Hochschule Esslingen
University of Applied Sciences

Campus Esslingen Flandernstraße

Fakultät Informationstechnik

Modulhandbuch SPO2

Master-Studiengang Angewandte Informatik (AIM)
Vertiefungsrichtungen

- Autonome Systeme (AIM-AS)
- Data Science (AIM-DS)
- IT Security (AIM-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 (Workload)
1 ECTS entspricht näherungsweise 30 Arbeitsstunden

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

Vertiefungsrichtungen und zugehörige Wahlmodule

Jedes Wahlmodul ist einem der drei Schwerpunkten

- Autonome Systeme,
- Data Science und
- IT Security

zugeordnet.

Werden ausschließlich Wahlmodule aus nur einer Vertiefungsrichtung absolviert, so kann die Vertiefungsrichtung im Abschlusszeugnis ausgewiesen werden. Der 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: 01.09.2019

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 Künstliche Intelligenz

Schlüsselworte: **Überwachtes Lernen, nicht überwachtes Lernen, maschinelles Lernen**

Zielgruppe:	**Semester AIM1** **Semester AIM2**	**Modulnummer:**	**AIM 119 xxxx**
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 wesentlichen 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

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. Andreas Rößler**		
Stand:	**01.09.2019**		

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 Projektmanagements.

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

Modulbeschreibung Wahlpflichtmodul 3 und 4

Schlüsselworte: Vertiefung im eigenen Studienprofil

Zielgruppe:	**Semester AIM2**	**Modulnummer:**	**AIM WM234**
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. Andreas Rößler**		
Stand:	**01.09.2019**		

Empfohlene Voraussetzungen:

Abhängig vom gewählten Modul

Gesamtziel:

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

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**
Arbeitsaufwand:	**25 ECTS**		**750 h**
Davon	**Kontaktzeit**		**30 h**
	Selbststudium		**660 h**
	Prüfungsvorbereitung		**60 h**
Unterrichtssprache:	**Deutsch oder Englisch**		
Modulverantwortung:	**Prof. Dr. Andreas Rößler**		
Stand:	**01.09.2019**		

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 Projektmanagements

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

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. Andreas Rößler**		
Stand:	**01.09.2019**		

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

- ein 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 (2013): Wissenschaftliches Arbeiten. Ethik Inhalt & Form wiss. Arbeiten Handwerkszeug Quellen Projektmanagement Präsentation. 2. Aufl., 1. korr. Nachdr. Herdecke, Witten: W3L-Verl. (Soft skills).
- Kornmeier, Martin (2013): Wissenschaftlich schreiben leicht gemacht. Für Bachelor Master und Dissertation. 6., aktualisierte Aufl. Bern, [Stuttgart]: Haupt (UTB, 3154 : Schlüsselkompetenzen).
- Theisen, Manuel René (2013): Wissenschaftliches Arbeiten. Erfolgreich bei Bachelor- und Masterarbeit. 16., vollst. überarb. Aufl. München: 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:

Bericht

Modulbeschreibung Advanced Control

Schlüsselworte: Fuzzy-Regelung; moderne Regelungstechnik PI-Zustandsregler, Zustands- und Störgrößenbeobachter, Optimale Regler und Zustandsschätzer

Zielgruppe:	**Semester AIM-AS1 Semester AIM-AS2**	**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:	**01.09.2019**		

Empfohlene Voraussetzungen:

Kenntnisse der Booleschen Schaltalgebra, der Wahrscheinlichkeitsrechnung sowie der Matrix-Vektor-Rechnung, Integral- und Differentialrechnung.
Kenntnisse der Regelungstechnik, Beschreibung dynamischer Systeme im Frequenz und Zeitbereich, Kenntnisse einer Simulationssprache

Modulziel – angestrebte Lernergebnisse:

Die Studierenden beherrschen die Modellierung von dynamischen Systemen in der Zustandsdarstellung und können die in der Systemtheorie gebräuchlichen Systembeschreibungen einsetzen. Die Studierenden können PID, P- und PI-Zustandsregler sowie Fuzzy-Regler und erweiterte Regler-Strukturen (2-DOF, IMC) auslegen und deren jeweilige Vor- und Nachteile für eine Anwendung einschätzen. Sie sind in der Lage, Zustandsschätzer (Luenberger, Kalman, Störgrößen) zu entwerfen und in Projekten einzusetzen. Dies gilt jeweils sowohl für die analytische als auch für die numerische (MATLAB/Simulink) Auslegung der jeweiligen Systeme.

Kenntnisse – fachliche Kompetenzen
Die Studierenden kennen

- die Grundlagen der Fuzzy-Logik (Einsatzgebiete, Fuzzifizierung, Regelbasis, Defuzzifizierung)
- Einsatz und Funktionsweise von Fuzzy-Reglern
- 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

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-Reglerstrukuren zu verstehen und z.B. mit Hilfe der Flachheitsmethodik auslegen
- IMC-Regler zu entwerfen

Literaturhinweise:

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

Wird angeboten:

In jedem Wintersemester

Teilgebiete und Leistungsnachweise:

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

Lehr- und Lernform:	Projektarbeit
Leistungskontrolle:	Referat (20 Minuten)
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
- sichere und zuverlässige Systeme im Automobil konzipieren und implementieren

Bildung der Modulnote:

Testat und Klausur

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 High Performance Computing

Schlüsselworte: Parallele Programmierung und Algorithmen, Performance

Zielgruppe:	**Semester AIM-AS1** **Semester AIM-AS2**	**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:	**01.09.2019**		

Empfohlene Voraussetzungen:

Programmierkenntnisse

Modulziel – angestrebte Lernergebnisse:

Die Studierenden lernen die Grundlagen der parallelen Programmierung.
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

Fertigkeiten – methodische Kompetenzen

Die Studierenden sind in der Lage

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

Übergreifende Kompetenzen

Die Studierenden können

- Die Performance von Softwaresysteme mit Hilfe paralleler Programmierung verbessern
- Fehler in parallelen Programmen erkennen und beheben

Inhalt:

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

Literaturhinweise:

- MPI-Standard 3.1
- OpenCL 2.2
- Butenhof: Programming with POSIX Threads, Addison-Wesley
- Resch, Keller, Himmler, Krammer, Schulz: Tools for High Performance Computing, Springer-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 Mobile Communication

Schlüsselworte: Automotive, Communication, Safety, Security

Zielgruppe:	**Semester AIM-AS1** **Semester AIM-AS2**	**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:	**01.09.2019**		

Empfohlene Voraussetzungen:

Grundlagen der Kommunikationstechnik
Kenntnisse aus den Modulen IT-Security Engineering and Advanced Software Engineering

Modulziel – angestrebte Lernergebnisse:

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen

- die Grundlagen der Intelligent Transportation Systems (ITS) und der Vehiculäre Ad-Hoc-Netzwerke (VANETs)
- C2I/V2I-Anwendungen
- Automotive wireless Netztechnologie (WLAN (IEEE 802.11p), CAM, DENM)
- relevante Standards (IEEE, ISO)
- Systeme zur Positionsbestimmung (GPS, …)
- den Zusammenhang Safety und Security

Fertigkeiten – methodische Kompetenzen

Die Studierenden sind in der Lage

- die Ziele von ITS und VANETs zu erklären
- die Sicherheit von VANETs einzuschätzen und Sicherheitsmaßnahmen vorzuschlagen
- die Architektur und Technologie von VANETs zu erklären

Übergreifende Kompetenzen

Die Studierenden können

- einfache Car2Infrastructure- und Car2-Car-Applikationen zu implementieren.

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
- Performanceverbesserungen und Berechtigungskonzepte

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 und Seminar
Leistungskontrolle:	Referat (20 Minuten)
Anteil Semesterwochenstunden:	3 SWS
Geschätzte studentische Arbeitszeit:	90 Stunden

Lehr- und Lernform:	Projektarbeit
Leistungskontrolle:	Testat
Anteil Semesterwochenstunden:	1 SWS
Geschätzte studentische Arbeitszeit:	60 Stunden

Bildung der Modulnote:

Referat (20 Minuten) und Testat

Wahlmodule der Vertiefungsrichtung Data Science

Modulbeschreibung Advanced Data Models

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

Zielgruppe:	**Semester AIM-DS1** **Semester AIM-DS2**	**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 Abstraktionsvermöge 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 zu lesen,
- sich in neue Themen eigenständig einzuarbeiten,
- einen 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., and Martin Fowler. "NoSQL distilled." AddisonWesley Professional (2012).
- Sullivan, Dan. NoSQL for mere mortals. Addison-Wesley Professional, 2015.
- Harrison, Guy. Next Generation Databases: NoSQLand 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 Advanced Data Mining

Schlüsselworte: Data Mining Process, Neuronale Netze, Deep Learning

Zielgruppe:	**Semester AIM-DS1** **Semester AIM-DS2**	**Modulnummer:**	**AIM 800 6626**
Arbeitsaufwand:	**5 ECTS**		**150 h**
Davon	**Kontaktzeit**		**120 h**
	Selbststudium		**15 h**
	Prüfungsvorbereitung		**15 h**
Unterrichtssprache:	**Deutsch**		
Modulverantwortung:	**Prof. Dr.-Ing. Steffen Schober**		
Stand:	**01.09.2019**		

Empfohlene Voraussetzungen:

Grundkenntnisse im Maschinellen Lernen

Modulziel – angestrebte Lernergebnisse:

Die Studierenden werden in die Lage versetzt

- ein vollständigen Lebenszyklus eines Datenanalytik/KI Projekt umzusetzen,
- sowie mit modernen neuronalen Netzarchitekturen selbstständig zu arbeiten.

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen:

- den Data Mining-Prozess, notwendige Daten-Vorverarbeitungstechniken
- Methoden der Visual Analytics: Scatter plots, Histogramme, Boxplots, Scatter plot matrix, Parallel Coordinates
- Techniken zur Datenbereinigung sowie zum Umgang mit nicht-gleichgewichteten Trainingsdaten
- Rekurrente und *feed-forward* neuronale Netze sowie deren Trainingsmethoden (tiefe Netzwerke zur Bildklassifikation, Rekurrente Netze, z.B. LSTMs, GRUs)
- Grundlagen der Bibliotheken Pandas, Tensorflow und Keras.

Fertigkeiten – methodische Kompetenzen

Die Studierenden sind in der Lage

- Daten mit ausgewählten Techniken zu visualisieren
- Daten für nachfolgende Maschinelle Lernverfahren aufzubereiten
- Neuronale Netzwerke mit Keras/Tensorflow zu trainieren und zu evaluieren

Übergreifende Kompetenzen

Die Studierenden können ein komplettes KI-/Datenanalytik-Projekt durchführen.

Inhalt:

- DataMining Prozess (am Beispiel CRISP-DM)
- Visual Analytics:
  - Scatter plots, Histogramme, Boxplots
  - Scatter plot matrix, Parallel Coordinates
- Datenvorverarbeitung mit Python-Werkzeugen
- Umgang mit nicht-gleichgewichteten Trainingsdaten
- Grundlagen Tensorflow und Keras
- *Feedforward*-Netzwerke
- *Rekurrente*-Netzwerke

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
- Goodfellow et al; Deep Learning

Wird angeboten:

In jedem Wintersemster

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 Business Intelligence

Schlüsselworte: Business Intelligence, Datawarehouse, OLAP, ETL

Zielgruppe:	**Semester AIM-DS1** **Semester AIM-DS2**	**Modulnummer:**	**AIM 800 xxxx**
Arbeitsaufwand:	**5 ECTS**		**150 h**
Davon	**Kontaktzeit**		**60 h**
	Selbststudium		**60 h**
	Prüfungsvorbereitung		**30 h**
Unterrichtssprache:	**Deutsch**		
Modulverantwortung:	**Prof. Dr. Dirk Hesse**		
Stand:	**01.09.2019**		

Empfohlene Voraussetzungen:

- Programmierkenntnisse (objektorientiert)
- Statistik
- Datenbanken

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
- 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 Business Intelligence 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
- Kemper, Hans-Georg, et al.: Business Intelligence - Grundlagen und praktische Anwendungen. Eine Einführung in die IT-basierte Managementunterstützung, Vieweg und Teubner, 3. Auflage 2010
- 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: Cloud computing, Lights out computing, IaaS, PaaS, Container, Continuous Integration / Continuous Delivery, Cloud Functions

Zielgruppe:	**Semester AIM-DS1** **Semester AIM-DS2**	**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,
- Programmieren und Softwareentwicklung.

Modulziel – angestrebte Lernergebnisse:

Die Studierenden werden in die Lage versetzt, grundlegende Kenntnisse und Fertigkeiten im Umgang mit gängigen Techniken und Werkzeugen aus allen Bereichen des Cloud Computing – aus der Sicht von Anwendern, Bereitstellern und Anwendungsentwicklern – anwenden zu können.

- Sie verstehen die Bedeutung von Cloud Computing für die heutige Softwareentwicklung.
- Sie verstehen Virtualisierung und Software defined (Networking, Infrastruktur) über alle Ebenen hinweg.
- Sie verstehen Cloud Automatisierung und Lights-out-Computingkonzepte.
- Sie sind mit unterschiedlichen Entwicklungsmethoden und Prozessen hinsichtlich Cloud Computing vertraut.
- 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

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen

- Grundbegriffe des Cloud Computing: IaaS, PaaS, SaaS
- Grundelemente einer Cloud: Server, Netzwerke, etc.
- Grundprinzipien des Software Engineerings: Modularisierung, Abstraktion, etc.
- Grundlegende Web-Konzepte: HTTP, REST, etc.

Fertigkeiten – methodische Kompetenzen

Die Studierenden sind in der Lage

- Brücken zwischen physikalischer und virtueller Infrastruktur zu bilden.
- Zu verstehen wie Infrastruktur programmiert werden könnte und welche Möglichkeiten dies bietet.
- 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. Den Studenten wird sowohl ein Überblick als auch ein detaillierter Einblick in all die Teilmengen des Themas vermittelt – beginnend von einer Begriffsklärung wird, anhand von praxisnahen Beispielen, jede Teilmenge des Themas Cloud Computing beleuchtet, erläutert und selbst erforscht. Dabei werden im Verlauf verschiedene Sichten auf das Thema eingenommen – vom einfachen Nutzer eines Cloud Dienst über einen Cloud Betreiber bis hin zum Anwendungsentwickler von Cloud Diensten.

Literaturhinweise:

- Thomas Erl: Cloud Computing: Concepts, Technology & Architecture (The Prentice Hall Service Technology Series from Thomas Erl)
- Betsy Beyer, Chris Jones, Jennifer Petoff and Niall Richard Murphy : Site Reliability Engineering - https://landing.google.com/sre/books/

Wird angeboten:

In jedem Sommersemester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Übungen
Leistungskontrolle:	Klausur (90 Minuten) oder mündliche Prüfung (20 Minuten)
Anteil Semesterwochenstunden:	4 SWS
Geschätzte studentische Arbeitszeit:	150 Stunden

Bildung der Modulnote:

Klausur oder mündliche Prüfung
Die Prüfungsform wird zu Semesterbeginn bekanntgegeben

Modulbeschreibung High Performance Computing

Schlüsselworte: Parallele Programmierung und Algorithmen, Performance

Zielgruppe:	**Semester AIM-AS1** **Semester AIM-AS2**	**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:	**01.09.2019**		

Empfohlene Voraussetzungen:

Programmierkenntnisse

Modulziel – angestrebte Lernergebnisse:

Die Studierenden lernen die Grundlagen der parallelen Programmierung.
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

Fertigkeiten – methodische Kompetenzen

Die Studierenden sind in der Lage

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

Übergreifende Kompetenzen

Die Studierenden können

- Die Performance von Softwaresysteme mit Hilfe paralleler Programmierung verbessern
- Fehler in parallelen Programmen erkennen und beheben

Inhalt:

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

Literaturhinweise:

- MPI-Standard 3.1
- OpenCL 2.2
- Butenhof: Programming with POSIX Threads, Addison-Wesley
- Resch, Keller, Himmler, Krammer, Schulz: Tools for High Performance Computing, Springer-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 Intelligent Data Analytics

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

Zielgruppe:	**Semester AIM-DS1** **Semester AIM-DS2**	**Modulnummer:**	**AIM 800 6616**
Arbeitsaufwand:	**5 ECTS**		**150 h**
Davon	**Kontaktzeit**		**60 h**
	Selbststudium		**60 h**
	Prüfungsvorbereitung		**30 h**
Unterrichtssprache:	**Englisch**		
Modulverantwortung:	**Prof. Dr. Gabriele Gühring**		
Stand:	**01.09.2019**		

Empfohlene Voraussetzungen:

Kenntnisse in

- Mathematik, Statistik und Optimierung,
- Informatik

.

Modulziel – angestrebte Lernergebnisse:

Die Studierenden besitzen Grundkenntnisse in „Data Mining auf Zeitreihen“ und im Umgang mit der Software R. 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

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 maschinellem 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
- Applied Data Mining for Hybrid Vehicle Powertrain

Literaturhinweise:

- T. Mitsa: Temporal Data Mining. Chapman & Hall/CRC Data Mining and Knowledge Discovery. 2010
- J. Han, M. Kamber, J. Pei: Data Mining – Concepts and Techniques (3rd Edition). Morgan Kaufman, 2012
- R. J. Hyndman, G. Athanasopoulos: Forecasting: principles and practice, available online at www.otexts.org/fpp, 2014

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

Wahlmodule der Vertiefungsrichtung IT Security (AIM-IS)

Modulbeschreibung Digitale Forensik

Schlüsselworte: Datenwiederherstellung, Forensics Field Set, Beweissicherung und Dokumentation

Zielgruppe:	**Semester AIM-IS1** **Semester AMI-IS2**	**Modulnummer:**	**AIM 800 xxxx**
Arbeitsaufwand:	**5 ECTS**		**150 h**
Davon	**Kontaktzeit**		**60 h**
	Selbststudium		**60 h**
	Prüfungsvorbereitung		**30 h**
Unterrichtssprache:	**Deutsch**		
Modulverantwortung:	**Prof. Dr.-Ing. Tobias Heer**		
Stand:	**01.09.2019**		

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 Daten 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 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:

- 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 Wintersemester / 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 Information Security Management

Schlüsselworte: **Informationssicherheit, Incident Response, Sicherheitmaßnahmen**

Zielgruppe:	**Semester AIM-IS1** **Semester AMI-IS2**	**Modulnummer:**	**AIM 800 xxxx**
Arbeitsaufwand:	**5 ECTS**		**150 h**
Davon	**Kontaktzeit**		**120 h**
	Selbststudium		**15 h**
	Prüfungsvorbereitung		**15 h**
Unterrichtssprache:	**deutsch oder englisch**		
Modulverantwortung:	**Markus Schlittenhardt**		
Stand:	**01.09.2019**		

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 all; Informationssicherheit und Datenschutz: Handbuch für Praktiker und Begleitbuch zum T.I.S.P. dpunkt Verlag, ISBN 978-3-86490-596-4, 2019

Wird angeboten:

in jedem Sommersemester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Übungen und Projektarbeit
Leistungskontrolle:	KL 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-IS1** **Semester AMI-IS2**	**Modul Number:**	**AIM 800 xxxx**
Workload:	**5 ECTS**		**150 h**
divided into	**Contact time**		**60 h**
	Self-study		**60 h**
	Exam preparation		**30 h**
Course language:	**English**		
Modul director:	**Prof. Dr.-Ing. Michael Scharf**		
Vaild from:	**01.09.2019**		

Recommended requirements:

Understanding of computer networks, IT security and cryptography fundaments, 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 (e. g., 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)
- Anonymous communication

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
- Anonymous communication (Mixes, TOR)

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

Schlüsselwörter: IT-Sicherheit, Pentesting, Offensive Security

Zielgruppe:	**Semester AIM-IS1** **Semester AMI-IS2**	**Modulnummer:**	**800 xxxx**
Arbeitsaufwand:	**5 ECTS**		**150 h**
Davon	**Kontaktzeit**		**60 h**
	Selbststudium		**90 h**
	Prüfungsvorbereitung		**0 h**
Unterrichtssprache:	**Deutsch**		
Modulverantwortung	**Thomas Fischer, M.Sc.**		
Stand:	**01.09.2019**		

Empfohlene Voraussetzungen:

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

Modulziel – angestrebte Lernergebnisse:
Um IT-Systeme erfolgreich gegen unbefugten Zugriff schützen zu können, 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 Offensive Security.
Sie sind in der Lage, verschiedene Schwachstellentypen in Web-Applikationen zu erkennen und auszunutzen.

Kenntnisse – fachliche Kompetenzen
Die Studierenden kennen
- die wichtigsten Schwachstellen von IT-Systemen.

Fertigkeiten – methodische Kompetenzen
Die Studierenden sind in der Lage
- die wichtigsten Tools des Penetration Testing anzuwenden.

Übergreifende Kompetenzen
Die Studierenden sind in der Lage
- Cyper-Attacken durchzuführen und die IT-Sicherheit von IT-Systemen zu bewerten.

Inhalt:

- Typische Schwachstellen in IT-Systemen
- Angriffstypen, Angriffsvektoren, Top 10 der gängigen Angriffe
- Die wichtigsten Tools des Penetration Testing
- Praktische Durchführung von Angriffen

Literaturhinweise:

- Hacking mit Metasploit: Das umfassende Handbuch zu Penetration Testing und Metasploit, Michael Messner. dpunkt.verlag GmbH, 2. Auflage 2015, ISBN-13: 978-3864902246
- The Hacker Playbook: Practical Guide to Penetration Testing, Peter Kim. CreateSpace Independent Publishing Platform, 2014, ISBN-13: 978-1494932633
- The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws, Dafydd Stuttard, Marcus Pinto. John Wiley & Sons, 2. Auflage 2011, ISBN-13: 978-1118026472

Wird angeboten:

In jedem Wintersemester

Teilgebiete und Leistungsnachweise:

Lehr-, Lernform:	Vorlesung und Projektarbeit
Leistungskontrolle:	Bericht und Fortschritt bei den praktischen Übungen
Anteil Semesterwochenstunden:	4 SWS
Geschätzte studentische Arbeitszeit:	150 Stunden

Bildung der Note:

benoteter Bericht mit Fortschritt bei praktischen Übungen

Modulbeschreibung Secure Coding

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

Zielgruppe:	**Semester AIM-IS1** **Semester AMI-IS2**	**Modulnummer:**	**AIM 800 xxxx**
Arbeitsaufwand:	**5 ECTS**		**150 h**
Davon	**Kontaktzeit**		**60 h**
	Selbststudium		**60 h**
	Prüfungsvorbereitung		**30 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 ausgenutzt werden können.
- Programmiertechniken, die Schwachstellen in Code vermeiden
- Methoden zum werkzeuggestützen 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, Speichermanipulation, 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; 1 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 AIM-IS1** **Semester AMI-IS2**	**Modulnummer:**	**AIM 800 xxxx**
Arbeitsaufwand:	**5 ECTS**		**150 h**
Davon	**Kontaktzeit**		**60 h**
	Selbststudium		**60 h**
	Prüfungsvorbereitung		**30 h**
Unterrichtssprache:	**Deutsch**		
Modulverantwortung:	**Dipl.-Ing. (FH) Bernhard Hirschmann**		
Stand:	**01.09.2019**		

Empfohlene Voraussetzungen:

- Rechnernetze
- Internet Technologien
- IT Sicherheit

Modulziel – angestrebte Lernergebnisse:

Die Studierenden werden in die Lage versetzt, sichere Webanwendungen und Web Server zu implementieren.

Kenntnisse – fachliche Kompetenzen
Die Studierenden kennen:

- die Angriffsvektoren und typische Schwachstellen von Webanwendungen und Webservern,
- die typischen Angriffsszenarien auf Webanwendungen

Fertigkeiten – methodische Kompetenzen
Die Studierenden beherrschen

- Design, Entwicklung, Bereitstellung und Betrieb von sicheren Webanwendungen und Webservern,
- Das Durchdringung von Webanwendungen mit manuellen und halbautomatischen Werkzeugen.

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:
  Worms, Botnets, Kerberos / SSL, Phishing, Intrusion Detection Systems, Browser Security
- Böswillige Aktivitäten und riskantes Verhalten in privaten Netzwerken
- Integration von Webanwendungen in Unternehmensanwendungslandschaften
- Einführung, Erklärung und Demonstration typischer Schwachstellen von Webanwendungen
- Maßnahmen zur Sicherung und Härtung von Webanwendungen, Webservern und Netzwerkinfrastrukturen
- Umgehen von Sicherheitsmaßnahmen
- Implementierung von sicheren Webanwendungen
- Einsatz statischer Codeanalyse
- Penetrationstest von Webanwendungen

Literaturhinweise:

- Joel Scambray, Mike Shema, Caleb Sima: Hacking Exposed Web Applications, 3rd ed., McGraw-Hill, 2010, ISBN 978-0072262995
- Michael Zalewski: Tangled Web - Der Security-Leitfaden für Webentwickler, dpunkt.verlag, 2013, ISBN 978-3864900020
- Open Web Application Security Project (OWASP), https://www.owasp.org

Wird angeboten:

in jedem Wintersemester / 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 Services

Schlüsselworte: Serviceorientierte Architekturen, Web-Anwendungen

Zielgruppe:	**Semester AIM-IS1** **Semester AMI-IS2**	**Modulnummer:**	**AIM 800 6608**
Arbeitsaufwand:	**5 ECTS**		**150 h**
Davon	**Kontaktzeit**		**60 h**
	Selbststudium		**90 h**
	Prüfungsvorbereitung		**0 h**
Unterrichtssprache:	**Deutsch oder Englisch**		
Modulverantwortung:	**Prof. Dr.-Ing. Andreas Rößler**		
Stand:	**01.09.2019**		

Empfohlene Voraussetzungen:

Grundlegende Web-Technologien, Rechnernetze, Softwareentwicklung

Modulziel – angestrebte Lernergebnisse:
Die Studierenden beherrschen die Architektur von mehrschichtigen und dienstorientierten Anwendungssystemen im Web. Sie können die Softwarearchitektur einer Webanwendung modellieren und implementieren.

Kenntnisse – fachliche Kompetenzen
Die Studierenden kennen

- Grundlagen von Web Services
- Methoden zur Identifikation und Spezifikation von Web Services
- Konzepte von HTTP, SOAP und WSDL
- Konzepte von REST
- Best Practices für den Entwurf von REST Web Services
- Methoden zur Absicherung von Web Services
- Varianten für den Betrieb von Web Services (bspw. Docker Images)
- Methoden zur Entwicklung von REST Web Service Clients
- 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

- Web Services auf Basis von SOAP zu entwickeln
- REST Web Services unter Berücksichtigung aktueller Best Practices zu entwickeln
- REST Web Services als Microservices in Docker Images zu deployen
- Web Service Clients zu entwickeln

Übergreifende Kompetenzen
Die Studierenden können
- verteilte Web-Architekturen mit Hilfe von Web Services konzipieren

Inhalt:

- Grundlagen der Web Services
- Spezifikation von Web Service Requirements
- Web Services mit SOAP und WSDL
- Web Services mit REST
- Best Practices für REST Web Services
- Web Service Security
- Web Service Deployment
- Entwicklung von Web Service Clients
- 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:

In jedem 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 verteilte Web-Architekturen und Web Services modellieren und die Technologien und Werkzeuge für Web Services in Projekten anwenden.

Bildung der Modulnote:

Referat (3) und Projektarbeit (2)