Trier University
of Applied Sciences

HOCHSCHULE TRIER

MODULHANDBUCH

der Master-Studiengänge im Fachbereich Informatik
Prüfungsordnung 2021

Stand: 29.06.2026

Inhaltsverzeichnis

Hinweise zu den Modulhandbucheinträgen 3
Abschlussarbeit 4
Anforderungsmanagement 5
Berechenbarkeit und Komplexität 6
Data Science 7
Data Warehouse 8
High Performance Computing 9
Informationssicherheit 10
Interactive Physical Simulation 12
Kooperative Systeme 13
Künstliche Intelligenz für Roboter 14
Künstliche neuronale Netze 15
Learning Analytics and Educational Data Mining 16
Lineare Optimierung 17
Maschinelles Lernen 18
Mensch-Computer-Interaktion 19
Mobile Robotik 20
Projektstudium 21
Safety KI-basierter Cyber Physical Systems 22
Seminar 23
Simulationstechnik und Reinforcement Learning 24
Software-Architekturen 25
Software-Qualitätsmanagement 26
Statistics and Learning Theory 27
Ubiquitous Computing 28
Verlässliche Echtzeitsysteme 29
Visuelle Navigation 30
Wissensrepräsentation mit Beschreibungslogiken 31

Hinweise zu den Modulhandbucheinträgen

- Die Dauer aller Module beträgt jeweils ein Semester.
- Die zeitliche Lage aller Module ergibt sich aus Anlage 1 bzw. Anlage 2 der Fachprüfungsordnung.
- Neben den jeweils genannten Voraussetzungen werden in allen Modulen Grundkenntnisse der Informatik erwartet, die in der Regel in einem Bachelor-Studium der Informatik vermittelt werden.
- Zur individuellen Vorbereitung kann der Besuch der im Feld „Empfohlene Voraussetzungen" genannten Module dienen, ebenso wie die Angaben im Feld „Literatur". Weitere Literaturangaben finden sich ggf. in den Vorlesungsunterlagen, die i.d.R. auf Stud.IP (https://studip.hochschule-trier.de) zur Verfügung gestellt werden.
- Falls in einem Modul eine Studienleistung zu erbringen ist, ist diese gemäß § 6 der allgemeinen Prüfungsordnung Voraussetzung für die Zulassung zur Prüfung. Die semesteraktuelle Form der Studienleistung wird zu Beginn der Lehrveranstaltung bekannt gegeben.
- Bei Angabe mehrerer alternativer Prüfungsformen für ein Modul wird die semesteraktuelle Prüfungsform zu Beginn der Lehrveranstaltung bekannt geben.
- Voraussetzung für die Vergabe von ECTS-Punkten ist eine Bewertung der Prüfungsleistung mit mindestens der Note „ausreichend".
- Die Gewichtung eines Modulergebnisses zur Bildung der Gesamtnote ergibt sich gemäß § 10 der Fachprüfungsordnung aus dem Verhältnis der ECTS-Punkte für das Modul und der Gesamtanzahl der ECTS-Punkte.

- Dieses Modulhandbuch spiegelt das jeweils aktuelle Lehrangebot wider. Die Modulbeschreibungen nicht mehr angebotener Module können im Fachbereich Informatik angefragt werden.
- Neben den oben genannten Modulen könnten weitere Module von anderen Fachbereichen angeboten werden, welche in den Master-Studiengängen des Fachbereichs Informatik als Pflicht- oder Wahlpflichtmodul zur Verfügung stehen.

Abschlussarbeit			
Inhalte	Die Abschlussarbeit umfasst die Bearbeitung einer qualifizierten Aufgabenstellung deren Schwierigkeitsgrad der späteren Berufspraxis eines Master-Absolventen entspricht mithilfe wissenschaftlicher Methoden. Die Aufgabenstellung kann theoretische, experimentelle, empirische oder praxisorientierte Probleme umfassen. Dabei werden systematische Vorgehensweisen und sinnvolle Arbeitstechniken eingeübt sowie die Verbindung zu Anwendungsgebieten der Informatik hergestellt. Die Studierenden präsentieren ihre Ergebnisse in einer Ausarbeitung unter Anwendung der Kriterien wissenschaftlichen Arbeitens sowie in einem Kolloquium vor einer Prüfungskommission. Dabei wird der Inhalt der Abschlussarbeit im Kontext des jeweiligen Themas hinterfragt.		
Lernergebnisse	Die Studierenden haben durch die erfolgreiche Bearbeitung gezeigt, dass sie in der Lage sind, innerhalb einer vorgegebenen Frist ein größeres Fachproblem selbstständig mit wissenschaftlichen Methoden zu bearbeiten. Sie verfügen über ein breites und integriertes Wissen, einschließlich der wissenschaftlichen Grundlagen sowie über ein kritisches Verständnis der wichtigsten Theorien und Methoden. Sie sind in der Lage, die im Studium erworbenen Kenntnisse, Fähigkeiten, Methoden und Werkzeuge auf neue Fragestellungen zu übertragen und darüber hinaus selbstständig um relevante Inhalte zu erweitern, zu bewerten und wissenschaftlich zu interpretieren. Sie leiten auf dieser Basis fundierte Lösungsansätze ab und formulieren eine dem Stand der Wissenschaft entsprechende Lösung für das Fachproblem. Sie können ihre Ergebnisse darüber hinaus professionell und unter Anwendung der Kriterien des wissenschaftlichen Arbeitens dokumentieren, in einem Kolloquium darlegen und argumentativ vertreten.		
Lehrform	☐ Vorlesung		
	☐ Übung		
	☐ Seminar/Seminaristischer Unterricht		
	☐ Labor		
	☒ Projekt		
Empfohlene Voraussetzungen	Kompetenzen gemäß der Lernergebnisse der Module „Seminar" und „Projektstudium". Weitere empfohlene Voraussetzungen abhängig von der Aufgabenstellung; werden vom Betreuer festgelegt.		
Literatur	Helmut Balzert, Marion Schröder, Christian Schäfer: Wissenschaftliches Arbeiten. W3L GmbH.		
Studienleistung	☐ Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation)		
	☐ Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation)		
	☐ Bestehen von Leistungsstandkontrollen		
Prüfungsform	☐ Mündliche Prüfung		
	☐ Klausur		
	☐ Prüfung am PC		
	☐ Hausarbeit (ggf. mit Präsentation)		
	☒ Projekt (ggf. mit Präsentation)		
Verwendbarkeit	Künstliche Intelligenz und Data Science		☒ PF ☐ WPF
Angebot	☒ Sommersemester ☒ Wintersemester ☐ Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	30	30 Stunden	870 Stunden
Lehrende(r)	Dozenten des Fachbereichs Informatik		
Modulverantwortliche(r)	Fachrichtungsleiter Informatik		
Änderungsdatum	13.02.2023		

Anforderungsmanagement			
Inhalte	In den meisten Unternehmen sind Anforderungen an Softwaresysteme oft unklar, widersprüchlich, unvollständig und nicht nachvollziehbar dokumentiert. Existierende Anforderungsspezifikationen (Lasten- und Pflichtenhefte) sind veraltet. Wichtige Anforderungen werden oft zu spät erkannt oder sogar übersehen. Darüber hinaus werden Anforderungen oft qualitativ unzureichend formuliert und lassen Spielraum für Interpretation. Die Folgen: unzufriedene Kunden, explodierende Kosten, weit überschrittene Projekttermine, unwartbare Systeme. Aufgabe des Anforderungsmanagements (engl.: Requirements Engineering oder RE) ist es, aus oft vagen und teilweise widersprüchlichen Ideen eine möglichst vollständige, korrekte widerspruchs- und redundanzfrei, nachverfolgbare und atomare Systemspezifikationen zu erzeugen, um diesen aufgeführten Problemen frühzeitig entgegenwirken zu können. Im zweiten Teil der Vorlesung wird auf aktuelle Entwicklungen im Bereich des Anforderungsmanagement eingegangen. Hierbei wird der Zusammenhang zwischen Anforderungsmanagement und Variantenmanagement behandelt. Im Mittelpunkt steht zunächst das Verstehen der variantenbezogenen Entwicklungstechniken (Domänen- und Applikationsengineering). Das Wissen, Verstehen und die Anwendung werden durch praxisorientierte Übungen, sowie den Einsatz aktueller Anforderungsmanagement- und Variantenmanagement-Werkzeuge vertieft.		
Lernergebnisse	Die Studierenden haben aufbauend auf der gleichnamigen Bachelor-Vorlesung die Kenntnisse bzgl. der Modellierung, Ermittlung und Dokumentation von Anforderungen vertieft. Die Studierenden können ▪ Systeme hinsichtlich Ihrer wesentlichen Eigenschaften erfassen und abstrahieren, ▪ Anforderungen (funktional, nicht-funktional) an Systeme sammeln und spezifizieren, ▪ Techniken zur Stakeholder-Identifikation und Interviews kennen und anwenden, ▪ Review-Arten unterscheiden und anwenden, ▪ Anforderungen auf deren Qualität beurteilen, ▪ Agile Vorgehensweisen im Requirements Engineering unterscheiden und anwenden, ▪ aktuelle Entwicklungen wie Standardisierungen im Anforderungsmanagement (z.B. ReqIF) beurteilen, und ▪ aktuelle Werkzeuge aus dem Bereich Variantenmanagement im Requirements Engineering anwenden.		
Lehrform	☒ Vorlesung		
	☒ Übung		
	☐ Seminar/Seminaristischer Unterricht		
	☒ Labor		
	☐ Projekt		
Empfohlene Voraussetzungen	Kompetenzen gemäß der Lernergebnisse des Bachelor-Moduls „Grundlagen des Anforderungsmanagements"		
Literatur	▪ Ian Sommerville: Software Engineering. Pearson Studium, 2007. ▪ Klaus Pohl: Requirements Engineering - Grundlagen, Prinzipien, Techniken. dpunkt.Verlag, 2. Auflage, 2008. ▪ Soren Lauesen: Software Requirements, Styles and Techniques. Addison-Wesley, 2002. ▪ Colin Hood, Simon Wiedemann, Stefan Fichtinger, Urte Pautz: Requirements Management. Springer, 2008. ▪ Helmuth Partsch: Requirements-Engineering systematisch. Springer, 2. Auflage, 2010.		
Studienleistung	☒ Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation)		
	☐ Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation)		
	☐ Bestehen von Leistungsstandkontrollen		
Prüfungsform	☒ Mündliche Prüfung (nur bei geringer Teilnehmerzahl)		
	☒ Klausur		
	☐ Prüfung am PC		
	☐ Hausarbeit (ggf. mit Präsentation)		
	☐ Projekt (ggf. mit Präsentation)		
Verwendbarkeit	Künstliche Intelligenz und Data Science		☐ PF ☒ WPF
Angebot	☒ Sommersemester ☐ Wintersemester ☐ Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	6	60 Stunden	120 Stunden
Lehrende(r)	Prof. Dr. G. Rock		
Modulverantwortliche(r)	Prof. Dr. G. Rock		
Änderungsdatum	03.02.2023		

Berechenbarkeit und Komplexität			
Inhalte	▪ Äquivalente Modelle der Computer-Berechenbarkeit: WHILE, Turingmaschinen ▪ Algorithmenbegriff, These von Church ▪ Berechenbare Funktionen und ihre grundlegenden Eigenschaften, z.B. s-m-n-Theorem, universelle Algorithmen, Gödelisierung ▪ Entscheidbarkeit, Semi-Entscheidbarkeit Aufzählbarkeit, äquivalente Charakterisierungen, die Klassen REC und RE ▪ Unentscheidbare Mengen, Halteproblem, Many-one-Reduzierbarkeit, RE-Vollständigkeit, Beispiel nicht-aufzählbarer Mengen, Reduktionsmethode, Satz von Rice ▪ Laufzeitanalyse, Polynomial- und Exponentialzeit, Katalog klassischer Entscheidungsprobleme ▪ Die Klassen P und NP, NP-Vollständigkeit, originärer Vollständigkeitsbeweis ▪ Reduktionen durch Komponentendesign, Vollständigkeitsbeweise für Auswahl-, Graphen- und Reihenfolgeprobleme ▪ Strenge NP-Vollständigkeit, Pseudo-Polynomialzeit, Large Number Problems, Dynamische Programmierung, pseudo-polynomielle Reduktionen ▪ NP-Optimierungsprobleme, die Klassen PO und NPO, Turing-Reduzierbarkeit, Begriff der (strengen) NP-Härte, PO-NPO-Frage ▪ Approximationsalgorithmen, Performanzrate, Beispiele mit konstanter Performanzrate, Approximierbarkeit, Approximationsschema		
Lernergebnisse	Die Studierenden können ▪ Berechenbarkeitsmodelle hinsichtlich ihrer prinzipiellen Leistungsfähigkeit einordnen, ▪ die Bedeutung eines präzisen Algorithmenbegriffs erläutern, ▪ Berechnungsprobleme hinsichtlich ihrer prinzipiellen algorithmischen Beherrschbarkeit analysieren und bewerten, ▪ die Bedeutung der P-NP-Theorie präsentieren und Berechnungsprobleme in dieser Hinsicht klassifizieren, ▪ die Methode der Reduktion auf neue Berechnungsprobleme anwenden und ▪ Optimierungsprobleme kategorisieren und deren praktische Lösbarkeit beurteilen.		
Lehrform	☒ Vorlesung		
	☒ Übung		
	☐ Seminar/Seminaristischer Unterricht		
	☐ Labor		
	☐ Projekt		
Empfohlene Voraussetzungen	Keine		
Literatur	▪ H. Rogers Jr.: Theory of Recursive Functions and Effective Computability. MIT–Press, 1967. 3rd printing 1992. ▪ Michael R. Garey and David S. Johnson: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman & Co., 1979. ▪ K. W. Wagner: Theoretische Informatik – Eine kompakte Einführung. Springer, 2nd edition, 2003.		
Studienleistung	☒ Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation)		
	☐ Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation)		
	☐ Bestehen von Leistungsstandkontrollen		
Prüfungsform	☒ Mündliche Prüfung (nur bei geringer Teilnehmerzahl)		
	☒ Klausur		
	☐ Prüfung am PC		
	☐ Hausarbeit (ggf. mit Präsentation)		
	☐ Projekt (ggf. mit Präsentation)		
Verwendbarkeit	Künstliche Intelligenz und Data Science		☐ PF ☒ WPF
Angebot	☐ Sommersemester ☒ Wintersemester ☐ Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	6	60 Stunden	120 Stunden
Lehrende(r)	Prof. Dr. H. Schmitz		
Modulverantwortliche(r)	Prof. Dr. H. Schmitz		
Änderungsdatum	22.11.2022		

Data Science			

Inhalte	▪ Grundlagen und Anwendungsfälle für Data Science
	▪ Daten
	  ▪ typische Datenquellen im Unternehmen
	  ▪ Arten von Daten: Stammdaten, Bewegungsdaten
	  ▪ strukturierte, unstrukturierte und semistrukturierte Daten
	  ▪ Netzwerkdaten/Graphen
	  ▪ Werkzeuge zur Datenanalyse und -vorverarbeitung
	  ▪ Datenqualität
	▪ Datenverarbeitung
	  ▪ Architektur von Datenverarbeitungsstrecken
	  ▪ Umgang mit großen Datenmengen
	  ▪ Datenintegration
	  ▪ Data Ingestion
	  ▪ Präsentationsschicht
	▪ Business Intelligence
	  ▪ Grundlagen Data Warehousing
	  ▪ Reporting
	  ▪ Visualisierung
	▪ Empfehlungssysteme
	▪ Social Network Analysis		

Lernergebnisse	Die Studierenden sind nach der Teilnahme in der Lage ▪ vorhandene Datenbestände in Unternehmen zu bewerten, zu verarbeiten und zu nutzen, ▪ die Qualität und Struktur von Daten einzuschätzen, ▪ Daten aus verschiedenen Quellen aufzubereiten und zu integrieren, ▪ Daten und daraus abgeleitetes Wissen für Fachanwender zu präsentieren, ▪ gängige Werkzeuge wie Kafka, Camel, PDI, Pandas, Jupyter einzusetzen und ▪ Wissen und Empfehlungen aus sozialen Netzwerken zu generieren. Die Vorlesung ist im Zusammenspiel mit der Veranstaltung „Maschinelles Lernen“ konzipiert, in der konkrete Lernverfahren vermittelt werden.		
Lehrform	☒ Vorlesung		
	☒ Übung		
	☐ Seminar/Seminaristischer Unterricht		
	☐ Labor		
	☐ Projekt		
Empfohlene Voraussetzungen	Kompetenzen gemäß der Lernergebnisse der Bachelor-Module „Datenbanken“ und „Big-Data-Technologien“		
Literatur	▪ Cady, F.: The Data Science Handbook. Wiley, 2017. ▪ O'Neil, C., Schutt, R.: Doing Data Science. O'Reilly, 2014. ▪ Kotu, V., Deshpande, B.: Data Science: Concepts and Practice. 2. Auflage, Morgan Kaufmann, 2019. ▪ Barabasi, A.-L., Posfai, M.: Network Science. Cambridge University Press, 2016. ▪ Aggarwal, C. C.: Recommender Systems: The Textbook. Springer, 2016. ▪ Kimball, R., Ross, M.: The Data Warehouse Toolkit. 3. Auflage, Wiley, 2013.		
Studienleistung	☒ Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation)		
	☐ Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation)		
	☐ Bestehen von Leistungsstandkontrollen		
Prüfungsform	☒ Mündliche Prüfung (nur bei geringer Teilnehmerzahl)		
	☒ Klausur		
	☐ Prüfung am PC		
	☐ Hausarbeit (ggf. mit Präsentation)		
	☐ Projekt (ggf. mit Präsentation)		
Verwendbarkeit	Künstliche Intelligenz und Data Science		☒ PF ☐ WPF
Angebot	☐ Sommersemester ☒ Wintersemester ☐ Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	6	60 Stunden	120 Stunden
Lehrende(r)	Prof. Dr. C. Schmitz		
Modulverantwortliche(r)	Prof. Dr. C. Schmitz		
Änderungsdatum	22.11.2022		

Data Warehouse			
Inhalte	▪ Motivation zum Einsatz von Data Warehouses (DWH) ▪ Einbettung von DWH in typische IT-Landschaften und -Prozesse ▪ SQL und PL/SQL ▪ Dimensionale Modellierung von DWH nach Kimball ▪ Implementierung von DWH in relationalen Datenbanksystemen ▪ OLAP und analytische Operatoren in SQL ▪ ETL-Prozesse (Extract-Transform-Load) und ETL-Tools ▪ Semistrukturierte Daten in XML und JSON		
Lernergebnisse	Die Studierenden können ▪ die Motivation für den Einsatz von DWH und typische Einsatzszenarien für DWH in IT-Landschaften erklären, ▪ geeignete Granularität, Dimensionen, Fakten und Metriken für gegebene Anwendungsszenarien von DWH auswählen, ▪ dimensionale Modelle nach Kimball in einer relationalen Datenbank implementieren, ▪ fortgeschrittene analytische Anfragen gegen relationale Datenbanken in SQL stellen, ▪ Daten mithilfe von PL/SQL, AWK und graphischen Werkzeugen wie z. B. PDI extrahieren und transformieren, sowie ▪ die Unterschiede zwischen semistrukturierten und strukturierten Daten verstehen und deren Verbindung innerhalb einer relationalen Datenbank umsetzen.		
Lehrform	☒ Vorlesung		
	☒ Übung		
	☐ Seminar/Seminaristischer Unterricht		
	☐ Labor		
	☐ Projekt		
Empfohlene Voraussetzungen	Kompetenzen gemäß der Lernergebnisse des Bachelor-Moduls „Datenbanken“		
Literatur	▪ William Inmon: Building the Data Warehouse. 4. Auflage, Wiley, 2005. ▪ Ralph Kimball, Margy Ross: The Data Warehouse Toolkit. 3. Auflage, Wiley, 2013.		
Studienleistung	☒ Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation)		
	☐ Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation)		
	☐ Bestehen von Leistungsstandkontrollen		
Prüfungsform	☒ Mündliche Prüfung (nur bei geringer Teilnehmerzahl)		
	☒ Klausur		
	☐ Prüfung am PC		
	☐ Hausarbeit (ggf. mit Präsentation)		
	☐ Projekt (ggf. mit Präsentation)		
Verwendbarkeit	Künstliche Intelligenz und Data Science		☒ PF ☐ WPF
Angebot	☒ Sommersemester ☐ Wintersemester ☐ Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	6	60 Stunden	120 Stunden
Lehrende(r)	Prof. Dr. C. Schmitz		
Modulverantwortliche(r)	Prof. Dr. C. Schmitz		
Änderungsdatum	04.01.2023		

High Performance Computing			

Inhalte	▪ General paradigms for parallel programming
	▪ Using the Threading Building Blocks Library for CPU parallelism
	  ▪ Concepts of the task stealing scheduler
	  ▪ Efficient memory management for parallel systems
	  ▪ General parallelism concepts in TBB
	▪ Using CUDA for GPGPU computing
	  ▪ Concepts of GPGPU computing
	  ▪ Writing simple CUDA programs
	  ▪ Synchronization in CUDA
	  ▪ Streaming and overlapping in CUDA		

Lernergebnisse	Students should be able to decide which HPC concept to use for a specific task at hand for compute intensive operations. They should be able to implement those in concrete programs.		
Lehrform	☒ Vorlesung		
	☒ Übung		
	☐ Seminar/Seminaristischer Unterricht		
	☐ Labor		
	☐ Projekt		
Empfohlene Voraussetzungen	Competences according to the learning outcome of the bachelor modules „C/C++-Programmierung" (C/C++ Programming) and „Technische Informatik" (Computer Engineering, especially Computer Architecture).		
Literatur	▪ Michael McCool, Arch D. Robinson, James Reinders: Structured Parallel Programming. Morgan Kaufmann, 2012. ▪ James Reinders: Intel Threading Building Blocks. O'Reilly, 2007. ▪ Jason Sanders, Edward Kandrot: CUDA by Example. Addison-Wesley Professional, 2010.		
Studienleistung	☐ Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation)		
	☒ Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation)		
	☐ Bestehen von Leistungsstandkontrollen		
Prüfungsform	☐ Mündliche Prüfung		
	☐ Klausur		
	☐ Prüfung am PC		
	☐ Hausarbeit (ggf. mit Präsentation)		
	☒ Projekt (ggf. mit Präsentation)		
Verwendbarkeit	Künstliche Intelligenz und Data Science		☐ PF ☒ WPF
Angebot	☐ Sommersemester ☐ Wintersemester ☒ Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	6	60 Stunden	120 Stunden
Lehrende(r)	Prof. Dr. C. Lürig		
Modulverantwortliche(r)	Prof. Dr. C. Lürig		
Änderungsdatum	22.11.2022		

Informationssicherheit			

Inhalte	▪ Einführung: Grundlegende Begriffe und Zusammenhänge
	▪ Kryptologie
	  ▪ Quanten-resistente kryptologische Verfahren wie CRYSTALS Kyber
	  ▪ Voll homomorphe Verfahren wie z.B. BFV
	▪ Secure AI
	  ▪ Grundlagen und Angriffe auf neuronale Netze
	  ▪ Adversarial Attacks
	▪ Organisatorische, konzeptionelle und prozessorientierte Sicherheit
	  ▪ Sicherheitskonzepte gemäß IT-Grundschutz des BSI
	  ▪ Sicherheitsmanagement und ISO 27000
	▪ Patch Management
	▪ Ggf. weitere Themen der Informationssicherheit wie z.B. Rechtliche Aspekte der Informationssicherheit		

Lernergebnisse	Die Studierenden haben einen gesamtheitlichen Einblick in die Informationssicherheit erlernt und verstanden. Dies umfasst neben technischen insbesondere organisatorische und konzeptionelle Aspekte. Zusätzlich werden vertiefend aktuelle Themen der Kryptologie und Angriffe auf neuronale Netze behandelt. Die Studierenden können ▪ strukturierte Vorgehensweisen zur Umsetzung von Informationssicherheit anhand von Standards wie z.B. ISO 27x oder BSI 100 benennen und anwenden, ▪ Risikoanalysen durchführen und berechnen, ▪ die Auswirkungen von Quantencomputern auf die Kryptologie einschätzen, ▪ die Theorie und Funktionsweise der behandelten Post-Quanten-Verfahren erklären, ▪ in kleinen Zahlenbeispielen die notwendigen Operationen der Post-Quanten-Verfahren durchführen und berechnen, ▪ die Taxonomie der Angriffe auf neuronale Netzwerke erklären, ▪ die wichtigsten Schritte der Angriffe auf neuronale Netzwerke in kleinen Beispielen rechnerisch durchführen, ▪ Angriffe auf neuronale Netze per vorgegebenem Code ausführen und auf neue Datensätze übertragen, ▪ die Grundprinzipien und Standards des Patch Managements beschreiben und in Beispielaufgaben anwenden.		
Lehrform	☐ Vorlesung		
	☒ Übung		
	☒ Seminar/Seminaristischer Unterricht		
	☒ Labor (Ausgewählter Vorlesungsstoff und die Übungen werden vertieft durch praktische Übungen.)		
	☐ Projekt		
Empfohlene Voraussetzungen	Besuch einer einführenden Veranstaltung zur IT-Sicherheit im Bachelor-Studium		
Literatur	▪ Ross Anderson, Security Engineering: A Guide to Building Dependable Distributed Systems, John Wiley & Sons ▪ Bundesamt für Sicherheit in der Informationstechnik, Standardreihe 200-x ▪ Claudia Eckert, IT-Sicherheit: Konzepte, Verfahren, Protokolle, Oldenbourg ▪ ISO 27001 und ISO 27002 ▪ Norbert Pohlmann: Cyber-Sicherheit, Springer, 2019 (Als eBook in der Bibliothek vorhanden)		
Studienleistung	☒ Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation)		
	☐ Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation)		
	☐ Bestehen von Leistungsstandkontrollen		
Prüfungsform	☒ Mündliche Prüfung (nur bei geringer Teilnehmerzahl)		
	☒ Klausur		
	☐ Prüfung am PC		
	☐ Hausarbeit (ggf. mit Präsentation)		
	☐ Projekt (ggf. mit Präsentation)		
Verwendbarkeit	Künstliche Intelligenz und Data Science		☒ PF ☐ WPF
Angebot	☐ Sommersemester ☒ Wintersemester ☐ Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	6	75 Stunden	105 Stunden
Lehrende(r)	Prof. Dr. K. Knorr		
Modulverantwortliche(r)	Prof. Dr. K. Knorr		

Änderungsdatum	29.06.2026

Interactive Physical Simulation			

Inhalte	▪ Collision detection
	  ▪ Coarse granular collision detection (sweep and prune, BSP trees, Octrees)
	  ▪ Fine granular collision detection (primitive collision, hierarchical OBBS, Minkowski sums, k-dops)
	  ▪ Tunnel Problem and Solutions like conservative advancement and retroactive detection
	▪ physical properties of moving rigid bodies (force, torque, acceleration, momentum, impulse, velocity...), Newton mechanic
	▪ numerical Integration of ODEs, L- and A- stability, implicit semi implicit and explicit solvers.
	▪ Collision reaction
	  ▪ Collision reaction between two rigid bodies with one point of contact
	  ▪ Multiple rigid bodies and multiple points of contact
	  ▪ Impulse base Simulation methods
	▪ Contraint based Simulation methods
	  ▪ Early methods like penalty and verlet integration
	  ▪ Lagrange Dynamics for contrained motion
	  ▪ linear complimentary problems
	  ▪ Simulation of breaking and tearing for games with lagrange dynamics
	  ▪ Coupling of Simulation and Animation for articulated characters		

Lernergebnisse	Students should be able to understand the techniques used in physical simulators for games, apply the proper technique for the specific situation at hand and implement portions of the system themselves.		
Lehrform	☒ Vorlesung		
	☒ Übung		
	☐ Seminar/Seminaristischer Unterricht		
	☐ Labor		
	☐ Projekt		
Empfohlene Voraussetzungen	Competences according to the learning outcome of the bachelor modules „Lineare Algebra" (Linear Algebra) and „Angewandte Mathematik" (Applied Mathematics).		
Literatur	▪ H. R. Schwarz: Numerische Mathematik. Springer Vieweg, 2011. ▪ David H. Eberly: Game Physics. Taylor & Francis, 2. Auflage, 2010. ▪ Gino van den Bergen: Collision Detection in Interactive 3D Environments. CRC Press, 2003.		
Studienleistung	☒ Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation)		
	☐ Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation)		
	☐ Bestehen von Leistungsstandkontrollen		
Prüfungsform	☒ Mündliche Prüfung		
	☐ Klausur		
	☐ Prüfung am PC		
	☐ Hausarbeit (ggf. mit Präsentation)		
	☐ Projekt (ggf. mit Präsentation)		
Verwendbarkeit	Künstliche Intelligenz und Data Science		☐ PF ☒ WPF
Angebot	☐ Sommersemester ☐ Wintersemester ☒ Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	6	60 Stunden	120 Stunden
Lehrende(r)	Prof. Dr. C. Lürig		
Modulverantwortliche(r)	Prof. Dr. C. Lürig		
Änderungsdatum	22.11.2022		

Kooperative Systeme			
Inhalte	▪ Soziale und technische Aspekte der Kooperation ▪ Klassifikation von Groupware-Lösungen ▪ Gestaltung von Groupware-Lösungen ▪ Awareness, Kommunikations- und Koordinationsunterstützung ▪ Cognitive Systems Engineering ▪ Mensch-Maschine-Kooperation ▪ Geteilte Kontrolle und Entscheidungsunterstützung ▪ Symbolische und subsymbolische Ansätze künstlicher Intelligenz ▪ Intelligent User Interfaces ▪ Mensch-Maschine-Integration		
Lernergebnisse	Die Studierenden können ▪ Grundlagen, Prinzipien und Anwendungsmöglichkeiten computergestützter kooperativer Arbeit erläutern, ▪ die wichtigsten Theorien und Modelle der Mensch-Maschine-Kooperation beschreiben, und ▪ kooperative Systeme analysieren, konzipieren, realisieren und evaluieren.		
Lehrform	☒ Vorlesung		
	☒ Übung		
	☐ Seminar/Seminaristischer Unterricht		
	☐ Labor		
	☐ Projekt		
Empfohlene Voraussetzungen	Keine		
Literatur	▪ T. Gross & M. Koch: Computer-Supported Cooperative Work. Oldenbourg, 2007. ▪ D. Coleman: Groupware. Prentice-Hall, 1997. ▪ G. Schwabe et al. (Hrsg.): CSCW-Kompendium. Springer, 2001. ▪ M. Beaudouin-Lafon (Hrsg.): Computer-Supported Cooperative Work. Wiley, 1998. ▪ C. Reuter: Sicherheitskritische Mensch-Computer-Interaktion. Interaktive Technologien und Soziale Medien im Krisen- und Sicherheitsmanagement. 2018		
Studienleistung	☒ Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation)		
	☐ Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation)		
	☐ Bestehen von Leistungsstandkontrollen		
Prüfungsform	☒ Mündliche Prüfung (nur bei geringer Teilnehmerzahl)		
	☒ Klausur		
	☐ Prüfung am PC		
	☐ Hausarbeit (ggf. mit Präsentation)		
	☐ Projekt (ggf. mit Präsentation)		
Verwendbarkeit	Künstliche Intelligenz und Data Science		☒ PF ☐ WPF
Angebot	☐ Sommersemester ☒ Wintersemester ☐ Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	6	60 Stunden	120 Stunden
Lehrende(r)	Prof. Dr. T. Mentler		
Modulverantwortliche(r)	Prof. Dr. T. Mentler		
Änderungsdatum	22.11.2022		

Künstliche Intelligenz für Roboter			
Inhalte	▪ Entscheidungsfindung und Planung für intelligente Roboter und Kognitive Systeme ▪ Probabilistisches Schlussfolgern ▪ Inferenz ▪ Parameterschätzung ▪ Bayessche Netze und Lernen von Strukturen ▪ Einfache und komplexe Entscheidungen ▪ Sequenzielle Probleme ▪ Entscheidungsfindung unter Unsicherheit ▪ Modellbasierte und Modellfreie Methoden ▪ Verhalten bei Zustandsunsicherheit ▪ Multi-Agenten-Systeme und Kollaboration		
Lernergebnisse	Die Studierenden können ▪ stochastische Modelle erstellen und implementieren, ▪ verstehen, was Bayessche Netze sind und diese auf verschiedene Probleme anwenden, ▪ Parameterschätzung nach Bayes anwenden und Explorationsstrategien herleiten, ▪ MAP-Schätzer verstehen und implementieren, ▪ MDPs und POMDPs anwenden, ▪ Q-Learning verstehen und anwenden, ▪ Dynamische Programme formulieren, ▪ Suchstrategien einordnen und effizient einsetzen, ▪ lokale Optimierungsstrategien anwenden und um Heuristiken ergänzen, sowie ▪ ggf. Planungsmethoden theoretisch verstehen und praktisch anwenden.		
Lehrform	☒ Vorlesung		
	☒ Übung		
	☐ Seminar/Seminaristischer Unterricht		
	☐ Labor		
	☐ Projekt		
Empfohlene Voraussetzungen	Grundlagenkenntnisse in Stochastik und Maschinellem Lernen sind von Vorteil, werden aber nicht vorausgesetzt.		
Literatur	▪ S. M. LaValle: Planning Algorithms. Cambridge University Press, 2006. ▪ M. J. Kochenderfer et al.: Algorithms for Decision Making. MIT Press, 2022. ▪ R. Neapolitan: Learning Bayesian Networks. Prentice Hall, 2003. ▪ S. Russell and P. Norvig: Artificial Intelligence: A Modern Approach. Pearson, 4. Auflage (2020).		
Studienleistung	☒ Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation)		
	☒ Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation)		
	☒ Bestehen von Leistungsstandkontrollen		
Prüfungsform	☒ Mündliche Prüfung (nur bei geringer Teilnehmerzahl)		
	☒ Klausur		
	☐ Prüfung am PC		
	☐ Hausarbeit (ggf. mit Präsentation)		
	☐ Projekt (ggf. mit Präsentation)		
Verwendbarkeit	Künstliche Intelligenz und Data Science	☐ PF ☒ WPF	
Angebot	☒ Sommersemester ☐ Wintersemester ☐ Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	6	60 Stunden	120 Stunden
Lehrende(r)	Prof. Dr. J. Graf		
Modulverantwortliche(r)	Prof. Dr. J. Graf		
Änderungsdatum	25.01.2023		

Künstliche neuronale Netze			
Inhalte	Künstliche neuronale Netze (KNN) sind eine zentrale Klasse von Funktionsmodellen, die in sämtlichen Anwendungsgebieten des maschinellen Lernens eingesetzt werden. In diesem Modul werden, ausgehend von den Anfängen der KNN, der Aufbau, das Trainieren und zentrale Architekturen eingeführt. Dabei werden auch wesentliche Punkte der konkreten Auswertung und Analyse besprochen. Weiterhin werden einige Aspekte der Theorie zu KNN insbesondere mit Bezug zu dem Gebiet der *Learning Theory* besprochen. Folgende Inhalte werden in diesem Modul behandelt: ▪ Architekturen: MLP, CNN, RNN, LSTM, Autoencoder, Attention, Transformer ▪ Algorithmen: Perzeptron-Algorithmus, Backpropagation, SGD und Varianten ▪ Theorie KNN: Approximationseigenschaften, VC-Dimension und Generalisierung von KNN		
Lernergebnisse	Studierende erlernen tiefergehende Konzepte, Methoden und Techniken im Bereich künstlicher neuronaler Netze, zur Durchführung anspruchsvoller Praxisprojekte in Anwendungsgebieten des maschinellen Lernens und Data Science. Studierende können sich selbstständig anhand von Primärliteratur, insbesondere in englischer Sprache, weitergehende wissenschaftliche Grundlagen erarbeiten und aneignen. Dabei erlernen die Studierenden auch sicherheitsrelevante und ethische Problemstellungen im Umgang mit KI-Methoden zu erkennen und kritisch einzuschätzen. Studierende können komplexe Problemstellungen analysieren und strukturieren, eigene Ideen zur Lösung entwickeln und die Lösung durch die Auswahl und Anwendung geeigneter Methoden erarbeiten.		
Lehrform	☒ Vorlesung		
	☐ Übung		
	☒ Seminar/Seminaristischer Unterricht		
	☐ Labor		
	☐ Projekt		
Empfohlene Voraussetzungen	Keine		
Literatur	▪ I. Goodfellow et al.: Deep Learning. MIT Press, 2016. ▪ G. Strang: Linear Algebra and Learning from Data. Wellesley-Cambridge Press, 2019.		
Studienleistung	☐ Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation)		
	☒ Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation)		
	☐ Bestehen von Leistungsstandkontrollen		
Prüfungsform	☒ Mündliche Prüfung (nur bei geringer Teilnehmerzahl)		
	☒ Klausur		
	☐ Prüfung am PC		
	☐ Hausarbeit (ggf. mit Präsentation)		
	☐ Projekt (ggf. mit Präsentation)		
Verwendbarkeit	Künstliche Intelligenz und Data Science		☒ PF ☐ WPF
Angebot	☒ Sommersemester ☐ Wintersemester ☐ Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	6	60 Stunden	120 Stunden
Lehrende(r)	Prof. Dr. H.-P. Beise		
Modulverantwortliche(r)	Prof. Dr. H.-P. Beise		
Änderungsdatum	23.01.2023		

Learning Analytics and Educational Data Mining			
Inhalte	Das Modul präsentiert und diskutiert die Analyse von Daten aus Lehr- und Lernsystemen als vielseitiges Anwendungsfeld von Data Science und KI-Methoden. Die Lerninhalte umfassen sowohl theoretische und praktische Grundlagen (z. B. Modelle, Architekturen, Rechtliche und Ethische Aspekte) als auch konkrete Anwendungsbeispiele durch die intensive Auseinandersetzung mit ausgewählten Verfahren und Algorithmen anhand realer Datensätze.		
Lernergebnisse	Die Studierenden können ▪ Konzepte und Anwendungsfelder von Learning Analytics und Educational Data Mining beschreiben, ▪ das „Learning Analytics Reference Model" sowie gängige Datenmodelle und Architekturen erklären, ▪ Verfahren und Algorithmen passend zu einem gegebenen Datensatz und einem gegebenen Analyseziel auswählen, ▪ ausgewählte Verfahren und Algorithmen auf konkrete Datensätze anwenden und die dadurch gewonnenen Erkenntnisse erklären, bewerten, präsentieren und diskutieren, ▪ Visualisierungen und Dashboards zur Unterstützung von Learning Analytics und Educational Data Mining entwerfen, ▪ Verfahren und Datensätze hinsichtlich rechtlicher und ethischer Bedenken einschätzen.		
Lehrform	☒ Vorlesung		
	☒ Übung		
	☐ Seminar/Seminaristischer Unterricht		
	☐ Labor		
	☐ Projekt		
Empfohlene Voraussetzungen	Kompetenzen gemäß der Lernergebnisse der Master-Module „Statistics and Learning Theory", „Data Science" und „Künstliche neuronale Netze"		
Literatur	▪ Lang, Charles, et al., eds. "Handbook of learning analytics." (2017). ▪ Clow, Doug. "An overview of learning analytics." Teaching in Higher Education 18.6 (2013): 683-695. ▪ Chatti, Mohamed Amine, et al. "A reference model for learning analytics." International Journal of Technology-Enhanced Learning 4.5-6 (2012): 318-331. ▪ Romero, Cristobal, and Sebastian Ventura. "Educational data mining and learning analytics: An updated survey." Wiley interdisciplinary reviews: Data mining and knowledge discovery 10.3 (2020): e1355.		
Studienleistung	☐ Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation)		
	☒ Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation)		
	☐ Bestehen von Leistungsstandkontrollen		
Prüfungsform	☐ Mündliche Prüfung		
	☐ Klausur		
	☐ Prüfung am PC		
	☐ Hausarbeit (ggf. mit Präsentation)		
	☒ Projekt (ggf. mit Präsentation)		
Verwendbarkeit	Künstliche Intelligenz und Data Science		☐ PF ☒ WPF
Angebot	☒ Sommersemester ☐ Wintersemester ☐ Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	6	60 Stunden	120 Stunden
Lehrende(r)	Prof. Dr. M. Striewe		
Modulverantwortliche(r)	Prof. Dr. M. Striewe		
Änderungsdatum	09.10.2025		

Lineare Optimierung			
Inhalte	▪ Lineare Programme ▪ Beispielprobleme und deren Modellierung, Standardform, Slackform ▪ Simplexverfahren in Grundform, Nachweis der Korrektheit, Eindeutigkeit optimaler Lösungen, Initialisierung, Hauptsatz der Linearen Programmierung, ökonomische Interpretation der optimalen Slackform ▪ LP-Dualität, schwache und starke Dualität, Satz vom komplementären Schlupf, Interpretation der Dualisierung in ökonomischen Modellen ▪ Revidiertes Simplexverfahren, temporäre Schattenpreise, Duales Simplexverfahren ▪ Geometrie Linearer Programme, konvexe Polyeder ▪ Einsatz von LP-Solvern		
Lernergebnisse	Die Studierenden können ▪ Berechnungsprobleme als lineare Programme modellieren, insbesondere Strukturvariablen identifizieren und Bedingungen mathematisch formulieren, ▪ die behandelten Verfahren reproduzieren und deren Korrektheit und Laufzeit erklären, ▪ Lineare Programme mit den behandelten Verfahren lösen und die erzielten Lösungen interpretieren, ▪ das Konzept der LP-Dualität wiedergeben und die zugrundeliegende Theorie begründen sowie ▪ den Einsatz von LP-Solvern am Beispiel praktisch umsetzen.		
Lehrform	☒ Vorlesung		
	☒ Übung		
	☐ Seminar/Seminaristischer Unterricht		
	☐ Labor		
	☐ Projekt		
Empfohlene Voraussetzungen	Grundkenntnisse der Python-Programmierung, Grundkenntnisse der linearen Algebra		
Literatur	▪ Vasek Chvatal: Linear programming. A Series of Books in the Mathematical Sciences, 1983. ▪ Dimitris Bertsimas and John N. Tsitsiklis: Introduction to Linear Optimization. January 1997.		
Studienleistung	☒ Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation)		
	☐ Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation)		
	☐ Bestehen von Leistungsstandkontrollen		
Prüfungsform	☒ Mündliche Prüfung (nur bei geringer Teilnehmerzahl)		
	☒ Klausur		
	☐ Prüfung am PC		
	☐ Hausarbeit (ggf. mit Präsentation)		
	☐ Projekt (ggf. mit Präsentation)		
Verwendbarkeit	Künstliche Intelligenz und Data Science		☒ PF ☐ WPF
Angebot	☒ Sommersemester ☐ Wintersemester ☐ Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	6	60 Stunden	120 Stunden
Lehrende(r)	Prof. Dr. H. Schmitz		
Modulverantwortliche(r)	Prof. Dr. H. Schmitz		
Änderungsdatum	22.11.2022		

Maschinelles Lernen			
Inhalte	Die Methoden des maschinellen Lernens stellen die Grundlage für die aktuelle Forschung und Entwicklungen im Bereich intelligenter Systeme dar und sind integraler Bestandteile einer großen Anzahl gegenwärtiger, industrieller Anwendungen. Hierbei werden Modelle anhand vorliegender Daten trainiert, um spezifische Aufgabenstellung zu lösen. ▪ Einfürung und grundlegende Konzepte ▪ Datenvorverarbeitung / Merkmalsextraktion (Zeit- und Frequenzbereich) / Dimensionsreduktion ▪ Hauptachsentransformation (PCA) / Whitening ▪ Sammon-Transformation ▪ Multidimensional Skalierung (MDS), T-SNE ▪ Entscheidungsbäume ▪ Bayes Klassifikator ▪ k-means, k-Nearest Neighour, Kerndichteschätzer ▪ Linear und nicht linear separierbare Probleme ▪ Quadratische Fehlerminimierung und Gradientenabstiegsverfahren ▪ Fischer lineare Diskriminanzanalyse (LDA) ▪ Linearer Klassifikator und generalisierte lineare Diskriminanzfunktionen ▪ Lineare und multiple lineare Regression ▪ Support Vector Machines ▪ Dynamic Time Warping ▪ Hidden-Markov-Modell ▪ Evaluierung von Lernverfahren ▪ Grundlagen neuronaler Netze		
Lernergebnisse	Die Teilnehmer der Lehrveranstaltung kennen den aktuellen Forschungsstand im Bereich Maschinelles Lernen. Sie sind fähig, aktuelle Entwicklungen zu verstehen, zu analysieren und kritisch zu reflektieren.		
Lehrform	☒ Vorlesung		
	☒ Übung		
	☐ Seminar/Seminaristischer Unterricht		
	☐ Labor		
	☐ Projekt		
Empfohlene Voraussetzungen	Keine		
Literatur	▪ Duda, Richard O.: Pattern Classification: Pattern Classification. Wiley-Interscience. ▪ Peter Flach: Machine Learning: The Art and Science of Algorithms that Make Sense of Data. Cambridge University Press. ▪ Ferdinand van der Heijden, de Ridder, Dick, Tax, David M. J.: Classification, Parameter Estimation and State Estimation: An Engineering Approach Using MATLAB. Wiley. ▪ Bishop, Christopher M.: Neural Networks for Pattern Recognition. Oxford University Press.		
Studienleistung	☐ Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation)		
	☒ Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation)		
	☐ Bestehen von Leistungsstandkontrollen		
Prüfungsform	☒ Mündliche Prüfung		
	☐ Klausur		
	☐ Prüfung am PC		
	☒ Hausarbeit (ggf. mit Präsentation)		
	☐ Projekt (ggf. mit Präsentation)		
Verwendbarkeit	Künstliche Intelligenz und Data Science		☒ PF ☐ WPF
Angebot	☐ Sommersemester ☒ Wintersemester ☐ Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	6	60 Stunden	120 Stunden
Lehrende(r)	Prof. Dr. J. Lohscheller		
Modulverantwortliche(r)	Prof. Dr. J. Lohscheller		
Änderungsdatum	12.05.2023		

Mensch-Computer-Interaktion			
Inhalte	▪ Grundlegende Begriffe der Mensch-Computer-Interaktion: Usability, User Experience, Accessibility ▪ Menschliche Informationsverarbeitung und Handlungsprozesse ▪ Ein-/Ausgabegeräte und Interaktionstechnologien ▪ Menschzentrierte Entwicklungsprozesse: ISO 9241-210, Contextual Design, Scenario-Based Design ▪ Usability und Software Engineering ▪ Methoden zur Analyse von Nutzungskontexten: Hierarchische Aufgabenanalyse, Personas, Prozessanalyse ▪ Low- und High-Fidelity-Prototyping ▪ Formative und summative Evaluationen ▪ Modelle für Mensch-Computer-Systeme ▪ Interaktion und Kooperation ▪ Normen und rechtliche Grundlagen		
Lernergebnisse	Die Studierenden können ▪ Prinzipien und Methoden menschzentrierter Entwicklung interaktiver Systeme erläutern. ▪ Kenntnisse über die menschliche Informationsverarbeitung in die Entwicklung interaktiver Systeme einbringen. ▪ Modelle und Kriterien zur Analyse und Bewertung interaktiver Systeme anwenden.		
Lehrform	☒ Vorlesung		
	☒ Übung		
	☐ Seminar/Seminaristischer Unterricht		
	☐ Labor		
	☐ Projekt		
Empfohlene Voraussetzungen	Keine		
Literatur	▪ Benyon, D. (2014). Designing interactive systems: A comprehensive guide to HCI, UX and interaction design (Third edition). Harlow, England: Pearson. ▪ Dahm, M. (2006). Grundlagen der Mensch-Computer-Interaktion. Informatik: Software-Ergonomie. München: Pearson Studium. ▪ Herczeg, M. (2018). Software-Ergonomie: Theorien, Modelle und Kriterien für gebrauchstaugliche interaktive Computersysteme: De Gruyter. ▪ Reuter, C. (Hrsg.) (2018). Sicherheitskritische Mensch-Computer-Interaktion. Wiesbaden: Springer Fachmedien Wiesbaden.		
Studienleistung	☒ Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation)		
	☐ Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation)		
	☐ Bestehen von Leistungsstandkontrollen		
Prüfungsform	☐ Mündliche Prüfung		
	☐ Klausur		
	☐ Prüfung am PC		
	☐ Hausarbeit (ggf. mit Präsentation)		
	☒ Projekt (ggf. mit Präsentation)		
Verwendbarkeit	Künstliche Intelligenz und Data Science		☐ PF ☒ WPF
Angebot	☐ Sommersemester ☐ Wintersemester ☒ Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	6	60 Stunden	120 Stunden
Lehrende(r)	Prof. Dr. T. Mentler		
Modulverantwortliche(r)	Prof. Dr. T. Mentler		
Änderungsdatum	23.03.2026		

Mobile Robotik			
Inhalte	▪ Koordinatentransformationen für die Lokalisierung ▪ Inertialnavigation ▪ Kalibrierung von Beschleunigungssensoren und Drehratensensoren ▪ Technologie und Kalibrierung ▪ Algorithmen für Strap Down-Verfahren ▪ Satellitennavigation (Signale und Systeme) ▪ Erweiterungen (bspw. RTK) ▪ Datenfusion von INS/GNSS (loosely- und tightly-coupled-Architekturen) in Theorie (Error State) und Praxis (Simulationsumgebung) ▪ Visuelle Navigation am Beispiel von Visueller Odometrie (VO) ▪ Möglichkeiten zur Integration von Inertialen VO und INS/GNSS/VO		
Lernergebnisse	Die Studierenden können ▪ geodätische Transformationen korrekt anwenden, ▪ Fehlermodelle für die Kalibrierung von Inertialen Systemen formulieren, ▪ Strap Down-Verfahren implementieren, ▪ Satellitennavigation auf Komponentenebene erklären, ▪ Lokalisierung mit Satelliten in Theorie und Praxis anwenden, ▪ Datenfusionsarchitekturen implementieren (loosely und tightly-coupled), ▪ die Grundlagen von Visueller Odometrie verstehen und implementieren, und ▪ INS/GNSS/VO-Architekturen entwerfen und diskutieren.		
Lehrform	☒ Vorlesung		
	☒ Übung		
	☐ Seminar/Seminaristischer Unterricht		
	☐ Labor		
	☐ Projekt		
Empfohlene Voraussetzungen	Kompetenzen gemäß der Lernergebnisse der Bachelor-Module „Einführung in die Robotik“, „Robotersehen“ und „Kognitive Sichtsysteme“		
Literatur	▪ R. Siegwart et al: Autonomous Mobile Robotics. Springer, 2. Auflage (2011). ▪ J. Hertzberg et al: Mobile Robotik. Springer, 2012. ▪ Groves: Navigation Systems. Artech House, 2013. ▪ Wendel: Integrierte Navigationssysteme. Oldenbourg, 2011. ▪ Jekeli: Inertial Navigation Systems with Geodetic Applications. De Gruyter, 2000. ▪ R. M. Rogers: Applied Mathematics in INS. Transatlantic Publishers, 2007. ▪ Titterton and Weston: Strapdown Inertial Navigation Technology. IET, 2. Auflage (2005). ▪ J. Graf: Vorlesungsunterlagen zum Modul Visuelle Navigation		
Studienleistung	☒ Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation)		
	☒ Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation)		
	☒ Bestehen von Leistungsstandkontrollen		
Prüfungsform	☒ Mündliche Prüfung (nur bei geringer Teilnehmerzahl)		
	☒ Klausur		
	☐ Prüfung am PC		
	☐ Hausarbeit (ggf. mit Präsentation)		
	☐ Projekt (ggf. mit Präsentation)		
Verwendbarkeit	Künstliche Intelligenz und Data Science		☐ PF ☒ WPF
Angebot	☒ Sommersemester ☐ Wintersemester ☐ Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	6	60 Stunden	120 Stunden
Lehrende(r)	Prof. Dr. J. Graf		
Modulverantwortliche(r)	Prof. Dr. J. Graf		
Änderungsdatum	25.01.2023		

Projektstudium			
Inhalte	Das Projektstudium umfasst die Bearbeitung einer qualifizierten Aufgabenstellung aus der Praxis unter intensiver Betreuung durch einen Dozenten. Der Schwierigkeitsgrad entspricht der späteren Berufspraxis eines Master-Absolventen. Dabei werden systematische Vorgehensweisen und sinnvolle Arbeitstechniken sowie die Verbindung zu Anwendungsgebieten der Informatik hergestellt. Die Studierenden präsentieren ihre Ergebnisse in einer Ausarbeitung unter Anwendung der Kriterien wissenschaftlichen Arbeitens.		
Lernergebnisse	Die Studierenden haben durch die erfolgreiche Bearbeitung gezeigt, dass sie in der Lage sind, innerhalb einer vorgegebenen Frist sich selbständig in ein Anwendungsgebiet der Informatik einzuarbeiten und ein praktisches Problem zu bearbeiten. Sie verfügen über ein breites und integriertes Wissen sowie über ein kritisches Verständnis der wichtigsten Theorien und Methoden. Sie sind in der Lage, die im Studium erworbenen Kenntnisse, Fähigkeiten, Methoden und Werkzeuge auf vorgegebene Fragestellungen anzuwenden. Sie leiten auf dieser Basis fundierte Lösungsansätze ab und formulieren eine dem Stand der Technik entsprechende Lösung für das praktische Problem. Sie verstehen die Erfolgsfaktoren und beherrschen Methoden der Definition, Planung sowie Kontrolle von Projekten. Daher können sie komplexe Aufgaben sinnvoll strukturieren und typische Schnittstellenprobleme sowohl auf technisch-fachlicher als auch auf sozialer Ebene bewältigen. Sie können Arbeitsergebnisse professionell und unter Anwendung der Kriterien des wissenschaftlichen Arbeitens dokumentieren.		
Lehrform	☐ Vorlesung		
	☐ Übung		
	☐ Seminar/Seminaristischer Unterricht		
	☐ Labor		
	☒ Projekt		
Empfohlene Voraussetzungen	Abhängig von der Aufgabenstellung; wird vom Betreuer festgelegt		
Literatur	Helmut Balzert, Marion Schröder, Christian Schäfer: Wissenschaftliches Arbeiten. W3L GmbH.		
Studienleistung	☐ Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation)		
	☐ Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation)		
	☐ Bestehen von Leistungsstandkontrollen		
Prüfungsform	☐ Mündliche Prüfung		
	☐ Klausur		
	☐ Prüfung am PC		
	☐ Hausarbeit (ggf. mit Präsentation)		
	☒ Projekt (ggf. mit Präsentation)		
Verwendbarkeit	Künstliche Intelligenz und Data Science		☒ PF ☐ WPF
Angebot	☒ Sommersemester ☒ Wintersemester ☐ Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	15	30 Stunden	420 Stunden
Lehrende(r)	Dozenten des Fachbereichs Informatik		
Modulverantwortliche(r)	Fachrichtungsleiter Informatik		
Änderungsdatum	13.02.2023		

Safety KI-basierter Cyber Physical Systems			
Inhalte	▪ Definition Cyber Physical Systems (CPS) ▪ Terminologie zu Safety, Dependability, Fault-Tolerance, Real-Time ▪ Beispiele KI-basierter Cyber Physical Systems, insbesondere aus den Bereichen Advanced Driver Assistance Systems (ADAS) und Automated Driving Systems (ADS) ▪ Zusammenhang Safety und Security ▪ Herausforderung Safety und KI ▪ Gesellschaftliche Bedeutung und ethische Fragestellungen ▪ Safety of the Intended Function (SOTIF) vs Functional Safety (FuSa) ▪ Einschlägige Standards und Normen (z.B. SAE J 3016, ISO 26262, ISO/DIS 21448) ▪ Übungsprojekt am Forschungsgroßgerät dynamischer Fahrsimulator FaSiMo: KI für ADAS bzw. ADS entwickeln gemäß SOTIF		
Lernergebnisse	Die Studierenden können die besonderen Herausforderungen bei der Entwicklung sicherheitsrelevanter KI-basierter Cyber Physical Systems erläutern. Sie können erklären, welche Rolle unterschiedliche Sicherheitsnormen bei der Entwicklung KI-basierter Lösungen für das automatisierte Fahren spielen und was der Unterschied zwischen SOTIF und FuSa ist. Sie kennen die grundlegenden Schritte, die bei der Entwicklung von Deep Learning/KI-Lösungen erforderlich sind, wissen um die Limitierungen solcher Systeme hinsichtlich Safety und kennen die standardisierten Herangehensweisen gemäß SOTIF. Sie sind in der Lage in interdisziplinären Teams unter Verwendung der einschlägigen Fachbegriffe Lösungsansätze für sicherheitsrelevante KI-basierte CPS gemäß dem SOTIF Prozess zu diskutieren, analysieren, entwerfen und zu gestalten.		
Lehrform	☐ Vorlesung		
	☒ Übung		
	☒ Seminar/Seminaristischer Unterricht		
	☐ Labor		
	☐ Projekt		
Empfohlene Voraussetzungen	Gute Kenntnisse der englischen Sprache; Programmierkenntnisse (z.B. in Python oder C/C++)		
Literatur	▪ ISO 21448 (wird in der Veranstaltung bereitgestellt) ▪ Algirdas Avižienis, Jean-Claude Laprie, Brian Randell and Carl Landwehr: Basic Concepts and Taxonomy of Dependable and Secure Computing. IEEE Transactions on Dependable and Secure Computing, 2004 & 2013.		
Studienleistung	☐ Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation)		
	☒ Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation)		
	☐ Bestehen von Leistungsstandkontrollen		
Prüfungsform	☐ Mündliche Prüfung		
	☐ Klausur		
	☐ Prüfung am PC		
	☐ Hausarbeit (ggf. mit Präsentation)		
	☒ Projekt (ggf. mit Präsentation)		
Verwendbarkeit	Künstliche Intelligenz und Data Science		☐ PF ☒ WPF
Angebot	☐ Sommersemester ☒ Wintersemester ☐ Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	6	60 Stunden	120 Stunden
Lehrende(r)	Prof. Dr. J. Schneider		
Modulverantwortliche(r)	Prof. Dr. J. Schneider		
Änderungsdatum	06.02.2023		

Seminar			
Inhalte	Das Seminar umfasst die selbstständige Erarbeitung eines vorgegebenen begrenzten Themenbereiches anhand von wissenschaftlicher Fachliteratur und anderen Quellen sowie dessen schriftliche und mündliche Darstellung. Es werden wechselnde aktuelle Themen aus der Informatik angeboten, die im Schwierigkeitsgrad für das Master-Studium angemessen sind.		
Lernergebnisse	Die Studierenden verfügen über ein vertieftes Verständnis spezifischer Theorien sowie Ansätze im gewählten Themenbereich und kennen den jeweils aktuellen Stand der Technik. Sie haben eigenständige Lernstrategien erfolgreich angewandt sowie ihre Kommunikations- und Kooperationsfähigkeit in Teams verbessert. Sie kennen wesentliche Merkmale von Präsentationstechniken und haben diese bei der verständlichen Darstellung ihrer Ergebnisse angewendet.		
Lehrform	☐ Vorlesung		
	☐ Übung		
	☒ Seminar/Seminaristischer Unterricht		
	☐ Labor		
	☐ Projekt		
Empfohlene Voraussetzungen	Empfohlene Voraussetzungen abhängig vom Thema des Seminars; werden vom Betreuer festgelegt.		
Literatur	Helmut Balzert, Marion Schröder, Christian Schäfer: Wissenschaftliches Arbeiten. W3L GmbH.		
Studienleistung	☐ Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation)		
	☐ Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation)		
	☐ Bestehen von Leistungsstandkontrollen		
Prüfungsform	☐ Mündliche Prüfung		
	☐ Klausur		
	☐ Prüfung am PC		
	☒ Hausarbeit (ggf. mit Präsentation)		
	☐ Projekt (ggf. mit Präsentation)		
Verwendbarkeit	Künstliche Intelligenz und Data Science		☒ PF ☐ WPF
Angebot	☒ Sommersemester ☒ Wintersemester ☐ Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	3	18 Stunden	72 Stunden
Lehrende(r)	Dozenten des Fachbereichs Informatik		
Modulverantwortliche(r)	Fachrichtungsleiter Informatik		
Änderungsdatum	13.02.2023		

Simulationstechnik und Reinforcement Learning			

Inhalte	▪ Modelling und Simulation
	  ▪ Projektablauf
	  ▪ Diskrete Simulation
	  ▪ Kontinuierliche Simulation
	  ▪ Empirische Modelle
	▪ Reinforcement Learning
	  ▪ Grundlagen
	  ▪ Tabellenverfahren
	  ▪ Approximation mit Neuronalen Netzen
	  ▪ Actor Critic
	  ▪ Alpha Zero		

Lernergebnisse	Die Studierenden können ▪ die Grundkonzepte von Modelling und Simulation sowie Reinforcement Learning erklären, ▪ die vorgestellten Techniken programmieren, sowie ▪ die vorgestellten Techniken in einem praktischen Projekt umsetzen.		
Lehrform	☒ Vorlesung		
	☐ Übung		
	☐ Seminar/Seminaristischer Unterricht		
	☐ Labor		
	☒ Projekt		
Empfohlene Voraussetzungen	▪ Kenntnisse in Python ▪ Kenntnisse in angewandter Mathematik ▪ Grundlagenkenntnisse in Neuronalen Netzen		
Literatur	▪ Louis G. Birta, Gilbert Arbez: Modelling and Simulation, Exploring Dynamic Systems Behaviour, Springer 2019 ▪ Richard S. Sutton, Andrew G.Barto: Reinforcement Learning, MIT Press, 2020 ▪ Maxim Lapan: Deep Reinforcement Learning Hands-On, Packt 2018		
Studienleistung	☐ Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation)		
	☒ Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation)		
	☐ Bestehen von Leistungsstandkontrollen		
Prüfungsform	☐ Mündliche Prüfung		
	☐ Klausur		
	☐ Prüfung am PC		
	☐ Hausarbeit (ggf. mit Präsentation)		
	☒ Projekt (ggf. mit Präsentation)		
Verwendbarkeit	Künstliche Intelligenz und Data Science		☒ PF ☐ WPF
Angebot	☒ Sommersemester ☐ Wintersemester ☐ Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	6	60 Stunden	120 Stunden
Lehrende(r)	Prof. Dr. C. Lürig		
Modulverantwortliche(r)	Prof. Dr. C. Lürig		
Änderungsdatum	03.01.2023		

Software-Architekturen			
Inhalte	Das Modul befasst sich sowohl theoretisch als auch praktisch mit dem Entwurf, der Dokumentation, der Analyse und dem Management von Software-Architekturen. Der erste Teil der Vorlesung führt dazu in grundlegende Begriffe und Konzepte ein, erläutert die Rolle des Software-Architekten in der Software-Entwicklung und stellt Notationen für Software-Architekturen vor. Der zweite Teil der Vorlesung demonstriert Vorgehensweisen, Stile und Muster für die Erstellung, Bewertung und Verwaltung von Software-Architekturen. Die Übung ergänzt beide Teile durch theoretische und praktische Aufgaben. Im dritten Teil des Moduls stellen die Studierenden selbst erarbeitete Inhalte zu vertiefenden Themen in kurzen Referaten vor.		
Lernergebnisse	Die Studierenden können ▪ die Rolle des Software-Architekten in der Software-Entwicklung beschreiben, ▪ grundlegende Begriffe und Konzepte der Software-Architektur beschreiben und gegeneinander abgrenzen, ▪ Techniken zur Notation von Software-Architekturen anwenden, ▪ Verfahren des Architekturentwurfs, der Architekturbewertung und des Managements von Unternehmensarchitekturen beschreiben und anwenden, ▪ Architekturstil und -muster benennen, zueinander in Beziehung setzen und gegeneinander abgrenzen, ▪ ausgewählte Themen der Software-Architektur eigenständig erarbeiten und präsentieren, ▪ für gegebene Szenarien Entwurfsalternativen abwägen, passende Architekturentscheidungen treffen und Architekturen dokumentieren.		
Lehrform	☒ Vorlesung		
	☒ Übung		
	☐ Seminar/Seminaristischer Unterricht		
	☐ Labor		
	☐ Projekt		
Empfohlene Voraussetzungen	Kompetenzen gemäß der Lernergebnisse der Bachelor-Module „Objektorientierte Programmierung - Einführung", „Software-Entwurf" und „Grundlagen des Anforderungsmanagements"		
Literatur	▪ Nick Rozanski, Eoin Woods: Software Systems Architecture. 2. Auflage, Addison-Wesley Professional, 2011. ▪ Len Bass, Paul Clements, Rick Kazman: Software Architecture in Practice. 3. Auflage, Addison-Wesley Longman, 2012. ▪ Ralf Reussner, Wilhelm Hasselbring: Handbuch der Software-Architektur. 2. Auflage, dpunkt.verlag, 2008.		
Studienleistung	☐ Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation)		
	☒ Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation)		
	☐ Bestehen von Leistungsstandkontrollen		
Prüfungsform	☐ Mündliche Prüfung		
	☒ Klausur		
	☐ Prüfung am PC		
	☐ Hausarbeit (ggf. mit Präsentation)		
	☐ Projekt (ggf. mit Präsentation)		
Verwendbarkeit	Künstliche Intelligenz und Data Science		☐ PF ☒ WPF
Angebot	☒ Sommersemester ☐ Wintersemester ☐ Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	6	60 Stunden	120 Stunden
Lehrende(r)	Prof. Dr. M. Striewe		
Modulverantwortliche(r)	Prof. Dr. M. Striewe		
Änderungsdatum	24.02.2025		

Software-Qualitätsmanagement			

Inhalte	▪ Einführung und Überblick
	▪ Qualitätssicherung
	▪ Manuelle Prüfmethoden
	▪ Verbesserung der Prozessqualität
	▪ Produktqualität - Komponenten
	  ▪ Testende Verfahren
	  ▪ Verifizierende Verfahren
	  ▪ Analysierende Verfahren
	▪ Produktqualität - Systeme
	  ▪ Integrationstest
	  ▪ System- und Abnahmetest
	▪ Konfigurations- und Änderungsmanagement		

Lernergebnisse	Die Studierenden können ▪ Qualitätsbegriffe definieren und einordnen, ▪ die Prinzipien der Software-Qualitätssicherung erklären und begründen, ▪ (Code-)Inspektionen durchführen, ▪ kontrollflussorientierte und datenflussorientierte Testverfahren einsetzen, ▪ die Konzepte der Verifikation und des symbolischen Testens verwenden und gegen testende Verfahren abgrenzen, ▪ für einfache Beispiele Integrations- und Abnahmetests durchführen, ▪ Testwerkzeuge einsetzen, sowie ▪ Werkzeuge und Verfahren des Konfigurationsmanagements einsetzen.		
Lehrform	☒ Vorlesung		
	☒ Übung		
	☐ Seminar/Seminaristischer Unterricht		
	☐ Labor		
	☐ Projekt		
Empfohlene Voraussetzungen	Keine		
Literatur	▪ Roman, Adam: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus: Test Techniques and Sample Mock Exams. Springer, 1. Auflage, 2018. ▪ Spillner, Andreas; Roßner, Thomas; Winter, Mario; Linz, Tilo: Praxiswissen Softwaretest – Testmanagement - Aus- und Weiterbildung zum Certified Tester – Advanced Level nach ISTQB-Standard. dpunkt Verlag, 4. Auflage, 2014. ▪ Liggesmeyer, Peter: Software-Qualität. Spektrum Verlag, 2. Auflage, 2009.		
Studienleistung	☒ Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation)		
	☐ Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation)		
	☐ Bestehen von Leistungsstandkontrollen		
Prüfungsform	☒ Mündliche Prüfung (nur bei geringer Teilnehmerzahl)		
	☒ Klausur		
	☐ Prüfung am PC		
	☐ Hausarbeit (ggf. mit Präsentation)		
	☐ Projekt (ggf. mit Präsentation)		
Verwendbarkeit	Künstliche Intelligenz und Data Science		☐ PF ☒ WPF
Angebot	☐ Sommersemester ☒ Wintersemester ☐ Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	6	60 Stunden	120 Stunden
Lehrende(r)	Prof. Dr. G. Rock		
Modulverantwortliche(r)	Prof. Dr. G. Rock		
Änderungsdatum	03.02.2023		

Statistics and Learning Theory			
Inhalte	Statistics and statistical learning theory constitute the theoretical foundation of machine learning and data science. The following content is covered in this module: ▪ Hypothesis tests: Binomial test, Chi-square test, t-test, ANOVA, Neyman-Pearson lemma, multi hypothesis testing, decision theory ▪ Estimation theory: MLE, MAP, Fisher information + Cramer-Rao lower bound, applications of exponential families, full-Bayesian approach, ▪ Statistical learning theory: Error decomposition and bias-variance tradeoff, PAC Learning bounds, growth functions, VC-dimension, Rademacher complexity, algorithmic stability, ensemble methods Hinweis: Die Vorlesung wird in Englisch gehalten.		
Lernergebnisse	Students learn fundamental principles and deeper concepts, methods and techniques in statistics and statistical learning theory. Students can independently use primary literature, especially in English, to acquire advanced scientific knowledge and competences. They also learn to evaluate the limitations of AI technologies and data science methods and to critically assess related safety and security issues. Students can analyze and structure complex problems, develop their own ideas and implement solutions by means of adequate methods and tools. In this module, students are prepared for further scientific qualifications.		
Lehrform	☒ Vorlesung		
	☐ Übung		
	☒ Seminar/Seminaristischer Unterricht		
	☐ Labor		
	☐ Projekt		
Empfohlene Voraussetzungen	Gute Englisch-Kenntnisse		
Literatur	▪ Henze, Norbert: Stochastik für Einsteiger. Springer, 13. Auflage (2021). ▪ L. Wasserman: All of Statistics. Springer, 2003. ▪ M. Hardt, B. Recht: Patterns, Predictions, and Actions: Foundations of Machine Learning. Princeton University Press, 2022.		
Studienleistung	☐ Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation)		
	☒ Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation)		
	☐ Bestehen von Leistungsstandkontrollen		
Prüfungsform	☒ Mündliche Prüfung (nur bei geringer Teilnehmerzahl)		
	☒ Klausur		
	☐ Prüfung am PC		
	☐ Hausarbeit (ggf. mit Präsentation)		
	☐ Projekt (ggf. mit Präsentation)		
Verwendbarkeit	Künstliche Intelligenz und Data Science		☒ PF ☐ WPF
Angebot	☐ Sommersemester ☒ Wintersemester ☐ Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	6	60 Stunden	120 Stunden
Lehrende(r)	Prof. Dr. H.-P. Beise		
Modulverantwortliche(r)	Prof. Dr. H.-P. Beise		
Änderungsdatum	25.01.2023		

Ubiquitous Computing			
Inhalte	Die folgenden Themen werden als Aspekte von Systemen im Bereich „Ubiquitous Computing" betrachtet: ▪ Beispielsysteme aus Forschung und Anwendung ▪ Erhebung, Verwendung und Verteilung von Kontextinformationen ▪ „Location" als besonderer Kontext ▪ Anpassung von Anwendungen an Benutzer und Situation ▪ Alternative Ein- und Ausgabe-, sowie Interaktionsmöglichkeiten ▪ Aufbau und Durchführung von Benutzerstudien ▪ Umgang und Unterstützung von Privacy als Kernproblematik von Ubicomp Systemen ▪ Drahtlose Datenkommunikation und Ad-hoc Protokolle ▪ Spracheingabe und Sprachausgabe ▪ Zukünftige Entwicklungen Weiterhin werden wissenschaftliche Papiere aus den relevanten wissenschaftlichen Konferenzen und Journalen diskutiert, um die Themen der Vorlesung an aktuellen Forschungsergebnissen zu spiegeln.		
Lernergebnisse	Studierende haben Kenntnis vom Bereich „Ubiquitous Computing", sowie den zugehörigen Anwendungen, Konzepten, Verfahrensweisen und aktuellen Forschungsthemen. Sie analysieren Situationen aus dem Blickwinkel des Forschungsgebietes und sind in der Lage aus den Beobachtungen und unter Verwendung der gängigen Methoden und Vorgehensweisen des Gebiets ein UbiComp-System zu konzipieren, zu erstellen und zu evaluieren. Sie sind in der Lage ihre Ergebnisse als wissenschaftliches Papier zu formulieren, im Stil, wie er im Forschungsgebiet üblich ist.		
Lehrform	☐ Vorlesung		
	☐ Übung		
	☒ Seminar/Seminaristischer Unterricht		
	☐ Labor		
	☐ Projekt		
Empfohlene Voraussetzungen	Kompetenzen gemäß der Lernergebnisse des Bachelor-Moduls „Web-Technologien". Die Veranstaltung ist teilnahmebeschränkt. Bei Überbuchung werden die Plätze durch einen Einstufungstest vergeben.		
Literatur	▪ John Krumm : Ubiquitous Computing Fundamentals. Taylor & Francis Ltd, 2009. ▪ Wissenschaftliche Papiere der Proceedings der Konferenzreihe UbiComp, bzw. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT) Journal werden im Rahmen der Übung vorgestellt und diskutiert.		
Studienleistung	☒ Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation)		
	☐ Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation)		
	☐ Bestehen von Leistungsstandkontrollen		
Prüfungsform	☒ Mündliche Prüfung		
	☐ Klausur		
	☐ Prüfung am PC		
	☐ Hausarbeit (ggf. mit Präsentation)		
	☐ Projekt (ggf. mit Präsentation)		
Verwendbarkeit	Künstliche Intelligenz und Data Science		☐ PF ☒ WPF
Angebot	☐ Sommersemester ☐ Wintersemester ☒ Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	6	60 Stunden	120 Stunden
Lehrende(r)	Prof. Dr. G. Schneider		
Modulverantwortliche(r)	Prof. Dr. G. Schneider		
Änderungsdatum	22.11.2022		

Verlässliche Echtzeitsysteme			
Inhalte	▪ Dependability ▪ Terminologie (z.B. nach Laprie) ▪ Echtzeitsysteme ▪ Systemeigenschaft Sicherheit ▪ Einschlägige Normen (z.B. IEC 61508, ISO 26262, DO 178 B) ▪ Fehlervermeidung vs. Fehlertoleranz ▪ Formale Methoden ▪ Echtzeitbetriebssysteme ▪ Parallelisierung von Echtzeitsystemen ▪ Anwendungsbeispiele, z.B. ESP, Motorsteuerung, Fahrerassistenzsysteme		
Lernergebnisse	Die Studierenden haben die Prinzipien verlässlicher Echtzeitsysteme und die Besonderheiten bei deren Entwicklung kennen gelernt. Sie sind in der Lage, in interdisziplinären Teams bei der Entwicklung von sicherheitsrelevanten Echtzeitsystemen mitzuwirken.		
Lehrform	☐ Vorlesung		
	☐ Übung		
	☒ Seminar/Seminaristischer Unterricht		
	☒ Labor		
	☐ Projekt		
Empfohlene Voraussetzungen	Kompetenzen gemäß der Lernergebnisse des Bachelor-Moduls „Technische Informatik“		
Literatur	▪ ISO 26262 (wird in der Veranstaltung bereitgestellt) ▪ Algirdas Avižienis, Jean-Claude Laprie, Brian Randell and Carl Landwehr: Basic Concepts and Taxonomy of Dependable and Secure Computing. IEEE Transactions on Dependable and Secure Computing, 2004 & 2013.		
Studienleistung	☐ Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation)		
	☒ Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation)		
	☐ Bestehen von Leistungsstandkontrollen		
Prüfungsform	☐ Mündliche Prüfung		
	☐ Klausur		
	☐ Prüfung am PC		
	☐ Hausarbeit (ggf. mit Präsentation)		
	☒ Projekt (ggf. mit Präsentation)		
Verwendbarkeit	Künstliche Intelligenz und Data Science		☐ PF ☒ WPF
Angebot	☐ Sommersemester ☐ Wintersemester ☒ Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	6	60 Stunden	120 Stunden
Lehrende(r)	Prof. Dr. J. Schneider		
Modulverantwortliche(r)	Prof. Dr. J. Schneider		
Änderungsdatum	06.02.2023		

Visuelle Navigation			
Inhalte	SLAM – Frontend, Backend, Mapping: ▪ Einführung in das Themengebiet ▪ Repräsentationen und Transformationen ▪ Lie-Gruppen und Lie-Algebren ▪ Kameramodelle und Bildentstehung ▪ Konvexe Optimierung (Einführung) ▪ Visuelle Odometrie I (Invariante Merkmale, Epipolar-Geometrie und Registrierung) ▪ Visuelle Odometrie II (Optischer Fluss und Direkte Methoden) ▪ Filter und Optimierung I (Bündelblockausgleich und graphenbasierte Optimierung) ▪ Filter und Optimierung II (Effiziente Löser und Posen-Graph-Optimierung) ▪ Identifikation von Schleifen in Trajektorien ▪ (Dichte) Rekonstruktion aus mehreren Ansichten ▪ Visuelle Stereo-Odometrie und weitere Verfahren Die Unterlagen sind in englischer Sprache verfasst, die Unterrichtssprache ist Deutsch.		
Lernergebnisse	Die Studierenden beherrschen die mathematischen Grundlagen für den Entwurf eines SLAM-basierten Navigationssystems und verfügen über die praktischen Fähigkeiten selbstständig ein ausgewähltes SLAM-Modell in Grundzügen zu implementieren (engl. Self Localization and Mapping). SLAM-basierte Navigation ermöglicht es einem mobilen Roboter durch Erkundung seiner Umwelt mit visuellen Sensoren selbständig ein Umgebungsmodell aufzubauen und dabei die zurückgelegte Trajektorie zu bestimmen. Die Analyse und Entwicklung eines derartigen Systems werden von den Studierenden beherrscht. Ausgehend von einem in der Vorlesung vorgestellten Systementwurfs werden im Rahmen der Vorlesung verschiedene Konzepte und Methoden diskutiert die anschließend im Rahmen der Übungen anhand ausgewählter Aufgaben implementiert werden. Damit ist der Erwerb der praktischen Fähigkeiten nach erfolgreichem Abschluss der Übungen sichergestellt. Visuelle Navigation stellt nach Einschätzung vieler Experten aktuell eines der wichtigsten und grundlegendsten Herausforderungen der Robotikforschung dar und wird zukünftig spannende und herausfordernde Arbeitsplätze im Bereich der Forschung und Entwicklung im industriellen und akademischen Umfeld schaffen (bspw. im beim autonomen Fahren).		
Lehrform	☒ Vorlesung		
	☒ Übung		
	☐ Seminar/Seminaristischer Unterricht		
	☐ Labor		
	☐ Projekt		
Empfohlene Voraussetzungen	Keine		
Literatur	▪ Yi Ma et al.: An Invitation to 3-D Vision: From Images to Geometric Models. 1. Auflage (2003), Springer. ▪ Richard Hartley & Andrew Zisserman: Multiple View Geometry in computer vision. 2. Auflage (2004), Cambridge University Press. ▪ Xiang Gao & Tao Thang: Introduction to Visual SLAM: From Theory to Practice. 2021, Springer. ▪ Wolfgang Förstner & Bernhard P. Wrobel: Photogrammetric Computer Vision: Statistics, Geometry, Orientation and Reconstruction. 2018, Springer.		
Studienleistung	☒ Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation)		
	☐ Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation)		
	☐ Bestehen von Leistungsstandkontrollen		
Prüfungsform	☒ Mündliche Prüfung (nur bei geringer Teilnehmerzahl)		
	☒ Klausur		
	☐ Prüfung am PC		
	☐ Hausarbeit (ggf. mit Präsentation)		
	☐ Projekt (ggf. mit Präsentation)		
Verwendbarkeit	Künstliche Intelligenz und Data Science		☐ PF ☒ WPF
Angebot	☐ Sommersemester ☒ Wintersemester ☐ Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	6	60 Stunden	120 Stunden
Lehrende(r)	Prof. Dr. J. Graf		
Modulverantwortliche(r)	Prof. Dr. J. Graf		
Änderungsdatum	12.01.2023		

Wissensrepräsentation mit Beschreibungslogiken			
Inhalte	Wissensrepräsentation befasst sich mit der Darstellung von Wissen in einer Form, die von Informationssystemen verarbeitet werden kann. Beschreibungslogiken haben eine lange Tradition in der Informatik und der Wissensrepräsentation. Sie wurden entwickelt, um Domänenwissen zu modellieren und Systemen die Möglichkeit zu geben, logische Schlussfolgerungen über dieses Wissen zu ziehen. In den letzten Jahren haben Beschreibungslogiken an Bedeutung gewonnen, da sie die logische Grundlage für weit verbreitete Ontologiesprachen bilden. In der Vorlesung werden verschiedene Beschreibungslogiken betrachtet und deren Eigenschaften näher untersucht. Betrachtete Themen sind: ▪ Überblick über den Bereich der Wissensrepräsentation ▪ Die Beschreibungslogik ALC ▪ Schlussfolgerungsprobleme ▪ Ausdrucksstärke und Eigenschaften der Modelle ▪ Erweiterungen von ALC ▪ Tableau-Algorithmen ▪ Erklärungen für Schlussfolgerungen (Justifications) ▪ Die Beschreibungslogik EL ▪ Anfragebeantwortung		
Lernergebnisse	Die Studierenden können ▪ Wissen in Form von beschreibungslogischen Ontologien repräsentieren, ▪ Ontologien in Protégé entwerfen, ▪ Schlussfolgerungen aus dem repräsentierten Wissen ziehen, ▪ Reasoner für das Schlussfolgern aus Ontologien verwenden, ▪ Erklärungen zu Schlussfolgerungen generieren.		
Lehrform	☒ Vorlesung		
	☒ Übung		
	☐ Seminar/Seminaristischer Unterricht		
	☐ Labor		
	☐ Projekt		
Empfohlene Voraussetzungen	Keine		
Literatur	▪ Baader, F., Horrocks I. Lutz C., Sattler U. An Introduction to Description Logic. Cambridge University Press; 2017 ▪ Baader, F., Calvanese, D., McGuinness, D., Nardi, D., & Patel-Schneider, P. (Eds.). (2007). The Description Logic Handbook: Theory, Implementation and Applications (2nd ed.). Cambridge: Cambridge University Press.		
Studienleistung	☒ Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation)		
	☐ Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation)		
	☐ Bestehen von Leistungsstandkontrollen		
Prüfungsform	☒ Mündliche Prüfung (nur bei geringer Teilnehmerzahl)		
	☒ Klausur		
	☐ Prüfung am PC		
	☐ Hausarbeit (ggf. mit Präsentation)		
	☐ Projekt (ggf. mit Präsentation)		
Verwendbarkeit	Künstliche Intelligenz und Data Science		☐ PF ☒ WPF
Angebot	☐ Sommersemester ☐ Wintersemester ☒ Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	6	60 Stunden	120 Stunden
Lehrende(r)	Prof. Dr. C. Schon		
Modulverantwortliche(r)	Prof. Dr. C. Schon		
Änderungsdatum	17.06.2024		