CSE 410
Spl Topic: AI Capstone
Weekly Meeting: T, Th, 200 pm-330 pm
Credits: 6

Dr. Sreyasee DAS BHATTACHARJEE
Email: sreyasee@buffalo.edu
Office: Davis Hall 349
Office Hours: Friday 530pm –630pm; in-person at the office

Course Overview and Objectives:

This intensive, six-credit *AI Capstone* transitions students from academic learners to capable computational engineers. Building upon foundational expertise in data architectures, optimization logic, and multi-dimensional modeling, students execute an accelerated 15-week computational development-to-validation cycle. The curriculum is anchored in an academic system-design model, where engineering success is defined by a dual mandate: achieving high-fidelity functional performance and ensuring systemic operational robustness against structural data variations and noise. The capstone instructor provides milestone-based progress management and structural oversight, while a project faculty supervisor guides the deep technical execution of the student team. Upon successful completion of this course, students will be able to:

1. **Complex Problem Analysis and System Synthesis (ABET 1):** Identify, define, and scope the technical requirements of a complex computing problem in coordination with a project faculty supervisor. Students apply advanced data science and engineering principles to architect multi-domain computational pipelines that prioritize operational efficiency and system-level reliability.

2. **Design, Implement, and Evaluate Advanced Computational Solutions (ABET 2):** Engineer a functional software prototype from initial scoping to final system integration. This involves implementing robust data-streaming pipelines, executing structural parameter tuning, and conducting rigorous evaluation phases where boundary stress testing and error-propagation checks serve as the primary metrics for validation.

3. **Professional Communication and Technical Defense (ABET 3):** Produce professional-grade technical documentation, including a comprehensive project report and a reproducibility-focused repository layout. Students defend their technical proof-of-concept through a rigorous oral presentation on the departmental Demo Day, tailored for faculty reviewers and invited external stakeholders, such as the Department Advisory Committee.

4. **Ethical Responsibility and Informed Judgment (ABET 4):** Make informed engineering judgments regarding the deployment of algorithmic systems by evaluating the ethical implications of data provenance, validation transparency, and public safety. Students use rigorous internal quality-assurance gates to enforce professional responsibility, preventing the validation of fragile or non-compliant models.

5. **Collaborative Leadership and Engineering Teamwork (ABET 5):** Function effectively as a member or leader of an engineering team. Students manage project milestones, synchronize technical tasks, balance physical system constraints with algorithmic logic, and

maintain consistent progress checks with the capstone instructor and faculty supervisor to deliver a unified technical solution.

Expected Student Outcomes and their Alignment with ABET Engineering Criteria:

The *AI Capstone* ensures every technical milestone maps directly to core ABET Student Outcomes. This begins with the project initiation and scoping phase, satisfying Criterion (1) by requiring the student team to analyze complex computational and engineering requirements in coordination with their project faculty supervisor. The functional prototype build addresses Criterion (2) as students design and integrate computational solutions that must meet performance benchmarks while passing rigorous boundary stress testing and validation checks. Professional communication (Criterion 3) is evaluated during the departmental Demo Day, where students demonstrate technical authority by presenting their final report and defending their results to faculty and invited stakeholders, such as the Department Advisory Committee. Ethical responsibilities are operationalized through systematic internal quality-assurance gates, mapping to Criterion (4) by embedding compliance and data-provenance standards directly into the pipeline validation process. Finally, collaborative team sprints satisfy Criterion (5) by requiring students to manage technical milestones, balance physical system constraints with algorithmic logic, and maintain structured progress checks managed by the capstone instructor.

Course Prerequisites:

General expertise in Python, good familiarity with Calculus, Linear Algebra, and Data Structures.
Course-specific expertise: Machine Learning(CSE474), Introduction to AI (CSE368), Software Security (CSE 418)
Good to have expertise: Computer Vision (CSE473)

Piazza Course Signup Link: TBF

Grading Policy & Other Course Details:

The absolute grading scale: A (93–100); A- (90–92); B+ (87–89); B (83–86); B- (80–82); C+ (77–79); C (73–76); C- (70–72), D+ (65–69),D (65–50), F (below 50).

Requirements:

The course grade will be based on all components discussed later in "Grade Compositions". All assignment submissions from the previous week will be in UBLearns and due by 11:59 PM every Monday.

The following items are designed to help you plan better for the course. Please read carefully:

- Grades are NON-NEGOTIABLE per UB policy.
- The deadlines are hard and cannot be extended without prior approval.
- You will need to submit your work in UBLearn.Email submissions (unless stated otherwise) will not be accepted

Grade Composition:

Phase 1: RFP Response & System Design	10%
Phase 2: Advanced Model Prototype	30%
Phase 3: Stress-Testing & Hardening	20%
Phase 4: Demo Day & Presentation to the Faculty Supervisor	40%

Selected Best Projects may be published on the UB Course page at the end of the semester

Intellectual Property & Confidentiality:

Students will interact with proprietary institutional datasets and novel research frameworks developed within the university ecosystem. All students must strictly adhere to the following data and asset protocols:

- **Data Provenance & Security:** All research data and internal code repositories must remain exclusively on approved, secure, UB-managed infrastructure. Under no circumstances may project data, internal pipelines, or algorithmic assets be uploaded to public repositories (e.g., public GitHub) or external commercial AI platforms without explicit written approval from the instructor.
- **IP Co-Ownership:** Any intellectual property, novel algorithms, or software frameworks generated during this course will be co-owned by the UB instructional and research team.
- **Commercialization Management:** All co-owned intellectual property generated within the scope of this course will be managed, protected, and commercialized through the University at Buffalo Technology Transfer (TT) office.

Policy on the Use of Generative AI: In this course, Generative AI (LLMs, GitHub Copilot) is encouraged for *productivity and boilerplate generation*. However, per ABET 4 requirements:

- All AI-generated code must be audited for *hallucinated* vulnerabilities.
- Students must disclose in their final Technical Report which components of their system were synthesized using GenAI.
- **The Core Value:** Your grade is based on your *validation* and *hardening* of the system, not just the generation of the model.

List of Projects Available this Semester: `https://buffalo.box.com/s/dcnnsrof4407ohppk5sdcv44rqbvvih6`

Rough Schedule:

1. **Phase 1: Project Initiation & Literature Review (Weeks 1–3)**
   Focus: Requirement Analysis and Systems Design (ABET 1)

   During the *first* week, students form engineering teams, receive their assigned dataset, and attend an introductory briefing by their project faculty supervisor to understand the core problem. By the *second* week, teams review the foundational literature provided by their supervisor and analyze the structure of the dataset to select an appropriate baseline computational architecture. This phase culminates in a straightforward System Design Review (SDR) in the *third* week, during which teams present their data loading strategy and initial model selection to the course instructor and faculty supervisor, establishing a clear roadmap for the semester's implementation goals.

2. **Phase 2: Model Development & Pipeline Implementation (Weeks 4–8)**
   Focus: Design and Implementation (ABET 2)

   In this phase, teams execute the core engineering of their computational solutions, moving from raw data to functional prototypes. Weeks *four* and *five* are dedicated to structural data engineering, where students address the challenges of raw or noisy datasets through programmatic cleaning, feature formatting, and input validation. As the timeline progresses into weeks *six* and *seven*, the focus shifts to core model training and system optimization, including parameter adjustment and structural tuning. The phase concludes in week *eight* with a *Midterm Prototype Demonstration* monitored by the capstone instructor and the faculty supervisor to validate that the primary computational task is operationally sound and ready for evaluation.

3. **Phase 3: System Testing & Validation Checkpoints (Weeks 9–12)**
   Focus: Evaluation and Ethical Practice (ABET 2, 4)

   During this phase, the primary focus shifts to extensive testing and validation of the developed models. Students write evaluation scripts to check how their system handles edge cases, data errors, and unexpected variations. Throughout weeks *ten* and *eleven*, teams run comprehensive tests to verify the accuracy and stability of their prototypes. Regular progress updates with the capstone instructor and technical review checkpoints with the faculty supervisor ensure that the system undergoes thorough testing, meets project requirements, and operates reliably before final deployment.

4. **Phase 4: Faculty Meeting & Departmental Demo Day (Weeks 13–15)**
   Focus: Communication and Team Leadership (ABET 3, 5)

   The capstone concludes with a high-intensity focus on technical communication and the formal presentation of findings. In week *thirteen*, teams finalize their comprehensive documentation, producing a formal Technical Report and repository layout that details their system architecture and reproducibility validation. Week *fourteen* is dedicated to refining the technical presentation with feedback from the capstone instructor to ensure clear communication of performance metrics. The semester culminates in a departmental *Demo Day* in week *fifteen*, a formal technical defense and showcase held within the department, where external stakeholders (such as members of the Department Advisory Committee) may be invited to evaluate the completed engineering projects alongside the faculty.

Computational Resources:

Teams will be granted access to UB's *Center for Computational Research (CCR)* or dedicated laboratory GPU workstations to support the training of Large Language Models (LLMs) and Vision Transformers. Misuse of high-performance computing (HPC) resources will result in immediate loss of access.

Accessibility Resources:

If you have any disability that requires reasonable accommodations to enable you to participate in this course, please get in touch with the Office of Accessibility Resources, 25 Capen Hall, 645-2608, and also the Teaching Assistants (with a copy to the instructor) of the course. The office will provide you with information and review appropriate accommodations. http://www.student-affairs.buffalo.edu/ods/

Academic Integrity:

(Short) Don't cheat! You will be caught and failing the course. Our department is serious about graduating ethical and upstanding computer scientists. The policy has recently been updated and will be enforced.

(Long) All academic work must be your own. Plagiarism, defined as copying or receiving materials from a source or sources and submitting this material as one's own without acknowledging the particular debts to the source (quotations, paraphrases, basic ideas), or otherwise representing the work of another as one's own, is never allowed. Collaboration, usually evidenced by unjustifiable similarity, is never permitted in individual assignments. Any submitted academic work may be subject to screening by software programs designed to detect evidence of plagiarism or collaboration. Also, do not post any course material outside the Course Piazza page. It will be interpreted as an attempt to get non-approved help. For the complete policy, please see:

https://engineering.buffalo.edu/computer-science-engineering/information-for-faculty-and-staff/academic-integrity.html