M.S. in Computer Science

Prereqs: AIM 5002 and 5001 or COM 5000, 5001 and 5002

Artificial Intelligence (AI) is an interdisciplinary field, integrating knowledge and methods from computer science, mathematics, philosophy, psychology, economics, neuroscience, linguistics, and biology. Intelligent agents mimic cognitive functions to implement intelligent behaviors such as perception, reasoning, communication, and acting in symbolic and computational models. AI is used in a wide range of narrow applications, from medical diagnosis to speech recognition to bot control. The autonomous single, multiple, and adversarial agents that students build in this course will support fully observable and partially observable decisions in both deterministic and stochastic environments. Topics covered include search, constraint satisfaction, Markov decision processes, planning, knowledge representation, reasoning under uncertainty, graphical models, and reinforcement learning. The techniques and technologies mastered here will provide the foundational knowledge for the ongoing study and application of AI in other applications across practice areas.

Data Acquisition and Management focuses on the data structures, data design patterns, algorithms, methods, and best practices for the pre-modeling phases of data science workflows, including problem formulation, gather, analyze, explore, model, and communicate. This

Arguably, most of data science is statistical learning, which requires strong foundational knowledge in probability and statistics. And applying computational methods such as direct simulation, shuffling, bootstrapping, and cross-validation to statistical problems is often more intuitive and can provide solutions where analytical methods would prove computationally intractable. This course introduces students to the statistical analysis of data using modern computational methods and software. Probability, descriptive statistics, inferential statistics and computation methods such as simulations sample distributions, shuffling, bootstrapping, and cross-validation will be covered.

Prereqs: AIM 5002 and 5001 or COM 5000, 5001 and 5002

In classical programming, answers are obtained from rules and data. In machine learning, rules are obtained from data and answers. The widespread availability and sharing of data, and improvements in computing capacity, processing methods, and algorithms have given machine learning the power to deliver game-changing systems and technologies to organizations that compete on predictive, prescriptive, and/or autonomous analytics. In this course, we'll look at methods for using, tuning, and comparing machine learning algorithms, based on measures of performance, accuracy, and explainability. We'll also look at recent advances and trends in automated machine learning.

Prereq: AIM 5005

Data scientists have been able to leverage better algorithms on faster hardware optimized with graphical processing units to deliver improved performance and accuracy in whole classes of applications that had been previously commercially unviable. The biggest beneficiaries are applications that require unstructured data, such as audio and or video processing. Deep neural networks have also provided gains for other complex applications, from recommendation systems to natural language processing. This course builds on the concepts in machine learning to train multi-layered neural networks. Main topics covered in this course are generalization, convolutional neural network, recurrent neural network, long short-term memory, and autoencoder.

Big-Tech development and management is project-based, and successful researchers and technologists are effective at managing projects and collaborating in cross-functional, geographically distributed project teams. This course teaches the methodologies and tools for large-sized (PMI) and small-scale (Agile) projects as well as how to adapt management methods to organizational culture and project team members' background and experience

Designing efficient algorithms is one of the most important tools computers scientists use to solve difficult problems. This course covers techniques for designing efficient algorithms, as well as advanced topics such as self-adjusting search trees, network flows, linear programming, approximation algorithms, and randomization algorithms. Students apply these tools and techniques to real-world problems, such as airline scheduling, image segmentation, social networking, genomic sequencing, and survey design.

The course provides students with a comprehensive understanding of the mathematical aspects of computer science as well as their application. Throughout the course, students will learn the theoretical foundations of computer science and gain knowledge with various topics including algorithms, computational models, and the fundamental principles underlying computation.

This advanced-level course explores innovative and cutting-edge programming concepts, languages, and methodologies. The course is designed for students who already possess a strong foundation in traditional programming paradigms and are eager to explore the latest trends and advancements in the field. Throughout the course, students will be exposed to various programming languages, frameworks, and tools to understand how they enable developers to tackle modern-day challenges effectively.

The course provides students with a comprehensive understanding of the fundamental principles and techniques used in computer graphics. It explores the creation, manipulation, and rendering of digital images and visual content using software and hardware technologies. Throughout the course, students will learn the theoretical foundations of computer graphics and gain hands-on experience with various tools and software used in the field. The course covers both 2D and 3D graphics, enabling students to develop skills in both domains.

The course is designed to provide participants with a comprehensive understanding of DevOps principles, practices, and tools. During this course, students will explore the fundamental concepts, methodologies, and technologies that drive the DevOps culture. They will learn how to bridge the gap between development and operations teams, enabling faster and more reliable software releases. The course will cover various aspects of the DevOps lifecycle, including continuous integration, continuous delivery, infrastructure automation, and monitoring. Through the course, the students are expected to have a solid understanding of DevOps concepts, tools, and practices, allowing them to contribute effectively to DevOps initiatives within their organizations.

The course provides students with a deep understanding of the principles, techniques, and processes involved in software system security. The course covers the measures and practices put in place to protect software applications and systems from potential threats and vulnerabilities. It emphasizes multiple layers of protection, each addressing different aspects of security, as software systems are often targeted by malicious actors seeking to exploit weaknesses and gain unauthorized access or control.

The course provides students with a deep understanding of hardware security that focuses on protecting the physical components and integrated circuits of electronic devices from threats and attacks. This course provides a comprehensive overview of hardware security principles, techniques, and best practices to ensure the confidentiality, integrity, and availability of digital systems. Students will gain a deep understanding of hardware vulnerabilities, attacks, and mitigation strategies.

and aroceerine

The Capstone in Computer Science integrates students’ prior coursework, research, colloquia, and professional experiences. It offers a unique opportunity to synthesize computer science theory with real-world practice through an applied project, thesis, approved internship, or equivalent activity. Students will work with their supervising faculty to identify deliverables for both Part 1 and Part 2 of the Capstone.

This course allows students to participate in an off-campus internship supervised by a staff person at the internship site and overseen by a faculty advisor. The internship site must be approved by the program director and the overall duration of student work must be no less than 150 hours (based on a 3-credit course). At the start of the internship, the student and faculty advisor will jointly develop specific learning objectives tailored to the nature of the internship. Over the course of the internship, students will be required to submit weekly reflections, and at the end of the internship, students write a final paper that represents the culmination of the work.

The course provides the student with the flexibility to learn more about a topic of interest outside of the formal course setting. The subject should be chosen in consultation with a faculty advisor who acts as the student's supervisor, and with the permission of the program director. The student is required to submit a course contract describing the course of study and its specific learning objectives. Course credit is determined in advance by the instructor with the approval of the program director.

This course provides the opportunity to offer special interest courses on emerging theory, phenomena, and technologies in computer science, in areas such as systems, human-computer interaction, machine learning, and artificial intelligence. This will be an advanced class, whether seminar style or project based. Students are required to complete an appropriate project or other deliverable in line with the number of credits awarded for the course.

Last updated: January 11, 2024