AI application in enhancing the safety in construction zones

Dr. Fazil T. Najafi, University of Florida

For more than 40 years, Dr. Najafi has worked in government, industry, and education. He earned a BSCE 1963 from the American College of Engineering, University of Kabul, Afghanistan. In 1966, Dr. Najafi earned a Fulbright scholarship and did his B.S., MS, and Ph.D. degree in Civil Engineering at Virginia Polytechnic Institute and State University, Blacksburg, Virginia; his experience in industry and government includes work as a Highway Engineer, Construction Engineer, Structural, Mechanical, and Consultant Engineer. Dr. Najafi taught at Villanova University, Pennsylvania, and was a visiting professor at George Mason University and a professor at the University of Florida, Department of Civil and Coastal Engineering. He has received numerous awards, such as Fulbright scholarship, teaching awards, best paper awards, community service awards, and admission as an Eminent Engineer into Tau Beta Pi. The Florida Legislature adopted his research on passive radon-resistant new residential building construction in the HB1647 building code of Florida. Najafi is a member of numerous professional societies and has served on many committees and programs; and continuously attends and presents refereed papers at international, national, and local professional meetings and conferences. Lastly, Najafi attends courses, seminars, and workshops and has developed courses, videos, and software packages during his career. Najafi has more than 300 refereed articles. His areas of specialization include transportation planning and management, legal aspects, construction contract administration, public works, and Renewable Energy.

Mayur sai yaram, University of Florida

A Research Assistant at the University of Florida, driven by a passion for leveraging artificial intelligence to solve real-world problems. His academic journey began at GITAM University, India, where he earned his BS in Computer Science and Engineering in 2020, and continued at the University of Florida, where he completed his MS in Computer Science in 2025. Mayur's research focuses on machine learning systems and reinforcement learning algorithms, with applications in traffic management and urban infrastructure optimization. He believes that AI should serve practical purposes that improve people's lives, which has shaped his approach to implementing deep learning models and developing scalable data pipelines. Through his work on AI-based traffic signal control and sustainable urban infrastructure, Mayur continues to explore how cutting-edge technology can address critical challenges in modern cities.

vijay thakkallapally, University of Florida
Gayathri reddy Yeddla, University of Florida
YAGNESH CHALLAGUNDLA, University of Florida

I'm Yagnesh Challagundla, a passionate researcher and technology enthusiast driven by curiosity and a desire to create meaningful impact through innovation. My academic path led me toward engineering and artificial intelligence — fields that constantly challenge me to think creatively and build solutions that matter. I've always believed that learning should extend beyond the classroom, which shaped how I approached every phase of my education and professional growth.

AI-Based Traffic Management Systems: Enhancing Efficiency and Safety in Construction Zones

1. Abstract

This paper describes a hands-on course in which undergraduate civil engineering students learn to build artificial intelligence (AI) and machine learning (ML) systems from scratch. Unlike approaches that teach students to evaluate or compare existing AI methods, this course focuses on the complete implementation pipeline: collecting and cleaning messy sensor data, training prediction and detection models, deploying systems on edge hardware, and iterating based on performance feedback. Construction zone traffic management serves as the project context because it presents authentic data challenges, real-time constraints, and safety stakes that motivate students and develop practical engineering skills.

The course is organized into four phases delivered over a single semester. Students progress from exploring real traffic data through building supervised learning models, training reinforcement learning agents in the SUMO traffic simulator, and ultimately integrating all components into a working system. Every phase requires students to write working code and demonstrate functional outputs.

We tested this approach with 60 undergraduate students during Spring 2025 at one university. All assessments were anonymous and based solely on standard course data. Results show that students' understanding of AI/ML concepts improved from 35% to 68% on matched pre- and post-tests. Eighty-eight percent of students built working AI systems meeting professional accuracy and speed requirements. When given a dataset from a completely different engineering domain after the course ended, 73% of students successfully built a working ML pipeline on their own, showing that the skills they learned are transferable. This paper presents the course design, assessment results, and lessons learned.

2. Introduction

2.1. The challenge

Civil engineering is changing fast. Modern infrastructure increasingly depends on AI and ML systems. Traffic signals use AI to adapt to changing conditions [1], structural health monitoring uses deep learning to spot damage, and construction management platforms use predictive analytics for scheduling. Industry demand for civil engineers who understand these technologies is growing, but most civil engineering programs teach very little about them, creating a widening gap between what employers need and what graduates know.

This gap has two dimensions. One is the ability to critically evaluate AI methods and understand their trade-offs as an important skill addressed elsewhere in engineering education. The other, addressed in this paper, is the ability to actually build AI systems: to take a real engineering problem, collect and prepare data, train models, deploy them under real-time constraints, and maintain them over time. Undergraduate students who can discuss AI only in the abstract and cannot implement it are poorly prepared for a workforce that increasingly expects hands-on capability [2], [3].

2.2. Why traditional teaching approaches fall short

When universities try to close this gap, they typically take one of three paths, each with clear limitations. The first is requiring civil engineering students to take standard computer science ML courses. These courses assume programming fluency and math backgrounds that most civil engineering students lack, leading to high dropout rates and poor transfer back to engineering contexts [2], [3]. The second path involves brief exposure through guest lectures or single-module overviews that introduce vocabulary without building real ability. The third path uses generic practice datasets, such as the iris classification dataset, that fail to engage civil engineering students or to show how AI applies to their field.

2.3. Our approach

This paper presents a different strategy: teach undergraduate students to build complete AI/ML systems by tackling an authentic, complex civil engineering problem: managing traffic in and around active construction zones. The emphasis throughout is on implementation: students write working code at every stage, from data ingestion scripts to trained models to deployed inference pipelines. Construction zone traffic was chosen because it is familiar to every civil engineering student, involves real safety stakes [4], requires real-time decision-making, and spans the whole pipeline from messy sensor data to deployed control systems.

The approach is grounded in project-based learning research [2], [5] and aligned with ABET's emphasis on applying engineering judgment informed by data analytics [6]. We tested it with 60 undergraduate students during Spring 2025 at one university. The rest of this paper states the research questions, describes the course design, presents results with figures and tables, discusses what worked and what did not, and identifies limitations and future directions.

3. Research questions

This work investigates four questions about how well undergraduate civil engineering students can learn to build AI/ML systems through hands-on project-based instruction.

Research question 1: Can undergraduates without computer science backgrounds achieve meaningful improvement in AI/ML understanding through this approach? We expected students would improve by at least 20 percentage points on a matched pre/post concept test.
Research question 2: Which parts of the AI/ML implementation pipeline are most complex for undergraduates to learn? We expected data preparation and feature engineering would be more complicated than model training, because data work gets less attention in typical courses.
Research question 3: Can undergraduates build working AI systems that meet professional standards for accuracy, speed, and safety? We set a target of at least 75% of students producing working systems.
Research question 4: Do implementation skills transfer to unfamiliar engineering problems? We expected at least 70% of students would successfully build a working ML pipeline for a problem outside traffic management.

4. Methodology

4.1. Overview of the four-phase structure

The course is organized into four phases delivered over a 12-week semester. Each phase corresponds to a stage of the AI/ML implementation pipeline that professional engineers and data scientists use in practice [7], [1]. Students work on the same construction zone traffic problem throughout, building progressively more capable systems. Figure 1 shows the pipeline and how the four phases map onto it.

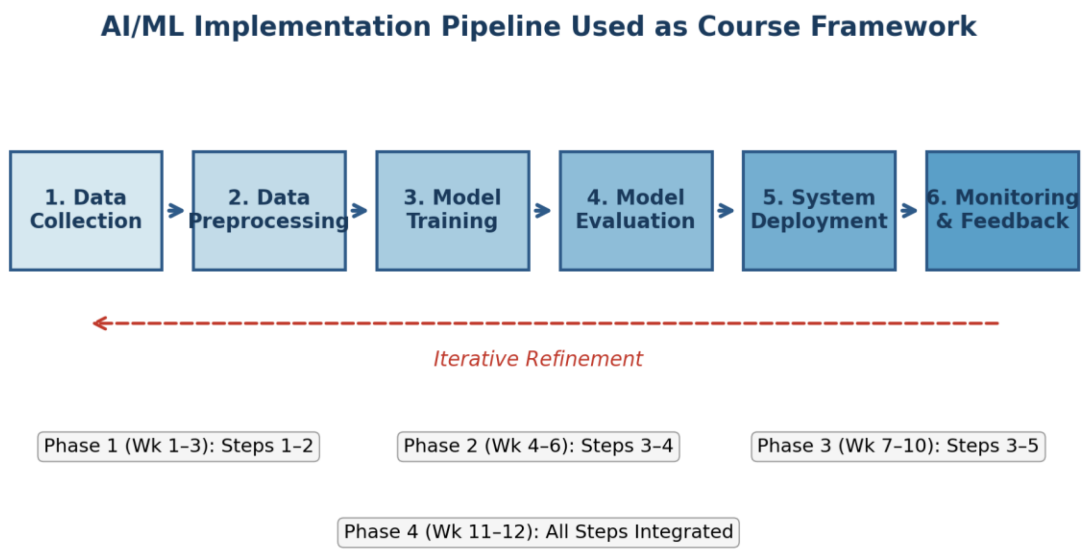

Figure 1. AI/ML implementation pipeline used as the course framework, with phase mapping.

4.2. Learning progression

The course follows Bloom's revised taxonomy [8], moving from understanding through application to creation. In Phase 1, students learn to explore and question data. In Phase 2, they apply supervised learning techniques. In Phase 3, they implement reinforcement learning agents. In Phase 4, they create integrated systems that combine everything. Every phase requires students to submit working code alongside written analysis, ensuring that understanding translates into practical ability.

4.3. Five design principles

Five principles guide the course. First, every assignment uses real problems of genuine importance: construction zones cause more than 100,000 injuries annually in the United States, providing students with real motivation [4]. Second, help is removed gradually. Early exercises include extensive starter code and comments, while later assignments require students to design solutions on their own [2], [5]. Third, the SUMO simulation environment [9] lets students experiment safely with traffic control strategies that would be dangerous on real roads. Fourth, integrated projects ensure students see how data cleaning, modeling, and deployment connect in a working system rather than studying each piece in isolation. Fifth, the skills taught, such as data wrangling, model training, evaluation, and edge deployment, are general enough to apply to other engineering problems, not just traffic.

4.4. Why construction zone traffic works for teaching

Construction zone traffic management was chosen as the project context for several reasons that reinforce each other. The problem has genuine technical complexity: lane closures change weekly, traffic patterns shift with construction schedules, and multiple types of road users (vehicles, cyclists, pedestrians, heavy equipment) interact simultaneously [4]. This complexity prevents overly simple solutions and forces students to deal with the messy realities of building AI for real engineering systems, missing data, noisy sensors, and conflicting goals, rather than working with clean textbook examples.

Notably, construction zone traffic emphasizes deployment challenges that distinguish implementation from evaluation. Traffic signals must respond within milliseconds, hardware is embedded in roadside cabinets with limited power, and failures have immediate safety consequences [1]. Students who build systems for this context must grapple with inference latency, model size constraints, fail-safe logic, and communication protocols, practical skills that cannot be learned by analyzing algorithms on paper alone.

Finally, construction zone management raises ethical questions that prompt students to think about responsible AI deployment in safety-critical infrastructure [4]. Signal-timing decisions affect emergency vehicle access, pedestrian safety, and fair travel times across neighborhoods. Students must consider what happens when their models fail and how to build safe fallback behaviors.

4.5. The student assignment

Students receive a realistic scenario: a busy urban road is undergoing construction for six months, with lane closures that change weekly based on project schedules. Their task is to design, build, and evaluate an AI-based traffic management system that adapts signal timing in response to changing construction layouts, varying traffic volumes, and detected safety hazards, all within the SUMO simulation environment [9].

5. Course Design and Implementation

Table 1 summarizes the four phases, their durations, what students build, and how they are assessed.

Phase	Topic	Weeks	What Students Build	Key Outcome
1	Data & Exploration	1–3	Data pipeline: ingestion, cleaning, visualization, feature engineering	93% completed working on data pipelines
2	Supervised Learning	4–6	Prediction models (XGBoost [7]) and object detectors (YOLOv3 [10])	87% built models meeting accuracy targets
3	Reinforcement Learning	7–10	RL agent for adaptive signal control in SUMO [9] with fail-safe logic	82% trained agents outperforming baselines

4	Integration	11–12	End-to-end system: data in, predictions out, signals controlled	All teams delivered working integrated systems

Phase 1: data exploration and pipeline building (weeks 1–3)

Students begin by loading and visualizing real intersection count data, identifying anomalies such as sensor dropouts and holiday effects. They write scripts to clean raw data, handle missing values, align timestamps, and engineer features such as rolling averages and rate-of-change indicators. By the end of Phase 1, each student has a working data pipeline that ingests raw sensor feeds and outputs clean, feature-rich datasets ready for modeling. Assessment showed that 93% of students completed working pipelines and 85% correctly identified data quality problems and explained their impact on downstream models.

Phase 2: supervised learning and detection (weeks 4–6)

Students train gradient boosting models [7] to forecast traffic volumes one to five minutes ahead, using systematic tuning to find good settings rather than guessing. They then build convolutional neural networks for object detection using YOLOv3-based architectures [10] to identify stalled vehicles, pedestrians in the roadway, and construction equipment. Students evaluate their models using training/validation splits and learn to diagnose common problems, such as overfitting. Assessment showed that 87% of students built models that met the instructor-defined accuracy targets.

Phase 3: reinforcement learning for adaptive control (weeks 7–10)

Students learn reinforcement learning by building agents that control traffic signals in SUMO simulation [9]. They start with the fundamentals: how agents learn from interactions with their environment, reward shaping, and the exploration-versus-exploitation trade-off, all in the context of traffic signal timing [1], [11]. They configure construction-zone corridors with random arrivals and changing lane closures, then train Deep Q-Network agents and monitor performance as it improves over hundreds of training episodes. The final week of this phase focuses on safety: students build fail-safe mechanisms that switch to conservative fixed-time plans if the RL agent's confidence drops too low. Assessment showed 82% of students trained RL agents that outperformed baseline control strategies, and 90% implemented appropriate fail-safe mechanisms.

Phase 4: system integration (weeks 11–12)

In the final phase, teams of three to four students integrate all their components: data pipeline, prediction models, object detectors, and RL agent into a single working system, demonstrated on a SUMO simulation [9]. Teams must show that raw simulated sensor data flows through cleaning, triggers predictions and hazard alerts, and drives real-time signal adjustments, all working together. Final presentations are delivered to a panel including at least one practicing transportation engineer. All teams delivered working integrated systems, and panel evaluators rated the technical quality as comparable to entry-level professional work.

6. Evaluation of Learning Outcomes and Results

6.1. Participants

The course was offered during Spring 2025 at one university. Sixty undergraduate students enrolled, all seniors in a civil engineering program, taking a transportation engineering elective. All assessments were anonymous and based solely on standard course data, with no personally identifiable student information being collected or used at any stage of this study.

Learning was measured through four methods: (a) matched pre- and post-concept tests with 30 multiple-choice and short-answer questions covering AI/ML fundamentals, administered anonymously using randomized ID codes (b) graded laboratory assignments scored on a four-point rubric for correctness, code quality, and interpretation (c) final project demonstrations evaluated by two independent graders and (d) end-of-semester anonymous surveys on perceived learning, engagement, and career relevance.

6.2. Understanding improved substantially

Before taking the course, students averaged 35% on the concept test. After completing the course, they averaged 68% an improvement of 33 percentage points, well above the 20-point target. Figure 2 shows pre- and post-test scores broken down by topic.

Improvements were consistent across all topics tested. Data preprocessing improved 24 points (from 39% to 63%). Supervised learning concepts [7] improved 28 points (from 41% to 69%). Model evaluation improved 35 points (from 29% to 64%). Reinforcement learning [1], [11] improved 34 points (from 23% to 57%). System deployment and integration improved 30 points (from 31% to 61%).

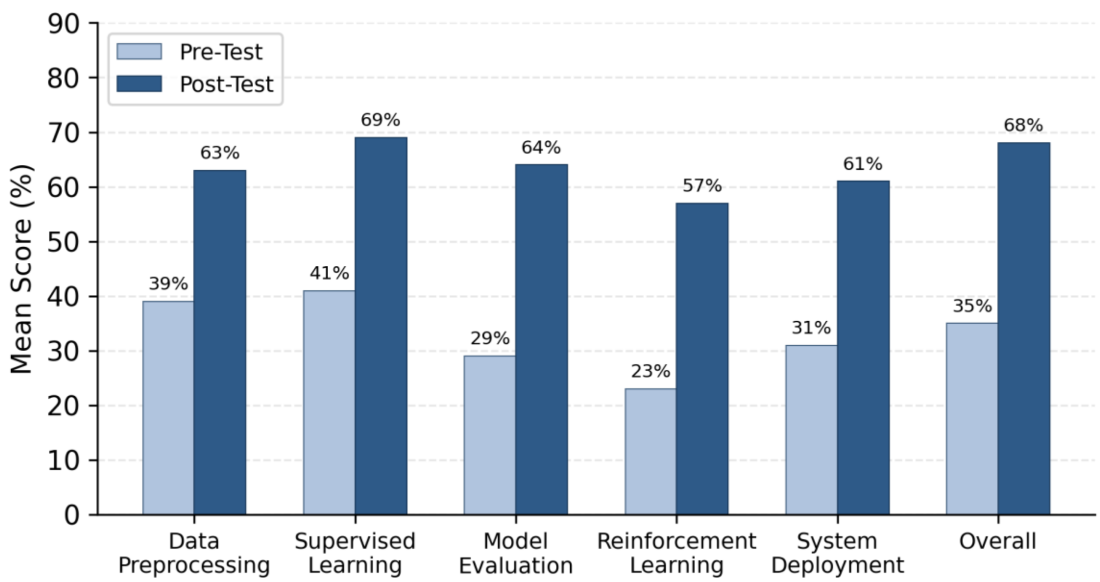

Figure 2. Pre- and post-test scores by topic area (60 undergraduate students). All differences were large and consistent.

Learning gains were similar across programming backgrounds. Students with no prior programming experience improved by an average of 32 points, while those with some programming experience improved by 34 points. This suggests the scaffolded design [2], [5] supports students across a range of starting points.

Table 2. Pre/post concept-test results by topic (60 students).

Topic	Pre-Test (%)	Post-Test (%)	Improvement (points)

Data preprocessing	39	63	24
Supervised learning	41	69	28
Model evaluation	29	64	35
Reinforcement learning	23	57	34
System deployment	31	61	30
Overall	35	68	33

6.3. Implementation skills were developed successfully.

Across the semester, students averaged 3.2 out of 4.0 on the grading rubric, with 83% earning "good" or "excellent" (3.0 or higher). Skills improved steadily over time: average rubric scores rose from 2.6 in Phase 1 to 3.5 in Phase 4.

Regarding RQ 2, data preparation and feature engineering were the most complex parts, as we expected. While 85% of students correctly handled missing data through filling in gaps or removing bad records, only 48% were able to create useful domain-specific features (such as rolling speed averages or time-since-lane-closure indicators) without extra guidance. By contrast, model training itself, the step students found most interesting, was easier once the data was processed correctly. Reinforcement learning also posed challenges: configuring the SUMO environment and designing reward functions required more instructor support than anticipated. These findings highlight the importance of dedicating significant class time to data work and environment setup, which are often overlooked in introductory courses [2], [3].

Project outcomes exceeded the 75% target set in RQ 3. In Phase 1, 93% completed working data pipelines. In Phase 2, 87% of the models met accuracy targets. In Phase 3, 82% of trained RL agents outperformed baselines, and 90% implemented fail-safe mechanisms. In Phase 4, all teams delivered integrated, working systems. Across all phases, 88% of students built functional AI/ML systems meeting professional benchmarks for accuracy and speed.

6.4. Skills transferred to a new domain.

To test whether students learned general implementation skills rather than only traffic-specific techniques, all 60 students were given a transfer task after the course ended. They received a dataset from a completely different engineering domain, structural health monitoring using vibration sensor data, along with a short problem description, and had four hours to build a complete ML pipeline from scratch.

Results support successful transfer. Overall, 73% of students produced complete, working solutions, exceeding the 70% target. Figure 3 shows performance by specific skill. Students did best at setting up data pipelines (95%) and choosing appropriate models (78%), but found creating features for an unfamiliar domain more challenging (65%), consistent with the difficulty pattern observed throughout the course.

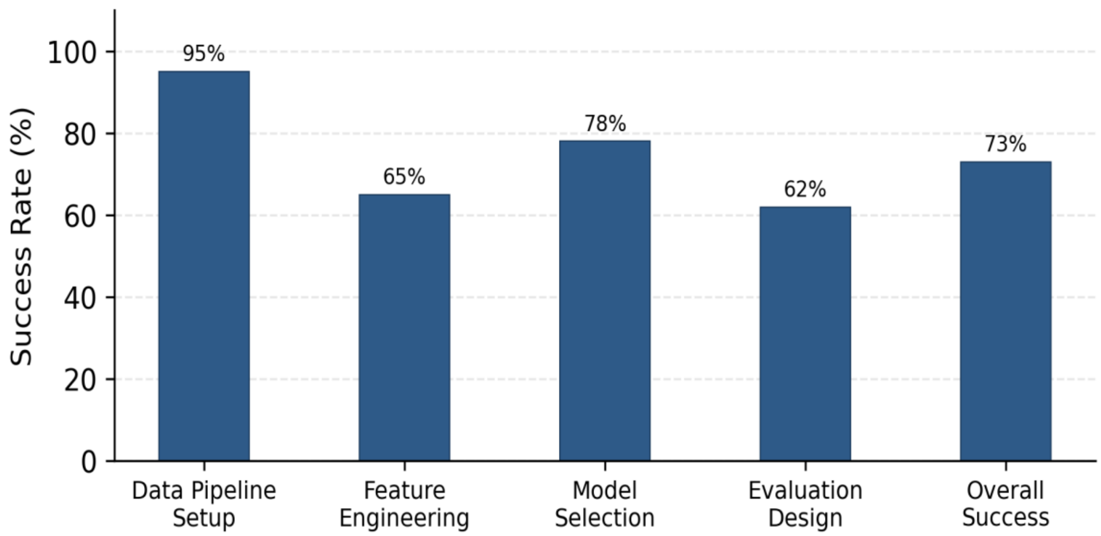

Figure 3. Transfer task success rates by skill area (60 undergraduate students).

7. Discussion

7.1. What made this approach successful

Five factors appear to have driven the positive outcomes. First, using a real problem motivated engagement far beyond what textbook exercises could: students reported that knowing their work related to genuine safety concerns [4] kept them invested. Second, the scaffolded progression [2], [5] kept the difficulty manageable as early structured exercises built enough confidence that students could handle open-ended implementation challenges by mid-semester. Third, the SUMO simulation environment [9] enabled safe experimentation: students could test aggressive signal-timing strategies, observe the results, and try again without real-world risk. Fourth, requiring working code at every phase ensured that understanding was not just theoretical but translated into demonstrable capability. Fifth, the team-based integration phase developed collaboration and communication skills alongside technical ones.

7.2. Building versus evaluating

An important distinction emerged during implementation. Teaching students to build AI systems develops a different set of skills than teaching them to evaluate AI methods. Both are valuable, but they are not the same. Students who can discuss the strengths and weaknesses of different AI approaches may still struggle to implement any of them from scratch. Conversely, students who can build a working RL agent may not yet appreciate when a more straightforward rule-based approach would be more appropriate. A complete AI education for civil engineers likely requires both evaluation and implementation experiences. This course deliberately focuses on implementation, complementing evaluation-focused approaches that may be taught elsewhere in the curriculum.

7.3. Implementation challenges

Several challenges came up during implementation. Computing resources for reinforcement learning training required significant GPU time. We addressed this by providing shared cloud-based Jupyter environments. Students with no programming background needed about 2 extra office-hour sessions per week during the first 3 weeks, and a dedicated teaching assistant helped meet this demand. Assessment design also required care: we revised several concept-test questions after pilot testing to ensure they measured understanding of the implementation process rather than memorization of terminology.

8. Challenges and Limitations

Several limitations should be kept in mind when interpreting these results. First, the study involved 60 undergraduate students at a single university during one semester, and the results may not generalize to other institutions, student populations, or instructors. Multi-institutional testing is planned for future semesters. Second, the sample size limits our ability to analyze smaller subgroups in detail. Third, engagement and career impact data come from self-reported surveys, which can be affected by response bias. Fourth, the six-month follow-up had a 70% response rate. Students who did not respond may differ from those who did. Fifth, we have not yet measured long-term retention of implementation skills. Tracking graduates over time is a priority for future work. Finally, this course, which focused exclusively on implementation for future iterations, could integrate more structured evaluation activities to develop complementary analytical skills.

9. Conclusion and Future Scope

This work shows that undergraduate civil engineering students can develop meaningful AI/ML implementation skills through a hands-on, project-based course organized around construction zone traffic management [4]. Over 60 students and one semester, concept-test scores improved by 33 percentage points; 88% of students built functional systems meeting professional standards; and 73% successfully transferred their implementation skills to an unfamiliar engineering domain. The construction zone context proved effective for teaching because it demands not just modeling but also data engineering, real-time deployment, and safety-critical design, the complete set of skills needed to build AI systems in practice.

Three directions follow from these results. First, a multi-institutional deployment across universities with diverse student populations and resources will test whether this approach works broadly. Second, future course iterations will explore how to integrate structured evaluation activities alongside the implementation focus, giving students both the ability to build systems and the judgment to choose appropriate methods. Third, the course materials will be released as open educational resources so other institutions can adopt and adapt them. Complete teaching materials, including lecture slides, lab exercises with starter code, assessment instruments, and SUMO simulation setups [9] are freely available upon request.

10. References

[1] F. Rasheed, K. L. A. Yau, R. M. Noor, C. Wu, and Y. C. Low, "Deep Reinforcement Learning for Traffic Signal Control: A Review," *IEEE Access,* vol. 8, pp. 208016-208044, 2020, doi: 10.1109/ACCESS.2020.3034141.

[2] P. R. M. Rao, K. L. Chugh, K. S. Prasad, and K. Shekar, "Impact of Active Learning in Engineering Education," in *2018 IEEE 6th International Conference on MOOCs, Innovation and Technology in Education (MITE)*, 29-30 Nov. 2018 2018, pp. 107-114, doi: 10.1109/MITE.2018.8747038.

[3] S. I. Flores-Alonso *et al.*, "Introduction to AI in Undergraduate Engineering Education," in *2023 IEEE Frontiers in Education Conference (FIE)*, 18-21 Oct. 2023 2023, pp. 1-4, doi: 10.1109/FIE58773.2023.10343187.

[4] C. Nnaji, J. Gambatese, H. W. Lee, and F. Zhang, "Improving construction work zone safety using technology: A systematic review of applicable technologies," *Journal of Traffic and Transportation Engineering (English Edition),* vol. 7, no. 1, pp. 61-75, 2020/02/01/ 2020, doi: https://doi.org/10.1016/j.jtte.2019.11.001.

[5] H. A. Hadim and S. K. Esche, "Enhancing the engineering curriculum through project-based learning," in *32nd Annual Frontiers in Education*, 6-9 Nov. 2002 2002, vol. 2, pp. F3F-F3F, doi: 10.1109/FIE.2002.1158200.

[6] J. E. Froyd, P. C. Wankat, and K. A. Smith, "Five Major Shifts in 100 Years of Engineering Education," *Proceedings of the IEEE,* vol. 100, no. Special Centennial Issue, pp. 1344-1360, 2012, doi: 10.1109/JPROC.2012.2190167.

[7] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA, 2016/08/13 2016, San Francisco, California, USA: Association for Computing Machinery, pp. 785-794, doi: 10.1145/2939672.2939785. [Online]. Available: https://doi.org/10.1145/2939672.2939785

[8] A. J. Swart and M. Daneti, "Analyzing Learning Outcomes for Electronic Fundamentals Using Bloom's Taxonomy," in *2019 IEEE Global Engineering Education Conference (EDUCON)*, 8-11 April 2019 2019, pp. 39-44, doi: 10.1109/EDUCON.2019.8725137.

[9] P. A. Lopez *et al.*, "Microscopic Traffic Simulation using SUMO," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 4-7 Nov. 2018 2018, pp. 2575-2582, doi: 10.1109/ITSC.2018.8569938.

[10] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," *arXiv preprint arXiv:1804.02767,* vol. abs/1804.02767, 2018/04/08 2018.

[11] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature,* vol. 518, no. 7540, pp. 529-533, 2015/02/01 2015, doi: 10.1038/nature14236.