vorkenntnisse <p>Mathematik im Bachelor-Studium</p>					
Lernziele (learning outcomes) Die Studierenden <ul style="list-style-type: none"> • erwerben Kenntnisse der Finiten-Elemente Methode und sind in der Lage geeignete Elementtypen sowie Diskretisierungen für unterschiedliche Problemstellungen zu identifizieren bzw. zu konstruieren • können Software kompetent für klassische Probleme des Ingenieurbaus zum Einsatz bringen • werden in die Lage versetzt, eigene Elementroutinen zu programmieren bzw. in einem Software Framework zu implementieren 					
Inhalt • Einführung in die Tensorrechnung und Kontinuumsmechanik kleiner Verzerrungen • Variationsprinzipien • Galerkin-Verfahren • Methodik der finiten Elemente • Dreidimensionale Stabwerkselemente und Assemblierungsoperator • Darstellung in Flächen- und Volumenkoordinaten • Isoparametrisches Konzept • Numerische Integration • Weitergehende Elementtypen wie Balken und Platten					
Lehrformen Vorlesung mit Übung					
Prüfungsformen • Klausur (120 Minuten) • Mündliche Prüfung (30 Minuten) Die Prüfungsform wird je nach Teilnehmerzahl am Anfang eines jeden Semesters festgelegt.					
Voraussetzungen für die Vergabe von Credits Bestandene Modulabschlussprüfung schriftlich oder mündlich.					
Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS) 5/105: M.Sc. Angewandte Informatik					

Titel des Moduls: High-Performance Computing on Clusters
High-Performance Computing on Clusters

Modul-Nr./Code	Credits 6 CP	Workload 180 h	Semester siehe Prüfungsordnung / see examination regulations	Turnus Wintersemester	Dauer 1 Semester
Lehrveranstaltungen High-Performance Computing on Clusters (127511)			Kontaktzeit 60 h	Selbststudium 120 h	Gruppengröße 50 Studierende
Unterrichtssprache Englisch			Teilnahmevoraussetzungen		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Andreas Vogel Lehrende: Prof. Dr. Andreas Vogel					
Verwendung des Moduls M.Sc. Computer Science M.Sc. Angewandte Informatik M.Sc. Bauingenieurwesen M.Sc. Computational Engineering					
Vorkenntnisse					
Lernziele (learning outcomes) Nach erfolgreicher Absolvierung des Moduls sind die Studierenden in der Lage <ul style="list-style-type: none"> • Programme für parallele Rechencluster zu entwerfen und zu erstellen • Distributed-Memory-Systeme und Programmiermuster kritisch zu bewerten • die mathematischen Eigenschaften iterativer Lösungsverfahren und deren Skalierbarkeit zu beurteilen. 					
Inhalt Die Vorlesung befasst sich mit der Parallelisierung auf Cluster-Rechnern. Es werden Konzepte der Distributed-Memory-Programmierung (MPI) vorgestellt und eine Best-Practice-Implementierung anhand von Anwendungen aus dem wissenschaftlichen Rechnen wie der Finite-Elemente-Methode und dem maschinellen Lernen präsentiert. Besonderes Augenmerk wird auf skalierbare Löser für Gleichungssysteme auf Systemen mit verteiltem Speicher gelegt, wobei der Schwerpunkt auf iterativen Verfahren wie einfachen Splitting-Methoden (Richardson, Jacobi, Gauß-Seidel, SOR), Krylov-Methoden (Gradientenabstieg, CG, BiCGStab) und insbesondere dem Mehrgitterverfahren liegt. Es werden die mathematischen Grundlagen für iterative Löser überprüft, geeignete objektorientierte Schnittstellenstrukturen entwickelt und eine Implementierung dieser Löser für moderne Parallelrechnerarchitekturen erarbeitet. Anhand von numerischen Experimenten und selbst entwickelten Softwareimplementierungen werden die theoretischen Ergebnisse diskutiert und illustriert.					
Lehrformen Beamer, Computerlabor, numerische Experimente					
Prüfungsformen Schriftliche Modulabschlussprüfung (120 Minuten)					
Voraussetzungen für die Vergabe von Credits Bestandene schriftliche Modulabschlussprüfung					
Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS) 6/97: M.Sc. Computer Science					

Titel des Moduls: High-Performance Computing on Multi- and Manycore Processors
High-Performance Computing on Multi- and Manycore Processors

Modul-Nr./Code	Credits 6 CP	Workload 180 h	Semester siehe Prüfungsordnung / see examination regulations	Turnus Sommersemester	Dauer 1 Semester
Lehrveranstaltungen High-Performance Computing on Multi- and Manycore Processors (126509)			Kontaktzeit 60 h	Selbststudium 120 h	Gruppengröße 40 Studierende
Unterrichtssprache Englisch			Teilnahmevoraussetzungen		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Andreas Vogel Lehrende: Prof. Dr. Andreas Vogel					
Verwendung des Moduls M.Sc. Computer Science M.Sc. Angewandte Informatik M.Sc. Bauingenieurwesen M.Sc. Computational Engineering					
Vorkenntnisse					
Lernziele (learning outcomes) Nach erfolgreichem Abschluss des Moduls sind die Studierenden <ul style="list-style-type: none"> • in der Lage, Programme für Multi- und Manycore-Prozessoren zu entwerfen und zu erstellen, • in der Lage Multi-Thread-Programme und Shared-Memory-Zugriffsmuster kritisch zu bewerten, • in der Lage, fortgeschrittene wissenschaftliche Themen selbstständig zu untersuchen und ihre Ergebnisse zu präsentieren. 					
Inhalt Die Vorlesung behandelt die Parallelisierung auf Multicore-Prozessoren. Thread-basierte Programmierkonzepte und -techniken, einschließlich pthreads, C++11 Threads, OpenMP und SYCL, werden vorgestellt und bewährte Praktiken anhand von Anwendungen aus dem wissenschaftlichen Rechnen aufgezeigt. Es wird ein Überblick über die relevanten Hardware-Aspekte einschließlich Multicore-Architekturen und Speicherhierarchien gegeben. Eine ausführliche Einführung in die Multi-Thread-Programmierung auf Multicore-Systemen mit besonderem Schwerpunkt auf Shared-Memory-Parallelisierung wird gegeben und Parallelisierungsmuster, Thread-Management und Speicherzugriffsstrategien werden diskutiert. In praktischen Sitzungen werden Programmierübungen verwendet, um die vorgestellten Inhalte zu diskutieren und zu illustrieren.					
Lehrformen Vorlesung mit Übung					
Prüfungsformen Schriftliche Modulabschlussprüfung (120 Minuten)					
Voraussetzungen für die Vergabe von Credits Bestandene schriftliche Modulabschlussprüfung.					

Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)

6/97: M.Sc. Computer Science

6/105: M.Sc. Angewandte Informatik

Titel des Moduls: Information Theory Information Theory					
Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester siehe Prüfungsordnung / see examination regulations	Turnus Sommersemester	Dauer 1 Semester
Lehrveranstaltungen Information Theory (211007)			Kontaktzeit 60 h	Selbststudium 90 h	Gruppengröße 30 Studierende
Unterrichtssprache Englisch			Teilnahmevoraussetzungen Keine		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Michael Walter Lehrende: Prof. Dr. Michael Walter					
Verwendung des Moduls B.Sc. Informatik (bis SS 23) B.Sc. IT-Sicherheit M.Sc. Informatik M.Sc. IT-Sicherheit/ Informationstechnik M.Sc. IT-Sicherheit/ Netze und Systeme (bis SS 23) M.Sc. Angewandte Informatik					
Vorkenntnisse Vertrautheit mit der diskreten Wahrscheinlichkeitsrechnung (wir werden Sie kurz an die wichtigsten Fakten erinnern). Einige Erfahrung mit präzisen mathematischen Aussagen und strengen Beweisen (da wir viele davon im Kurs sehen werden). Ein Teil der Hausaufgaben wird die Programmierung in Python erfordern.					
Lernziele (learning outcomes) Sie werden grundlegende Konzepte, Algorithmen und Ergebnisse der Informationstheorie kennenlernen. Nach erfolgreichem Abschluss dieses Kurses kennen Sie das mathematische Modell der Informationstheorie, wissen, wie man Algorithmen für eine Vielzahl von Informationsverarbeitungsaufgaben entwirft und analysiert, und wie man sie in Python implementiert. Sie haben sich selbstständig in ein Thema der Informationstheorie eingelesen und dieses vor Ihren Kommilitonen präsentiert. Sie werden auf einen weiterführenden Kurs oder ein Forschungs- oder Abschlussprojekt in diesem Bereich vorbereitet. Eine genaue Auflistung der Lernziele finden Sie auf der Homepage des Kurses.					
Inhalt Dieser Kurs gibt eine Einführung in die Informationstheorie - die mathematische Theorie der Information. Seit ihren Anfängen hat die Informationstheorie einen tiefgreifenden Einfluss auf die Gesellschaft gehabt. Sie bildet die Grundlage für wichtige technologische Entwicklungen, von zuverlässigen Speichern bis hin zu Mobilfunkstandards, und ihr vielseitiges mathematisches Instrumentarium findet Anwendung in der Informatik, dem maschinellen Lernen, der Physik, der Elektrotechnik, der Mathematik und vielen anderen Disziplinen. Ausgehend von der Wahrscheinlichkeitstheorie werden wir erörtern, wie man Informationsquellen und Kommunikationskanäle mathematisch modelliert, wie man Informationen optimal komprimiert und wie man fehlerkorrigierende Codes entwirft, die uns eine zuverlässige Kommunikation über verrauschte Kommunikationskanäle ermöglichen. Wir werden auch sehen, wie die in der Informationstheorie verwendeten Techniken allgemeiner angewendet werden können, um Vorhersagen aus verrauschten Daten zu treffen.					
Vorläufiger Lehrplan:					

- Begrüßung, Einführung in die Informationstheorie
- Auffrischung der Wahrscheinlichkeitstheorie
- Numerische Zufallsvariablen, Konvexität und Konkavität, Entropie
- Symbol-Codes: Verlustfreie Komprimierung, Huffman-Algorithmus
- Block-Codes: Shannons Quellencodierungstheorem, sein Beweis und Variationen
- Strom-Codes: Lempel-Ziv-Algorithmus
- Strom-Codes: Arithmetische Kodierung
- Gemeinsame Entropien & Kommunikation über verrauschte Kanäle
- Shannons Theorem der verrauschten Kodierung
- Beweis des Theorems der verrauschten Kodierung (Noisy Coding Theorem)
- Beweis der Umkehrung, Shannons Theorie und Praxis
- Reed-Solomon-Codes
- Nachrichtenübermittlung für Dekodierung und Inferenz, Ausblick
- Studentische Präsentationen

Weitere Informationen finden Sie auf der Kurs-Homepage https://qi.rub.de/it_ss23.

Lehrformen

Vorlesung mit Übung

Prüfungsformen

Schriftliche (180 Minuten) oder mündliche (30 Minuten) Modulabschlussprüfung, abhängig von der Teilnehmerzahl. Wird zum Kursbeginn bekanntgegeben.

Voraussetzungen für die Vergabe von Credits

Passed Exam

Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)

5/158: B.Sc. Informatik [PO 22]

5/170: B.Sc. Informatik [PO 20]

5/150: B.Sc. IT-Sicherheit [PO 22]

5/149: B.Sc. IT-Sicherheit [PO 20]

5/97: M.Sc. Informatik

5/91: M.Sc. IT-Sicherheit/ Informationstechnik [PO 22]

5/84: M.Sc. IT-Sicherheit/ Informationstechnik [PO 20]

5/99: M.Sc. IT-Sicherheit/ Netze und Systeme [PO 22]

5/105: M.Sc. Angewandte Informatik

Titel des Moduls: Inklusives IT-Design Inclusive IT design					
Modul-Nr./Code	Credits 6 CP	Workload 180 h	Semester siehe Prüfungsordnung	Turnus Sommersemester	Dauer 1 Semester
Lehrveranstaltungen			Kontaktzeit 60 h	Selbststudium 120 h	Gruppengröße 20 Studierende
Unterrichtssprache Deutsch			Teilnahmevoraussetzungen Keine		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Christian Meske Lehrende: Prof. Dr. Christian Meske Markus Jelonek, M.Sc.					
Verwendung des Moduls B.Sc. Angewandte Informatik M.Sc. Angewandte Informatik					
Vorkenntnisse					
Lernziele (learning outcomes) Die Veranstaltung richtet sich an Studierende mit Interesse an Themen der Inklusion und dem Nutzerzentrierten Design. Die Studierenden <ul style="list-style-type: none"> • ... kennen die Perspektiven von Endnutzer:innen mit geistiger und/oder körperlicher Behinderung hinsichtlich Interaktionsdesign, Usability und User Experience und können diese bei der Entwicklung eines Prototyps anwenden, begründen und beurteilen. • ... leisten einen Dienst an der Gesellschaft, indem sie reale Probleme bzgl. der Inklusion von Menschen mit Behinderungen im Rahmen der digitalen Transformation aufgreifen, analysieren, beurteilen und gemeinsam prototypische Lösungen konstruieren (Service Learning). • ... können partizipative und Design Science Methoden mit Aspekten der Inklusion und des Wohlbefindens in zukünftigen IT-Design-Prozessen zusammenführen und beurteilen. • ... können theoretisch-konzeptionelle Design-Science-Vorgehensmodelle mit einer praxisorientierten Anwendung im Themenfeld des Positive Computing anwenden. • ... kennen Anforderungen an inklusives IT-Design und können diese praktisch anwenden. • ... können Fachwissen, welches in anderen Fächern erworben wurde (UX, Software-Ergonomie und Usability Engineering, Positive Computing, etc.), problemorientiert anwenden und beurteilen sowie neues Wissen selbstständig aneignen. • ... können zielgerichtet und interdisziplinär zusammenarbeiten sowie Lösungen in einem festen Zeitraum eigenständig konstruieren, analysieren und präsentieren. • ... können die Vorteile und Herausforderungen bei der Entwicklung von Anwendungen mit und für die spezifische Zielgruppe beschreiben. • ... können Aspekte des agilen IT-Projektmanagements für Ihren Praxisteil anwenden. 					

- ... können Aspekte des Positive Computing in ihrer Prototypenwicklung anwenden.

Inhalt

- Inclusive und accessible IT-Design
- Universal Design vs. Inclusive and Accessible Design
- Barrierefreiheit
- UX und Usability in inclusive und accessible IT-Design
- Design Science Research
- Partizipative Methoden und Co-Creation

Lehrformen

- Vorlesung (Flipped Classroom, hybrid) mit praktischen Phasen in Kooperation mit Studierenden der Hochschule Ruhr-West (digital und analog)
- Praxisorientierte Blockveranstaltungen und Projektphasen mit Praxispartner vor Ort und digital

Prüfungsformen

Projektarbeit

Voraussetzungen für die Vergabe von Credits

- Regelmäßige Teilnahme an Pflichtterminen
- Bestandene Zwischen- und Abschlusspräsentation
- Bestandene Abgabe der Dokumentation und Präsentation

Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)

6/168: B.Sc. Angewandte Informatik [PO22]

6/170: B.Sc. Angewandte Informatik [PO20]

6/105: M.Sc. Angewandte Informatik

Titel des Moduls: Introduction to Computational Linguistics in Python Introduction to Computational Linguistics in Python					
Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester siehe Prüfungsordnung	Turnus Wintersemester	Dauer 1 Semester
Lehrveranstaltungen Introduction to Computational Linguistics in Python (050825)			Kontaktzeit 60 h	Selbststudium 90 h	Gruppengröße Studierende
Unterrichtssprache Englisch			Teilnahmevoraussetzungen		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Ralf Klabunde Lehrende: Prof. Dr. Ralf Klabunde					
Verwendung des Moduls M.Sc. Angewandte Informatik					
Vorkenntnisse Idealerweise Vorkenntnisse in Linguistik (kann auch durch Literaturlektüre angeeignet werden) , Programmierung und Mathematik					
Lernziele (learning outcomes) <ul style="list-style-type: none"> • Die Studierenden kennen Grundlagen der Programmierung mit Python mit einem besonderen Fokus auf kommentiertem und strukturiertem Code. • Sie können Fragen zu Programmierproblemen angemessen formulieren und beantworten. • Sie können Anforderungen an Programme formulieren und umsetzen. • Sie haben gelernt, bestehende Komponenten einzubinden und angemessen auf Quellen zu verweisen. • Sie verstehen den Unterschied zwischen Computational Linguistics, Informatik und NLP-Verfahren. • Sie kennen grundlegende Fragestellungen aus der Computerlinguistik und kennen die linguistische Perspektive auf NLP-Verfahren. 					
Inhalt Das Modul bietet eine Einführung in die Programmiersprache R aus der Perspektive der Computerlinguistik (und Linguistic Data Science). Es beleuchtet Grundlagenverfahren und spezielle Fallstricke kennen, auf die sie bei der Verarbeitung von Sprachdaten für linguistische Fragestellungen achten müssen. Dabei werden einerseits externe Module genutzt und eingesetzt, andererseits aber vor allem Funktionen auch selbst entwickelt, um ihre Funktion nachzuvollziehen und zu verstehen, welche Überlegungen sie auch bei der reinen Anwendung von „fertigen“ Programmen berücksichtigen müssen. In den „Vorlesungen“ werden die vorher hochgeladenen Folien besprochen, nicht vorgetragen. Für die Klausur werden nur Inhalte der Folien vorausgesetzt. Die begleitenden Übungen zur Vorlesung sind nicht verpflichtend, aber sind zur Selbstkontrolle offen für alle Studierenden.					
Lehrformen Die Veranstaltung setzt sich aus einer Präsenzkomponente (Fragen und Antworten zu den Folien, Diskussionen, praktische Übungen und Überlegungen zum aktuellen Thema) und Übungen zum Kurs zusammen. Die Studierenden müssen sich eigenständig auf die Sitzungen vorbereiten und entscheiden, ob die Teilnahme ihr Verständnis fördert.					
Prüfungsformen Schriftliche Modulabschlussprüfung					
Voraussetzungen für die Vergabe von Credits					
Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS) 5/105: M.Sc. Angewandte Informatik					

Titel des Moduls: Introduction to Linguistic Data Science with R Introduction to Linguistic Data Science with R					
Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester siehe Prüfungsordnung	Turnus Wintersemester	Dauer 1 Semester
Lehrveranstaltungen Introduction to Linguistic Data Science with R (050826)			Kontaktzeit 60 h	Selbststudium 90 h	Gruppengröße 20 Studierende
Unterrichtssprache Englisch			Teilnahmevoraussetzungen		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Tibor Kiss Lehrende: Prof. Dr. Tibor Kiss					
Verwendung des Moduls M.Sc. Angewandte Informatik					
Vorkenntnisse					
Lernziele (learning outcomes) <ul style="list-style-type: none"> • Die Studierenden erwerben grundlegende Modelle und Methoden der Linguistic Data Science und können sie kritisch auf Fragestellungen und Probleme anwenden. • Sie werden mit der Fachsprache vertraut und können sich gut schriftlich und mündlich zu den behandelten Themen ausdrücken. • Sie verstehen die Position von Linguistic Data Science in Abgrenzung zu verwandten Fächern wie reiner Linguistik oder allgemeiner Data Science und haben einen ersten Überblick über mögliche Vertiefungen in Abgrenzung zu Computerlinguistik und theoretischer Linguistik. • Sie verstehen einige grundlegende Evaluationsmethoden und können sie auf kleine Beispielfälle anwenden. • Sie vollziehen den Einfluss von Entscheidungen bei der Wahl von Datensätzen, Datentypen oder der Struktur von Experimenten auf Daten in einfachen Beispielen nach. • Sie können sich über kleine fachspezifische Problemstellungen mit andere austauschen und diese gemeinsam lösen. • Sie erwerben den Umgang mit der Programmiersprache R und können Programme schreiben, die einzelne Funktionen erfüllen, sowie Programme, die mehrere Module/Funktionen kombinieren. 					
Inhalt Das Modul bietet eine Einführung in die Verfahren, Methoden und Modelle sowie Theorien aus dem Bereich Linguistic Data Science. Es werden zunächst diejenigen Konzepte der Programmiersprache R vorgestellt, die eine Repräsentation und Manipulation linguistischer Daten (Rohdaten, Annotationen, experimentell erhobene Daten, Datenübersichten) ermöglichen. Anschließend erfolgt die Einführung in grundlegende lineare Modelle, sowie deren Grundannahmen und Verfahren zur Evaluation. Darüber hinaus werden Entwicklungsverfahren und Projektkonzeption vorgestellt. In den Übungen werden kleine Aufgaben von den Studierenden in Gruppen oder allein bearbeitet.					
Lehrformen Die Veranstaltung setzt sich aus einer Präsenzkomponekte (Vortrag der Lehrenden im Plenum mit anschließender Diskussion, gemeinsames Lösen von Aufgaben) und Übungen zum Kurs zusammen. Dabei können insbesondere die Übungen als eLearning-Einheiten angeboten werden.					
Prüfungsformen Schriftliche Modulabschlussprüfung					

Voraussetzungen für die Vergabe von Credits

Bestandene schriftliche Modulabschlussprüfung

Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)

5/105: M.Sc. Angewandte Informatik

Titel des Moduls: Klbox: Lösung von Nachhaltigkeits-Challenges durch den Einsatz künstlicher Intelligenz

Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester siehe Prüfungsordnung	Turnus Jedes Semester (nicht SS 23!)	Dauer 1 Semester
Lehrveranstaltungen Klbox: Lösung von Nachhaltigkeits-Challenges durch den Einsatz künstlicher Intelligenz			Kontaktzeit 60 h	Selbststudium 90 h	Gruppengröße Studierende
Unterrichtssprache Deutsch			Teilnahmevoraussetzungen		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Christian Meske Lehrende: Prof. Dr. Jens Pöppelbuß Prof. Dr. Christian Meske					
Verwendung des Moduls M.Sc. Angewandte Informatik					
Vorkenntnisse Es werden keinerlei Vorkenntnisse benötigt, aber Interesse an Methoden der künstlichen Intelligenz. Erste Erfahrungen mit Python sind hilfreich und gern gesehen, jedoch keine Voraussetzung.					
Lernziele (learning outcomes) Die Studierenden erwerben durch das Absolvieren der Lehrveranstaltung folgende Kompetenzen: <ul style="list-style-type: none"> • Die „Kickbox“-Innovationsmethode erläutern und anwenden, • Beispielhafte Nachhaltigkeits-Challenges identifizieren und analysieren, • Potenziale von KI zur Lösung von Nachhaltigkeits-Challenges identifizieren, • Kreativitätstechniken, Ideation- und Prototyping-Methoden (z. B. aus dem Design Thinking sowie Lean-Start-Up) für den eigenen Innovationsprozess auswählen und anwenden, • Eigene kreative Ideen präsentieren und testen und mittels Feedbacks überarbeiten, • Ziele definieren, Projekte managen und Meilensteine eigenständig erreichen, Teamarbeit unter Einsatz von begrenzten Ressourcen koordinieren, • KI-Lösungen prototypisch implementieren, • Unternehmerisches Handeln, • Nachhaltigkeit anhand etablierter Dimensionen (z. B. Triple-Bottom-Line und Corporate Social Responsibility) zur Bewertung eigener Ideen nutzen. 					
Inhalt <ul style="list-style-type: none"> • Basierend auf dem Kickbox-Innovationsprozess sollen Studierende für gesellschafts- und praxisrelevante Herausforderungen eigenständig Problemlösungen durch den Einsatz künstlicher Intelligenz (KI) entwickeln. • Als Ausgangspunkt des Lehrformats werden Nachhaltigkeits-Challenges formuliert, bei denen KI zu einem umweltfreundlicheren und wettbewerbsfähigeren Europa beitragen kann. • Die für das Lehrformat spezifisch zu gestaltende Kickbox (Klbox) unterstützt die Studierenden im Innovationsprozess. 					
Lehrformen <ul style="list-style-type: none"> • Vereinzelt Vorlesungen/ Inputs von Dozierenden • Projektarbeit und Projektveranstaltungen als Schwerpunkt • Feedback und Möglichkeiten zum Austausch mit Dozierenden 					
Prüfungsformen Es sind zwei Prüfungsleistungen zu erbringen. Abschlusspräsentation (Gruppenleistung, Dauer: 15 Minuten plus anschließende Diskussion von 10 Minuten; 100 % der Endnote) Abschlussbericht (Gruppenleistung; Bestehen notwendig zum Abschluss des Moduls und zum Erwerb der Leistungspunkte)					

Prüfungstermine: Die Prüfungstermine werden zu Beginn des Semesters bekanntgegeben.

Voraussetzungen für die Vergabe von Credits

- Prüfungsleistung bestanden und
- Um das Modul erfolgreich abzuschließen ist es außerdem notwendig, an mindestens 60% der Treffen im Plenum teilzunehmen

Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)

5/105: M.Sc. Angewandte Informatik

Titel des Moduls: Kryptographie Cryptography					
Modul-Nr./Code	Credits 8 CP	Workload 240 h	Semester siehe Prüfungsordnung / see examination regulations	Turnus Wintersemester	Dauer 1 Semester
Lehrveranstaltungen Kryptographie (212017)			Kontaktzeit 90 h	Selbststudium 150 h	Gruppengröße 100 Studierende
Unterrichtssprache Deutsch			Teilnahmevoraussetzungen		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Jun.-Prof. Nils Fleischhacker Lehrende: Jun.-Prof. Nils Fleischhacker					
Verwendung des Moduls B.Sc. IT-Sicherheit/ Informationstechnik M.Sc. IT-Sicherheit/ Netze und Systeme M.Sc. Computer Science M.Sc. Angewandte Informatik					
Vorkenntnisse Inhalte der Vorlesungen Einführung in die Kryptographie 1 und 2					
Lernziele (learning outcomes) Die Studierenden haben ein Verständnis der wesentlichen mathematischen Methoden und Verfahren, auf denen moderne kryptographische Verfahren beruhen. Die Tiefe der Behandlung der Verfahren geht deutlich über das in den vorhergehenden Veranstaltungen vermittelte Maß hinaus. Die Teilnehmer sind zur Analyse und dem Design aktueller und zukünftiger kryptographischer Methoden befähigt. Zudem weisen sie ein Bewusstsein für Methodik und Mächtigkeit verschiedenster Angriffsszenarien auf.					
Inhalt Es wird eine Einführung in moderne Methoden der symmetrischen und asymmetrischen Kryptographie geboten. Dazu wird ein Angreifermodell definiert und die Sicherheit der vorgestellten Verschlüsselungs-, Hash- und Signaturverfahren unter wohldefinierten Komplexitätsmaßnahmen in diesem Angreifermodell nachgewiesen. Themenübersicht: <ul style="list-style-type: none">• Sichere Verschlüsselung gegenüber KPA-, CPA- und CCA-Angreifern• Pseudozufallsfunktionen und -permutationen• Message Authentication Codes• Kollisionsresistente Hashfunktionen• Blockchiffren• Konstruktion von Zufallszahlengeneratoren• Diffie-Hellman Schlüsselaustausch• Trapdoor Einwegpermutationen• Public Key Verschlüsselung: RSA, ElGamal, Goldwasser-Micali, Rabin, Paillier• Einwegsignaturen• Signaturen aus kollisionsresistenten Hashfunktionen• Random-Oracle Modell					
Lehrformen Vorlesung und Übungen					

Prüfungsformen

Schriftliche Modulabschlussprüfung (120 Minuten)

Voraussetzungen für die Vergabe von Credits

Bestandene schriftliche Modulabschlussprüfung.

Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)

8/150: B.Sc. IT-Sicherheit/ Informationstechnik [PO 22]

8/149: B.Sc. IT-Sicherheit/ Informationstechnik [PO 20]

8/99: M.Sc. IT-Sicherheit/ Netze und Systeme [PO 22]

8/96: M.Sc. IT-Sicherheit/ Netze und Systeme [PO 20]

8/97: M.Sc. Computer Science

8/105: M.Sc. Angewandte Informatik

Titel des Moduls: Linguistic Data Science Linguistic Data Science					
Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester siehe Prüfungsordnung	Turnus Wintersemester	Dauer 1 Semester
Lehrveranstaltungen Linguistic Data Science (050826)			Kontaktzeit 60 h	Selbststudium 90 h	Gruppengröße Studierende
Unterrichtssprache Englisch			Teilnahmevoraussetzungen Erfolgreicher Abschluss des Moduls "Introduction to Linguistic Data Science with R"		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Tibor Kiss Lehrende: Prof. Dr. Tibor Kiss					
Verwendung des Moduls M.Sc. Angewandte Informatik					
Vorkenntnisse					
Lernziele (learning outcomes) <ul style="list-style-type: none"> • Die Studierenden erwerben grundlegende Modelle und Methoden der Linguistic Data Science und können sie kritisch auf Fragestellungen und Probleme anwenden. • Sie werden mit der Fachsprache vertraut und können sich gut schriftlich und mündlich zu den behandelten Themen ausdrücken. • Sie verstehen die Position von Linguistic Data Science in Abgrenzung zu verwandten Fächern wie reiner Linguistik oder allgemeiner Data Science und haben einen ersten Überblick über mögliche Vertiefungen in Abgrenzung zu Computerlinguistik und theoretischer Linguistik. • Sie verstehen einige grundlegende Evaluationsmethoden und können sie auf kleine Beispielfälle anwenden. • Sie vollziehen den Einfluss von Entscheidungen bei der Wahl von Datensätzen, Datentypen oder der Struktur von Experimenten auf Daten in einfachen Beispielen nach. • Sie können sich über kleine fachspezifische Problemstellungen mit andere austauschen und diese gemeinsam lösen. • Sie erwerben den Umgang mit der Programmiersprache R und können Programme schreiben, die einzelne Funktionen erfüllen, sowie Programme, die mehrere Module/Funktionen kombinieren. 					
Inhalt Das Modul bietet eine Vertiefung in die Verfahren, Methoden und Modelle sowie Theorien aus dem Bereich Linguistic Data Science. Auf der Basis der erfolgten Einführung in grundlegende lineare Modelle, sowie deren Grundannahmen und Verfahren zur Evaluation werden schrittweise Generalized Linear Models und Generalized Linear Mixed Models vorgestellt und durch Entwicklungskonzepte und Fragestellungen der Projektkonzeption unterfüttert. In den Übungen werden kleine Aufgaben von den Studierenden in Gruppen oder allein bearbeitet.					
Lehrformen Die Veranstaltung setzt sich aus einer Präsenzkomponente (Vortrag der Lehrenden im Plenum mit anschließender Diskussion, gemeinsames Lösen von Aufgaben) und Übungen zum Kurs zusammen. Dabei können insbesondere die Übungen als eLearning-Einheiten angeboten werden.					
Prüfungsformen Schriftliche Modulabschlussprüfung					
Voraussetzungen für die Vergabe von Credits Bestandene schriftliche Modulabschlussprüfung					
Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS) 5/105: M.Sc. Angewandte Informatik					

Titel des Moduls: Machine Learning: Evolutionary Algorithms
Machine Learning: Evolutionary Algorithms

Modul-Nr./Code	Credits 6 CP	Workload 180 h	Semester siehe Prüfungsordnung / see examination regulations	Turnus Wintersemester	Dauer 1 Semester
Lehrveranstaltungen Machine Learning: Evolutionary Algorithms (212008)			Kontaktzeit 60 h	Selbststudium 120 h	Gruppengröße Studierende

Unterrichtssprache Englisch	Teilnahmevoraussetzungen
---------------------------------------	---------------------------------

Modulbeauftragte/r und hauptamtlich Lehrende
 Modulbeauftragte/r: Prof. Dr. Tobias Glasmachers
 Lehrende: Prof. Dr. Tobias Glasmachers

Verwendung des Moduls
 M.Sc. Angewandte Informatik
 M.Sc. Computer Science

Vorkenntnisse
 Der Kurs richtet sich an Masterstudierende technischer Fächer der (angewandten) Informatik, Mathematik, Ingenieur- und Naturwissenschaften. Er setzt solide Kenntnisse der mathematischen Grundlagen, insbesondere der linearen Algebra (z.B. Eigenwertzerlegung) und der Wahrscheinlichkeitsrechnung, sowie etwas Python-Programmierung voraus.

Lernziele (learning outcomes)
 Internationalisierung: Die Veranstaltung wird auf Englisch durchgeföhrt. Digitalisierung: Inhalte werden durch Vidoes und Lesematerial vermittelt. Übungsaufgaben mit Programmieranteilen werden in Form von Jupyter-Notebooks bereitgestellt. Nach dem erfolgreichen Abschluss des Moduls

- kennen die Teilnehmer die wichtigsten Klassen direkter Optimierungsverfahren und ihre algorithmischen Komponenten,
- haben die Teilnehmer ein tiefes Verständnis evolutionärer Algorithmen, insbesondere für kontinuierliche Probleme,
- kennen die Teilnehmer eine Reihe spezifischer Problemschwierigkeiten und die zugehörigen algorithmischen Komponenten, welche diese Adressieren,
- können die Teilnehmer elementare Laufzeitanalysen durchführen und verstehen die wichtigsten Konvergenzklassen ∙ können die Teilnehmer Optimierungsverfahren selbst implementieren und zur Lösung neuer Probleme anwenden.

Inhalt
 Dieser Kurs bietet eine gründliche Einführung in die praktische Optimierung mit Algorithmen aus dem Bereich der evolutionären Berechnung.

Evolutionäre Algorithmen sind randomisierte Such- und Optimierungsheuristiken, die von Prinzipien der biologischen Evolution inspiriert sind. Das Gebiet zielt darauf ab, das Prinzip des "Überlebens des Stärkeren" für die Lösung technischer Probleme zu nutzen. Die daraus resultierenden Optimierungsalgorithmen sind konzeptionell einfach, breit anwendbar und leicht zu implementieren. Die evolutionäre Suche wird in Wissenschaft und Technik zur näherungsweise Lösung schwieriger "Black Box"-Probleme eingesetzt.

Der Kurs beginnt mit einem allgemeinen Überblick über das weite Feld der Optimierung, einschließlich der Modellierung von Problemen. Anschließend werden verschiedene Arten der evolutionären Berechnung vorgestellt, insbesondere genetische Algorithmen, genetische Programmierung und Evolutionsstrategien einschließlich der Anpassung der Kovarianzmatrix. Die Vorlesung entwickelt das grundlegende Modell der evolutionären Optimierung. Verschiedene Aspekte der evolutionären Suche in diskreten und kontinuierlichen Suchräumen werden im Detail diskutiert, was zu einer systematischen Taxonomie von weitgehend modularen Bausteinen führt.

Der zweite Teil des Kurses behandelt eine Reihe typischer Herausforderungen, wie den Umgang mit Nebenbedingungen, hochgradig multimodalen Problemen, verrauschten Zielfunktionen und mehreren Zielen. Den Abschluss bildet die Neuroevolution, eine Methode zum Training von neuronalen Netzreglern, als moderner Anwendungsbereich.

Lehrformen

Vorlesung mit Übung im flipped classroom Format.

Prüfungsformen

Schriftliche Modulabschlussprüfung (90 Minuten).

Voraussetzungen für die Vergabe von Credits

Bestandene schriftliche Modulabschlussprüfung.

Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)

6/105: M.Sc. Angewandte Informatik

6/97: M.Sc. Computer Science

Titel des Moduls: Machine Learning: Supervised Methods
Machine Learning: Supervised Methods

Modul-Nr./Code	Credits 6 CP	Workload 180 h	Semester siehe Prüfungsordnung / see examination regulations	Turnus Sommersemester	Dauer 1 Semester
Lehrveranstaltungen Machine Learning: Supervised Methods (211024)			Kontaktzeit 60 h	Selbststudium 120 h	Gruppengröße 80 Studierende
Unterrichtssprache Englisch			Teilnahmevoraussetzungen		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Tobias Glasmachers Lehrende: Prof. Dr. Tobias Glasmachers					
Verwendung des Moduls M.Sc. Computer Science M.Sc. Angewandte Informatik M.Sc. IT-Sicherheit/ Informationstechnik					
Vorkenntnisse empfohlen: Vorlesung "Mathematics for Modeling and Data Analysis"					
Lernziele (learning outcomes) Internationalisierung: Die Veranstaltung wird auf Englisch durchgeführt. Digitalisierung: Inhalte werden durch Videos und Lesematerial vermittelt. Übungsaufgaben mit Programmieranteilen werden in Form von Jupyter-Notebooks bereitgestellt. Nach dem erfolgreichen Abschluss des Moduls <ul style="list-style-type: none"> • verstehen die Teilnehmer die Grundlagen der statistischen Lerntheorie • kennen die Teilnehmer die wichtigsten Algorithmen des überwachten statistischen Lernens und können diese auf Lernprobleme anwenden, • kennen die Teilnehmer Stärken und Beschränkungen verschiedenen Lernverfahren, • können die Teilnehmer Standardsoftware zum maschinellen Lernen zur Lösung neuer Probleme einsetzen. 					
Inhalt Grundlagen der statistischen Lerntheorie, Querschnitt der wichtigsten Algorithmen des maschinellen Lernens, konkrete Problemlösung mit Standardsoftware.					
Lehrformen Vorlesung mit Übung im flipped classroom Format					
Prüfungsformen Schriftliche Modulabschlussprüfung (90 Minuten)					
Voraussetzungen für die Vergabe von Credits Bestandene schriftliche Modulabschlussprüfung					
Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS) 6/105: M.Sc. Angewandte Informatik 6/97: M.Sc. Computer Science 6/91: M.Sc IT-Sicherheit/ Informationstechnik [PO 22]					

Titel des Moduls: Machine Learning: Unsupervised Methods (kein Angebot im WS 23/24)
Machine Learning: Unsupervised Methods (no offer in WS 23/24)

Modul-Nr./Code	Credits 9 CP	Workload 270 h	Semester siehe Prüfungsordnung / see examination regulations	Turnus Wintersemester (kein Angebot im WS 23/24)	Dauer 1 Semester
Lehrveranstaltungen Machine Learning: Unsupervised Methods (212501)			Kontaktzeit 60 h	Selbststudium 210 h	Gruppengröße 40 Studierende
Unterrichtssprache Englisch			Teilnahmevoraussetzungen		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Laurenz Wiskott Lehrende: Prof. Dr. Laurenz Wiskott					
Verwendung des Moduls M.Sc. Computer Science M.Sc. Angewandte Informatik M.Sc. Elektro-und Informationstechnik					
Vorkenntnisse Das mathematische Niveau des Kurses ist gemischt, aber im Allgemeinen hoch, einschließlich Infinitesimalrechnung (Funktionen, Ableitungen, Integrale, Differentialgleichungen, ...), lineare Algebra (Vektoren, Matrizen, inneres Produkt, orthogonale Vektoren, Basissysteme, ...) und ein wenig Wahrscheinlichkeitstheorie (Wahrscheinlichkeiten, Wahrscheinlichkeitsdichten, Bayes-Theorem, ...). Die Programmierung erfolgt in Python, so dass die Studierenden auch hier über Grundkenntnisse verfügen oder zumindest eine andere Programmiersprache fließend beherrschen sollten.					
Lernziele (learning outcomes) Nach dem erfolgreichen Abschluss dieses Kurses: <ul style="list-style-type: none"> • kennen Studierende eine Reihe von wichtigen unüberwachten Lernmethoden, • können sie diskutieren und entscheiden, welche der Methoden für einen gegebenen Datensatz geeignet sind, • verstehen sie die Mathematik dieser Methoden, • wissen sie, wie man diese Methoden in Python implementiert und anwendet, • haben sie Erfahrung in der Organisation und Arbeit in einem Team gesammelt, • kennen sie Problemlösungsstrategien wie Brainstorming, • können sie über all dies auf Englisch kommunizieren. 					
Inhalt Dieser Kurs behandelt eine Vielzahl von flachen, unüberwachten Methoden des maschinellen Lernens wie Hauptkomponentenanalyse, unabhängige Komponentenanalyse, Vektorquantisierung, Clustering, Bayes'sche Theorie und grafische Modelle.					
Lehrformen Dieser Kurs wird in einer Mischung aus Inverted Classroom und problemorientiertem Lernen durchgeführt. Der Kurs beginnt mit einer zweiwöchigen Einführung in unüberwachte Methoden des maschinellen Lernens, die einen Überblick gibt. Anschließend arbeiten die Studierenden in Gruppen von etwa 4 Personen an realistischen Problemen, die mit diesen Methoden gelöst werden können. In der ersten Woche eines Problems entwickeln sie Hypothesen und Strategien für eine Lösung und legen fest, welche Methoden sie erlernen wollen. Dann einigt sich der Kurs auf eine Methode, auf die er sich theoretisch konzentrieren will, was dann in einem umgekehrten Klassenraumformat geschieht. Die Studierenden versuchen dann, das Problem zu lösen und präsentieren ihre Ergebnisse in einem kurzen Vortrag mit Folien. Auf diese Weise lernen die Studierenden nicht nur etwas über maschinelles Lernen, sondern auch über Soft Skills.					
Prüfungsformen Semesterbegleitend; Die Prüfung ist eine Kombination aus benoteten Präsentationen für die Probleme und					

benoteten Quizfragen für die Theorie. 40% der Note ergeben sich aus der durchschnittlichen Gruppenleistung bei der Lösung der Aufgaben. 10% gehen in die Präsentationen ein, wobei die Folien und der Präsentationsstil berücksichtigt werden, dies ist eine individuelle Note des Vortragenden. 50% ergeben sich aus einem digitalen Quiz über die Theorie der behandelten Methoden. Somit entfallen 60% der Note auf die Einzelnote und 40% auf die Gruppennote. Darüber hinaus können Sie bis zu 8 Bonuspunkte für (i) die Wahl zum "Most Valuable Player (MVP)" für ein Projekt und (ii) die Erstellung nützlicher Quizfragen erhalten.

Voraussetzungen für die Vergabe von Credits

Kontinuierliche Teilnahme und bestandene Prüfung.

Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)

9/105: M.Sc. Angewandte Informatik

9/97: M.Sc. Computer Science

Titel des Moduls: Master-Praktikum Deep Learning and Natural Language Processing					
Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester	Turnus Sommersemester	Dauer 1 Semester
Lehrveranstaltungen			Kontaktzeit 30 h	Selbststudium 120 h	Gruppengröße 3 Studierende
Unterrichtssprache Englisch			Teilnahmevoraussetzungen		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Asja Fischer Lehrende: Prof. Dr. Asja Fischer					
Verwendung des Moduls M.Sc. Angewandte Informatik					
Vorkenntnisse					
Lernziele (learning outcomes) After successful completion of this module, <ul style="list-style-type: none"> • The students should be able to develop applications using deep learning techniques for machine learning tasks. • The students should be able to measure the efficacy of the developed machine learning applications using scientific methods. • The students should be familiar with PyTorch, a library for implementing neural network based machine learning applications and experiments 					
Inhalt This praktikum teaches the basics of the development of applications for machine learning tasks. The praktikum is aimed at but not necessarily limited to natural language processing (NLP) tasks, such as machine translation or sentiment analysis. NLP tasks are aimed at the ability to process or generate natural language (e.g. English) texts. The course is started with a short introduction to deep learning, NLP, Python programming and a development environment setup (installing dependencies and setting up an IDE (e.g. PyCharm)). Subsequently, teams of up to 3 students will choose (and may propose) topics that consist of a machine learning task. These tasks have to be solved by the students over the course of one semester, with regular meetings to discuss progress.					
Lehrformen Introductory lectures in the first weeks, followed by independent work on the chosen practical task, supported by regular meetings.					
Prüfungsformen Written report describing the motivation, methodology and result of the chosen task in Latex (max. 10 pages in ACL format), code for the task and its documentation, final presentation using slides (15 min. + 5 min. discussion)					
Voraussetzungen für die Vergabe von Credits Written report, documented code, participation in meetings, final presentation					
Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS) 5/105: M.Sc. Angewandte Informatik					

Titel des Moduls: Master-Praktikum: Autonomous Robotics					
Modul-Nr./Code	Credits 3 CP	Workload 90 h	Semester	Turnus Jedes Semester	Dauer 1 Semester
Lehrveranstaltungen Autonomous Robotics			Kontaktzeit 40h Blockpraktikum	Selbststudium 50 h	Gruppengröße Studierende
Unterrichtssprache Deutsch			Teilnahmevoraussetzungen		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Greogor Schöner Lehrende: Prof. Dr. Greogor Schöner					
Verwendung des Moduls M.Sc. Angewandte Informatik					
Vorkenntnisse Vorlesung Autonomous Robotics: Action, Perception, and Cognition					
Lernziele (learning outcomes) <ul style="list-style-type: none"> • Einüben des praktischen Arbeitens im Umfeld der autonomen Robotik, insbesondere der Entwurf von mathematischen Verfahren und ihre algorithmische Implementation zur Erzeugung von zeitlich strukturierten Abläufen • Einüben der Dokumentation von technischen Programmieraufgaben. • Verständnis der Grundlagen der autonomen Robotik anhand des einfachsten Robotersystems, eines autonomen Kleinstvehikels. • Einüben des Lesens und Schreibens von wissenschaftlich-technischen Artikeln 					
Inhalt Ziel des Praktikums ist es, die Grundlagen der autonomen Robotik anhand von Experimenten zu vermitteln, die eine enge Kopplung an aktuelle wissenschaftliche Probleme dieses Feldes aufweisen. Am Beispiel der Kleinstroboter vom Typ E-Puck wird hierbei die Erzeugung und Organisation von künstlichem Verhalten behandelt. Unter Verwendung der robotereigenen Sensorik werden experimentelle Aufgaben durch Implementierungen in der Programmierumgebung MATLAB gelöst. Der theoretische Hintergrund der Lösungen ist der dynamische Systeme Ansatz der kognitiven Robotik. Eine optionale Erweiterung des Praktikums durch das dritte CP erzielt wird, ist das Lesen einer wissenschaftlichen Veröffentlichung im Umfeld der im Praktikum behandelten Methoden und das Schreiben eines Essays, in dem Fragen zu dem Artikel durch einen längeren, strukturierten und illustrierten Text beantwortet werden.					
Lehrformen Praktikum					
Prüfungsformen Praktische Prüfung					
Voraussetzungen für die Vergabe von Credits Erfolgreiche Bearbeitung der Programmieraufgaben; Abgabe von schriftlichen Dokumentationen					
Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS) 3/105: M.Sc. Angewandte Informatik					

Titel des Moduls: Master-Praktikum: Big Data in der Bioinformatik					
Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester	Turnus Jedes Semester	Dauer 1 Semester
Lehrveranstaltungen Big Data in der Bioinformatik (202621 bzw. 190801)			Kontaktzeit 60 h	Selbststudium 90 h	Gruppengröße Studierende
Unterrichtssprache Deutsch oder Englisch			Teilnahmevoraussetzungen		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Axel Mosig Lehrende: Prof. Dr. Axel Mosig					
Verwendung des Moduls M.Sc. Angewandte Informatik					
Vorkenntnisse praktische Programmiererfahrung in einer oder mehreren Sprachen (z. B. Java, R, Python, Matlab) Vertiefungsmodul(e) aus dem Themengebiet der Bioinformatik 					
Lernziele (learning outcomes) Nach dem erfolgreichen Abschluss des Moduls <ul style="list-style-type: none"> • können Studierende praktischen Herausforderungen bei der Entwicklung von Bioinformatik-Anwendungen begegnen, • haben die Studierenden Programmier-Bibliotheken aus dem Bereich Bioinformatik kennen gelernt • haben Studierenden die Verwendung von Workflow-Systemen eingeübt • haben Studierende die Mechanismen von Code- und Programm-Dokumentation eingeübt • haben Studierende Verfahren zur Bereitstellung eigener Libraries (z. B. R-Pakete, .jar-Files) angewendet 					
Inhalt Die Bioinformatik wendet naturgemäß Informatik-Methoden auf die Daten eines lebenswissenschaftlichen Anwendungsfaches an, stellt also per se angewandte Digitalisierung dar. Das Praktikum vermittelt Grundlagen der Programmierung mit Bezug zu lebenswissenschaftlichen Anwendungen mit großen Datenmengen. Dies geschieht anhand aktueller Beispiele aus den Themengebieten Bildverarbeitung und Sequenzanalyse. Nach einer kurzen Einführung (Präsenztreffen) in Programmierung und Entwicklungsumgebungen (z. B. Java, R, C++, eclipse, RStudio, Python, Matlab) werden praktische Programmieraufgaben ausgegeben und im Laufe des Semesters bearbeitet, ggfls. mit weiteren Präsenztreffen zur Diskussion des Fortschritts.					
Lehrformen Einführung als seminaristischer Unterricht, Bearbeitung der praktischen Aufgabe selbständig oder als Gruppenarbeit.					
Prüfungsformen Semesterbegleitend; Praktische Prüfung: Protokoll max. 10 Seiten (wenn Abgabe bis zwei Wochen nach Praktikumsende, dann Korrekturrunde vor Benotung), Dokumentation der praktischen Aufgabe					
Voraussetzungen für die Vergabe von Credits Protokoll (siehe Prüfungsformen), Bearbeitung / Dokumentation der praktischen Aufgabe, Teilnahme an den Präsenztreffen, Abschlusspräsentation (15 min. + 5 min. Diskussion)					
Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS) 5/105: M.Sc. Angewandte Informatik					

Titel des Moduls: Master-Praktikum: Computational Proteomics					
Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester	Turnus Jedes Semester	Dauer 1 Semester
Lehrveranstaltungen Master-Praktikum: Computational Proteomics (201914)			Kontaktzeit 60 h	Selbststudium 90 h	Gruppengröße Studierende
Unterrichtssprache Deutsch oder Englisch			Teilnahmevoraussetzungen		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Martin Eisenacher Lehrende: Prof. Dr. Martin Eisenacher					
Verwendung des Moduls M.Sc. Angewandte Informatik					
Vorkenntnisse praktische Programmiererfahrung in einer oder mehreren Sprachen (z. B. Java, R, Python, Matlab) Vertiefungsmodul(e) aus dem Themengebiet der Bioinformatik 					
Lernziele (learning outcomes) Nach dem erfolgreichen Abschluss des Moduls <ul style="list-style-type: none"> • können Studierende praktischen Herausforderungen bei der Entwicklung von Bioinformatik-Anwendungen begegnen, • haben die Studierenden Programmier-Bibliotheken aus dem Bereich Bioinformatik kennen gelernt • haben Studierenden die Verwendung von Workflow-Systemen eingeübt • haben Studierende die Mechanismen von Code- und Programm-Dokumentation eingeübt • haben Studierende Verfahren zur Bereitstellung eigener Libraries (z. B. R-Pakete, .jar-Files) angewendet 					
Inhalt Die Bioinformatik wendet naturgemäß Informatik-Methoden auf die Daten eines lebenswissenschaftlichen Anwendungsfaches an, stellt also per se angewandte Digitalisierung dar. Das Praktikum vermittelt Grundlagen der Programmierung mit Bezug zu lebenswissenschaftlichen Anwendungen mit großen Datenmengen. Dies geschieht anhand aktueller Beispiele aus dem Themengebiet Hochdurchsatz Massenspektrometrie/ Proteomics. Nach einer kurzen Einführung (Präsenztreffen) in Programmierung und Entwicklungsumgebungen (z. B. Java, R, C++, eclipse, RStudio, Python, Matlab, KNIME) werden praktische Programmieraufgaben ausgegeben und im Laufe des Semesters bearbeitet, ggfls. mit weiteren Präsenztreffen zur Diskussion des Fortschritts.					
Lehrformen Einführung als seminaristischer Unterricht, Bearbeitung der praktischen Aufgabe selbständig oder als Gruppenarbeit.					
Prüfungsformen Semesterbegleitend; Praktische Prüfung: Protokoll max. 10 Seiten (wenn Abgabe bis zwei Wochen nach Praktikumsende, dann Korrekturrunde vor Benotung), Dokumentation der praktischen Aufgabe					
Voraussetzungen für die Vergabe von Credits Protokoll (siehe Prüfungsformen), Bearbeitung / Dokumentation der praktischen Aufgabe, Teilnahme an den Präsenztreffen, Abschlusspräsentation (15 min. + 5 min. Diskussion)					
Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS) 5/105: M.Sc. Angewandte Informatik					

Titel des Moduls: Model Checking Model Checking					
Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester siehe Prüfungsordnung	Turnus Sommersemester	Dauer 1 Semester
Lehrveranstaltungen Model Checking (211000)			Kontaktzeit 60 h	Selbststudium 90 h	Gruppengröße Studierende
Unterrichtssprache Deutsch			Teilnahmevoraussetzungen		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Thomas Zeume Lehrende: Dr. Nils Vortmeier Marko Schmellenkamp					
Verwendung des Moduls B.Sc. Informatik M.Sc. Informatik M.Sc. Angewandte Informatik B.Sc. IT-Sicherheit/ Informationstechnik M.Sc. Mathematik					
Vorkenntnisse <p>Grundlagenvorlesungen Mathematik,</p> <p>Einführung in die Theoretische Informatik (ggf. kann das nötige Wissen auch nachgeholt werden),</p> <p>Hilfreich: Logik in der Informatik, Datenstrukturen und elementare Programmierkenntnisse</p>					
Lernziele (learning outcomes) Die Studierenden lernen wie sich verteilte Systeme durch Transitionssysteme modellieren und Eigenschaften in logischen Spezifikationssprachen wie LTL und CTL spezifizieren lassen. Sie sollen elementare Algorithmen zur Überprüfung von Eigenschaften in Transitionssystemen kennenlernen. Sie sollen ein Verständnis für die Möglichkeiten und Grenzen des Model Checking entwickeln, und in die Lage versetzt werden, sich eigenständig mit fortgeschrittenen Methoden des Model Checkings auseinanderzusetzen.					
Inhalt Wie kann die Korrektheit von Software und Hardware formal überprüft werden? Im Model Checking werden Software- und Hardware-Module durch Transitionssysteme formalisiert; gewünschte Eigenschaften mit Hilfe logischer Formalismen formal beschrieben; und mit Hilfe von Algorithmen automatisiert überprüft, ob ein Transitionssystem eine formal spezifizierte Eigenschaft besitzt. In dieser Veranstaltung werden die theoretischen Grundlagen des Model Checkings vermittelt, mit einem Fokus auf logik-basierten Spezifikationssprachen. Die Spezifikationssprachen LTL und CTL werden eingeführt, ihre Ausdrucksstärke untersucht, und die wichtigsten algorithmischen Ansätze für das Model Checking vorgestellt.					
Lehrformen Vorlesung mit Übungen					
Prüfungsformen Abschlussprüfung; mündliche Prüfung (20-30min) oder schriftliche Klausur (120min) in Abhängigkeit der Teilnehmerzahl					
Voraussetzungen für die Vergabe von Credits Erfolgreiches Bestehen der Modulprüfung.					
Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)					

5/170: B.Sc. Informatik [PO 22]

5/158: B.Sc. Informatik [PO 20]

5/97: M.Sc. Informatik

5/105: M.Sc. Angewandte Informatik

5/150: B.Sc. IT-Sicherheit /Informationstechnik [PO 22]

5/149: B.Sc. IT-Sicherheit/ Informationstechnik [PO 22]

Titel des Moduls: Netzsicherheit 1
Network Security 1

Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester 3. Semester	Turnus Wintersemester	Dauer 1 Semester
Lehrveranstaltungen Netzsicherheit 1 (212012)			Kontaktzeit 60 h	Selbststudium 90 h	Gruppengröße Studierende
Unterrichtssprache Deutsch			Teilnahmevoraussetzungen		

Modulbeauftragte/r und hauptamtlich Lehrende

Modulbeauftragte/r: Prof. Jörg Schwenk
 Lehrende: Prof. Dr. Jörg Schwenk

Verwendung des Moduls

B.Sc. IT-Sicherheit / Informationstechnik
 M.Sc. IT-Sicherheit / Netze und Systeme
 M.Sc. Angewandte Informatik

Vorkenntnisse

Grundkenntnisse in TCP/IP, Grundkenntnisse der Sicherheitstechnik, Kenntnisse der Grundlagen von Kommunikationssystemen auf dem Niveau der Informatik (z.B. c't).

Lernziele (learning outcomes)

Studierende verfügen nach erfolgreichem Abschluss des Moduls über ein umfassendes Verständnis der technischen Aspekte von Netzsicherheit. Sie haben erkannt, dass Kryptographie alleine nicht ausreicht, um sicherheitstechnische Probleme zu lösen. Sie haben ein umfassendes Systemverständnis für komplexe IT-Systeme erworben. Durch eigenständige Überlegungen zur Verbesserung der Netzsicherheit bereiten sich die Studierenden auf ihre Rolle im Berufsleben vor. Sie können neue Probleme analysieren und neue Lösungsmöglichkeiten entwickeln. Sie können im Gespräch den Nutzen der von ihnen erarbeiteten Lösungen argumentativ begründen. Sie haben verstanden, dass nicht-technische Faktoren wie Fragen der Haftung und der entstehenden Kosten Entscheidungen zur IT-Sicherheit maßgeblich mit beeinflussen.

Inhalt

You can find our Moodle course via the [Moodle Search](#)

Wenn Kryptographie in einer technischen Umgebung wie einem Computer-, Daten- oder Telefonnetz eingesetzt wird, hängt die Sicherheit außer von rein kryptographischen Faktoren auch von der technischen Einbettung der Verschlüsselungs- und Signaturalgorithmen ab. Prominente Beispiele (für fehlerhafte Einbettungen) sind EFAIL (efail.8203.de), Angriffe auf die WLAN-Verschlüsselungssysteme WEP und WPA (KRACK) und diverse Angriffe auf TLS (Bleichenbacher, POODLE, DROWN, ROBOT). Das Modul „Netzsicherheit 1“ beschäftigt sich mit konkreten Netzen zur Datenübertragung und beleuchtet diese von allen Seiten auf ihre Sicherheit hin. Es umfasst folgende Teile:

- Einführung: Internet
- Einführung: Vertraulichkeit
- Einführung: Integrität
- Einführung: Kryptographische Protokolle
- PPP-Sicherheit (insb. PPTP), EAP-Protokolle
- WLAN-Sicherheit (WEP, WPA, Wardriving, KRACK)
- GSM- und UMTS-Mobilfunk (Authentisierung und Verschlüsselung)
- IPSec (ESP und AH, IKEv1 und v2, Angriffe auf IPSec)
- Dateiverschlüsselung mit OpenPGP (Datenformat, Efail, Klima-Rosa)
- E Mail-Verschlüsselung mit S/MIME (SMTP, Datenformat, Efail, POP3, IMAP)

Neben den Systemen selbst werden dabei auch publizierte Angriffe auf diese Systeme besprochen; die

Studierenden stellen selbst wissenschaftliche Überlegungen zur Verbesserung der Sicherheit an.

Lehrformen

Den aktuellen Moodle Kurs finden Sie über die [Moodle Suche](#)

Der Inhalt der Vorlesung wird über Youtube-Videos und Materialien in Moodle zur Verfügung gestellt. Ergänzend dazu gibt es in Präsenz eine Vertiefungsvorlesung. Dort werden keine neuen Themen vorgestellt, sondern die Themen der Online-Materialien werden vertiefend behandelt. Ob eine Aufzeichnung der Präsenzveranstaltung möglich ist, muss noch geklärt werden. Durch diese Mischform aus Online-Materialien und Vertiefung in Präsenz soll die Teilnahme aller Studierenden auch bei möglicherweise erhöhtem Krankenstand im Winter gewährleistet werden.

Prüfungsformen

Schriftliche Modulabschlussprüfung (120 Minuten)

Voraussetzungen für die Vergabe von Credits

Bestandene schriftliche Modulabschlussprüfung

Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)

5/150: B.Sc. IT-Sicherheit/ Informationstechnik [PO 22]

5/99: M.Sc. IT-Sicherheit/ Netze und Systeme [PO 22]

5/96: M.Sc. IT-Sicherheit/ Netze und Systeme [PO 20]

5/105: M.Sc. Angewandte Informatik

Titel des Moduls: Netzsicherheit 2
Network Security 2

Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester Siehe Prüfungsordnung	Turnus Sommersemester	Dauer 1 Semester
Lehrveranstaltungen Netzsicherheit 2 (211013)			Kontaktzeit 60 h	Selbststudium 90 h	Gruppengröße 150 Studierende
Unterrichtssprache Deutsch			Teilnahmevoraussetzungen		

Modulbeauftragte/r und hauptamtlich Lehrende

Modulbeauftragte/r: Prof. Dr. Jörg Schwenk
 Lehrende: Prof. Dr. Jörg Schwenk

Verwendung des Moduls

B.Sc. IT-Sicherheit/ Informationstechnik
 M.Sc. IT-Sicherheit/ Netze und Systeme
 M.Sc. Angewandte Informatik

Vorkenntnisse

Grundkenntnisse in TCP/IP, Grundkenntnisse der Sicherheitstechnik von Computernetzen auf dem Niveau der Informatik (z.B. C++, Java)

Lernziele (learning outcomes)

Studierende verfügen nach erfolgreichem Abschluss des Moduls über ein umfassendes Verständnis der technischen Aspekte von Netzsicherheit. Sie haben erkannt, dass Kryptographie alleine nicht ausreicht, um sicherheitstechnische Probleme zu lösen. Sie haben ein umfassendes Systemverständnis für komplexe IT-Systeme erworben. Durch eigenständige Überlegungen zur Verbesserung der Netzsicherheit bereiten sich die Studierenden auf ihre Rolle im Berufsleben vor. Sie können neue Probleme analysieren und neue Lösungsmöglichkeiten entwickeln. Sie können im Gespräch den Nutzen der von ihnen erarbeiteten Lösungen argumentativ begründen. Sie haben verstanden, dass nicht-technische Faktoren wie Fragen der Haftung und der entstehenden Kosten Entscheidungen zur IT-Sicherheit maßgeblich mit beeinflussen.

Inhalt

Wenn Kryptographie in einer technischen Umgebung wie einem Computer-, Daten- oder Telefonnetz eingesetzt wird, hängt die Sicherheit außer von rein kryptographischen Faktoren auch von der technischen Einbettung der Verschlüsselungs- und Signaturalgorithmen ab. Prominente Beispiele (für fehlerhafte Einbettungen) sind EFAIL (efail.de), Angriffe auf die WLAN-Verschlüsselungssysteme WEP und WPA (KRACK) und diverse Angriffe auf TLS (Bleichenbacher, POODLE, DROWN, ROBOT). Das Modul „Netzsicherheit“ beschäftigt sich mit konkreten Netzen zur Datenübertragung und beleuchtet diese von allen Seiten auf ihre Sicherheit hin. Es umfasst folgende Teile:

- Sicherheit von HTTP (HTTP Authentication, Secure HTTP, Architektur von SSL/TLS)
- Transport Layer Security (TLS1.2, Versionen SSL 2.0 bis TLS 1.3)
- Angriffe auf SSL und TLS (BEAST, CRIME, POODLE, Lucky13, Bleichenbacher, DROWN, Heartbleed, Invalid Curve)
- Secure Shell - SSH
- das Domain Name System und DNSSEC (faktorisierte Schlüssel)
- Sicherheit von Webanwendungen (HTML, URI, XSS, CSRF, SQLi, SSO)
- XML- und JSON-Sicherheit

Neben den Systemen selbst werden dabei auch publizierte Angriffe auf diese Systeme besprochen; die Studierenden stellen selbst wissenschaftliche Überlegungen zur Verbesserung der Sicherheit an.

Lehrformen

Vorlesung mit Übung

Prüfungsformen

Schriftliche Modulabschlussprüfung (120 Minuten)

Voraussetzungen für die Vergabe von Credits

Bestandene schriftliche Modulabschlussprüfung.

Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)

5/150: B.Sc. IT-Sicherheit/ Informationstechnik [PO 22]

5/149: B.Sc. IT-Sicherheit/ Informationstechnik [PO 20]

5/96 : M.Sc. IT-Sicherheit/ Netze und Systeme [PO20]

5/99 : M.Sc. IT-Sicherheit/ Netze und Systeme [PO22]

5/105: M.Sc. Angewandte Informatik

Titel des Moduls: Parallel Computing					
Modul-Nr./Code	Credits 6 CP	Workload 180 h	Semester	Turnus Sommersemester	Dauer 1 Semester
Lehrveranstaltungen Parallel Computing (126519)			Kontaktzeit 60 h	Selbststudium 120 h	Gruppengröße Studierende
Unterrichtssprache Englisch			Teilnahmevoraussetzungen		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Markus König Lehrende: Prof. Dr. Markus König					
Verwendung des Moduls M.Sc. Angewandte Informatik					
Vorkenntnisse Knowledge in the programming languages C, C++ or Java					
Lernziele (learning outcomes) Goals of the course "Parallel Computing" include a basic understanding of the theoretical foundation for efficient parallel algorithms, the architecture of parallel hardware systems and current parallel software paradigms used in research and industry today. Also, students learn how to develop parallel algorithms and implement them using state-of-the-art software systems and programming tools. In particular, numerically intensive engineering applications are a focus within the lecture itself as well as a target for the team projects carried out by the students of this course. Further, students must clearly present their projects results in a classroom setting to an audience with various technical background (the course is attended by students from applied computer science as well as by students from computational engineering). When the students have successfully complete this course, <ul style="list-style-type: none"> students will be familiar with the development of parallel algorithms and parallel software systems and have a good understanding of how to realize efficient parallel programs; students will have a good understanding of the basics of parallel programming to be able to, for example, select a proper parallel software system or tool that is appropriate for a given numerical engineering task, or be able to discuss possible solutions concerning necessary hardware or software requirements at a competent level. 					
Inhalt <ul style="list-style-type: none"> Introduction to parallel computing using examples from the field of engineering Theoretical foundations of parallel computing (concurrency, parallel processes, dead-locks, Amdahl's Law, Flynn's taxonomy, efficiency metrics, memory models, etc.) Parallel programming based on "shared memory" using the OpenMP application programming interface Parallel programming based on "distributed memory" based the Message Passing Interface (MPI) Hardware based parallel programming based on the General Purpose Graphical Processing Unit (GPGPU) Application of parallel programming paradigms to solve engineering tasks as a teamproject 					
Lehrformen <ul style="list-style-type: none"> lectures exercises using computers team projects 					
Prüfungsformen completion of a team project (2-4 students) presentation of project results (end of semester) oral examination (end of semester)					

Voraussetzungen für die Vergabe von Credits

- successful completion of a team project
- presentation of project results in the classroom
- oral examination

Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)

6/105: M.Sc. Angewandte Informatik

Titel des Moduls: Performance Engineering Praktikum Performance Engineering lab					
Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester siehe Prüfungsordnung	Turnus Wintersemester	Dauer 1 Semester
Lehrveranstaltungen Performance Engineering Praktikum (212422)			Kontaktzeit 30 h	Selbststudium 120 h	Gruppengröße 1 Studierende
Unterrichtssprache Deutsch und Englisch			Teilnahmevoraussetzungen Keine		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Timo Hönig Lehrende: Prof. Dr. Timo Hönig Benedict Herzog (M.Sc.)					
Verwendung des Moduls M.Sc. Angewandte Informatik					
Vorkenntnisse Inhalte des Moduls Betriebssysteme werden empfohlen.					
Lernziele (learning outcomes) Die Studierenden erlangen bei erfolgreichem Abschluss die folgenden Lernziele: <ul style="list-style-type: none"> Fachkompetenz: vertieftes Wissen, Analyse- und Evaluationsfähigkeiten im Umgang und der Entwicklung von Software zur Visualisierung von nicht-funktionalen Eigenschaften Methodenkompetenz: sichere Anwendung gängiger Werkzeuge zur Analyse von Software hinsichtlich ihrer nicht-funktionaler Eigenschaften Selbstkompetenz: Kompetenzen zur eigenständigen Planung, Bearbeitung und Evaluation von komplexen Softwarekomponenten; Umgang mit kritischen Feedback und Kompetenzen zur Fehlersuche in komplexen Softwarekomponenten 					
Inhalt Tools zur Visualisierung nicht-funktionaler Eigenschaften von Software konzentrieren sich heutzutage hauptsächlich auf Eigenschaften, die die Skalierbarkeit oder zeitlichen Eigenschaften wie Latenzen von Software analysieren. Die Betrachtung anderer nicht-funktionaler Eigenschaften wie zum Beispiel Energieverbrauch oder CO2-Emissionen haben bisher wenig Aufmerksamkeit erfahren, werden aber zunehmend Teil einer Betrachtung der „Performance“ von Software. Ziel dieses Praktikums ist die Konzeption und Weiterentwicklung existierender und neuer Tools zur Visualisierung solcher nicht-funktionaler Eigenschaften.					
Lehrformen Selbständige Bearbeitung der gestellte Aufgaben. Regelmäßige Treffen zur Fortschrittskontrolle und weitergehenden Planung der nächsten Arbeitspakete.					
Prüfungsformen praktische Prüfung (semesterbegleitend)					
Voraussetzungen für die Vergabe von Credits Erfolgreiche Teilnahme am Praktikum					
Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS) 5/105: M.Sc. Angewandte Informatik					

Titel des Moduls: Product Lifecycle Management
Product Lifecycle Management

Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester siehe Prüfungsordnung	Turnus Sommersemester	Dauer 1 Semester
Lehrveranstaltungen Product Lifecycle Management (138577)			Kontaktzeit 60 h	Selbststudium 90 h	Gruppengröße Studierende
Unterrichtssprache Deutsch			Teilnahmevoraussetzungen		

Modulbeauftragte/r und hauptamtlich Lehrende

Modulbeauftragte/r: Prof. Dr.-Ing. Detlef Gerhard
 Lehrende: Prof. Dr.-Ing. Detlef Gerhard

Verwendung des Moduls

M.Sc. Angewandte Informatik
 M.Sc. Maschinenbau
 M.Sc. Sales Engineering and Product Management

Vorkenntnisse

Lernziele (learning outcomes)

Nach dem erfolgreichen Abschluss des Moduls

- verfügen Studierende über ein breites, integriertes Wissen über die Herausforderungen des ganzheitlichen Informationsmanagements im Produktlebenszyklus und die resultierenden Anforderungen an Softwaresysteme zur Unterstützung von PLM.
- kennen und verstehen Studierende die Teilprozesse des Produktlebenszyklus, die Methoden des Product Lifecycle Management (PLM) sowie die wissenschaftlichen Grundlagen der zugrundeliegenden IT-Systeme. Indem sie praktische Beispiele und Aufgaben mit entsprechender Anwendungssoftware bearbeiten, können sie die erlernten Fertigkeiten im Umgang mit PLM Software auf konkrete und praxisorientierte PLM Problemstellungen übertragen.
- haben Studierende ein umfassendes Verständnis vom Zusammenwirken der PLM Softwaresysteme und können kritisch die Eignung von Methoden zum Objektmanagement, zum Produktstruktur- und Konfigurationsmanagement sowie zum Projekt- und Prozessmanagement differenzieren und beurteilen.
- können Studierende prozessorientiert an PLM Aufgabenstellungen herangehen, diese reflektieren und bewerten sowie selbstgesteuert verfolgen.
- können Studierende kooperativ PLM-Aufgabenstellungen in heterogenen Gruppen bearbeiten, Abläufe und Ergebnisse begründen sowie über Sachverhalte umfassend kommunizieren.

Inhalt

Das Modul vermittelt Methoden und Werkzeuge zum Product Lifecycle Management (PLM), insbesondere das dazu erforderliche Grundlagenwissen und die relevanten methodischen Aspekte von Produktinnovationsprozessen. Schwerpunkte bilden dabei die verschiedenen PLM-Funktionen entsprechender Softwaresysteme (z.B. Teile-, Dokumenten- und Produktstrukturmanagement, Klassifizierung, Konfigurationsmanagement, Projekt- und Prozessmanagement). Weiterhin werden allgemeine Methoden zur Organisation und Handhabung von Produktdaten und Benutzerinformationen sowie Methoden des Collaborative Engineerings und die Vorgehensweise bei der PLM-Einführung vermittelt.

Lehrformen

Vorlesung mit Übung

Prüfungsformen

Klausur 'Modulabschlussprüfung: Product Lifecycle Management' (90 Min., Anteil der Modulnote 100 %)

Studienbegleitende Aufgaben: Hausarbeiten (Sofern die Hausarbeiten vor der Modulabschlussprüfung absolviert werden, sind optional Bonuspunkte für die Klausur möglich) (Umfang und Abgabefristen wird am Anfang des Semesters bekanntgegeben)

Voraussetzungen für die Vergabe von Credits

- Bestandene Modulabschlussprüfung: Klausur
- Bestandene studienbegleitende Aufgaben: Hausarbeiten (Umfang und Abgabefristen wird am Anfang des Semesters bekanntgegeben)

Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)

6/105: M.Sc. Angewandte Informatik

Titel des Moduls: Quantum Information and Computation
Quantum Information and Computation

Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester siehe Prüfungsordnung / see examination regulations	Turnus Wintersemester	Dauer 1 Semester
Lehrveranstaltungen Quantum Information and Computation (212011)			Kontaktzeit 60 h	Selbststudium 90 h	Gruppengröße 40 Studierende
Unterrichtssprache Deutsch oder Englisch (depends on audience)			Teilnahmevoraussetzungen Keine		

Modulbeauftragte/r und hauptamtlich Lehrende

Modulbeauftragte/r: Prof. Dr. Michael Walter
 Lehrende: Prof. Dr. Michael Walter

Verwendung des Moduls

B.Sc. Informatik
 B.Sc. IT-Sicherheit/ Informationstechnik
 M.Sc. Angewandte Informatik
 M.Sc. IT-Sicherheit/ Informationstechnik (auf Antrag)
 M.Sc. IT-Sicherheit/ Netze und Systeme (auf Antrag)
 M.Sc. Computer Science

Vorkenntnisse

<p>Familiarity with linear algebra (in finite dimensions) and
probability (with finitely many outcomes) at the level of a first Bachelors course; we will briefly remind you of the more difficult bits in class. In addition, some mathematical maturity, since we will discuss precise mathematical statements and rigorous proofs. No background in physics is required.</p>

Lernziele (learning outcomes)

You will learn fundamental concepts, algorithms, and results in quantum information and computation. After successful completion of this course, you will know the theoretical model of quantum information and computation, how to generalize computer science concepts to the quantum setting, how to design and analyze quantum algorithms and protocols for a variety of computational problems, and how to prove complexity theoretic lower bounds. You will be prepared for an advanced course or a research or thesis project in this area.

Inhalt

This course will give an introduction to quantum information and quantum computation from the perspective of theoretical computer science.

Topics to be covered will likely include:

- Fundamentals of quantum computing: quantum bits, states and operations
- The power of quantum entanglement: nonlocal games
- Entanglement as a resource: superdense coding and teleportation
- Quantum circuit model of computation
- Quantum computing with oracles: Deutsch-Jozsa, Bernstein-Vazirani, Simon
- Quantum Fourier transform and phase estimation
- Shor's factoring algorithm
- Grover's search algorithm and beyond: how to solve SAT on a quantum computer?
- From no cloning to quantum money: a peek at quantum cryptography

The course should be of interest to students of computer science, mathematics, physics, and related disciplines. Students interested in a BSc or MSc project in quantum information, computing, cryptography, etc. are particularly encouraged to participate.

Lehrformen

Lecture with Exercise

Prüfungsformen

Final written module exam (180 minutes)

Voraussetzungen für die Vergabe von Credits

Passed written exam

Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)

5/158: B.Sc. Informatik [PO 22]

5/165: B.Sc. Informatik [PO 20]

5/150: B.Sc. IT-Sicherheit/ Informationstechnik [PO 22]

5/149: B.Sc. IT-Sicherheit/ Informationstechnik [PO 20]

5/105: M.Sc. Angewandte Informatik

5 /91: M.Sc. IT-Sicherheit/ Informationstechnik [PO 22]

5/ 99: M.Sc. IT-Sicherheit/ Netze und Systeme [PO 22]

5/97: M.Sc. Computer Science

Titel des Moduls: Schwerpunktseminar Computational Linguistics
Computational Linguistics

Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester siehe Prüfungsordnung	Turnus Jedes Semester	Dauer 1 Semester
Lehrveranstaltungen siehe Vorlesungsverzeichnis			Kontaktzeit 30 h	Selbststudium	Gruppengröße Studierende
Unterrichtssprache Englisch			Teilnahmevoraussetzungen		

Modulbeauftragte/r und hauptamtlich Lehrende

Modulbeauftragte/r: Prof. Dr. Tibor Kiss
 Lehrende: wechselt ja nach Seminar

Verwendung des Moduls

M.Sc. Angewandte Informatik

Vorkenntnisse

Idealerweise Vorkenntnisse in Linguistik (kann auch durch Literaturlektüre angezeigt werden), Programmierung und Mathematik.

Die Kenntnisse der Grundkurse Introduction to Computer Linguistic in Python und Introduction to Linguistic Models (with R) müssen vorausgesetzt werden. In den Seminaren im Sommersemester müssen zusätzlich Kenntnisse aus den Grundkursen Introduction to Linguistic Data Science (with R) und Introduction to Computational Linguistics vorausgesetzt werden.

Lernziele (learning outcomes)

- Die Studierenden sind in der Lage, kleinere Problemstellungen der Linguistic Data Science im jeweiligen Schwerpunktbereich angemessen zu formulieren.
- Sie kennen verschiedene Ansätze zur Problemlösung im jeweiligen Schwerpunktbereich und können einige davon kritisch auf Fragestellungen und Probleme anwenden.
- Sie verstehen, dass es in vielen Bereichen keine einzig richtige Antwort gibt, und können sich mit anderen konstruktiv über Lösungsansätze austauschen und Vor- und Nachteile von unterschiedlichen Perspektiven abwägen.
- Die Studierenden kennen weiterführende computerlinguistische Methoden, Modelle und Theorien und können sie kritisch auf Fragestellungen und Probleme anwenden.
- Sie sind im Umgang mit den Programmiersprachen vertraut und können mit weiterführenden Anleitungen oder Dokumentationen umgehen.
- Sie planen einfache Programme strukturiert und können sie gemäß ihrer Planung implementieren.
- Sie lesen und verstehen Programmcode von anderen und schreiben und dokumentieren ihren eigenen Code für andere nachvollziehbar.
- Sie diskutieren Theorien und Umsetzungen mit anderen kritisch.
- Typische Bias-Fallen durch Datencodierung oder Implementierung von Algorithmen sind ihnen vertraut und sie können bewusste Entscheidungen über kritische Fälle treffen.

Inhalt

In Seminaren des Schwerpunkts Computational Linguistics lernen die Studierenden Fragestellungen, Theorien und Methoden aus den Bereichen Computerlinguistik und Sprachmodellierung kennen. In einzelnen Seminaren beschäftigen sie sich auch mit spezifischen Tools, Frameworks oder Anwendungsbereichen.

Ein besonderes Augenmerk soll dabei auf den Konsequenzen von perspektivischem oder technischem Bias liegen und der Verantwortung von Computerlinguisten im Umgang mit teils persönlichen Sprachdaten.

Lehrformen

Das Modul besteht aus einem Seminar, in denen unterschiedliche Lernformen angewendet werden, wie beispielsweise

- Vortrag der Lehrenden im Plenum
- Gastvorträge
- Vortrag der Studierenden im Plenum

- Diskussionen im Plenum
- Arbeitsgruppen
- Arbeitsaufgaben
- Selbststudium

Prüfungsformen

Seminararbeit (semesterbegleitend)

Voraussetzungen für die Vergabe von Credits

Die zu erbringende Studienleistung für ein Seminar, die mögliche Prüfungsleistung in diesem Seminar und die Schwerpunktzuordnung der Veranstaltung werden im Vorlesungsverzeichnis angegeben.

Dozierende können für die Seminare die aktive Teilnahme durch Teilnahme an Diskussions oder Reflektionsrunden in den Sitzungen verpflichtend machen. Ergibt sich dadurch eine Anwesenheitspflicht zu Veranstaltungsterminen, muss dies im Vorlesungsverzeichnis zur Veranstaltung angemerkt werden.

Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)

Titel des Moduls: Schwerpunktseminar Linguistic Data Science

Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester siehe Prüfungsordnung	Turnus Jedes Semester	Dauer 1 Semester
Lehrveranstaltungen siehe Vorlesungsverzeichnis			Kontaktzeit 30 h	Selbststudium	Gruppengröße Studierende
Unterrichtssprache Englisch			Teilnahmevoraussetzungen		

Modulbeauftragte/r und hauptamtlich Lehrende

Modulbeauftragte/r: Prof. Dr. Tibor Kiss

Lehrende: wechselt ja nach Seminar

Verwendung des Moduls

M.Sc. Angewandte Informatik

Vorkenntnisse

Idealerweise Vorkenntnisse in Linguistik (kann auch durch Literaturlektüre angezeigt werden), Programmierung und Mathematik.

Die Kenntnisse der Grundkurse Introduction to Computer Linguistic in Python und Introduction to Linguistic Models (with R) müssen vorausgesetzt werden. In den Seminaren im Sommersemester müssen zusätzlich Kenntnisse aus den Grundkursen Introduction to Linguistic Data Science (with R) und Introduction to Computational Linguistics vorausgesetzt werden.

Lernziele (learning outcomes)

- Die Studierenden sind in der Lage, kleinere Problemstellungen der Linguistic Data Science im jeweiligen Schwerpunktbereich angemessen zu formulieren.
- Sie kennen verschiedene Ansätze zur Problemlösung im jeweiligen Schwerpunktbereich und können einige davon kritisch auf Fragestellungen und Probleme anwenden.
- Sie verstehen, dass es in vielen Bereichen keine einzig richtige Antwort gibt, und können sich mit anderen konstruktiv über Lösungsansätze austauschen und Vor- und Nachteile von unterschiedlichen Perspektiven abwägen.
- Die Studierenden kennen weiterführende computerlinguistische Methoden, Modelle und Theorien und können sie kritisch auf Fragestellungen und Probleme anwenden.
- Sie sind im Umgang mit den Programmiersprachen vertraut und können mit weiterführenden Anleitungen oder Dokumentationen umgehen.
- Sie planen einfache Programme strukturiert und können sie gemäß ihrer Planung implementieren.
- Sie lesen und verstehen Programmcode von anderen und schreiben und dokumentieren ihren eigenen Code für andere nachvollziehbar.
- Sie diskutieren Theorien und Umsetzungen mit anderen kritisch.
- Typische Bias-Fallen durch Datencodierung oder Implementierung von Algorithmen sind ihnen vertraut und sie können bewusste Entscheidungen über kritische Fälle treffen.

Inhalt

In Seminaren des Schwerpunkts Computational Linguistics lernen die Studierenden Fragestellungen, Theorien und Methoden aus den Bereichen Computerlinguistik und Sprachmodellierung kennen. In einzelnen Seminaren beschäftigen sie sich auch mit spezifischen Tools, Frameworks oder Anwendungsbereichen.

Ein besonderes Augenmerk soll dabei auf den Konsequenzen von perspektivischem oder technischem Bias liegen und der Verantwortung von Computerlinguisten im Umgang mit teils persönlichen Sprachdaten.

Lehrformen

Das Modul besteht aus einem Seminar, in denen unterschiedliche Lernformen angewendet werden, wie beispielsweise

- Vortrag der Lehrenden im Plenum
- Gastvorträge
- Vortrag der Studierenden im Plenum

- Diskussionen im Plenum
- Arbeitsgruppen
- Arbeitsaufgaben
- Selbststudium

Prüfungsformen

Seminararbeit (semesterbegleitend)

Voraussetzungen für die Vergabe von Credits

Die zu erbringende Studienleistung für ein Seminar, die mögliche Prüfungsleistung in diesem Seminar und die Schwerpunktzuordnung der Veranstaltung werden im Vorlesungsverzeichnis angegeben.

Dozierende können für die Seminare die aktive Teilnahme durch Teilnahme an Diskussions oder Reflektionsrunden in den Sitzungen verpflichtend machen. Ergibt sich dadurch eine Anwesenheitspflicht zu Veranstaltungsterminen, muss dies im Vorlesungsverzeichnis zur Veranstaltung angemerkt werden.

Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)

Titel des Moduls: Simulationstechnik					
Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester siehe Prüfungsordnung	Turnus Wintersemester	Dauer 1 Semester
Lehrveranstaltungen Simulationstechnik (127022)			Kontaktzeit 60 h	Selbststudium 90 h	Gruppengröße Studierende
Unterrichtssprache Deutsch			Teilnahmevoraussetzungen		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Markus K Lehrende: Prof. Dr. Markus K					
Verwendung des Moduls M.Sc. Angewandte Informatik					
Vorkenntnisse <p>Objektorientierte Modellierung</p>					
Lernziele (learning outcomes) Dieses Modul vermittelt Kompetenzen zum Einsatz von rechnergestützten Simulationskonzepten zur Lösung praxisbezogener Aufgabenstellungen im Bau- und Umweltingenieurwesen. Im Rahmen der Übungsveranstaltungen erfolgt eine Einführung in aktuelle Simulations- und Optimierungssoftware. Die Projektarbeit wird als Gruppenarbeit durchgeführt.					
Inhalt Es werden Vorgehensweisen zur Simulation komplexer Systeme vermittelt. Folgende Themen werden behandelt: <ul style="list-style-type: none"> • Systemanalyse und Modellbildung • System Dynamics • Ereignisdiskrete Simulation • Multiagentensimulation • Aufbereitung von Eingangsdaten • Stochastische Simulation • Simulationsgestützte Optimierung • Einführung in die Software AnyLogic <p>Innerhalb der Projektarbeit werden aktuelle Fragestellungen aus den Bau- und Umweltingenieurwissenschaften (Montage- und Logistikprozesse, Fußgängersimulation, Schadstoffausbreitung, etc.) aufgearbeitet und mit Hilfe einer objektorientierten Simulationssoftware analysiert. Den Studierenden werden Softwarelizenzen durch den Lehrstuhl zur Verfügung gestellt. Die Programmierung erfolgt unter Verwendung der Programmiersprache Java.</p>					
Lehrformen Vorlesung mit Übung					
Prüfungsformen Projektarbeit mit Abschlusspräsentation					
Voraussetzungen für die Vergabe von Credits Bestandene Projektarbeit mit Abschlusspräsentation					
Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS) 5/105: M.Sc. Angewandte Informatik					

Titel des Moduls: Software Security Software Security					
Modul-Nr./Code	Credits 9 CP	Workload 270 h	Semester siehe Prüfungsordnung	Turnus Wintersemester	Dauer 1 Semester
Lehrveranstaltungen Software Security (212026)			Kontaktzeit 60 h	Selbststudium 210 h	Gruppengröße Studierende
Unterrichtssprache Englisch			Teilnahmevoraussetzungen Systemsicherheit und Betriebssysteme		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Kevin Borgolte Lehrende: Prof. Kevin Borgolte					
Verwendung des Moduls M.Sc. IT-Sicherheit/ Informationstechnik M.Sc. IT-Sicherheit/ Netze und Systeme M.Sc. Angewandte Informatik					
Vorkenntnisse Keine					
Lernziele (learning outcomes)					
Inhalt					
Lehrformen Vorlesung mit Übung					
Prüfungsformen Schriftliche Hausarbeit (Take-Home-Exam) am Ende der Vorlesungszeit.					
Voraussetzungen für die Vergabe von Credits Bestandene Hausarbeit.					
Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS) 5/91: M.Sc. IT-Sicherheit/ Informationstechnik 5/99: M.Sc. IT-Sicherheit/ Netze und Systeme 5/105: M.Sc. Angewandte Informatik					

Titel des Moduls: Studienprojekt (Master)					
Modul-Nr./Code	Credits 10 CP	Workload 300 h	Semester 3	Turnus Jedes Semester	Dauer 1 Semester
Lehrveranstaltungen Individuelle Team- bzw. Gruppenprojekte			Kontaktzeit 30 h	Selbststudium 270 h	Gruppengröße Studierende
Unterrichtssprache Deutsch oder Englisch			Teilnahmevoraussetzungen		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Studiendekan Angewandte Informatik Lehrende: Dozierende im Studiengang Angewandte Informatik					
Verwendung des Moduls M.Sc. Angewandte Informatik					
Vorkenntnisse Gute Programmierkenntnisse (aus den Modulen Informatik 1 und 2, Software-Engineering) und ggf. weitere Kompetenzen je nach Wahl des Projektes					
Lernziele (learning outcomes) Ziele des Studienprojekts sind: <ul style="list-style-type: none"> • Die Anwendung des erlernten Fachwissens. • Der Erwerb zusätzlicher Fachkompetenz gemäß der jeweiligen projektspezifischen Aufgabenstellung. • Die Schulung hinsichtlich der Erarbeitung eigener Lösungsstrategien. • Die Schulung hinsichtlich Arbeitsteilung und Zusammenarbeit im Team (Teamfähigkeit und Projektorganisation). • Erwerb weiterer Kompetenz im Hinblick auf die Dokumentation und die Präsentation von Ergebnissen. 					
Inhalt Im Rahmen des Studienprojekts soll eine Aufgabe aus Bereichen der Angewandten Informatik in Teamarbeit unter Anleitung eines Betreuers gelöst werden. Die angebotenen Projekte decken dabei thematisch die gesamte Bandbreite der Vertiefungsmodule ab. Darüber hinaus werden auch interdisziplinäre Studienprojekte angeboten. In diesen arbeiten z.B. Angewandte Informatiker mit Studierenden der Sozialwissenschaften oder der Sportwissenschaft zusammen.					
Lehrformen Betreutes Projekt					
Prüfungsformen Projektarbeit mit Abschlussbericht					
Voraussetzungen für die Vergabe von Credits Aktive Teilnahme am Projekt, fristgerechte Abgabe des Abschlussberichts und Bewertung des Abschlussberichts mit min. ausreichend. Hinweis: Jeder Abschlussbericht muss im Umfang von einer DinA4 Seite eine Reflexion über das methodische Vorgehen enthalten.					
Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS) 10/105: M.Sc. Angewandte Informatik					

Titel des Moduls: Systemsicherheit System Security					
Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester siehe Prüfungsordnung	Turnus Sommersemester	Dauer 1 Semester
Lehrveranstaltungen Systemsicherheit (211011)			Kontaktzeit 60 h	Selbststudium 90 h	Gruppengröße Studierende
Unterrichtssprache Englisch			Teilnahmevoraussetzungen keine		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Ghassan Karame Lehrende: Prof. Dr. Ghassan Karame					
Verwendung des Moduls B.Sc. IT-Sicherheit/ Informationstechnik B.Sc. Informatik M.Sc. Angewandte Informatik M.Sc. IT-Sicherheit/ Netze und Systeme					
Vorkenntnisse Background in Cryptographic primitives (encryption methods, signatures, MACs, hash functions), principles of communication networks, is recommended.					
Lernziele (learning outcomes) At the end of this course, students will be able to <ul style="list-style-type: none"> • classify and describe vulnerabilities and protection mechanisms of popular systems and protocols, and • analyze / reason about basic protection mechanisms for modern OSs, software, and hardware systems. Students will also develop the ability to reason about the security of a given protocol and independently develop appropriate security defenses and security models. 					
Inhalt While clearly beneficial, the large-scale deployment of online services has resulted in the increase of security threats against existing services. As the size of the global network grows, the incentives of attackers to abuse the operation of online applications also increase and their advantage in mounting successful attacks becomes considerable. These cyber-attacks often target the resources, availability, and operation of online services. With an increasing number of services relying on online resources, integrating proper security measures therefore becomes integral to ensure the correct functioning of every online service. In this course, we discuss important theoretical and analytical aspects in system security. The focus of the course is to understand basic attack strategies on modern systems and platforms, with a focus on side-channel attacks, software-based attacks, malware analysis, as well as software-based defenses (e.g., address space randomization and non-executable memory) and hardware-based defenses (e.g., using TPMs and TEEs). Other topics of the course include analyzing the security of modern cryptocurrencies and ML platforms, and similar aspects in system security. An integral part of this course are exercises and homeworks, which aim to deepen the understanding of the material with practical examples.					
Lehrformen					
Prüfungsformen					

Voraussetzungen für die Vergabe von Credits

Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)

5/150: B.Sc. IT-Sicherheit/ Informationstechnik [PO 22]

5/149: B.Sc. IT-Sicherheit/ Informationstechnik [PO 20]

5/158: B.Sc. Informatik [PO 22]

5/165: B.Sc. Informatik [PO 20]

5/105: M.Sc. Angewandte Informatik

5/99: M.Sc. IT-Sicherheit/ Netze und Systeme [PO 22]

5/96: M.Sc. IT-Sicherheit/ Netze und Systeme [PO 20]

Titel des Moduls: Theorie des maschinellen Lernens
Theory of machine learning

Modul-Nr./Code	Credits 9 CP	Workload 270 h	Semester siehe Prüfungsordnung / see examination regulations	Turnus Sommersemester	Dauer 1 Semester
Lehrveranstaltungen Theorie des maschinellen Lernens (211052)			Kontaktzeit 90 h	Selbststudium 180 h	Gruppengröße 20 Studierende
Unterrichtssprache Deutsch			Teilnahmevoraussetzungen		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Asja Fischer Lehrende: Prof. Dr. Asja Fischer					
Verwendung des Moduls M.Sc. Angewandte Informatik M.Sc. Computer Science					
Vorkenntnisse					
Lernziele (learning outcomes) Die Studierenden werden mit mathematischen Modellen für das maschinelle Lernen vertraut gemacht. Sie erwerben die Fähigkeit, Lernalgorithmen zu beurteilen und zu vergleichen anhand des Grades, in welchem diese (exakt beschriebene) Erfolgskriterien erreichen. Sie erwerben Techniken sowohl zum Design effizienter Lernalgorithmen als zum Nachweis der inhärenten Härte eines Problems. Nach dem erfolgreichen Abschluss des Moduls <ul style="list-style-type: none"> • kennen Studierende die wichtigsten Lernmaschinen (wie zum Beispiel Support Vector Machines und verwandte Modelle), • verstehen Studierende den Unterschied zwischen empirischer und realer Fehlerrate und kennen Techniken zum Umgang mit dem Problem des overfitting der Daten (mit einem zu komplexen Modell), • können Studierende zwischen unformer und nicht unformer Lernbarkeit einer Hypothesenklasse unterscheiden und kennen die dazu passenden Theorien und Lernregeln. 					
Inhalt Gegenstand der Vorlesung ist die statistik-basierte Theorie des maschinellen Lernens. Insbesondere wird die Methode der strukturierten Risikominimierung vermittelt sowie die ihr zugrunde liegenden statistischen Lehrsätze. Es werden sowohl Techniken zum Entwurf effizienter Lernalgorithmen besprochen als auch informations- oder berechnungstheoretische Barrieren, die bestimmte Lernprobleme als nicht effizient lösbar erscheinen lassen.					
Lehrformen Vorlesung mit Übung					
Prüfungsformen mündliche Modulabschlussprüfung (30 Minuten)					
Voraussetzungen für die Vergabe von Credits Bestandene mündliche Modulabschlussprüfung.					
Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS) 9/105: M.Sc. Angewandte Informatik 9/97: M.Sc. Computer Science					

Titel des Moduls: Verkehrstechnik					
Modul-Nr./Code	Credits 6 CP	Workload 180 h	Semester	Turnus Sommersemester	Dauer 1 Semester
Lehrveranstaltungen a) Verkehrssteuerung (128026) b) Modellierung und Simulation des Verkehrsflusses (128025)			Kontaktzeit 60 h	Selbststudium 120 h	Gruppengröße Studierende
Unterrichtssprache Deutsch			Teilnahmevoraussetzungen		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr.-Ing. Justin Geistefeldt Lehrende: Prof. Dr.-Ing. Justin Geistefeldt, Prof. Dr.-Ing. Ning Wu					
Verwendung des Moduls M.Sc. Angewandte Informatik M.Sc. Bau- und Umweltingenieurwesen					
Vorkenntnisse <p>Höhere Mathematik</p>					
Lernziele (learning outcomes) a) Die H6rer verfugen 6ber das aktuelle technische Wissen und besitzen erweiterte Kenntnisse 6ber die Methoden der verkehrstechnischen Analyse und Steuerung von Knotenpunkten. Sie haben die F6higkeit, die in der Praxis angewandten Planungstechniken f6r Lichtsignalanlagen zu verstehen und komplexe Anlagen einschlie6lich einer Koordinierung praxisgerecht zu entwerfen. b) Die H6rer verfugen 6ber differenzierte Kenntnisse der Gesetzm66igkeiten des Verkehrsflusses auf Stra6en. Sie sind in der Lage, wissenschaftliche Beschreibungsm66glichkeiten dieser Gesetzm66igkeiten zu reflektieren und ihre praktische Anwendbarkeit zu erkennen. Sie haben die F6higkeit, selbst6ndig Erweiterungen oder Anpassungen von Verkehrsflussmodellen zu entwickeln.					
Inhalt a) Es werden Methoden der verkehrstechnischen Analyse und Bemessung von Stra6en-knotenpunkten sowie Steuerungssysteme f6r Knotenpunkte und die zu ihrem Betrieb erforderlichen Einrichtungen behandelt. Die in der Praxis 6blichen Verfahren werden in der 6bung an einigen Beispielen veranschaulicht. Dabei werden EDV-Verfahren eingesetzt. Im Einzelnen werden behandelt: Wartezeitermittlung an Knotenpunkten, vorfahrtgeregelte Knotenpunkte, Festzeitsteuerung von Signalanlagen, Gr6ne Welle, Koordinierung im Netz, verkehrabh6ngige Steuerung einschlie6lich Signalprogrammbildung, Signaltechnik, Steuerungskriterien. b) Die theoretischen Grundlagen f6r die Beschreibung des Verkehrsflusses auf Stra6en werden mit Hilfe mathematischer Verfahren erarbeitet. Die zu Grunde liegenden Gesetzm66igkeiten werden hergeleitet. Im Einzelnen werden behandelt: Kenngr66en des Verkehrsablaufs und deren Zusammenh6nge, Fundamentaldiagramm, Kapazit6t, freier Verkehrsfluss, Kontinuumstheorie, Abstandsmodelle, Fahrzeugfolgetheorie, mikroskopische Verkehrsflusssimulation.					

Lehrformen Powerpoint-Präsentationen, Tafel, Vorführungen und Übungen am PC
Prüfungsformen Schriftliche Modulabschlussprüfung (120 Minuten); Klausur über das gesamte Modul.
Voraussetzungen für die Vergabe von Credits Bestandene schriftliche Modulabschlussprüfung.
Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS) 6/105: M.Sc. Angewandte Informatik

Titel des Moduls: Freie Wahlmodule					
Modul-Nr./Code	Credits 15 CP	Workload 450 h	Semester	Turnus Jedes Semester	Dauer 1 Semester
Lehrveranstaltungen			Kontaktzeit abhängig von Veranstaltungswahl	Selbststudium Je nach Veranstaltungswahl	Gruppengröße Studierende
Unterrichtssprache Je nach Veranstaltungswahl			Teilnahmevoraussetzungen		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Studienfachberatung Angewandte Informatik Lehrende:					
Verwendung des Moduls M.Sc. Angewandte Informatik					
Vorkenntnisse abhängig von Veranstaltungswahl					
Lernziele (learning outcomes) Die Teilnehmer erwerben so genannte Schlüsselfähigkeiten in den freien Wahlfächern					
Inhalt Studierende müssen Veranstaltungen im Gesamtumfang von 15 CP absolvieren. Je nach Veranstaltungswahl werden unterschiedliche Inhalte vermittelt. Studierende haben die Möglichkeit unter den Freien Wahlmodulen auch Fächer jenseits der Informatik zu absolvieren und ihre Soft Skills zu erweitern. Z.B. Die freien Wahlfächer erweitern die Soft Skills. Z.B. wird die englische Fachsprache verbessert, in die Grundlagen der Rechtswissenschaften eingeführt oder Grundkenntnisse der Betriebswirtschaft vermittelt. Die Studierenden haben die Möglichkeit eine Auswahl entsprechend der eigenen Interessen zu treffen.					
Lehrformen abhängig von Veranstaltungswahl					
Prüfungsformen abhängig von Veranstaltungswahl					
Voraussetzungen für die Vergabe von Credits abhängig von Veranstaltungswahl					
Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS) 0/105 (unbenotet): M.Sc. Angewandte Informatik					

Titel des Moduls: Masterarbeit und Kolloquium (AI)					
Modul-Nr./Code	Credits 30 CP	Workload 900 h	Semester 4	Turnus Jedes Semester	Dauer 1 Semester
Lehrveranstaltungen			Kontaktzeit 15 h	Selbststudium 885 h	Gruppengröße 1 Studierende
Unterrichtssprache Deutsch oder Englisch			Teilnahmevoraussetzungen M.Sc. Angewandte Informatik PO 22: Erfolgreich abgeschlossene Module im Umfang von 70 CP. M.Sc. Angewandte Informatik PO 20: Erfolgreich abgeschlossene Module im Umfang von 60 CP		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Studiendekan Angewandte Informatik Lehrende: Lehrende im Studiengang Angewandte Informatik					
Verwendung des Moduls M.Sc. Angewandte Informatik					
Vorkenntnisse Abhängig von der Themenwahl.					
Lernziele (learning outcomes) Nach erfolgreichem Abschluss des Moduls: <ul style="list-style-type: none"> • können Studierende selbstständig und fristgerecht ein wissenschaftliches Thema bearbeiten von der Recherche bis zur Dokumentation der Resultate • können Studierende geeignete wissenschaftliche Verfahren und Methoden, die sie im Studium kennengelernt haben, auswählen, anwenden und weiterentwickeln, um ein konkretes Problem zu lösen • können Studierende ihre Ergebnisse kritisch mit dem Stand der Forschung vergleichen und evaluieren • können Studierende ihre eigenen Ergebnisse angemessen in Wort und Schrift darstellen. 					
Inhalt Es soll eine anspruchsvolle Fragestellung der Angewandten Informatik bearbeitet und dokumentiert werden. Im Anschluss an die Bearbeitung der Masterarbeit werden die Ergebnisse in Form eines Kolloquiumsvortrags mit anschließender Diskussion präsentiert.					
Lehrformen Abschlussarbeit					
Prüfungsformen Masterarbeit und Kolloquiumsvortrag					
Voraussetzungen für die Vergabe von Credits Sowohl die Masterarbeit als auch der Kolloquiumsvortrag müssen bestanden sein. Der Anteil der Kolloquiumsnote an der Gesamtnote beträgt 10%					
Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS) 30/105: M.Sc. Angewandte Informatik					

