

Modulcode <small>(1)</small>	Modulbezeichnung <small>(2)</small>	Zuordnung <small>(3)</small>
BAI7350	Deep Learning und Generative KI (DLGKI)	
	Studiengang <small>(4)</small>	Bachelor Angewandte Informatik
	Fakultät <small>(5)</small>	Gebäudetechnik und Informatik

Modulverantwortlich <small>(6)</small>	Prof. Dr. Marcel Spehr
Modulart <small>(7)</small>	Pflicht
Angebotshäufigkeit <small>(8)</small>	SS
Regelbelegung / Empf. Semester <small>(9)</small>	BA6
Credits (ECTS) <small>(10)</small>	5 CP
Leistungsnachweis <small>(11)</small>	PrP(N) SL
Unterrichtssprache <small>(12)</small>	Deutsch
Voraussetzungen für dieses Modul <small>(13)</small>	BAI0101: Mathematik 1 BAI0104: Grundkonzepte der Programmierung BAI0105: Datenbanken 1 BAI0201: Mathematik 2 BAI0205: Datenbanken 2 BAI0305: Statistik/Stochastik
Modul ist Voraussetzung für <small>(14)</small>	
Moduldauer <small>(15)</small>	1 Semester
Notwendige Anmeldung <small>(16)</small>	-
Verwendbarkeit des Moduls <small>(17)</small>	-

Lehrveranstaltung <small>(18)</small>	Dozent/in <small>(19)</small>	Art <small>(20)</small>	Teilnehmer (maximal) <small>(21)</small>	Anzahl Gruppen <small>(22)</small>	SWS <small>(23)</small>	Workload	
						Präsenz <small>(24)</small>	Selbststudium <small>(25)</small>
1 DLGKI	Spehr	S	20	1	4	60	65
				Summe	4	60	65
Workload für das Modul				<small>(26)</small>		125	

Qualifikationsziele	(27.)	<p>Die Studierenden können...</p> <ul style="list-style-type: none"> • DL-Grundlagen sicher anwenden: Autodiff, Optimierung, Regularisierung. • mit PyTorch und dem Hugging-Face-Stack trainieren, feintunen und evaluieren. • Kernarchitekturen auswählen und anpassen: CNN, Transformer. • generative Modelle nutzen: LLM-Prompting, LoRA-Finetuning, Diffusion-Inference. • kleine RAG-Systeme mit Quellenbelegen bauen und testen. • Modelle schlank deployen und rechtlich sicher betreiben (DSGVO, AI Act-Überblick).
Inhalte	(28.)	<ul style="list-style-type: none"> • Grundlagen und Werkzeuge <ul style="list-style-type: none"> ◦ Tensoren, Loss/Optim, Overfitting, Datenpipeline. PyTorch-Basics, HF-Transformers/Datasets. • Supervised DL und Kernarchitekturen <ul style="list-style-type: none"> ◦ CNN-Pattern, Transfer Learning. Transformer-Encoder/Decoder. Tokenisierung, Sampling. • Generative KI <ul style="list-style-type: none"> ◦ LLM-Nutzung, Prompting, Adapter/LoRA. Diffusionsmodelle auf vortrainierten Gewichten. • Evaluation, Betrieb, Verantwortung <ul style="list-style-type: none"> ◦ Klassische und generative Metriken, Red-Team-Checks. Quantisierung, Export, einfache API. Datenschutz, Urheberrecht, Risikoabschätzung. • Praxisprojekt <ul style="list-style-type: none"> ◦ Erstellung eines Prototypen, der die gelernten Techniken realisiert
Vorleistungen und Modulprüfung	(29.)	<p>Vorleistungen:</p> <ul style="list-style-type: none"> - Keine <p>Modulprüfung:</p> <ul style="list-style-type: none"> - 60% Projekt mit Präsentation (Gruppenarbeit) - 40% Testat 60min
Literatur	(30.)	<ul style="list-style-type: none"> • Goodfellow, Bengio, Courville: Deep Learning. MIT Press, 2016. • Géron: Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow (3rd). O'Reilly, 2022. • Chollet: Deep Learning with Python (2nd). Manning, 2021. • Foster: Generative Deep Learning (2nd). O'Reilly, 2023. • Tunstall, von Werra, Wolf: Natural Language Processing with Transformers (Revised). O'Reilly, 2022. • Deisenroth, Faisal, Ong: Mathematics for Machine Learning. CUP, 2020.