

Fakultät Informationstechnik

Modulhandbuch SPO 2 Master-Studiengang **Angewandte Informatik**

Vertiefungsrichtungen:

- Autonome Systeme (AS)
- Data Science (DS)
- IT-Security (IS)

Hinweise

Die in den Modulbeschreibungen genannten Voraussetzungen sind nicht zwingend, aber sehr hilfreich für das Verständnis der vermittelten Lerninhalte.

Abkürzungen:

SWS Semesterwochenstunden

ECTS European Credit Transfer and Accumulation System

Europäisches System zur Übertragung und Akkumulierung von Studienleistungen

ECTS ist ein Maß für den erforderlichen Arbeitsaufwand im Studium

1 ECTS entspricht näherungsweise 30 Arbeitsstunden

Die Angabe der ECTS-Punkte in den Modulbeschreibungen soll den aufzubringenden Arbeitsaufwand transparent machen.

Vertiefungsrichtungen und zugehörige Wahlmodule

Jedes Wahlmodul ist einem der drei Schwerpunkten

- Autonome Systeme (AS),
- Data Science (DS) und
- IT Security (IS)

zugeordnet.

Werden ausschließlich Wahlmodule aus nur einer Vertiefungsrichtung absolviert, so kann die Vertiefungsrichtung im Abschlusszeugnis ausgewiesen werden. Der formlose Antrag dafür ist zu Beginn des 3. Fachsemesters beim Studiengangleiter einzureichen.

Werden Wahlmodule aus verschiedenen Vertiefungsrichtungen absolviert, so wird im Abschlusszeugnis keine Vertiefungsrichtung ausgewiesen.

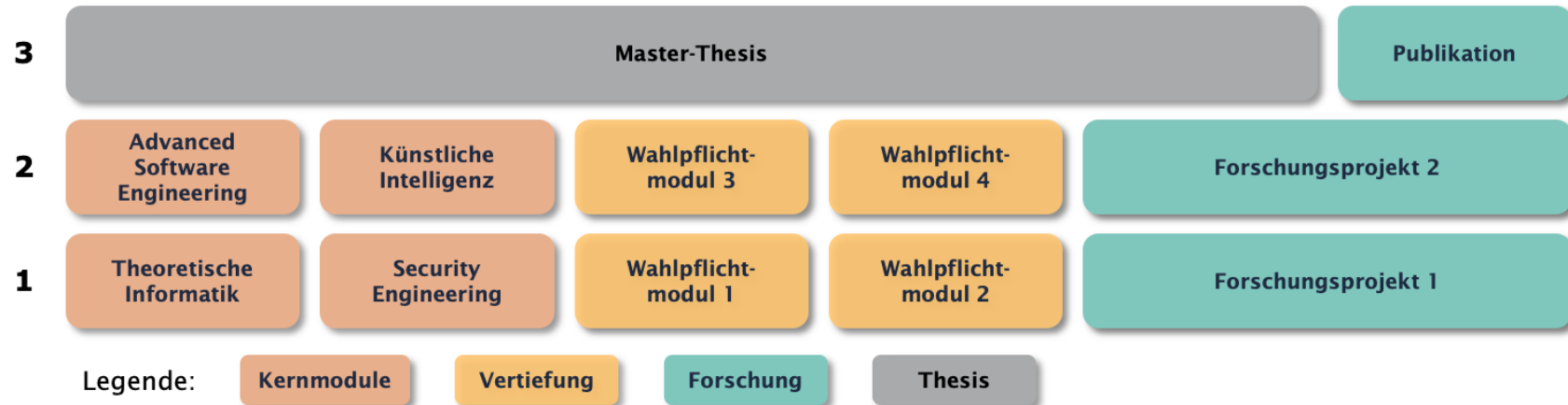
Die Zuordnung der Wahlmodule zu den Schwerpunkten wird durch den Vorsitzenden des Prüfungsausschusses jeweils vor Semesterbeginn per Aushang veröffentlicht.

Version: 23.03.22

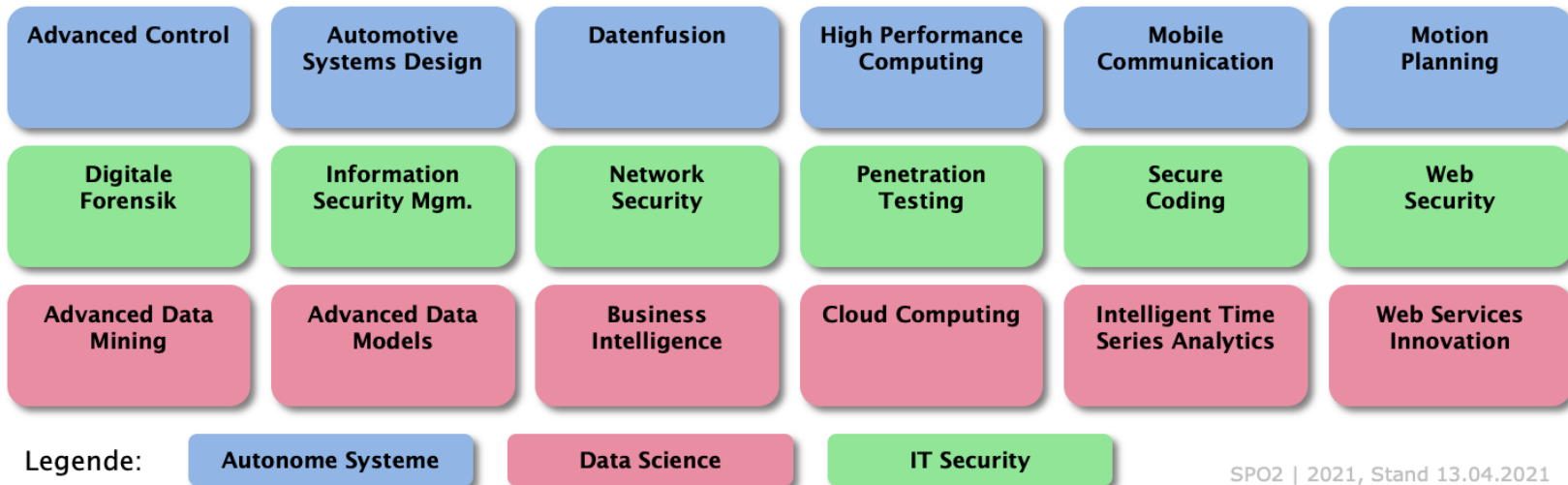
Inhaltsverzeichnis

Prüfungsnummer	Modul	Seite
	Übersicht Modulplan	
1. Semester		
AIM 119 1002	Security Engineering	5
AIM 119 1004	Theoretische Informatik	7
AIM 119 1003	Forschungsprojekt 1	9
AIM WM12	Wahlpflichtmodul 1 und 2	11
2. Semester		
AIM 119 2001	Advanced Software Engineering	12
AIM 119 2004	Künstliche Intelligenz	14
AIM 119 2003	Forschungsprojekt 2	16
AIM WM34	Wahlpflichtmodul 3 und 4	18
3. Semester		
AIM 119 3000	Masterarbeit	19
AIM 119 3001	Publikation	21
Vertiefungsrichtung Autonome Systeme		
AIM 800 6614	Advanced Control	24
AIM 800 6610	Automotive System Design	30
AIM 800 6615	Data Fusion	36
AIM 800 6618	High Performance Computing	40
AIM 800 6601	Mobile Communication	46
AIM 800 6622	Motion Planning for autonomous Systems	48
Vertiefungsrichtung Data Science		
AIM 800 6626	Advanced Data Mining	26
AIM 800 6619	Advanced Data Models	28
AIM 800 6603	Business Intelligence	32
AIM 800 6620	Cloud Computing	34
AIM 800 6618	High Performance Computing	40
AIM 800 6632	Intelligent Time Series Analytics	44
Vertiefungsrichtung IT-Security		
AIM 800 6631	Digitale Forensik	38
AIM 800 6624	Information Security Management	42
AIM 800 6625	Network Security	50
AIM 800 6630	Penetration Testing	52
AIM 800 6623	Secure Coding	54
AIM 800 6629	Web Security	56
AIM 800 6627	Web Services Innovationen	58

Übersicht Modulplan



Wahlpflichtmodule / Vertiefung



SPO2 | 2021, Stand 13.04.2021

Modulbeschreibung Security Engineering

Schlüsselworte: Security, Engineering, Software, Design, Standards

Zielgruppe:	Semester AIM1 und AIM2	Modulnummer:	AIM 119 1002
Arbeitsaufwand:	5 ECTS		150 h
Davon	Kontaktzeit		60 h
	Selbststudium		60 h
	Prüfungsvorbereitung		30 h
Unterrichtssprache:	Englisch		
Modulverantwortung:	Prof. Dr. Dominik Schoop		
Stand:	14.01.2021		

Empfohlene Voraussetzungen:

Grundlagen des Software Engineerings

Modulziel – angestrebte Lernergebnisse:

Die Studierenden werden in die Lage versetzt, nach der Einschätzung des Sicherheitsrisikos, Technologien und Vorgehensweisen anzuwenden, um ein sicheres, komplexes und heterogenes IT-System zu entwickeln. Sie beherrschen die Methoden zur Einschätzung von Risiken und sind in der Lage, der Situation angemessene Methoden für den sicheren Entwurf, Implementierung und Betrieb eines IT-Systems auszuwählen und anzuwenden. Sie werden befähigt, die IT-Sicherheit von Anwendungen zu analysieren und zu bewerten. Sie sind in der Lage angemessene IT-Sicherheitslösungen systematisch zu entwickeln und darzustellen.

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen:

- Moderne Verschlüsselungsverfahren
- Kryptographische Protokolle
- Security System Modelling
- Security Requirements Engineering
- Security Design Principles
- Verifikation von Sicherheitskomponenten

Fertigkeiten – methodische Kompetenzen

Die Studierenden sind in der Lage

- Sicherheitssysteme zu analysieren,
- Sicherheitsrisiken zu analysieren,
- Systeme auf Sicherheit zu validieren und zu verifizieren
- Informationstechnisch sichere Systeme zu entwerfen

Übergreifende Kompetenzen

Die Studierenden können

- nach der Einschätzung des Sicherheitsrisikos, Vorgehensweisen und Technologien anwenden, um ein sicheres, komplexes und heterogenes IT-System zu entwickeln.

Inhalt:

- Einführung in die Kryptographie
- Moderne Verschlüsselungsverfahren
- Kryptographische Protokolle
- Blockchain-Technologie
- Elliptische Kurven
- Pseudorandom Generatoren, Pseudorandom Permutationen, Pseudorandom Funktionen
- Beweisbar sichere symmetrische und asymmetrische Kryptographie
- Quantenkryptographie

- Sicherheitsziele
- Safety vs. Security
- Security Engineering als Prozess
- Security Requirements Engineering (Angreifermodelle, Bedrohungsidentifikation und -modellierung)
- Risikoanalyse
- Bedrohungen und Angriffsmuster
- Security Design Principles
- Security Patterns
- Verifikation von Sicherheitseigenschaften von Protokollen

Literaturhinweise:

- Ross Anderson: Security Engineering: A Guide to Building Dependable Distributed Systems, Wiley, 3. Aufl., 2020
- Cas Cremers, Sjouke Mauw: Operational Semantics and Verification of Security Protocols, Springer 2012
- Jonathan Katz and Yehuda Lindell: Introduction to Modern Cryptography, Chapman and Hall/CRC, 2014
- Nigel P. Smart: Cryptography made simple, Springer 2016
- Bruce Schneier: Angewandte Kryptographie, Pearson Studium, 2006

Wird angeboten:

in jedem Wintersemester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Übungen und Security Case Study
Leistungskontrolle:	Klausur (90 Minuten)
Anteil Semesterwochenstunden:	4 SWS
Geschätzte studentische Arbeitszeit:	150 Stunden

Bildung der Modulnote:

Klausur

Modulbeschreibung Theoretische Informatik

Schlüsselworte: Theoretische Informatik, Berechenbarkeitstheorie, Komplexitätstheorie, Informationstheorie

Zielgruppe:	Semester AIM1 und AIM2	Modulnummer:	AIM 119 1004
Arbeitsaufwand:	5 ECTS		150 h
Davon	Kontaktzeit		60 h
	Selbststudium		60 h
	Prüfungsvorbereitung		30 h
Unterrichtssprache:	Deutsch		
Modulverantwortung:	Prof. Dr.-Ing. Steffen Schober		
Stand:	01.09.2019		

Empfohlene Voraussetzungen:

Diskrete Mathematik

Modulziel – angestrebte Lernergebnisse:

Einführung in die theoretischen Grundlagen der Informatik, im einzelnen Theorie der formalen Sprachen und Automatentheorie, Komplexitäts-, Berechenbarkeits- und Informationstheorie.

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen und verstehen

- Sprachen der Chomsky-Hierarchie mit den zugehörigen Berechnungsmodellen
- Begriffe der Berechenbarkeit und Bedeutung der Unentscheidbarkeit
- Das Halteproblem und den Satz von Gödel
- Klassen P / NP, sowie das Konzept der NP-schweren Probleme
- Ausgewählte NP-schwere Probleme
- Grundlagen der Informationstheorie
 - Informationsmaß (Entropie)
 - Codierungsverfahren (z.B. Huffman und Shannon-Fano Code)

Fertigkeiten – methodische Kompetenzen

Die Studierenden sind in der Lage

- Grammatiken für Sprachen zu formulieren
- Deterministische von nicht-deterministischen Modellen zu unterscheiden und ihre Mächtigkeit gegeneinander abschätzen
- Kombinatorisch schwere Berechnungsprobleme zu erkennen und zu klassifizieren (Reduktion auf bekannte Probleme)
- Die Grenzen der Informationsverarbeitung zu verstehen und richtig einzuordnen

Übergreifende Kompetenzen

Die Studierenden können

- Systeme formal spezifizieren und Eigenschaften beweisen
- Erlernete Beweistechniken auf andere Probleme anwenden

Inhalt:

- Sprachen und Automaten:
 - Chomsky 3 Sprachen und endliche Automaten
 - Chomsky 2 Sprachen und endliche Automaten
 - Chomsky 1+0 Sprachen und Turingmaschinen
- Grundlagen der Berechenbarkeitstheorie
 - Berechnungsmodelle und deren Beziehung; These von Church
 - Halteproblem
 - Satz von Gödel

- Grundlagen der Komplexitätstheorie
 - P und NP und NP-schwere Probleme
 - Ausgewählte NP-schwere Probleme
- Grundlagen der Informationstheorie
 - Entropie als Informationsbegriff
 - Quellencodierungsverfahren

Literaturhinweise:

- Schöning: Theoretische Informatik – kurz gefasst, Spektrum Akademischer Verlag
- Hopcroft, u.a.: Einführung in die Automatentheorie, Formale Sprachen und Komplexitätstheorie, Pearson Studium

Wird angeboten:

in jedem Wintersemester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Übungen
Leistungskontrolle:	Klausur (90 Minuten)
Anteil Semesterwochenstunden:	4 SWS
Geschätzte studentische Arbeitszeit:	120 Stunden

Bildung der Modulnote:

Klausur

Modulbeschreibung Forschungsprojekt 1

Schlüsselworte: Wissenschaftliches Arbeiten im Team

Zielgruppe:	Semester AIM1	Modulnummer:	AIM 119 1003
Arbeitsaufwand:	10 ECTS		300 h
Davon	Kontaktzeit		30 h
	Selbststudium		270 h
	Prüfungsvorbereitung		0 h
Unterrichtssprache:	Deutsch oder Englisch		
Modulverantwortung:	Prof. Dr.-Ing. Rainer Keller		
Stand:	14.01.2021		

Empfohlene Voraussetzungen:

Anwendung der Methoden des Softwareentwicklung, Kenntnisse in der gewählten Vertiefungsrichtung, Grundkenntnisse wissenschaftlichen Arbeitens

Modulziel – angestrebte Lernergebnisse:

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen

- die Qualitätskriterien für wissenschaftliches Arbeiten
- die Methoden des Projektmanagement

Fertigkeiten – methodische Kompetenzen

Die Studierenden sind in der Lage

- wissenschaftliche Projekte im Team zu planen und durchzuführen,
- die in den Kern- und Vertiefungsfächern erworbenen Kenntnisse zur Lösung von Problemen aus dem Bereich der Forschung einzusetzen,
- Lösungsansätze (Stand der Technik) zu recherchieren und zu verstehen,
- gefundene Lösungsansätze bewerten,
- die Ergebnisse ihres wissenschaftlichen Arbeitens nachvollziehbar dokumentieren.

Übergreifende Kompetenzen

Die Studierenden können

- unter Anleitung komplexe Problemstellungen aus der Forschung oder aus der Industrie innerhalb einer vorgegebenen Frist zu lösen,
- neue Kenntnisse in der Informatik zu gewinnen und neue Verfahren zu entwickeln,
- Wissen aus verschiedenen Domänen integrieren,
- in einem Team gemeinsam eine Aufgabe erfolgreich umzusetzen.

Inhalt:

Im Forschungsprojekt bearbeiten Studierende in einem Team unter Anleitung eines Dozenten aktuelle Forschungsthemen aus wissenschaftlichen Einrichtungen oder forschungsnahe Themen aus der Industrie. Die Projekte sind auf ein Jahr angelegt, wobei alle Phasen eines Softwareprojekts durchlaufen werden sollen: Problem- und Anforderungsanalyse, Recherche des Standes der Technik, Projektplanung, Erarbeitung von Lösungsansätzen, Softwareentwurf, Implementierung, Testphase. Die Studierenden erarbeiten Arbeits- und Zeitpläne und berichten regelmäßig über ihren Fortschritt. Am Ende der Semester tragen die Studierenden jeweils Zwischen- bzw. Endergebnisse vor.

Literaturhinweise:

Abhängig von der gewählten Problemstellung

Wird angeboten:

Sommer- und Wintersemester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Projektarbeit
Leistungskontrolle:	Bericht und Referat (20 Minuten)
Anteil Semesterwochenstunden:	2 SWS
Geschätzte studentische Arbeitszeit:	300 Stunden

Bildung der Modulnote:

Bericht und Referat

Modulbeschreibung Wahlpflichtmodul 1 und 2

Schlüsselworte: Vertiefung im eigenen Studienprofil

Zielgruppe:	Semester AIM1	Modulnummer:	AIM WM12
Arbeitsaufwand:	5 ECTS	Abhängig vom gewählten Modul	
Davon	Kontaktzeit	Abhängig vom gewählten Modul	
	Selbststudium	Abhängig vom gewählten Modul	
	Prüfungsvorbereitung	Abhängig vom gewählten Modul	
Unterrichtssprache:	Deutsch oder Englisch		
Modulverantwortung:	Prof. Dr.-Ing. Rainer Keller		
Stand:	01.09.2019		

Empfohlene Voraussetzungen:

Abhängig vom gewählten Modul

Gesamtziel:

Die Studierenden erlangen eine wissenschaftliche und fachliche Vertiefung auf dem Gebiet der gewählten Module.

Inhalt:

Abhängig vom gewählten Modul

Literaturhinweise:

Abhängig vom gewählten Modul

Wird angeboten:

Die zur Auswahl stehenden Wahlfächer werden zu Semesterbeginn öffentlich bekannt gegeben.

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform: Abhängig vom gewählten Modul

Leistungskontrolle: Abhängig vom gewählten Modul

Anteil Semesterwochenstunden: Abhängig vom gewählten Modul

Geschätzte studentische Arbeitszeit: 150 Stunden

Lernergebnisse:

Abhängig vom gewählten Modul

Bildung der Modulnote:

Abhängig vom gewählten Modul

Modulbeschreibung Advanced Software Engineering

Schlüsselworte: Software Engineering, Metriken, Empirie

Zielgruppe:	Semester AIM1 und AIM2	Modulnummer:	AIM 119 2001
Arbeitsaufwand:	5 ECTS		150 h
Davon	Kontaktzeit		60 h
	Selbststudium		60 h
	Prüfungsvorbereitung		30 h
Unterrichtssprache:	Deutsch		
Modulverantwortung:	Prof. Dr. Mirko Sonntag		
Stand:	15.01.2021		

Empfohlene Voraussetzungen:

Grundlagen des Softwareengineering

Modulziel – angestrebte Lernergebnisse:

Die Studierenden werden in die Lage versetzt, komplexe Anforderungen mit ingenieurmäßigen Methoden des Software Engineering zu lösen. Sie beherrschen fortgeschrittene Methoden des Software Engineerings.

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen

- Agile Vorgehensweisen in Großprojekten: Scrum/Agile@Scale,
- DevOps
- Spezielle Techniken des Software Engineering: Aspektorientierung, SOA und Microservices, Resilience Patterns
- Agiles Testen
- Securityaspekte bei der Systementwicklung

Fertigkeiten – methodische Kompetenzen

Die Studierenden sind in der Lage

- Methoden des Projekt- & Risikomanagement in realen Softwareprojekten anzuwenden
- Hochqualitative Software mit Hilfe automatisierte Buildchains zu erstellen
- spezielle Techniken des Software Engineerings situativ auszuwählen und anzuwenden

Übergreifende Kompetenzen

Die Studierenden können

- Probleme des Softwareengineerings in Projekten lösen
- Arbeiten an Software in global verteilten Teams durchführen

Inhalt:

Verschiedenste praxisrelevante Techniken beim Software Engineering

- Agile@Scale
- DevOps und Buildchain
- SOA und Microservices
- Distributed Software Engineering
- Resilience Engineering
- Security Engineering

Literaturhinweise:

- Sommerville: Engineering Software Products. Pearson 2019
- Sommerville: Software Engineering. Pearson 2018

- Ludwig, Lichter: Software Engineering. dpunkt.verlag 2013.

Wird angeboten:

in jedem Sommersemester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Übungen und Projektarbeit
Leistungskontrolle:	Klausur (90 Minuten)
Anteil Semesterwochenstunden:	4 SWS
Geschätzte studentische Arbeitszeit:	90 Stunden

Bildung der Modulnote:

Klausur

Modulbeschreibung Künstliche Intelligenz

Schlüsselworte: Überwachtes Lernen, nicht überwachtes Lernen, maschinelles Lernen

Zielgruppe:	Semester AIM1 und AIM2	Modulnummer:	AIM 119 2004
Arbeitsaufwand:	5 ECTS		150 h
Davon	Kontaktzeit		60 h
	Selbststudium		60 h
	Prüfungsvorbereitung		30 h
Unterrichtssprache:	Deutsch		
Modulverantwortung:	Prof. Dr.-Ing. Steffen Schober		
Stand:	01.09.2019		

Empfohlene Voraussetzungen:

- Kenntnisse der Programmiersprache Python
- Lineare Algebra
- Statistik

Modulziel – angestrebte Lernergebnisse:

Die Studierenden beherrschen die wesentliche Methoden der künstlichen Intelligenz.

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen

- Grundlagen der Statistischen Lerntheorie
 - *Bias-Variance Tradeoff*
 - *Maximum-Likelihood (ML)* Schätzer und Regularisierung (penalized ML)
 - Beurteilung der Güte (z.B. *accuracy*)
- Verfahren des überwachten Lernens:
 - Lineare Klassifikations-/Regressionsverfahren, z.B. *k-nearest neighbours*, lineare Modelle, (lineare) *support vector machines*
 - Nichtlineare Verfahren: Polynomielle Regression, Kernel basierte Verfahren, Entscheidungsbäume, neuronale Netzwerke (*feed-forward*)
 - Ensemble Ansätze: *Bagging*, *Random Forests*, *GradientBoosting*
- Verfahren des unüberwachten Lernens:
 - partitionierendes Clustering, hierarchisches Clustering
 - PCA
 - *Gaussian Mixture-Models* und EM-Algorithmus
- Grundlagen des Python Data-Science Stacks (numpy, scikit-learn)

Fertigkeiten – methodische Kompetenzen

Die Studierenden sind in der Lage

- geeignete Verfahren für bestimmte Probleme auszuwählen
- die erlernten Verfahren mit Hilfe der Programmiersprache Python einzusetzen
- die Ergebnisse der Verfahren zu interpretieren.

Übergreifende Kompetenzen

Die Studierenden können

- Verfahren des maschinellen Lernens für Problemlösungen in anderen Domänen einsetzen

Inhalt:

- Grundlagen der Wahrscheinlichkeitstheorie
- Grundlagen Lineare Algebra
- Grundbegriffe der statistischen Lerntheorie
- Verfahren des überwachten Lernens

- Verfahren des unüberwachten Lernens
- Python Bibliotheken für Maschinelles Lernen

Literaturhinweise:

- James et al; Introduction to Statistical Learning ISBN 9781461471370
- Hastie, Trevor J: The elements of statistical learning: data mining, inference, and prediction. - New York, NY: Springer, 2013. ISBN: 9780387848570

Wird angeboten:

in jedem Sommersemester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Übungen
Leistungskontrolle:	Klausur (90 Minuten)
Anteil Semesterwochenstunden:	4 SWS
Geschätzte studentische Arbeitszeit:	150 Stunden

Modulbeschreibung Forschungsprojekt 2

Schlüsselworte: Wissenschaftliches Arbeiten im Team

Zielgruppe:	Semester AIM2	Modulnummer:	AIM 119 2003
Arbeitsaufwand:	10 ECTS		300 h
Davon	Kontaktzeit		30 h
	Selbststudium		270 h
	Prüfungsvorbereitung		0 h
Unterrichtssprache:	Deutsch oder Englisch		
Modulverantwortung:	Prof. Dr.-Ing. Rainer Keller		
Stand:	14.01.2021		

Voraussetzungen nach Prüfungsordnung:

Keine

Empfohlene Voraussetzungen:

Anwendung der Methoden des Softwareentwicklung, Kenntnisse in der gewählten Vertiefungsrichtung, Grundkenntnisse wissenschaftlichen Arbeitens

Modulziel – angestrebte Lernergebnisse:

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen

- die Qualitätskriterien für wissenschaftliches Arbeiten
- die Methoden des Projektmanagement

Fertigkeiten – methodische Kompetenzen

Die Studierenden sind in der Lage

- wissenschaftliche Projekte im Team zu planen und durchzuführen,
- die in den Kern- und Vertiefungsfächern erworbenen Kenntnisse zur Lösung von Problemen aus dem Bereich der Forschung einzusetzen,
- Lösungsansätze (Stand der Technik) zu recherchieren und zu verstehen,
- gefundene Lösungsansätze bewerten,
- die Ergebnisse ihres wissenschaftlichen Arbeitens nachvollziehbar dokumentieren.

Übergreifende Kompetenzen

Die Studierenden können

- unter Anleitung komplexe Problemstellungen aus der Forschung oder aus der Industrie innerhalb einer vorgegebenen Frist zu lösen,
- neue Kenntnisse in der Informatik zu gewinnen und neue Verfahren zu entwickeln,
- Wissen aus verschiedenen Domänen integrieren,
- in einem Team gemeinsam eine Aufgabe erfolgreich umzusetzen.

Inhalt:

Im Forschungsprojekt bearbeiten Studierende in einem Team unter Anleitung eines Dozenten aktuelle Forschungsthemen aus wissenschaftlichen Einrichtungen oder forschungsnahe Themen aus der Industrie. Die Projekte sind auf ein Jahr angelegt, wobei alle Phasen eines Softwareprojekts durchlaufen werden sollen: Problem- und Anforderungsanalyse, Recherche des Standes der Technik, Projektplanung, Erarbeitung von Lösungsansätzen, Softwareentwurf, Implementierung, Testphase. Die Studierenden erarbeiten Arbeits- und Zeitpläne und berichten regelmäßig über ihren Fortschritt. Am Ende der Semester tragen die Studierenden jeweils Zwischen- bzw. Endergebnisse vor.

Literaturhinweise:

Abhängig von der gewählten Problemstellung

Wird angeboten:

In jedem Semester

Teilgebiete und Leistungsnachweise:**Lehr- und Lernform:**

Projektarbeit

Leistungskontrolle:

Bericht und Referat (20 Minuten)

Anteil Semesterwochenstunden:

2 SWS

Geschätzte studentische Arbeitszeit:

300 Stunden

Bildung der Modulnote:

Bericht und Referat

Modulbeschreibung Wahlpflichtmodul 3 und 4

Schlüsselworte: Vertiefung im eigenen Studienprofil

Zielgruppe:	Semester AIM2	Modulnummer:	AIM WM34
Arbeitsaufwand:	5 ECTS		150 h
Davon	Kontaktzeit	Abhängig vom gewählten Modul	
	Selbststudium	Abhängig vom gewählten Modul	
	Prüfungsvorbereitung	Abhängig vom gewählten Modul	
Unterrichtssprache:	Deutsch oder Englisch		
Modulverantwortung:	Prof. Dr.-Ing. Rainer Keller		
Stand:	01.09.2019		

Empfohlene Voraussetzungen:

Abhängig vom gewählten Modul

Gesamtziel:

Die Studierenden erlangen eine wissenschaftliche und fachliche Vertiefung auf dem Gebiet der gewählten Module.

Inhalt:

Abhängig vom gewählten Modul

Literaturhinweise:

Abhängig vom gewählten Modul

Wird angeboten:

Die zur Auswahl stehenden Wahlfächer werden zu Semesterbeginn öffentlich bekannt gegeben.

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Abhängig vom gewählten Modul
Leistungskontrolle:	Abhängig vom gewählten Modul
Anteil Semesterwochenstunden:	Abhängig vom gewählten Modul
Geschätzte studentische Arbeitszeit:	150 Stunden

Lernergebnisse:

Abhängig vom gewählten Modul

Bildung der Modulnote:

Abhängig vom gewählten Modul

Modulbeschreibung Masterarbeit

Schlüsselworte: Selbstständiges wissenschaftliches Arbeiten

Zielgruppe:	Semester AIM3	Modulnummer:	AIM 119 3000 / 2
Arbeitsaufwand:	25 ECTS		750 h
Davon	Kontaktzeit		30 h
	Selbststudium		660 h
	Prüfungsvorbereitung		60 h
Unterrichtssprache:	Deutsch oder Englisch		
Modulverantwortung:	Prof. Dr.-Ing. Rainer Keller		
Stand:	10.01.2021		

Empfohlene Voraussetzungen:

Kenntnisse der Methoden wissenschaftlichen Arbeitens, sichere Anwendung der Methoden des Softwareengineering, umfassende Kenntnisse in der gewählten Vertiefungsrichtung.

Modulziel – angestrebte Lernergebnisse:

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen

- die Qualitätskriterien für wissenschaftliches Arbeiten
- die Methoden des Projektmanagement

Fertigkeiten – methodische Kompetenzen

Die Studierenden sind in der Lage

- wissenschaftliche Fragestellungen zu formulieren,
- wissenschaftliche Methoden anzuwenden,
- wissenschaftliche Projekte zu planen und durchzuführen,
- die in den Kern- und Vertiefungsfächern erworbenen Kenntnisse zur Lösung von Problemen einzusetzen,
- Lösungsansätze (Stand der Forschung) zu recherchieren und zu verstehen, und zu bewerten,
- eigene Lösungsansätze zu entwickeln und umzusetzen,
- die Ergebnisse ihres wissenschaftlichen Arbeitens nachvollziehbar dokumentieren.

Übergreifende Kompetenzen

Die Studierenden können

- eine komplexe Problemstellung der Informatik selbstständig, wissenschaftlich, innerhalb einer vorgegebenen Frist zu bearbeiten,
- den dazugehörigen Stand der Forschung zu recherchieren, zu strukturieren und zu verstehen,
- geeignete Methoden und Verfahren auszuwählen, diese korrekt einzusetzen und falls notwendig sie anzupassen oder weiter zu entwickeln,
- ihre Ergebnisse mit anderen Ergebnissen zu vergleichen und ihre Lösungsansätze kritische zu überprüfen,
- ihre Ergebnisse strukturiert zu dokumentieren und in wissenschaftlicher Form zu veröffentlichen.

Inhalt:

- Problemanalyse und Eingrenzung des Themas
- Literaturrecherche
- Planung der Vorgehensweise, Erarbeitung eines Lösungsansatzes
- Zeit- und Projektmanagement
- Herstellen eines Bezugs zwischen eigenen Ansätze und dem Stand der Forschung
- Wissenschaftliche Darstellung der Ergebnisse
- Verteidigung

Literaturhinweise:

Abhängig von der gewählten Problemstellung

Wird angeboten:

in jedem Semester

Teilgebiete und Leistungsnachweise:

Modulteil:	Masterarbeit
Lehr- und Lernform:	Wissenschaftliche Arbeit mit Bericht und Verteidigung
Leistungskontrolle:	Schriftlicher Bericht, Testat und Referat (20 Minuten)
Anteil Semesterwochenstunden:	2 SWS
Geschätzte studentische Arbeitszeit:	750 Stunden
Lernergebnisse:	

Die Studierenden verfügen über **Kenntnisse** der grundlegenden Techniken des wissenschaftlichen Arbeitens, insbesondere Gliederung, Zitieren und formale Aspekte. Sie besitzen die **Fertigkeiten** Probleme zu analysieren, zu strukturieren und zu lösen. Dazu sind sie in der Lage, den dazugehörigen Stand der Forschung zu recherchieren, zu strukturieren und zu verstehen sowie geeignete Methoden und Verfahren auszuwählen, diese korrekt einzusetzen und falls notwendig sie anzupassen oder weiter zu entwickeln. Sie können ihre Ergebnisse mit anderen Ergebnissen vergleichen und ihre Lösungsansätze kritisch überprüfen. Die Studierenden verfügen über die **Kompetenzen** zur selbstständigen Projekt- und Zeitplanung und haben Verantwortungsbewusstsein in Bezug auf wissenschaftliche Qualitätskriterien wie Überprüfbarkeit und Verlässlichkeit von wissenschaftlichen Ergebnissen erworben.

Die Studierenden publizieren einen Kurzbericht ihrer Masterarbeit als Testat.

Bildung der Modulnote:

Gemittelte Note aus Bericht (Faktor 22) und Referat (Faktor 3)

Modulbeschreibung Publikation

Schlüsselworte: Selbstständiges wissenschaftliches Schreiben

Zielgruppe:	Semester AIM3	Modulnummer:	AIM 119 3001
Arbeitsaufwand:	5 ECTS		150 h
Davon	Kontaktzeit		15 h
	Selbststudium		135 h
Unterrichtssprache:	Deutsch oder Englisch		
Modulverantwortung:	Prof. Dr.-Ing. Rainer Keller		
Stand:	23.10.2020		

Empfohlene Voraussetzungen:

Kenntnisse der Methoden wissenschaftlichen Arbeitens, erfolgreiche Teilnahme am Forschungsprojekt 1 und 2.

Modulziel – angestrebte Lernergebnisse:

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen

- die formalen Aspekte einer wissenschaftlichen Veröffentlichung
- geeignete Journalen und Konferenzen

Fertigkeiten – methodische Kompetenzen

Die Studierenden sind in der Lage

- eine Thema für eine Veröffentlichung einzugrenzen,
- den Stand der Forschung zu recherchieren, zu strukturieren, zu verstehen und wiederzugeben,
- Bezüge zwischen eigenen Ansätzen und dem Stand der Forschung herzustellen

Übergreifende Kompetenzen

Die Studierenden können

- Forschungsergebnisse strukturiert dokumentieren und in eine publikationsreife Form bringen.

Literaturhinweise:

- Balzert, Helmut; Schröder, Marion; Schaefer, Christian (2017): Wissenschaftliches Arbeiten: Ethik, Inhalt & Form wiss. Arbeiten, Handwerkszeug, Quellen, Projektmanagement, Präsentation. 2. Aufl., Springer.
- Kornmeier, Martin (2018): Wissenschaftlich schreiben leicht gemacht: Für Bachelor, Master und Dissertation. 8. überarb. Aufl., Haupt.
- Theisen, Manuel René (2017): Wissenschaftliches Arbeiten: Erfolgreich bei Bachelor- und Masterarbeit. 17., akt. und überarb. Aufl., Vahlen.

Wird angeboten:

in jedem Semester

Lehr- und Lernform:	Verfassen einer wissenschaftlichen Publikation
Leistungskontrolle:	Bericht (Veröffentlichungsreifer Artikel)
Anteil Semesterwochenstunden:	1 SWS
Geschätzte studentische Arbeitszeit:	150 Stunden

Lernergebnisse:

Die Studierenden verfügen über Kenntnisse darüber, wie und wo wissenschaftliche Ergebnisse publiziert werden können. Sie sind in der Lage, die Ergebnisse ihrer wissenschaftlichen Tätigkeit nachvollziehbar nach wissenschaftlichen Kriterien zusammenzufassen. Sie verfügen über die Kompetenz, selbstständig für die Qualitätssicherung der Publikation zu sorgen und diese fristgerecht einzureichen.

Bildung der Modulnote:

Bewertung einer veröffentlichungsreifen Publikation

Wahlpflichtmodule

Modulbeschreibung Advanced Control

Schlüsselworte: Fuzzy-Regelung; moderne Regelungstechnik (P-, PI-Zustandsregler, Zustands- und Störgrößenbeobachter), Optimale Regler und Kalman-Filter, Flachheit

Zielgruppe:	Semester AIM1 und AIM2	Modulnummer:	AIM 800 6614
Arbeitsaufwand:	5 ECTS		150 h
Davon	Kontaktzeit		60 h
	Selbststudium		60 h
	Prüfungsvorbereitung		30 h
Unterrichtssprache:	Deutsch		
Modulverantwortung:	Prof. Dr. Walter Lindermeir		
Stand:	11.12.2020		

Empfohlene Voraussetzungen:

Die Studierenden beherrschen die Modellierung von dynamischen Systemen in der Zustandsdarstellung und können die in der Systemtheorie gebräuchlichen Systembeschreibungen einsetzen. Die Studierenden können PID sowie P-Zustandsregler auslegen und deren jeweilige Vor- und Nachteile für eine Anwendung einschätzen. Sie sind in der Lage Luenberger Beobachter zu entwerfen und in Projekten einzusetzen. Dies gilt jeweils sowohl für die analytische als auch für die numerische (MATLAB/Simulink) Auslegung der jeweiligen Systeme.

Modulziel – angestrebte Lernergebnisse:

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen

- die Grundlagen der Fuzzy-Logik (Einsatzgebiete, Fuzzifizierung, Regelbasis, Defuzzifizierung)
- Einsatz und Funktionsweise von Fuzzy-Systemen
- Entwurf und Einsatz von PI-Zustandsreglern und Störgrößenbeobachtern (Polvorgabe)
- Optimale Regler (LQR-Regelung) und optimale Zustandsschätzung dynamischer Systeme (Kalman-Filter)
- Regler in der Zwei-Freiheitsgrad-Struktur (2-DOF)
- Steuerungen bzw. Vorsteuerungen basierend auf der Flachheitsmethode
- IMC-Regler
- verschiedene Approximationen für das Totzeitverhalten

Fertigkeiten – methodische Kompetenzen

Die Studierenden sind in der Lage

- Fuzzy-Inferenz-Systeme zu entwerfen und einzusetzen, z.B. unter Einsatz der Matlab Fuzzy-Logic-Toolbox
- PI-Zustandsregler zu entwickeln (einfache Systeme von Hand, bzw. unter Einsatz von Matlab/Simulink) und in Betrieb zu nehmen
- Störgrößenmodelle zu definieren und Störgrößenbeobachter zu entwickeln
- Kalman-Filter zu entwerfen und sinnvoll zu parametrieren
- 2-DOF-Reglerstrukturen zu verstehen und z.B. mit Hilfe der Flachheitsmethodik auslegen
- IMC-Regler zu entwerfen

Übergreifende Kompetenzen

Die Studierenden können

- die in der Systemtheorie gebräuchlichen Systembeschreibungen (Laplace-Übertragungsfunktionen und Zustandsbeschreibungen) vorteilhaft einsetzen
- Regelungen (PID-/Zustands-/IMC-/Fuzzy-Regler) eigenständig auslegen

- Zustandsschätzer (Luenberger, Kalman, Störgrößen) eigenständig auslegen
- Erweiterte Regler-Strukturen (2-DOF-Struktur, IMC) sinnvoll einsetzen
- Sich im Umfeld von eingebetteten Systemen z.B. in den Bereichen der Automobiltechnik oder der Industrieautomatisierung fachlich einbringen

Inhalt:

- Laplace-Übertragungsfunktionen und Zustandsbeschreibungen
- Auslegung von PID, P- und PI-Zustandsregler sowie Fuzzy-Regler und erweiterte Regler-Strukturen (2-DOF, IMC)
- Entwurf von Zustandsschätzer (Luenberger, Kalman, Störgrößen)
- analytische Auslegung als auch numerische Auslegung der Systeme mit MATLAB/Simulink
- Anwendung der Vorlesungsinhalte am Beispiel der Regelung eines inversen Pendels

Literaturhinweise:

- Föllinger, O.: Regelungstechnik. Hüthig Verlag
- Lunze, J.: Regelungstechnik. 2 Bände, Springer Verlag
- Unbehauen, H.: Regelungstechnik. 3 Bände, Vieweg Verlag
- Simon, D.: Optimal State Estimation, Wiley
- Mann, Schiffelgen, Frieriep, Webers: Einführung in die Regelungstechnik, Hanser Verlag

Wird angeboten:

In jedem Sommersemester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Übungen
Leistungskontrolle:	Klausur (90 Minuten)
Anteil Semesterwochenstunden:	4 SWS
Geschätzte studentische Arbeitszeit:	150 Stunden

Bildung der Modulnote:

Klausur

Modulbeschreibung Advanced Data Mining

Schlüsselworte: Data Mining, Maschinelles Lernen, Visual Analytics, Data Science, Python

Zielgruppe:	Semester AIM1 und AIM2	Modulnummer:	AIM 800 6626
Arbeitsaufwand:	5 ECTS		150 h
Davon	Kontaktzeit		60 h
	Selbststudium		60 h
	Prüfungsvorbereitung		30 h
Unterrichtssprache:	Deutsch		
Modulverantwortung:	Prof. Dr.-Ing. Steffen Schober		
Stand:	01.10.2020		

Empfohlene Voraussetzungen:

Kenntnisse in

- grundlegende Programmierkenntnisse
- mathematische Grundlagen

Modulziel – angestrebte Lernergebnisse:

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen

- die Grundlagen des Data Mining: den Data Mining-Prozess, notwendige Daten-Vorverarbeitung
- Methoden der Visual Analytics: Scatter plots, Histogramme, Boxplots, Scatter plot matrix, Parallel Coordinates
- Verfahren des maschinellen Lernens
 - unüberwachtes Lernen: partitionierendes Clustering, hierarchisches Clustering
 - überwachtes Lernen: Klassifikations-/Regressionsverfahren, z.B. k-nearest neighbours, support vector machines, Entscheidungsbäume, Lineare Modelle)
- Grundlagen der Programmiersprache Python sowie des Python Data-Science Stacks (numpy, pandas, scikit-learn)

Fertigkeiten – methodische Kompetenzen

Die Studierenden sind in der Lage

- vorhandene Datensätze und Strukturen zu analysieren,
- geeignete Verfahren für bestimmte Probleme auszuwählen
- die erlernten Verfahren mit Hilfe der Programmiersprache Python einzusetzen

Übergreifende Kompetenzen

Die Studierenden können

- konkrete Probleme mit Hilfe von Data Mining Verfahren selbstständig lösen
- selbstständig Wissen aus Daten extrahieren

Inhalt:

Grundlagen des Data Mining:

- Data Mining-Prozess
- Datenvorverarbeitung
- Statistische Grundlagen

Maschinelles Lernen:

- Unüberwachtes Lernen
 - partitionierendes Clustering
 - hierarchisches Clustering
- Überwachtes Lernen

- k-nearest neighbours, support vector machines, Entscheidungsbäume und Erweiterungen, Lineare Modelle

Visual Analytics:

- Scatter plots, Histogramme, Boxplots
- Scatter plot matrix, Parallel Coordinates

Praxis:

- Data Mining-Verfahren mit Python

Literaturhinweise:

Han, Jiawei: Data mining : concepts and techniques. - Amsterdam: Elsevier, 2012. - ISBN: 9780123814791

Hastie, Trevor J: The elements of statistical learning: data mining, inference, and prediction. - New York, NY: Springer, 2013. - ISBN: 9780387848570

Mazza, Riccardo: Introduction to information visualization. - London: Springer, 2009. - ISBN: 9781848002180

Wird angeboten:

In jedem Wintersemester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Übungen, Nachbereitung und Prüfungsvorbereitung
Leistungskontrolle:	Klausur (90 Minuten)
Anteil Semesterwochenstunden:	4 SWS
Geschätzte studentische Arbeitszeit:	150 Stunden

Lernergebnisse:

Das Modul besteht aus theoretischer Vorlesung und einem praktischen Anteilen.

Nach erfolgreicher Absolvierung des Moduls sollen die Studierenden in der Lage sein, ausgewählte Data Mining-Verfahren selbstständig anzuwenden.

Die Studierenden haben Wissen erworben zu häufig verwendeten Data Mining-Verfahren und Big Data-Technologien und sollen in der Lage sein die geeignete Technologie für eine Aufgabenstellung zu wählen. Die Studierenden sollen Wissen erworben haben, das es ihnen erlaubt, selbstständig Wissen aus Daten zu extrahieren.

Bildung der Modulnote:

Klausur

Modulbeschreibung Advanced Data Models

Schlüsselworte: Data Models, NoSQL, Graph Database, Document Store, Semantic Web, Skill Based Engineering, Knowledge Graphs

Zielgruppe:	Semester AIM1 und AIM2	Modulnummer:	AIM 800 6619
Arbeitsaufwand:	5 ECTS		150 h
Davon	Kontaktzeit		50 h
	Selbststudium		100 h
Unterrichtssprache:	Deutsch oder Englisch		
Modulverantwortung:	Dr.-Ing. Jan R. Seyler		
Stand:	01.09.2019		

Empfohlene Voraussetzungen:

Sehr gutes Abstraktionsvermögen, Programmierkenntnisse, Grundverständnis über Datenbanken, Kenntnisse über Modellierungssprachen (bspw. UML)

Modulziel – angestrebte Lernergebnisse:

Die Studierenden lernen sowohl Grundlagen als auch Praxisbeispiele für Systeme, Protokolle und Verfahren zum Management von Kommunikationsnetzen. Sie kennen verschiedene Datenarten und wissen mit welcher Datenbankart sich welche Daten besonders effizient speichern lassen. Des Weiteren haben die Studierenden einen Einblick in den aktuellen Stand der Technik und Forschung und offene Problemstellungen. Das Abstraktionsvermögen und die Problemlösungsfähigkeit werden trainiert. Es werden Forschungsthemen zu neusten Technologien im Bereich Datenmodelle aufgegriffen. Die zwei besten Arbeiten zu einem Forschungsthema werden als Paper auf renommierten Konferenzen eingereicht.

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen

- NoSQL Datenbankmodelle,
- Semantische Webtechnologien wie Knowledge Graphen,
- den Stand der Wissenschaft im Bereich Datenmodelle und Datenmanagement,
- aktuelle, offene Problemstellungen aus Wissenschaft und Industrie im Bereich Datenmodellierung.

Fertigkeiten – methodische Kompetenzen

Die Studierenden können

- zu vorliegenden Daten das richtige Datenmanagementsystem zu finden,
- NoSQL Datenbanken aufzusetzen,
- Queries in N1QL, SPARQL und Cypher formulieren,
- mit LaTeX wissenschaftliche Arbeiten verfassen.

Übergreifende Kompetenzen

Die Studierenden sind in der Lage

- wissenschaftliche Artikel effizient lesen,
- sich in neue Themen eigenständig einarbeiten,
- ein wissenschaftlichen Artikel zu erstellen,
- ihr neues Fachwissen vor einem Team zu präsentieren.

Inhalt:

- Einführung in NoSQL Datenbankmodelle
- Vorstellung von Datenarten
- Das funktionale Modell
- Pitch & Verteilung der Themen
- Regelmäßige Zwischenmeetings während des Selbststudiums bei Festo
- Vorstellung der erarbeiteten Ergebnisse pro Gruppe in einem 30 minütigen Vortrag und mit Hilfe eines funktionalen Prototypen

Literaturhinweise:

- Sadalage, Pramod J., Martin Fowler. "NoSQL distilled." AddisonWesley Professional (2012).
- Sullivan, Dan. "NoSQL for mere mortals" Addison-Wesley Professional, 2015.
- Harrison, Guy. "Next Generation Databases: NoSQL and Big Data". Apress, 2015.
- Yu, Liyang. Introduction to the semantic web and semantic web services. Chapman and Hall/CRC, 2007.
- Siegel, David. Pull: The power of the semantic web to transform your business. Penguin, 2009.
- West, Matthew. Developing high quality data models. Elsevier, 2011.
- Bubenko, Janis A. "From information algebra to enterprise modelling and ontologies—a historical perspective on modelling for information systems." Conceptual Modelling in Information Systems Engineering. Springer, Berlin, Heidelberg, 2007. 1-18.

Wird angeboten:

In jedem Sommersemester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Seminar
Leistungskontrolle:	Referat (30 Minuten), Bericht
Anteil Semesterwochenstunden:	4 SWS
Geschätzte studentische Arbeitszeit:	150 Stunden

Bildung der Modulnote:

Bericht (1), Prototyp mit Referat (1)

Modulbeschreibung Automotive System Design

Schlüsselworte: System- und Softwarearchitekturen sowie Netzwerke und Kommunikationsprotokolle in der automobilen Anwendung, Funktionale Sicherheit

Zielgruppe:	Semester AIM1 und AIM2	Modulnummer:	AIM 800 6610
Arbeitsaufwand:	5 ECTS		150 h
Davon	Kontaktzeit		60 h
	Selbststudium		60 h
	Prüfungsvorbereitung		30 h
Unterrichtssprache:	Englisch oder Deutsch		
Modulverantwortung:	Prof. Dr.-Ing. Werner Zimmermann		
Stand:	01.01.2021		

Empfohlene Voraussetzungen:

Kenntnisse der Informatik und Netzwerke auf Bachelorniveau, Softwareentwicklung in einer gängigen Programmiersprache bevorzugt, C/C++

Modulziel – angestrebte Lernergebnisse:

Die Studierenden können sichere und zuverlässige Elektrik/Elektronik-Systeme im Automobil analysieren, konzipieren und implementieren.

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen

- die Grundlagen der Anwendungsdomänen wie Powertrain, Chassis, Body, Infotainment, Driver Assistance und Automated Driving,
- Grundlagen verteilter Systeme, Kommunikationsanforderungen, Analyse und Gestaltung unter Echtzeit- und Sicherheitsanforderungen,
- die Architektur von Systemen im Automobil und deren Software.

Fertigkeiten – methodische Kompetenzen

Die Studierenden sind in der Lage

- die Zuverlässigkeit und Verfügbarkeit von Systemen zu analysieren,
- funktionale Sicherheitsanforderungen aufzustellen zu implementieren,
- die Netzwerkkommunikation von Systemen im Automobil zu gestalten und quantitativ abzuschätzen.

Übergreifende Kompetenzen

Die Studierenden sind in der Lage

- Systeme im Automobil hinsichtlich Sicherheit und Zuverlässigkeit abschätzen,
- sichere und zuverlässige Systeme im Automobil konzipieren und implementieren.

Inhalt:

- Powertrain, Chassis, Body, Infotainment, Driver Assistance
- Grundlagen verteilter Systeme, Kommunikationsanforderungen
- Analyse und Gestaltung unter Echtzeit- und Sicherheitsanforderungen
- Analysen zur Zuverlässigkeit und Sicherheit
- Architektur von Systemen im Automobil und deren Software
- Funktionale Sicherheitsanforderungen

Literaturhinweise:

- J. Schäufele, T. Zurawka: Automotive Software Engineering. 2016, Springer-Vieweg
- P. Supke: Diagnostic Communication with Road-Vehicles. 2019, VDE-Verlag
- W. Zimmermann, R. Schmidgall: Bussysteme in der Fahrzeugtechnik – Protokolle, Standards und Softwarearchitektur. 2014, Springer-Vieweg

- K. Reif: Bosch Fachinformation Automobil, 2021, Springer-Vieweg

Wird angeboten:

In jedem Wintersemester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung
Leistungskontrolle:	Klausur (60 Minuten)
Anteil Semesterwochenstunden:	3 SWS
Geschätzte studentische Arbeitszeit:	120 Stunden

Lehr- und Lernform:	Projektarbeit
Leistungskontrolle:	Referat (20 Minuten) mit Testat
Anteil Semesterwochenstunden:	1 SWS
Geschätzte studentische Arbeitszeit:	30 Stunden

Lernergebnisse:

Die Studierenden sind in der Lage

- funktionale Sicherheitsanforderungen aufzustellen und zu implementieren
- die Netzwerkkommunikation von Systemen im Automobil zu gestalten und quantitativ abzuschätzen
- zuverlässige Systeme im Automobil konzipieren und implementieren

Bildung der Modulnote:

Testat und Klausur

Modulbeschreibung Business Intelligence

Schlüsselworte: Business Intelligence, Datawarehouse, OLAP, ETL

Zielgruppe:	Semester AIM1 und AIM2	Modulnummer:	AIM 800 6603
Arbeitsaufwand:	5 ECTS		150 h
Davon	Kontaktzeit		60 h
	Selbststudium		60 h
	Prüfungsvorbereitung		30 h
Unterrichtssprache:	Deutsch		
Modulverantwortung:	Prof. Dr. Dirk Hesse		
Stand:	02.01.2021		

Empfohlene Voraussetzungen:

- Programmierkenntnisse (objektorientiert)
- Statistik
- Datenbanken
- Python, R

Modulziel – angestrebte Lernergebnisse:

Kenntnisse – fachliche Kompetenzen

Die Studierenden verfügen über Kenntnisse der grundlegenden Konzepte des Business Intelligence. Sie haben die Fertigkeit verschiedene Ansätze, Methoden und Werkzeuge des Business Intelligence zu unterscheiden und können Unternehmens- /Wettbewerbs- und Kundendaten analysieren. Sie verfügen über die Kompetenz, die vorgestellten Konzepte in das unternehmensweite Informations- und Wissensmanagement zu integrieren.

Die Studierenden kennen

- die Grundbegriffe der Business Intelligence
- die Bedeutung von BI für die unternehmerische Praxis
- die Grundlagen entscheidungsunterstützender Systeme
- verschiedene Konzepte und Methoden der BI

Fertigkeiten – methodische Kompetenzen

Die Studierenden sind in der Lage

- heterogene Daten im Rahmen des ETL Prozesses aufzubereiten,
- multidimensionale Speicherstrukturen zu erstellen,
- Grundlegende Charakteristiken von Datawarehouse und Datamart zu definieren,
- verschiedene Instrumente und Anwendungen der BI zielgerichtet einzusetzen,
- Entscheidungsmodelle mit Hilfe einer höheren Programmiersprache zu entwickeln.

Übergreifende Kompetenzen

Die Studierenden können

- für spezifische Anwendungsfelder geeignete BI-Verfahren auswählen und umsetzen.

Inhalt:

Vertiefung der Methodenkompetenz im Bereich Business Analytics:

- Verarbeitung (semi-) strukturierter Daten im ETL Prozess
- Logische Modellierung (Star Schema, Snowflake Schema etc.)
- Einrichtung von multidimensionalen Modellen (OLAP Cubes)
- Reporting und Analyse mittels verschiedener Tools, Queries und Webreports
- Programmierung entscheidungsunterstützender Modelle

Literaturhinweise:

- Business Intelligence and Analytics: Systems for Decision Support, Sharda, Turba, Delen, Pearson Education Limited, 10. Auflage 2014.
- Analytics, Data Science, & Artificial Intelligence: Systems for Decision Support, Sharda et al. 2019
- Business Intelligence & Analytics – Grundlagen und praktische Anwendungen: Eine Einführung in die IT-basierte Managementunterstützung, Kemper, Baars, Springer Vieweg, 4. Auflage, 2020.

Wird angeboten:

Wintersemester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform: Vorlesung mit Seminar
Leistungskontrolle: Klausur (90 Minuten)
Anteil Semesterwochenstunden: 3 SWS
Geschätzte studentische Arbeitszeit: 120 Stunden

Lehr- und Lernform: Projektarbeit (Labor)
Leistungskontrolle: Referat, Testat
Anteil Semesterwochenstunden: 1 SWS
Geschätzte studentische Arbeitszeit: 30 Stunden

Lernergebnisse:

Die Studierenden haben die Fertigkeit BI-Systeme zu modellieren und zu implementieren. Sie können ETL Prozesse aufsetzen und parametrisieren. Sie haben erste Erfahrungen beim Einsatz von BI-Tools und Programmen. Sie sind in der Lage Programme zur Visualisierung und Entscheidungsunterstützung selbständig in einer höheren Programmiersprache zu erstellen.

Bildung der Modulnote:

Klausur

Modulbeschreibung Cloud Computing

Schlüsselworte: Lights out computing, IaaS, PaaS, Container, Continuous Integration / Continuous Delivery, Cloud Functions

Zielgruppe:	Semester AIM1 und AIM2	Modulnummer:	AIM 800 6620
Arbeitsaufwand:	5 ECTS		150 h
Davon	Kontaktzeit		120 h
	Selbststudium		15 h
	Prüfungsvorbereitung		15 h
Unterrichtssprache:	Deutsch		
Modulverantwortung:	Dipl.-Ing. Simon Moser		
Stand:	01.09.2019		

Empfohlene Voraussetzungen:

Kenntnisse in:

- Grundlagen der Informatik, Web-Technologien, Rechnernetze,
- Programmierung und Softwareentwicklung.

Modulziel – angestrebte Lernergebnisse:

Die Studierenden werden grundlegende Kenntnisse und Fertigkeiten im Umgang mit gängigen Techniken und Werkzeugen aus allen Bereichen des Cloud Computing erwerben und anwenden können. Dazu werden im Laufe des Moduls drei unterschiedliche Perspektiven auf das Thema Cloud Computing eingenommen:

1. Cloud Computing Sicht eines Cloud Providers („Wie baut man eine Cloud?“)
 - Sie verstehen Virtualisierung, Software defined Networking und programmierbare Infrastruktur über alle Ebenen hinweg.
 - Sie verstehen Cloud Automatisierung und Lights-out-Computing Konzepte.
2. Cloud Computing aus der Sicht eines Anwendungsentwicklers („Wie entwickelt man Software für eine Cloud?“)
 - Sie verstehen die Bedeutung von Cloud Computing für die heutige Softwareentwicklung (Stichwort: Microservices)
 - Sie sind mit unterschiedlichen Entwicklungsmethoden und Prozessen hinsichtlich Cloud Computing vertraut, z.B. CI/CD und DevOps
 - Sie sind in der Lage ein Konzept zur Migration von Anwendungen in die Cloud zu erarbeiten und können eine Cloud-Anwendungsarchitektur entwickeln, sowie deren Implementierungsentscheidung detailliert begründen
3. Cloud Computing aus der Sicht eines Nutzers („Was muss man bei der Nutzung einer Cloud beachten?“)

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen

- Grundbegriffe des Cloud Service Models: *aaS, z.B. IaaS, PaaS, SaaS
- Grundlagen der Cloud Betriebsformate: Public, Private und Hybrid Clouds
- Elemente einer (Infrastruktur) Cloud: Virtualisierung, Server, Netzwerke, Storage, ...
- Grundprinzipien des Software Engineerings: Modularisierung, Abstraktion, ...
- Grundlegende Web-Konzepte: HTTP, REST, ...

Fertigkeiten – methodische Kompetenzen

Die Studierenden sind in der Lage

- Brücken zwischen physikalischer und virtueller Infrastruktur zu bilden.
- Zu verstehen wie Infrastruktur programmiert werden kann und welche Möglichkeiten dies bietet.
- Zu verstehen, was DevOps für die Rolle eines Softwareentwicklers bedeutet
- Zu verstehen, welche Möglichkeiten das *-as-a-service Model für einen Softwareentwickler bedeutet

- REST Web Services als Microservices in z.B. Docker Images zu deployen.

Übergreifende Kompetenzen

Die Studierenden können

- Wissen aus verschiedenen Domänen integrieren.
- in einem Team gemeinsam eine Aufgabe erfolgreich umzusetzen.
- verteilte Web-Architekturen mit Hilfe von Web Services konzipieren.

Inhalt:

Cloud Computing ist ein umgangssprachlich sehr überladener Begriff, der eine Obermenge verschiedenster Technologien und Use Cases bildet. In der Vorlesung wird dem Studenten sowohl ein Überblick als auch ein detaillierter Einblick in die Teilmengen des Themas vermittelt – beginnend von einer Begriffsklärung werden, anhand von praxisnahen Beispielen, drei Teilmengen des Themas „Cloud Computing“ beleuchtet, erläutert und selbst erforscht. Dabei werden im Verlauf drei Sichten auf das Thema eingenommen:

1. Cloud Computing aus der Sicht eines Cloud Providers („Wie baut man eine Cloud?“)
2. Cloud Computing aus der Sicht eines Anwendungsentwicklers („Wie entwickelt man Software für eine Cloud?“)
3. Cloud Computing aus der Sicht eines Nutzers („Was muss man bei der Nutzung einer Cloud beachten?“)

Jedes Teilgebiet wird zunächst konzeptionell und theoretisch aufgearbeitet und anschließend in mehreren Übungen praktisch vertieft.

Literaturhinweise:

- Thomas Erl: Cloud Computing: Concepts, Technology & Architecture (The Prentice Hall Service Technology Series from Thomas Erl)
- Sjaak Laan: IT Infrastructure Architecture – Infrastructure Building Blocks and Concepts, Third Edition
- Betsy Beyer, et al: Site Reliability Engineering, <https://landing.google.com/sre/books/>
- Gene Kim: The Phoenix Project – a novel about IT, DevOps and Helping Your Business Win, 2018, IT Revolution Press
- Bernd Öggl, Michael Kofler: Docker – Das Praxisbuch für Entwickler und DevOps-Teams, 2. Auflage, 2020, Rheinwerk
- Kelsey Hightower, Brendan Burns, Joe Beda: Kubernetes – eine kompakte Einführung, 2018, d.Punkt Verlag
- Publikationen zum Thema „Serverless Computing“

Wird angeboten:

In jedem Sommersemester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Übungen
Leistungskontrolle:	Klausur (90 Minuten)
Anteil Semesterwochenstunden:	4 SWS
Geschätzte studentische Arbeitszeit:	150 Stunden

Bildung der Modulnote:

Klausur oder mündliche Prüfung
Die Prüfungsform wird zu Semesterbeginn bekanntgegeben.

Modulbeschreibung Data Fusion

Schlüsselworte: Kalman-Filter, Bayes Netze, Markov Modelle

Zielgruppe:	Semester AIM1 und AIM2	Modulnummer:	AIM 800 6615
Arbeitsaufwand:	5 ECTS		150 h
Davon	Kontaktzeit		45 h
	Selbststudium		75 h
	Prüfungsvorbereitung		30 h
Unterrichtssprache:	Deutsch		
Modulverantwortung:	Prof. Dr.-Ing. Reiner Marchthaler		
Stand:	01.09.2019		

Empfohlene Voraussetzungen:

Grundlagen Statistik und Automatentheorie, im Speziellen in Zustandsautomaten

Modulziel – angestrebte Lernergebnisse:

Die Studierenden beherrschen die mathematischen Konzepte zum Entwurf von Kalman-Filtern, Markov-Ketten und Bayes-Netzwerke. Darüber hinaus sind Sie in der Lage selbstständig ein Kalman-Filter zur Sensordatenfusion zu entwerfen.

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen

- die Grundlagen der Kalman-Filter-Gleichungen,
- die hier zu Grunde liegenden Annahmen,
- die Grundlagen von Markov-Modellen, Partikelfilter und Bayes-Netzen.

Fertigkeiten – methodische Kompetenzen

Die Studierenden sind in der Lage Zustandsmodelle zu modellieren.

Übergreifende Kompetenzen

Die Studierenden können in praktischen Übungen Kalman-Filter in einem Mikrocontroller implementieren.

Inhalt:

- Kalman-Filter und Partikelfilter
- Markov Decision Processes und Hidden Markov Modelle
- Bayes Netzwerke

Literaturhinweise:

- Richard E. Neapolitan: „Learning Bayesian Networks“. Prentice Hall, 2003
- Olle Häggström: “Finite Markov Chains and Algorithmic Applications“. Cambridge University Press, 2002
- Brown, R. G.; Hwang, P. Y. C.: „Introduction to Random Signals and Applied Kalman Filtering“. Third Edition. John Wiley & Sons, Inc., 2012
- Simon, Dan: “Optimal State Estimation“, John Wiley & Sons, Inc., 2006
- Thrun, S.; u.a. „Probabilistic Robotics“, MIT Press, 2005
- Marchthaler, R.; Dingler S. „Kalman-Filter - Einführung in die Zustandsschätzung und ihre Anwendung“, Springer-Vieweg Verlag, 2016

Wird angeboten:

In jedem Sommersemester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform: Vorlesung mit Übungen
Leistungskontrolle: Klausur (90 Minuten)
Anteil Semesterwochenstunden: 4 SWS
Geschätzte studentische Arbeitszeit: 150 Stunden

Bildung der Modulnote:
Klausur

Modulbeschreibung Digitale Forensik

Schlüsselworte: Datenwiederherstellung, Forensics Field Set, Beweissicherung und Dokumentation

Zielgruppe:	Semester AIM1 und AIM2	Modulnummer:	AIM 800 6631
Arbeitsaufwand:	5 ECTS		150 h
Davon	Kontaktzeit		60 h
	Selbststudium		60 h
	Prüfungsvorbereitung		30 h
Unterrichtssprache:	Deutsch		
Modulverantwortung:	Prof. Dr.-Ing. Tobias Heer		
Stand:	09.02.2021		

Empfohlene Voraussetzungen:

- Betriebssysteme, Dateisysteme
- Rechnernetze
- IT Sicherheit

Modulziel – angestrebte Lernergebnisse:

Die Studierenden werden in die Lage versetzt, sich am digitalen Tatort rechtskonform zu verhalten und forensische Indizien zu einer Beweiskette zusammenzufassen.

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen

- Die Grundlagen der digitalen Forensik,
- Die Entstehung, der Manipulier- und Kopierbarkeit sowie der Personenbezogenheit von digitalen Spuren,
- Konzepte und Eigenschaften von Dateisystemen.

Fertigkeiten – methodische Kompetenzen

Die Studierenden können

- eine Dateisystemanalyse durchführen,
- gelöschte Dateien von Speichermedien wiederherstellen,
- allgemeine und spezielle forensische Tools sicher anwenden,
- forensische Analysen von Anwendungen durchführen,
- eine Analyse und Auswertung von Smartphones entwickeln,
- ein Forensics Field Set selbstständig planen und aufbauen,
- evidenzbasierte Hinweise bewerten und einfache Beweiskette zusammenfassen.

Übergreifende Kompetenzen

Die Studierenden sind in der Lage

- sich am digitalen Tatort rechtskonform zu verhalten,
- Indizien auswerten und Beweisketten zu synthetisieren.

Inhalt:

- Aufgaben der digitalen Forensik
- Rechtskonformes Verhalten am digitalen Tatort
- Field Set: Die Werkzeuge der digitalen Forensik
- Methoden und Werkzeuge zur Datensicherung und Datenanalyse
- Analyse der Windows Registry, Dateisystem, Browser Forensics
- Flash Speicher, Struktur und Inhalt wichtiger Verzeichnisse und Dateien
- Übersicht Cloud Forensik, Post Mortem und Live Analyse,
- Dokumentation in Form eines umfassendes forensischen Handbuchs

Literaturhinweise:

- T Lorenz Kuhlee, Victor Völzow: Computer Forensik Hacks. O'Reilly, 2012, ISBN 978-3868991215.
- John R. Vacca, K. Rudolph: Computer Forensics: Computer Crime Scene Investigation, Jones & Bartlett Publ., 2010, ISBN 978-0763791346.
- Cory Altheide, Harlan Carvey: Digital Forensics with Open Source Tools. Syngress, ASIN B00LI84Y28.

Wird angeboten:

In jedem Sommersemester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Übungen und Projektarbeit
Leistungskontrolle:	Klausur (90 Minuten)
Anteil Semesterwochenstunden:	4 SWS
Geschätzte studentische Arbeitszeit:	150 Stunden

Bildung der Modulnote:

Klausur

Modulbeschreibung High Performance Computing

Schlüsselworte: Parallele Programmierung und Algorithmen, Performance

Zielgruppe:	Semester AIM1 und AIM2	Modulnummer:	AIM 800 6618
Arbeitsaufwand:	5 ECTS		150 h
Davon	Kontaktzeit		50 h
	Selbststudium		60 h
	Prüfungsvorbereitung		40 h
Unterrichtssprache:	Englisch oder Deutsch		
Modulverantwortung:	Prof. Dr.-Ing. Rainer Keller		
Stand:	14.01.2021		

Empfohlene Voraussetzungen:

Programmierkenntnisse in C / C++

Modulziel – angestrebte Lernergebnisse:

Die Studierenden sind in der Lage, eigenen parallelen Code für CPU und GPU mit verschiedenen Programmierparadigmen für Systeme mit gemeinsamen und verteilten Speicher zu schreiben und vorhandenen Code zu parallelisieren. Sie haben eine Auswahl an Parallelisierungsparadigmen und einen Überblick und praktische Erfahrung in der Nutzung von Tools um diese anzuwenden.

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen

- Grundlagen paralleler und nebenläufiger Programme:
 - Kommunikation über schnelle Netzwerke
 - Moderne Synchronisationsmechanismen
 - Fallstricke wie Deadlocks, Priority Inversion, etc.
- Werkzeuge, Technologien und Frameworks zur parallelen Programmierung,
- Die architektonischen Unterschiede von CPU, GPU und weiteren Beschleunigern.

Fertigkeiten – methodische Kompetenzen

Die Studierenden sind in der Lage

- Strategien zur Parallelisierung von Softwaresystemen beurteilen und auswählen,
- Verschiedene Programmierschnittstellen anzuwenden, z.B. OpenMP, MPI, OpenCL und OpenACC für GPU-Programmierung,
- Werkzeuge zum parallelen Debuggen anzuwenden,
- Performance Analysen bei parallelen Systemen durchzuführen.

Übergreifende Kompetenzen

Die Studierenden können

- Die Performance von Softwaresysteme mit Hilfe paralleler Programmierung verbessern,
- Fehler in parallelen Programmen erkennen und beheben,
- Die erlernten Fähigkeiten auf andere Programmierparadigmen und -sprachen anwenden.

Inhalt:

- Einführung in die parallele Programmierung
- Parallele Programmierung mit Threads & OpenMP und nebenläufigem Code
- Parallele Programmierung mit MPI
- Parallele Programmierung für GPUs mittels OpenCL und OpenACC
- Effizienz von parallelen Algorithmen
- Performance Analyse Tools
- Verwendung von parallelen Debuggern

Literaturhinweise:

- David R. Butenhof: Programming with POSIX Threads, Addison-Wesley, ISBN 0201633922
- Paul Butcher: Seven Concurrency Models in Seven Weeks, The Pragmatic Programmer, ISBN 1937785653
- Georg Hager, Gerhard Wellein: Introduction to High Performance Computing for Scientists and Engineers. Chapman&Hall, ISBN 978-1439811924
- OpenMP Spezifikation
- MPI-Standard
- OpenCL Standard

Wird angeboten:

In jedem Sommersemester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Übungen
Leistungskontrolle:	Klausur (90 Min.)
Anteil Semesterwochenstunden:	4 SWS
Geschätzte studentische Arbeitszeit:	150 Stunden

Bildung der Modulnote:

Klausur

Modulbeschreibung Information Security Management

Schlüsselworte: Informationssicherheit, Incident Response, Sicherheitsmaßnahmen

Zielgruppe:	Semester AIM1 und AIM2	Modulnummer:	AIM 800 6624
Arbeitsaufwand:	5 ECTS		150 h
Davon	Kontaktzeit		120 h
	Selbststudium		15 h
	Prüfungsvorbereitung		15 h
Unterrichtssprache:	Deutsch oder Englisch		
Modulverantwortung:	Markus Schlittenhardt		
Stand:	23.01.2021		

Empfohlene Voraussetzungen:

Kenntnisse in Rechnernetzen, Netzwerke und Betriebssysteme

Modulziel – angestrebte Lernergebnisse:

Die Studierenden werden in die Lage versetzt, die Informationssicherheit in einer Organisation zu managen. Die Studierenden sind in der Lage zu verstehen, was Informationssicherheit innerhalb einer globalen Organisation bedeutet und wie diese aufgebaut werden kann. Zusätzlich werden die Studenten in die Lage versetzt zu verstehen, wie eine Organisation auf Sicherheitsvorfälle innerhalb der IT angemessen reagieren kann und welche Voraussetzungen dafür geschaffen werden müssen.

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen

- die komplexen Auswirkungen von Sicherheitsanforderungen innerhalb der IT auf eine Organisation,
- die Funktionsweise eines Managementsystems für Informationssicherheit,
- die Auswirkungen von Risiken innerhalb der Informationstechnik auf eine Organisation und wie diese damit umgehen kann,
- die Voraussetzungen, die geschaffen werden müssen, dass eine Organisation auf Sicherheitsvorfälle innerhalb der IT reagieren kann,
- die Vorgehensweise auf Sicherheitsvorfälle angemessen zu reagieren und wie diese Vorfälle systematisch analysiert werden können.

Fertigkeiten – methodische Kompetenzen

Die Studierenden sind in der Lage

- auf Managementebene über Risiken innerhalb der Informationssicherheit zu diskutieren,
- auf Sicherheitsvorfälle innerhalb der Informationssicherheit zu reagieren sowie diese zu analysieren,
- Schwachstellen innerhalb einer Infrastruktur zu identifizieren und angemessen damit umzugehen.

Übergreifende Kompetenzen

Die Studierenden können

- komplexe Zusammenhänge und Auswirkungen der Informationssicherheit auf eine Organisation zu verstehen.

Inhalt:

- Einführung in die Informationssicherheit
- Technische Grundlagen für das Verständnis von Informationssicherheit
- Organisatorische Grundlagen für das Verständnis von Informationssicherheit
- Angriffe verstehen, analysieren und verhindern

Literaturhinweise:

- Thomas W. Harich; IT-Sicherheitsmanagement: Praxiswissen für IT Security Manager, mitp-Verlag, ISBN 978-3-95845-275-6, 2018
- André Domnick, et al; Informationssicherheit und Datenschutz: Handbuch für Praktiker und Begleitbuch zum T.I.S.P. dpunkt Verlag, ISBN 978-3-86490-596-4, 2019
- Peter H. Gregory; Certified Information Security Manager, McGraw-Hill, ISBN 978-1-26002-7037, 2018

Wird angeboten:

In jedem Wintersemester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Übungen und Projektarbeit
Leistungskontrolle:	Klausur (90 Minuten)
Anteil Semesterwochenstunden:	4 SWS
Geschätzte studentische Arbeitszeit:	150 Stunden

Bildung der Modulnote:

Klausur

Modulbeschreibung Intelligent Time Series Analytics

Schlüsselworte: Big Data, Data Mining, Zeitreihen, Klassifikation, Vorhersage, Querying, Machine Learning

Zielgruppe:	Semester AIM1 und AIM2	Modulnummer:	AIM 800 6632
Arbeitsaufwand:	5 ECTS		150 h
Davon	Kontaktzeit		60 h
	Selbststudium		60 h
	Prüfungsvorbereitung		30 h
Unterrichtssprache:	Englisch		
Modulverantwortung:	Prof. Dr. Gabriele Gühring		
Stand:	24.01.2021		

Empfohlene Voraussetzungen:

Kenntnisse in

- Mathematik, Statistik und Optimierung.
- Kenntnisse in Informatik, Programmieren

Modulziel – angestrebte Lernergebnisse:

Die Studierenden besitzen Grundkenntnisse in „Data Mining auf Zeitreihen“ und im Umgang mit der Software R oder Python. Sie sind in der Lage, ausgewählte Verfahren aus den Funktionalitäten Querying, Klassifikation und Vorhersage auf Zeitreihen anzuwenden. Die gelernten Methoden und Konzepte können zum Zwecke des Data Mining auch auf andere Datentypen angewandt werden.

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen

- Grundlagen der Zeitreihen,
- Anwendungen, in denen Zeitreihen generiert und aufgezeichnet werden,
- Verfahren der Klassifikation von Zeitreihendaten,
- Verfahren zur Regressionsanalyse und zur Vorhersage,
- Grundlagen der künstlichen Neuronalen Netze zur Verarbeitung von Zeitreihen.

Fertigkeiten – methodische Kompetenzen

Die Studierenden sind in der Lage

- geeignete Analyseverfahren auszuwählen und anzuwenden

Übergreifende Kompetenzen

Die Studierenden können

- Zeitreihen mit Hilfe von Algorithmen aus den Bereichen Data Mining und maschinelles Lernen intelligent analysieren.

Inhalt:

- Introduction to Data Mining with a focus on Time Series Data (Temporal Data Mining)
- Fundamentals of Time Series Data
- Classification, Time Series Querying, Regression/Forecasting
- Visualization of Time Series
- Artificial Neural Networks

Literaturhinweise:

- T. Mitsa: Temporal Data Mining. Data Mining and Knowledge Discovery. Chapman & Hall/CRC, 2010
- J. Han, M. Kamber, J. Pei: Data Mining – Concepts and Techniques, 3rd ed., Morgan Kaufman, 2012
- R. J. Hyndman, G. Athanasopoulos: Forecasting: principles and practice. Available online at <https://www.otexts.org/fpp>, 3rd ed., 2014
- R. H. Shumway, D. S. Stoffer: Time Series Analysis and Its Applications, 4th ed., Springer, 2027

Wird angeboten:

In jedem Sommersemester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Übungen
Leistungskontrolle:	Klausur (90 Minuten)
Anteil Semesterwochenstunden:	4 SWS
Geschätzte studentische Arbeitszeit:	150 Stunden

Bildung der Modulnote:

Klausur

Modulbeschreibung Mobile Communication

Schlüsselworte: Automotive, Communication, Safety, Security

Zielgruppe:	Semester AIM1 und AIM2	Modulnummer:	AIM 800 6601
Arbeitsaufwand:	5 ECTS		150 h
Davon	Kontaktzeit		60 h
	Selbststudium		60 h
	Prüfungsvorbereitung		30 h
Unterrichtssprache:	Englisch		
Modulverantwortung:	Prof. Dr.-Ing. Harald Melcher, Prof. Dr. Dominik Schoop		
Stand:	14.01.2021		

Empfohlene Voraussetzungen:

Grundlagen Kommunikationstechnik
Kenntnisse aus den Modulen Security Engineering und Advanced Software Engineering

Modulziel – angestrebte Lernergebnisse:

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen

- die Grundlagen der drahtlosen Übertragungstechnik
- Drahtlose Techniken rund um's Fahrzeug: DAB, RFID, Bluetooth LE, ZigBee
- Die Grundlagen der Intelligent Transportation Systems (ITS) und der Vehikuläre Ad-Hoc-Netzwerke (VANETs)
- Technologie von V2X-Anwendungen
- Automotive wireless Netztechnologie (WLAN / IEEE 802.11p, CAM, DENM)
- relevante Standards (IEEE, ETSI)
- Systeme zur Positionsbestimmung (GPS, ...)
- den Zusammenhang Safety und Security für ITS
- Routingverfahren und speziell Routing in VANETs
- Authentizität und Anonymität in VANETs

Fertigkeiten – methodische Kompetenzen

Die Studierenden sind in der Lage

- die Ziele von ITS und VANETs zu erklären
- die Architektur und Technologie von VANETs zu erklären
- Simulation von Straßenverkehr und Funkverkehr

Übergreifende Kompetenzen

Die Studierenden können

- einfache V2X-Applikationen zu implementieren.

Inhalt:

- Radio Grundlagen
- IEEE 802.11p
- IPv6
- Routing
- Location Finding
- Digital Audio Broadcast (DAB)
- Radio Frequency Identification (RFID)
- Bluetooth LE
- ZigBee
- Einführung in ITS
- V2X Use Cases
- V2X Technologie und Routing

- Authentizität und Anonymität mit V2X

Literaturhinweise:

- Christoph Sommer, Falko Dressler: Vehicular Networking. Cambridge University Press, 2014
- Erdal Cayirci, Chunming Rong: Security in Wireless Ad Hoc and Sensor Networks, John Wiley & Sons, 2009
- Srikanta Patnaik, Xiaolong Li, Yeon-Mo Yang: Recent Development in Wireless Sensor and Ad-hoc Networks, Springer, 2014

Wird angeboten:

In jedem Wintersemester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung
Leistungskontrolle:	Klausur (60 Minuten)
Anteil Semesterwochenstunden:	3 SWS
Geschätzte studentische Arbeitszeit:	120 Stunden

Lehr- und Lernform:	Projektarbeit
Leistungskontrolle:	Bericht und Referat (20 Minuten)
Anteil Semesterwochenstunden:	1 SWS
Geschätzte studentische Arbeitszeit:	30 Stunden

Bildung der Modulnote:

Klausur (3) und Testat (2)

Modulbeschreibung Motion Planning for autonomous Systems

Schlüsselworte: Robotik, Bewegungsplanung, Pfadplanung, Verhaltensplanung

Zielgruppe:	Semester AIM1 und AIM2	Modulnummer:	AIM 800 6622
Arbeitsaufwand:	5 ECTS		150 h
Davon	Kontaktzeit		120 h
	Selbststudium		15 h
	Prüfungsvorbereitung		15 h
Unterrichtssprache:	deutsch		
Modulverantwortung:	Prof. Dr.-Ing. Thao Dang		
Stand:	01.01.2020		

Empfohlene Voraussetzungen:

- Lineare Algebra
- Algorithmen und Datenstrukturen
- Grundlegende Programmierkenntnisse in Python

Modulziel – angestrebte Lernergebnisse:

Die Studierenden beherrschen grundlegende Methoden der Bewegungsplanung in der Robotik. Im Speziellen verstehen sie die speziellen Anforderungen der Bewegungsplanung bei nicht-holonomen Robotern wie z.B. automatisierten Fahrzeugen. Sie sind in der Lage, Algorithmen der Bewegungsplanung für selbstfahrende Fahrzeuge selbst zu entwerfen und zu implementieren.

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen:

- die Grundbegriffe der Bahn- und Trajektorienplanung in der Robotik,
- Bewegungsmodelle für selbstfahrende Fahrzeuge,
- grundlegende Algorithmen zur Bestimmung eines kostenoptimalen Pfades (diskrete Suchverfahren, probabilistische Verfahren, Methoden der Optimalsteuerung),
- Verhaltensgenerierung mit State Charts.

Fertigkeiten – methodische Kompetenzen

Die Studierenden sind in der Lage

- ein Bahn- oder Trajektorienplanungsproblem für mobile Roboter als diskretes oder kontinuierliches Optimierungsproblem mit Nebenbedingungen zu formulieren
- Lösungsverfahren für solche Optimierungsprobleme auszuwählen und zu implementieren

Übergreifende Kompetenzen

Die Studierenden können

- Planungs- und Regelungskomponenten für Roboter entwerfen und bewerten.

Inhalt:

- Übersicht Pfad- und Trajektorienplanung
- Einführung Robot Operating System (ROS)
- Verhaltensplanung für autonome Fahrzeuge
- Fahrzeugdynamik und Fahrzeugregelung
Ackermann, Einspurmodell, Deichselregler, Stanley
- Grundlagen Bewegungsplanung in der Robotik
Grundbegriffe, Konfigurationsraum, Kollisionsberechnung
- Grundlegende Planungsverfahren
Bug Approaches, Potentialverfahren, Suchverfahren, A*
- Probabilistische Planungsverfahren
Probabilistic Roadmaps, RRTs
- Bewegungsplanung und Regelung mit State Lattice Planern
- Einführung Reinforcement Learning
- Ausgewählte Themen der Absicherung, Implementierung von Planungsalgorithmen in ROS und Systemintegration zur Verhaltenssteuerung automatisierter Fahrzeuge

Literaturhinweise:

- Howie Choset, Principles of Robot Motion - Theory, Algorithms and Implementation, MIT Press, 2005.
- Steven M. LaValle, Planning Algorithms, Cambridge University Press, 2006.

Wird angeboten:

In jedem Wintersemester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Übungen
Leistungskontrolle:	Klausur (90 Minuten)
Anteil Semesterwochenstunden:	4 SWS
Geschätzte studentische Arbeitszeit:	150 Stunden

Bildung der Modulnote:

Klausur

Course Description Network Security

Keywords: Secure Protocols, Authentication, Identity Management, Firewalls, Intrusion Detection

Audience:	Semester AIM 1 and AIM 2	Modul Number:	AIM 800 6625
Workload:	5 ECTS		150 h
divided into	Contact time		90 h
	Self-study		30 h
	Exam preparation		30 h
Course language:	English		
Modul director:	Prof. Dr. Tobias Heer		
Valid as of:	17.12.2020		

Recommended requirements:

Understanding of computer networks, IT security and cryptography fundamentals, basic programming skills

Desired learning outcomes of the module:

Students understand how to protect networks using both basic and advanced security methods.

Knowledge - professional competences

Students know:

- Network security objectives and basic attacks
- Security models for network protocols
- Cryptographic basics for network security protocols
- Security mechanisms on different network layers (PPP, IPsec, TLS, SSH)
- Authentication frameworks and identity management (e.g. OAuth, Kerberos, RADIUS)
- Basic protection solutions and devices (e.g., firewalls, VLAN, VPN, network monitoring, fail2ban)
- Advanced security mechanisms and algorithms (e.g., intrusion detection, honeypots)

Skills - methodical competences

Students are able to

- Perform a security risk analysis for complex network deployments
- Select and implement network security methods
- Segment networks into security zones
- Design networks with regard to security
- Understand and use network security devices
- Understand anonymization techniques and their limitations

Comprehensive Competencies

Students be able to

- Deploy secure networked applications and IT services
- Leverage advanced concepts in network security

Contents:

- Network security goals, attacks and protection mechanisms
- Security mechanisms in the Internet (e.g., VLAN, IEEE 802.1X, IPsec, OpenVPN, TLS, SSH)
- Design and functions of network security protocols
- Authentication frameworks and identity management (e.g., Single-Sign-On, OAuth, Kerberos, PKI)
- Network attacks and counter-measures (e.g., firewalls, intrusion detection systems,)
- Advanced security solutions and research (e.g., intrusion detection, honeypots)
- Secure network operation and network monitoring

Literature:

- W. Stallings: Network Security Essentials, Pearson Prentice Hall, 2007
- N. Ferguson, B. Schneier: Practical Cryptography John Wiley & Sons, 2003
- G. Schäfer, M. Roßberg: Netzsicherheit, 2. Auflage, dpunkt Verlag, 2014
- C. Eckert: IT-Sicherheit, Konzepte-Verfahren-Protokolle, Oldenbourg-Verlag, 2011
- R. Anderson: Security Engineering, Wiley, 2009
- B. Schneier: Applied Cryptography. Protocols, Algorithms, and Source Code in C. Wiley, New York 1996.

Offered:

Every summer semester

Submodules and Assessment:

Type of instruction:	Lecture with exercises and project work
Type of assessment:	Exam (90 minutes)
Hours per week:	4 SWS
Estimated student workload:	150 hours

Generation of the module grade:

Exam

Modulbeschreibung Penetration Testing

Schlüsselworte: IT-Sicherheit, Pentesting, Offensive Security

Zielgruppe:	Semester AIM1 und AIM2	Modulnummer:	AIM 800 6630
Arbeitsaufwand:	5 ECTS		150 h
Davon	Kontaktzeit		60 h
	Selbststudium		90 h
	Prüfungsvorbereitung		0 h
Unterrichtssprache:	Deutsch		
Modulverantwortung:	Prof. Dr.-Ing. Tobias Heer		
Stand:	26.01.2021		

Empfohlene Voraussetzungen:

Kenntnisse über den Aufbau von Web-Applikationen und grundlegender Umgang mit dem Betriebssystem Linux

Modulziel – angestrebte Lernergebnisse:

Um IT-Systeme auf Schwachstellen zu prüfen, ist ein Einblick in die Denkweise und Techniken von Angreifern unverzichtbar. Das Modul gibt einen Überblick über die offensive Seite der IT-Sicherheit und behandelt typische Schwachstellen und Angriffsmethoden.

Die Studierenden haben einen Überblick über die Vorgehensweise bei Angriffen auf IT-Systeme. Sie wissen um die verfügbaren Tools und Methoden im Bereich der offensiven Sicherheit. Sie sind in der Lage, sicherheitsrelevante Informationen aus öffentlichen Quellen zu beschaffen (OSINT), verschiedene Schwachstellentypen in Web-Applikationen und Binäranwendungen zu erkennen, zu bewerten und auszunutzen sowie Gegenmaßnahmen vorzuschlagen.

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen

- offensive Sicherheitsmethoden und ihre Ziele im Kontext der IT-Sicherheit,
- die wichtigsten Schwachstellen von IT-Systemen,
- die Methodik von Penetrationstests,
- gängige Risikofaktoren und Angriffsarten,
- die rechtlichen und moralischen Rahmenbedingungen bei der Anwendung offensiver Sicherheitsmethoden.

Fertigkeiten – methodische Kompetenzen

Die Studierenden sind in der Lage

- die wichtigsten Tools des Penetration Testing anzuwenden,
- sich Zugang zu Systemen über Malware und offensive Sicherheitswerkzeuge zu verschaffen,
- den Zugang zu übernommenen Systemen zu verstetigen,
- relevante Informationen aus öffentlichen Quellen ermitteln und eine Analyse der Informationsfläche eines Ziels/Unternehmens durchzuführen,
- Code auf Schwachstellen hin zu analysieren,
- Schwachstellen zu entdecken und anhand von CVSS und anderen Metriken bewerten,
- eine Sicherheitsüberprüfung durch einen Penetrationstest durchzuführen.

Übergreifende Kompetenzen

Die Studierenden sind in der Lage

- Die IT-Sicherheit von IT-Systemen zu prüfen und zu bewerten.

Inhalt:

- Methodik und Ablauf eines Penetrationstests
- Rechtliche Stolperstricke und Rechtslage
- Angriffsfläche und des Informations-Footprint eines Unternehmens
- Erkennen und Bewerten typischer Schwachstellen in IT-Systemen
- Angriffstypen, Angriffsvektoren von Windows und Linux Systemen
- Programmatische Erzeugung von Malware
- Malware-Verschleierung und Angriffswerkzeuge
- Persistenten Zugang über verschiedene Methoden erreichen
- Praktische Durchführung von Angriffen
- Pivoting und Lateral Movement
- Praktische Übungen basierend auf realistischen Angriffsszenarien

Literaturhinweise:

- Kim, P.: The Hacker Playbook 2, A practical Guide to Penetration Testing, Secure Planet LLC, 2015
- Hadnagy, C.: Social Engineering, The Art of Human Hacking, Wiley Publishing Platform, ISBN 978-1494932633, 2011
- Stuttard, D., Pinto, M.: The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws, John Wiley & Sons, 2. Auflage, 2011, ISBN 978-1118026472 2011
- Erickson, J.: Hacking, The Art of Exploitation, No Starch Press, 2. Auflage, 2007
- Messner, M.: Metasploit: Das Handbuch zum Penetration-Testing-Framework, dpunkt.Verlag, 2015

Wird angeboten:

In jedem Wintersemester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung und Projektarbeit
Leistungskontrolle:	Klausur (90 Minuten)
Anteil Semesterwochenstunden:	4 SWS
Geschätzte studentische Arbeitszeit:	150 Stunden

Bildung der Modulnote:

Klausur

Modulbeschreibung Secure Coding

Schlüsselworte: Schwachstellen in Software, Code-Analyse, automatisierte Sicherheitstest, Verteidigungsmaßnahmen für Software

Zielgruppe:	Semester AIM 1 und AIM2	Modulnummer:	AIM 800 6623
Arbeitsaufwand:	5 ECTS		150 h
Davon	Kontaktzeit		120 h
	Selbststudium		15 h
	Prüfungsvorbereitung		15 h
Unterrichtssprache:	Deutsch		
Modulverantwortung:	Prof. Dr. Dominik Schoop		
Stand:	01.09.2019		

Empfohlene Voraussetzungen:

Kenntnisse in

- Programmiersprachen, Programmierung
- IT Security
- Betriebssysteme

Modulziel – angestrebte Lernergebnisse:

Die Studierenden werden in die Lage versetzt, sichere Software zu erstellen. Sie können Schwachstellen in Source Code identifizieren, analysieren und eliminieren. Sie können Sicherheitssoftwaretests durchführen und können Schutzmaßnahmen von Compilern und Betriebssystemen anwenden.

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen:

- die Arten von Schwachstellen im Code und wissen, wie diese Schwachstellen ausgenutzt werden können,
- Programmiertechniken, die Schwachstellen in Code vermeiden,
- Methoden zum werkzeuggestützten Auffinden von Schwachstellen in Code,
- Schutzmaßnahmen von Compilern und Betriebssystemen.

Fertigkeiten – methodische Kompetenzen

Die Studierenden können:

- Schwachstellen im Code finden,
- Schwachstellen im Code beseitigen,
- Sicherheitstests durchführen.

Übergreifende Kompetenzen

Die Studierenden sind in der Lage,

- Regeln für eine sichere Programmierung zu befolgen, auf Schwachstellen zu testen und das Sicherheitsniveau des Codes zu erhöhen.

Inhalt:

- Sicherheitsvorfälle wegen Softwareschwächen
- Arten von Sicherheitsschwächen in Software
- Sicherheitseigenschaften von Klassen von Programmiersprachen
- Über- und Unterläufe, Speicher manipulation, unzureichende Flusskontrolle, ...
- Techniken statischer Code Analyse (Kontrollfluss, Datenfluss, semantische Analyse)
- Techniken dynamischer Programmanalyse
- Fuzzing
- Sicherheitsmaßnahmen von Compilern und Betriebssystemen

Literaturhinweise:

- John Viega, Gary McGraw, Building Secure Software: How to Avoid Security Problems the Right Way, Addison-Wesley, 2001
- Jason Grembi, Secure Software Development: A Security Programmer's Guide Delmar Cengage Learning; 1 Edition, 2008
- Robert C. Seacord, The CERT C Secure Coding Standard Addison-Wesley Professional; 1st Edition, 2008
- Robert Seacord, Secure Coding in C and C++, Addison-Wesley Professional; 2 Edition, 2013
- Software Engineering Institute, Carnegie Mellon University, SEI CERT C Coding Standard, 2016

Wird angeboten:

In jedem Sommersemester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Übungen und Projektarbeit
Leistungskontrolle:	Klausur (90 Minuten)
Anteil Semesterwochenstunden:	4 SWS
Geschätzte studentische Arbeitszeit:	150 Stunden

Bildung der Modulnote:

Klausur

Modulbeschreibung Web Security

Schlüsselworte: Sichere Webanwendungen und Web Server

Zielgruppe:	Semester AIM1 und AIM2	Modulnummer:	AIM 800 6629
Arbeitsaufwand:	5 ECTS		150 h
Davon	Kontaktzeit		60 h
	Selbststudium		60 h
	Prüfungsvorbereitung		30 h
Unterrichtssprache:	Deutsch		
Modulverantwortung:	LB Dipl.-Ing. Sebastian Auwärter		
Stand:	26.01.2021		

Empfohlene Voraussetzungen:

- Betriebssysteme (Linux/Windows)
- Netzwerktechnologien
- Entwicklung von Webanwendungen

Modulziel – angestrebte Lernergebnisse:

Die Studierenden werden in die Lage versetzt, sichere Webanwendungen zu implementieren und die Sicherheit von Webanwendungen und deren Infrastruktur zu bewerten.

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen

- die Angriffsvektoren und typische Schwachstellen von Webanwendungen und Webservern,
- die typischen Angriffsszenarien auf Webanwendungen,
- Methoden und Werkzeugen zur Entwicklung sicherer Webanwendungen.

Fertigkeiten – methodische Kompetenzen

Die Studierenden beherrschen

- Design, Entwicklung, Bereitstellung und Betrieb von sicheren Webanwendungen und Webservern,
- Durchdringung von Webanwendungen mit manuellen und halbautomatischen Werkzeugen,
- Erkennung und Vermeidung von Schwachstellen bei der Entwicklung und Konfiguration von Webanwendungen

Übergreifende Kompetenzen

Die Studierenden sind in der Lage

- sichere Webserver und Webanwendungen zu entwickeln und zu implementieren,
- das Sicherheitsniveau von Webservern und Webanwendungen analysieren und bewerten.

Inhalt:

- Funktionsweise von Webservern und Webanwendungen
- Übersicht Web (In)Security:
OWASP Top10, Angriffs-Proxies, HTML/HTTP, DOM, Web-Technologien
- Angriffe auf Authentisierung und Authentifizierung
- Enumeration von Webanwendungen
- Cross-Site Scripting, Cross-Site Request Forgery
- Angriff auf Backendsysteme
- Command Injection und Path Traversal
- OS Command Injection, Path Traversal und Serialisierungs-Schwachstellen
- Ausnutzung bekannter Sicherheitslücken
- HTTPS, TLS und PKI

- Testprozesse während der Entwicklung von Webanwendungen

Literaturhinweise:

- Stuttard, D., Pinto, M.: The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws, John Wiley & Sons, 2. Auflage, 2011, ISBN 978-1118026472 2011
- Open Web Application Security Project (OWASP), <https://www.owasp.org>

Wird angeboten:

In jedem Wintersemester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Übungen und Projektarbeit
Leistungskontrolle:	Klausur (90 Minuten)
Anteil Semesterwochenstunden:	4 SWS
Geschätzte studentische Arbeitszeit:	150 Stunden

Bildung der Modulnote:

Klausur

Modulbeschreibung Web Services Innovationen

Schlüsselworte: Serviceorientierte Architekturen, Web-Anwendungen

Zielgruppe:	Semester AIM1 und AIM2	Modulnummer:	AIM 800 6627
Arbeitsaufwand:	5 ECTS		150 h
Davon	Kontaktzeit		60 h
	Selbststudium		90 h
	Prüfungsvorbereitung		0 h
Unterrichtssprache:	Deutsch oder Englisch		
Modulverantwortung:	Dr.-Ing. Michael M. Gebhart		
Stand:	18.01.2021		

Empfohlene Voraussetzungen:

Grundlegende Web-Technologien, Softwareentwicklung, Innovative Konzepte

Modulziel – angestrebte Lernergebnisse:

Die Studierenden verstehen die Konzepte, den Bedarf und die Einsatzmöglichkeiten von Web Services. Sie kennen bestehende Web Services für bspw. Datenhaltung und -analyse, Blockchain oder Künstliche Intelligenz, können diese in Anwendungen nutzen und verknüpfen sowie deren Mehrwerte erklären. Ebenso sind sie in der Lage eigene Web Services zu konzipieren und unter Berücksichtigung verbreiteter Best Practices zu entwickeln.

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen

- Grundlagen von Web Services,
- Methoden zur Identifikation und Spezifikation von Web Services,
- Bestehende Web Services für bspw. Datenhaltung und -analyse, Blockchain oder Künstliche Intelligenz,
- Konzepte von HTTP, SOAP, REST sowie Best Practices für REST Web Services,
- Methoden zur Entwicklung von REST Web Service Clients,
- Methoden zur Absicherung von Web Services,
- Varianten für den Betrieb von Web Services (bspw. Docker Images),
- Grundlagen serviceorientierter Architekturen,
- Unterschiede zwischen serviceorientierten Architekturen als Integrations- und strategische IT-Architekturen,
- Grundlagen des Business Process Management.

Fertigkeiten – methodische Kompetenzen

Die Studierenden sind in der Lage

- bestehende Web Services zu verstehen und Mehrwerte zu erkennen,
- Web Services in eigenen Anwendungen einzusetzen,
- Bedarfe für neue Web Services zu identifizieren,
- REST Web Services unter Berücksichtigung aktueller Best Practices zu entwickeln.

Übergreifende Kompetenzen

Die Studierenden können

- verteilte Web-Architekturen mit Hilfe von Web Services konzipieren,
- Business Cases aufbauend auf den obigen Architekturkonzepten entwerfen.

Inhalt:

- Grundlagen der Web Services
- Spezifikation von Web Service Requirements
- Web Services mit SOAP und REST
- Best Practices für REST Web Services
- Entwicklung von Web Service Clients
- Web Service Security
- Web Service Deployment
- Serviceorientierte Architekturen und Business Process Management

Literaturhinweise:

- Erl, Th. et.al.: SOA with REST. Prentice Hall 2012.
- Erl, Th.: Service-Oriented Architecture: Concepts, Technology, and Design. Prentice Hall 2015
- Papazoglou, M.P.: Web Services: Principles and Technology, Pearson Education, 2008.

Wird angeboten:

Wintersemester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung / Seminar
Leistungskontrolle:	Referat (20 Minuten)
Anteil Semesterwochenstunden:	2 SWS
Geschätzte studentische Arbeitszeit:	90 Stunden

Lernergebnisse:

Die Studierenden kennen die wichtigsten Technologien und Standards zur Entwicklung von Web Services. Sie sind in der Lage die Architektur serviceorientierter Web-Anwendungen zu verstehen, zu beurteilen und anzuwenden.

Lehr- und Lernform:	Projektarbeit
Leistungskontrolle:	Projektarbeit
Anteil Semesterwochenstunden:	2 SWS
Geschätzte studentische Arbeitszeit:	60 Stunden

Lernergebnisse:

Die Studierenden können Anforderungen an Web Services modellieren und die Technologien und Werkzeuge für Web Services in Projekten anwenden.

Bildung der Modulnote:

Referat (3) und Projektarbeit (2)