Engineering Educators Bringing the World Together
2025 ASEE Annual Conference & Exposition
Palais des congrès de Montréal, Montréal, QC • June 22–25, 2025
Paper ID #45845

Integrating AI/ML Learning in Senior Projects for Mechanical Engineering Technology Students

Dr. Wenhai Li, State University of New York, College of Technology at Farmingdale

Assistant Professor in Department of Mechanical Engineering Technology, Farmingdale State College, Farmingdale, NY 11735

Dr. Yue Hung, Farmingdale State College

Dr. Yue (Jeff) Hung holds degrees in engineering and technology disciplines (Ph.D. in Materials Science and Engineering, M.S in Mechanical Engineering, and B.S in Manufacturing Engineering Technology). He has over 20 yearsâ€™ experience in Computer-Aided

Dr. Gonca Altuger-Genc, State University of New York, College of Technology at Farmingdale

Dr. Gonca Altuger-Genc is an Associate Professor at State University of New York - Farmingdale State College in the Mechanical Engineering Technology Department.

Sen Zhang, State University of New York, Oneonta

Sen Zhang has been teaching Computer Science at SUNY Oneonta since 2004. The recent courses he has been teaching include Python, Artificial Intelligence, Intro to Machine Learning as special topics, Intro to Robotics, Internet Programming, Linux, and Software Design and Development (which typically contains a sizable team term-project as capstone experience). He publishes on data mining algorithms, conducts educational research and values project-based learning.

Dr. Akin Tatoglu, University of Hartford

Akin Tatoglu is an Assistant Professor of Mechanical Engineering at University of Hartford, CT. He received his Ph.D. degree in mechanical engineering from Stevens Institute of Technology, NJ, in 2015. His research focuses on robotics, collaborative navig

Dr. Zhou Zhang, State University of New York, College of Technology at Farmingdale

I am an Assistant Professor at SUNY Farmingdale State College. My teaching and research interests include robotics and virtual reality in engineering education. I have a Ph.D. and a bachelor's degree in Mechanical Engineering, and my master's degree is in Electrical Engineering. I have over seven years of industrial experience as an electrical and mechanical engineer. I also have extensive teaching and research experience with respect to various interdisciplinary topics involving Mechanical Engineering, Electrical Engineering, and Computer Science.

Integrating AI/ML Learning in Senior Projects for Mechanical Engineering Technology Students

Wenhai Li, Yue Hung, Gonca Altuger-Genc, Zhou Zhang
Department of Mechanical Engineering Technology, Farmingdale State College, Farmingdale, NY 11735

Sen Zhang
Department of Mathematics, Computer Science and Statistics, State University of New York College at Oneonta, Oneonta, NY 13820

Akin Tatoglu
Department of Mechanical, Aerospace, and Acoustical Engineering, University of Hartford, West Hartford, CT 06117

Abstract

This study explores the integration of Artificial Intelligence (AI) and Machine Learning (ML) education into the Senior Projects course for Mechanical Engineering Technology (MET) students, addressing the growing demand for AI/ML skills in engineering fields. In the absence of a dedicated AI/ML course within the current MET curriculum, the initiative bridges this gap through a dual approach: weekly lectures tailored to MET students, focusing on accessible tools and practical applications, and senior projects specifically designed to apply AI/ML concepts to solve engineering problems. A comprehensive assessment plan, incorporating pre- and post-course identical quizzes, topic-specific quizzes, self-evaluations and reflections, demonstrated significant learning gains. The successful completion of these AI-focused senior projects highlights the effectiveness of this approach in equipping students with essential AI/ML skills. This innovative strategy not only addresses the curriculum gap but also offers a scalable model for integrating emerging technologies into undergraduate engineering education.

Introduction

The integration of Artificial Intelligence (AI) and Machine Learning (ML) into modern engineering practices has created an urgent need for engineers with AI/ML skills to tackle challenges in automation, robotics, preventive maintenance, defect detection, system optimization, and beyond. This integration underscores the transformative potential of AI/ML in engineering education, necessitating curriculum advancements to prepare students for the evolving technological landscape [1]. This need is driven not only by industry demands but also by students, who increasingly see AI/ML expertise as vital for their future careers and expect opportunities to apply these skills in real-world engineering projects. Numerous national reports, including those by the McKinsey Global Institute [2], the National Institute of Standards and Technology (NIST) [3], and the Society of Manufacturing Engineers (SME) [4], highlight the growing demand for engineers who can integrate AI tools into manufacturing and automation workflows. Through regular interactions with Mechanical Engineering Technology (MET) students, it has become clear that there is a strong interest in learning the fundamentals of AI/ML. Many students frequently ask whether these topics are covered in existing courses, highlighting the gap between their aspirations and the current curriculum. However, traditional MET programs often lack dedicated AI/ML courses, constrained by the already packed schedule of core MET subjects. To bridge this gap, we introduced a novel teaching approach in this study, which enhances the current Senior Projects course by incorporating AI/ML-focused lectures and designing capstone projects that apply AI/ML concepts to solve real-world engineering problems.

MET students face distinct challenges when engaging with AI/ML concepts, largely due to their specialized academic preparation. While the traditional MET curriculum excels in applied engineering principles, it typically offers limited exposure to the advanced mathematical foundations of AI/ML, such as multivariable calculus and linear algebra. Additionally, MET students often have programming experience confined to basic engineering applications, which can create significant obstacles when working with complex AI/ML algorithms and modern development tools. Furthermore, the conceptual framework of AI/ML, including neural network architectures, optimization techniques, and data modeling paradigms, adds another layer of difficulty, as these topics are traditionally grounded in computer science rather than engineering. These challenges highlight the need for a thoughtfully structured pedagogical framework that bridges the gap between traditional MET competencies and essential AI/ML knowledge, while maintaining practical relevance to engineering applications.

Given the challenges outlined above, teaching AI/ML to MET students requires a specialized approach that differs significantly from traditional computer science frameworks. Conventional AI/ML education often focuses on theoretical foundations, advanced mathematics, and extensive programming, which can be misaligned with the specialized academic background of MET students. Instead, a methodology that builds on their strengths in applied engineering and practical problem-solving should be far more effective. The Senior Projects course offers an ideal platform for this adapted approach. Project-Based Learning (PBL) has been recognized as an effective pedagogical approach in engineering education, fostering critical thinking and real-world problem-solving skills [5]. As a capstone experience, it naturally incorporates project-based learning that aligns with the applied focus of MET education. Positioned at the culmination of the curriculum, this course ensures that students have the technical foundation necessary to integrate AI/ML concepts into their projects. This thoughtful integration bridges the gap between understanding theory and applying it in practice, enabling MET students to acquire essential AI/ML skills in a way that is accessible and directly relevant to their professional goals.

To address these challenges, we implemented a two-part teaching strategy. First, weekly ninety-minute lectures introduce fundamental AI/ML concepts in the context of real-world engineering applications. These lectures focus on using established AI/ML tools, preparing students to be practitioners who can apply existing solutions rather than developers creating new algorithms. This aligns with industry needs for practical expertise. Second, AI/ML-focused capstone projects allow students to work with accessible AI platforms to solve real engineering problems. This approach simplifies learning while building skills that are immediately useful. By prioritizing practical applications, this framework helps students connect their academic work with the expectations of modern engineering careers.

To evaluate the effectiveness of this approach, a comprehensive assessment framework was developed. This includes pre- and post-course assessments to measure knowledge gains, topic-specific evaluations to assess comprehension, and reflective exercises to gather qualitative feedback. Additionally, capstone project outcomes provide insights into students' ability to apply AI/ML concepts to real-world challenges, emphasizing creativity, problem-solving strategies, and the integration of learned concepts. The results of these evaluations highlight the success of this methodology in bridging curricular gaps and preparing MET students for the modern engineering careers.

The subsequent sections of this paper present a detailed examination of the redesigned Senior Projects course structure and its outcomes. We will first lay out the details of our design, including descriptions of the weekly AI/ML lectures and capstone projects, followed by an outline of the assessment methodology and analysis of results. Finally, we conclude with a discussion of implications for curriculum development and key takeaways for broader applications.

Contents of Course

Building on the challenges and needs identified earlier, the course design focuses on a dual approach: structured weekly lectures and hands-on senior projects. The lectures aim to provide MET students with a practical understanding of AI/ML concepts using accessible tools, while the senior projects offer an opportunity to apply this knowledge to real-world engineering problems. The detailed design for each part is outlined below.

A) Weekly Lectures

The weekly lecture component serves as a foundational pillar of the redesigned Senior Projects course, equipping students with the essential theoretical frameworks needed for meaningful engagement with their AI/ML-focused capstone projects. Providing this structured knowledge is vital, as diving directly into project work without a theoretical background risks reducing the learning experience to surface-level tool usage, rather than enabling the deeper understanding required for practical applications in industry.

The primary goal of these lectures is not to train students to become AI/ML experts, which would be impractical within the scope of a single course. Instead, they aim to develop informed practitioners who can effectively utilize AI/ML technologies. By introducing core AI/ML principles, demonstrating systematic problem-solving approaches, and building familiarity with user-friendly AI/ML tools, the lectures provide students with an accessible entry point into this field. For many MET students, this course marks their first exposure to AI/ML concepts. These lectures are intentionally introductory, serving as a gateway to AI/ML learning by lowering barriers and equipping students with the confidence to explore these technologies further.

The weekly lectures are designed to equip MET students with practical AI/ML skills tailored to their future engineering careers. Rather than adopting the traditional computer science approach to AI/ML education, which spans multiple courses and focuses on advanced modeling, these lectures emphasize foundational concepts. To guide the curriculum, we identified a set of essential AI/ML skills that MET students should develop, as illustrated in Figure 1. These skills include understanding basic ML concepts, mastering the training pipeline (e.g., data preprocessing, model selection, training, and evaluation), interpreting outputs like accuracy and confidence, and applying pre-trained models to solve engineering problems. The lectures also introduce existing basic Python libraries, ensuring students gain hands-on experience with accessible platforms. With the inclusion of this targeted content, MET students acquire practical AI/ML skills in a streamlined and accessible manner, preparing them to meet modern engineering challenges effectively.

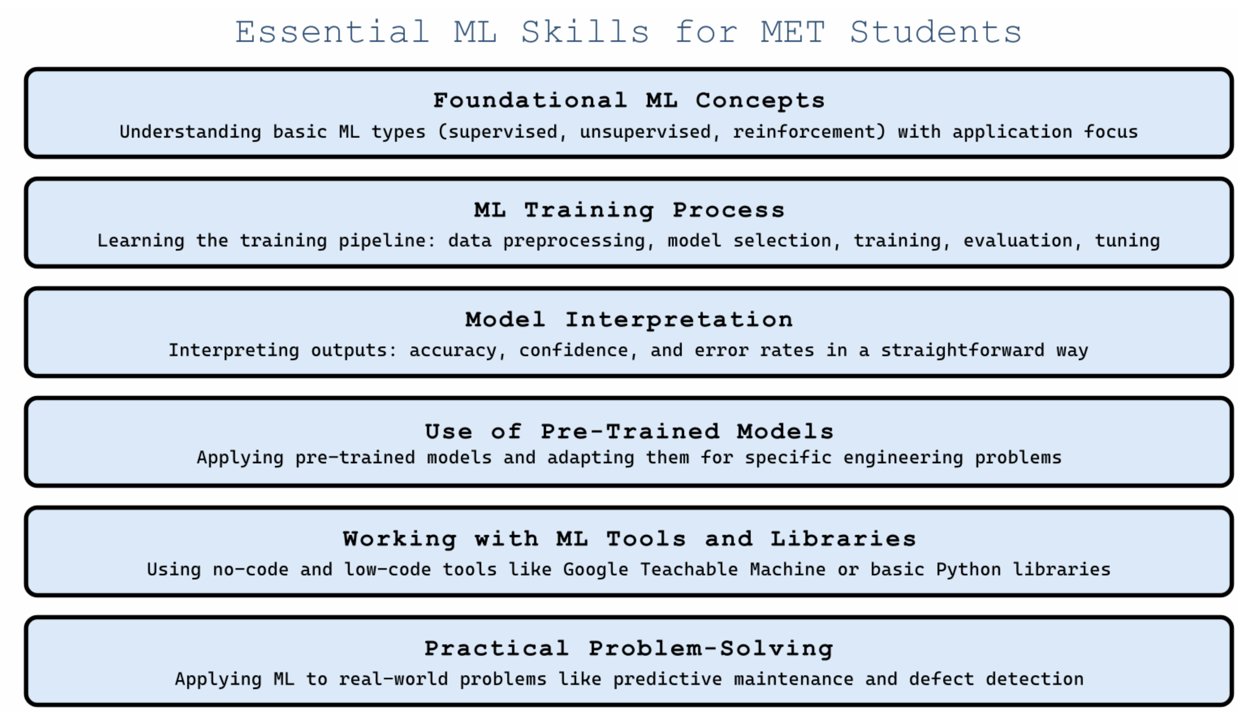

Figure 1. Essential ML Skills for MET Students

The lecture series is structured around eight key topics. Below, we detail the content and pedagogical strategies for each topic, highlighting their alignment with MET students' needs and the course's primary objectives:

- <u>Topic 1: Introduction to Machine Learning</u> – This topic lays the groundwork by introducing core ML concepts and their relevance to engineering. It uses AI/ML applications in engineering to illustrate the capabilities of ML and explain the differences between supervised, unsupervised, and reinforcement learning. By focusing on practical, relatable examples, the lecture helps MET students understand how AI/ML methods are applied to solve real-world engineering problems.
- <u>Topic 2: Supervised Learning</u> – This topic introduces the fundamental concepts of supervised learning, focusing specifically on classification problems. To lower the learning barrier for MET students, the lecture covers the simplest model, the Perceptron, while briefly mentioning more advanced models without delving into detail. A complete mathematical breakdown of the Perceptron model is provided, covering key elements like input features, weights, biases, and the activation function. This is followed by an explanation of how the model adjusts weights during training using gradient descent and

cost functions, ensuring students grasp these foundational concepts without unnecessary complexity. To solidify understanding, a Python coding example demonstrates the implementation of the Perceptron model, its training process using a simplified dataset, and its application for classification predictions. The step-by-step example covers essential aspects like data preparation, weight updates, and performance evaluation, providing a hands-on understanding of the concepts. Additionally, the lecture ties these ideas to real-world engineering scenarios, such as classifying defective parts on an assembly line or sorting materials based on predefined characteristics, reinforcing the practical relevance of supervised learning. This structured approach equips MET students with a strong foundational understanding of supervised learning and prepares them to apply these techniques confidently in their projects.

- Topic 3: Dataset Preparation and Model Evaluation – This topic emphasizes the critical role of data quality and preparation, as well as the model performance evaluation, in the machine learning process. It begins with an overview of data types and encoding techniques, explaining why encoding is necessary and introducing common encoding methods. The importance of high-quality data is emphasized, along with characteristics that define it. Essential pre-processing techniques, including feature selection, dimensionality reduction, shuffling, and scaling, are introduced, with advanced topics such as handling data imbalance and detecting outliers briefly discussed. The lecture then explains the rationale behind splitting datasets into training, validation, and test sets, with a focus on effective split ratios. Key model evaluation metrics such as accuracy, precision, recall, and confusion matrix are introduced, along with guidance on selecting appropriate metrics for different applications. The concept of overfitting and underfitting, and their relationship to bias and variance, is introduced as a pivotal topic that will be revisited throughout the course, with strategies to address these challenges briefly covered. Common training hyperparameters, such as learning rate, batch size, and epochs, are explained in detail, highlighting their impact on training and model performance. The process of hyperparameter tuning is also discussed, illustrating its role in optimizing model outcomes. To solidify these abstract concepts, a Python example using the scikit-learn package demonstrates data pre-processing, dataset splitting, model evaluation, and hyperparameter tuning with a simple dataset, providing MET students with practical and relatable insights into the machine learning workflow.

- Topic 4: Regression and Unsupervised Learning – This topic bridges the gap from Topic 2 by introducing the regression aspect of supervised learning. The discussion begins by differentiating classification and regression within supervised learning, using relatable engineering application examples to illustrate these distinctions for MET students. The focus is placed on the simplest regression model, linear regression, with its mathematical details fully explained, while other regression models are only briefly mentioned to avoid overwhelming students with complex mathematics. A practical Python example demonstrates how linear regression is modeled and applied, giving students a concrete understanding of its implementation. The second part of the lecture transitions to unsupervised learning, which, while less critical for engineering applications, is still an important foundational concept. Due to its secondary importance, unsupervised learning is covered concisely and combined with regression in a single lecture. The primary focus is on K-Means Clustering, the most intuitive unsupervised learning technique. The lecture provides a detailed explanation of K-Means, followed by a Python example that shows how clustering can be implemented, helping MET students grasp this concept through hands-on experience.

- Topic 5: Deep Learning and Neural Networks – This topic introduces MET students to neural networks by first drawing analogies between biological neurons and artificial neurons, as well as between human neural networks and artificial neural networks, to illustrate how these systems handle complex tasks. The lecture then explains the operation of a simple single-layer neural network, covering inputs, weights, biases, and outputs with a clear mathematical layout. This foundation is further extended to multi-layer neural networks, emphasizing the roles of input, hidden, and output layers while avoiding advanced mathematical complexities to keep the material accessible for MET students. The effects of network depth and layer width are discussed in the context of overfitting and underfitting, helping students understand the practical implications of network architecture. The lecture also introduces key activation functions, such as Sigmoid, Tanh, ReLU, and Softmax, with a focus on selecting appropriate functions based on the application and layer context. Forward propagation is explained in detail with Python code to reinforce understanding, while backward propagation, being more mathematically complex, is introduced at a high level to provide conceptual clarity without overwhelming students. For those interested, optional resources with detailed math and Python code for backward propagation are provided. To further enhance comprehension, a practical example using TensorFlow is included, demonstrating how to construct a neural network for a simple image classification task. This hands-on example showcases the ease of implementing neural networks using existing tools, helping MET students gain confidence in applying artificial neural networks to solve engineering problems.

- Topic 6: Convolutional Neural Networks (CNNs) – This topic introduces MET students to CNNs as specialized artificial neural networks designed for image processing tasks. The foundational structure of CNNs is explained, covering input layers, convolutional layers, pooling layers, fully connected layers, and output layers. The concept of convolution is emphasized, with key terms like kernel, padding, and stride briefly introduced. Practical examples demonstrate how different kernels can modify images to extract meaningful features, aiding students in understanding the functionality of convolutional layers. The lecture also explains the pooling operation and the distinct effects of max pooling and average pooling on images, highlighting the role of pooling layers in reducing data dimensionality while preserving critical information. To provide a broader perspective, a quick overview of popular CNN architecture is included, giving students insight into how real-world CNNs are structured. The application of CNNs to Computer Vision (CV) is then explored, introducing key CV tasks such as image classification, object detection, and object segmentation, which demonstrate the versatility of CNNs in engineering problems. Students are introduced to OpenCV [6], a widely used library for computer vision tasks, and its compatibility with machine learning tools. The session concludes with two hands-on projects: first, using TensorFlow [7] to build a simple CNN for an image classification task, showcasing the advantages of CNNs over traditional artificial neural networks; and second, leveraging OpenCV to capture images from a webcam and perform basic image classification using a pre-trained model created earlier. These exercises help MET students gain confidence in applying CNNs and computer vision tools to solve real-world problems.

- Topic 7: Natural Language Processing (NLP) – This topic introduces MET students to the basics of NLP, starting with Recurrent Neural Networks (RNNs) and their ability to handle sequential data such as human language. The structure of RNNs is explained conceptually without complex mathematics, making it accessible for MET students. The limitations of RNNs are discussed, followed by the introduction of improvements through Long Short-Term Memory (LSTM) networks and Gated Recurrent Units (GRUs), which enhance sequence-processing capabilities. A practical Python example demonstrates building a simple LSTM RNN using TensorFlow for sentiment analysis, highlighting essential text preprocessing techniques like text vectorization and character mapping. The session then transitions to the transformative role of Transformers in NLP, focusing on their improvements over RNNs without delving into advanced mathematical details. This leads to the discussion of Large Language Models (LLMs), such as ChatGPT [8], emphasizing their engineering applications. Students learn to use ChatGPT's API to integrate NLP into workflows, with a Python example showing how to send prompts, receive responses, and maintain conversational context by including prior interactions. These hands-on examples help MET students understand the practical applications of NLP tools like LLMs in solving engineering problems.
- Topic 8: Reinforcement Learning (RL) – The last topic introduces the foundational concepts of reinforcement learning, emphasizing its applications in robotics, autonomous driving, and other engineering fields. The basic structure of RL is explained, including key concepts such as value-based and policy-based RL, along with a brief introduction to the mathematical foundations like Markov Decision Processes and the Bellman Equation. Due to the mathematical complexity of RL, the discussion of these topics is kept at an introductory level to provide MET students with essential background knowledge without delving into overwhelming details. RL is framed as an optimization process aimed at maximizing cumulative rewards, with discussions on the balance between immediate and future rewards. The lecture focuses on Q-learning as a simple yet powerful RL algorithm, explaining the creation of a Q-table, the application of the Bellman Equation, and the significance of key parameters such as learning rate and discount factor. Additionally, the exploration-exploitation tradeoff is introduced, with an explanation of how the "ϵ-greedy policy" can effectively balance exploration and exploitation. To consolidate these challenging concepts, the lecture concludes with a hands-on example using Q-learning to solve a maze problem. This practical exercise demonstrates how RL can find the shortest path from a start to a goal point while avoiding obstacles, showcasing RL's potential and providing students with a tangible understanding of its applications.

One major challenge of the lecture series lies in balancing the additional AI/ML content with the existing demands of the Senior Projects course, as students must complete capstone projects to meet graduation requirements. Given the time constraints and the students' already busy schedules nearing graduation, the lectures are designed as voluntary seminar-like sessions rather than a mandatory standalone AI/ML course. To minimize the added workload, no additional assignments are required for the lectures, ensuring that the students can engage with the content without feeling overburdened. Each weekly lecture is strictly limited to 90 minutes, with larger topics spanning two weeks as necessary. This format relies heavily on students' self-motivation to dedicate extra time outside of class to deepen their understanding of AI/ML concepts. To gauge learning outcomes, each topic is accompanied by a short quiz comprising multiple-choice questions focused on key concepts. Small extra credit incentives are offered to students who complete these quizzes, encouraging active participation and reinforcing their engagement with the material.

B) AI/ML-Focused Senior Projects

In the MET program at Farmingdale State College, the capstone project serves as a culminating academic experience, requiring students to apply the knowledge and skills acquired throughout the program to solve practical engineering problems. These projects aim not only to assess and demonstrate students' learning outcomes and technical capabilities but also to act as a dynamic learning platform, enabling students to acquire additional MET-related skills that prepare them for their future careers. AI/ML-focused capstone projects need to be carefully designed to align with these objectives, ensuring their relevance to MET by addressing real-world engineering challenges instead of focusing on purely computer science-based topics. To accommodate the complexity and breadth of these projects, they are usually completed in groups, which allows for larger project scopes and provides students with invaluable experience in teamwork, collaborative problem-solving, and project management, which are critical skills for success in professional engineering environments.

In the Fall 2024 semester, this teaching plan was executed, and students successfully carried out several AI/ML-related senior projects. The selection of project topics resulted from a collaborative process between students and advisors. In some cases, students proposed initial ideas that were refined and approved by their advisors. In other instances, advisors suggested foundational concepts, which students then modified and tailored to align with their interests and skills. This joint effort ensured that the projects were both challenging and relevant to the students' career aspirations while meeting the program's educational goals. Below, we outline the details of each project and how they were implemented.

Project #1: Automated Sorting System Using Computer Vision:

This project focused on developing an automated sorting system that integrates computer vision and robotics to classify and organize objects based on their visual features. The system utilized a UR3 robotic [9] arm equipped with a Logitech C920 camera [10] to identify objects labeled with distinct letters. Students aimed to implement a real-time solution that could detect, classify, and sort objects into designated bins, simulating industrial automation scenarios.

A central feature of this project was the use of Google Teachable Machine [11], a no-code platform designed to simplify the training of AI models. As shown in Figure 3, this tool allowed students to create an effective classification model by uploading labeled image datasets and leveraging the platform's user-friendly interface to configure and train the model. By removing the complexity of manual coding and algorithm development, Google Teachable Machine enabled MET students, many of whom have limited programming experience, to focus on the application and deployment of AI/ML tools.

The trained model was integrated using Python, with OpenCV handling real-time image capture and the *ur_rtde* package [12] managing robotic arm control. This hands-on implementation provided students with practical exposure to AI/ML workflows, emphasizing the power of no-code tools in reducing the learning curve and enabling rapid prototyping. The project exemplified how

AI/ML technologies can be applied to solve engineering challenges in automation and equipped students with skills that align with industry needs.

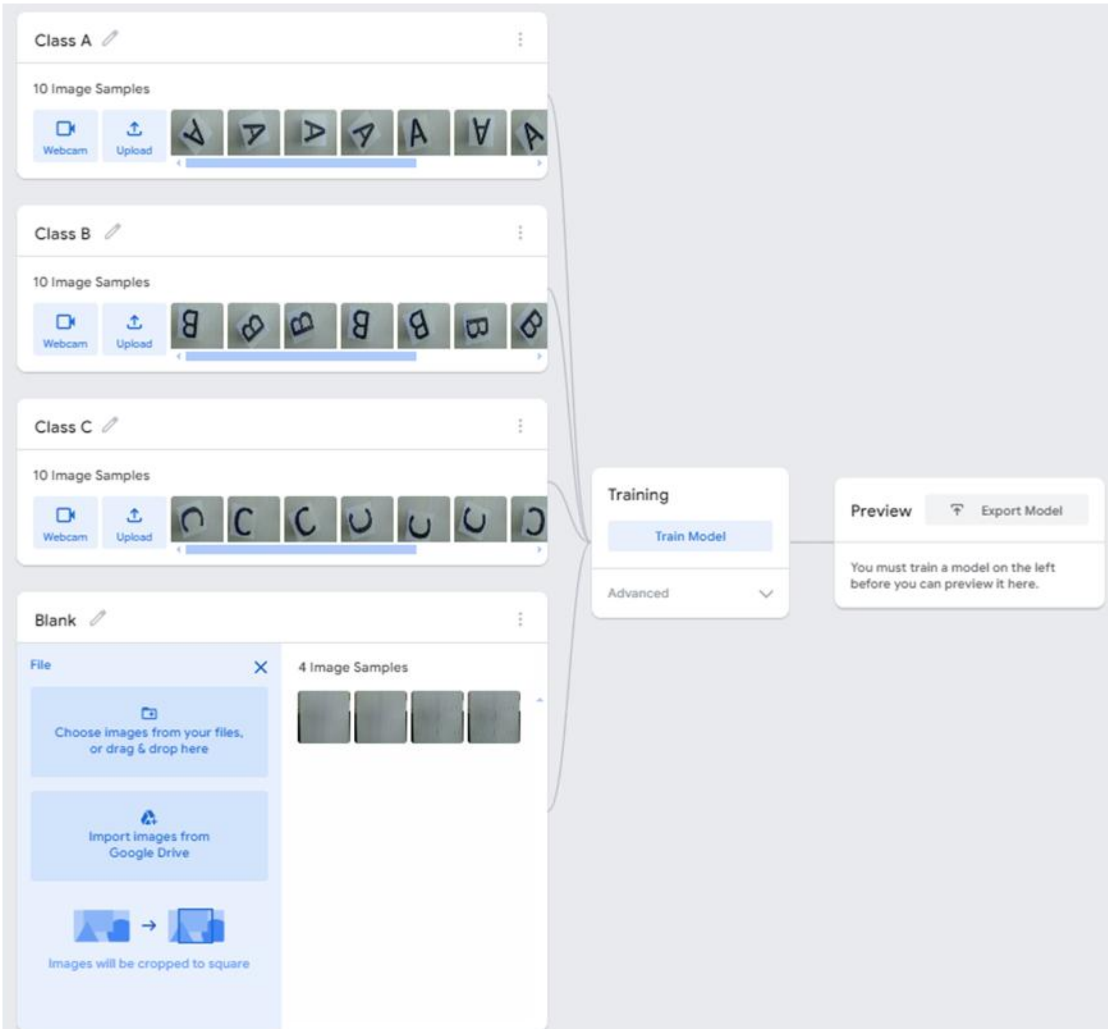

Figure 3. Project #1 - Demonstration of the model training process using Google Teachable Machine

Project #2: Automated Wind Turbine Blade Inspection

This project focused on developing an automated system to inspect wind turbine blades (WTBs) for defects such as dents and cracks, leveraging machine learning and robotics. The students designed a solution that integrated robotics and computer vision to tackle the challenge of identifying and categorizing defects on real-world blade images and data.

The project utilized Roboflow [13], a no-code platform for object detection, to streamline the machine learning pipeline. Students uploaded images of wind turbine blades to the platform, annotated the defects directly through an intuitive interface, and used Roboflow's automated tools to preprocess the data and train a custom object detection model. Figure 4 shows the annotation process in Roboflow. This process enabled students to generate a high-quality model without diving into the complexities of traditional coding or ML pipelines. The trained model was deployed via Roboflow's API for real-time defect detection using a webcam.

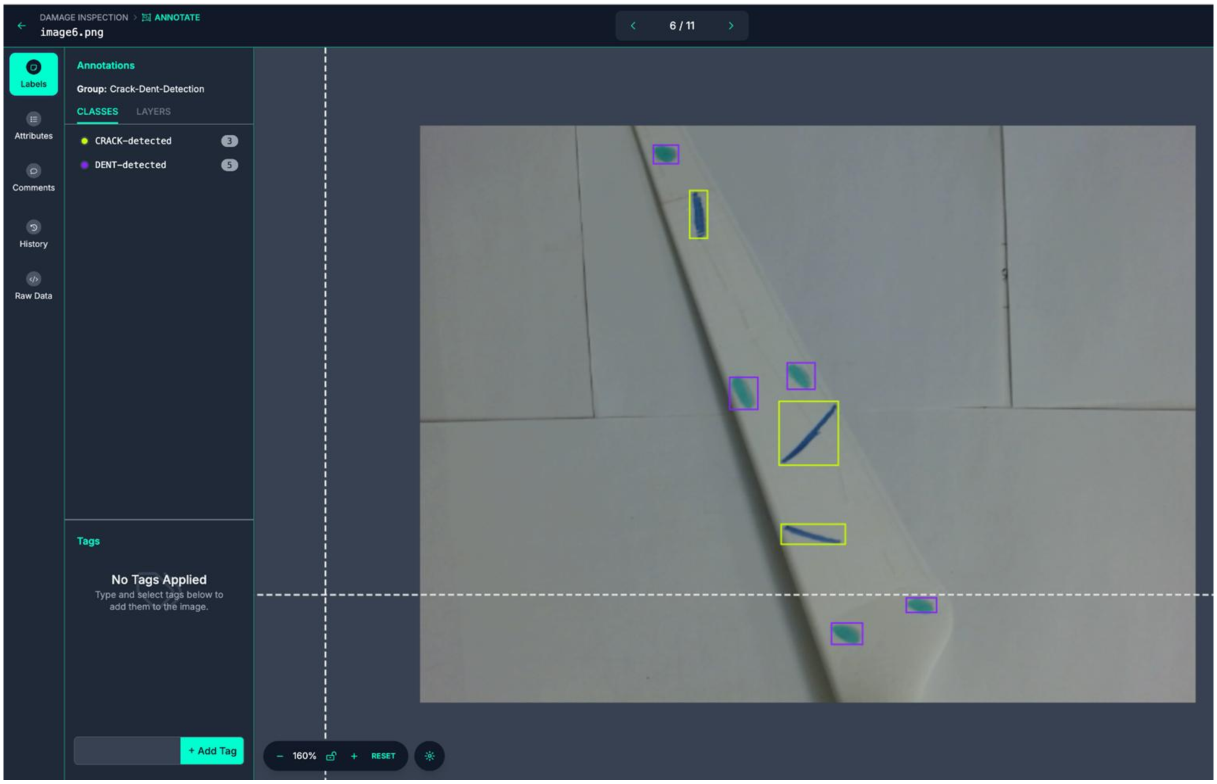

Figure 4. Project #2 - Annotation Process in Roboflow

Python programming was employed for essential tasks, including camera calibration and interfacing the detection model with the robotic system. The universal robot performed inspection routines guided by the AI model, showcasing how robotics and AI can work together in real-time to solve engineering challenges.

By incorporating accessible tools like Roboflow, this project significantly reduced the learning curve for MET students, allowing them to focus on understanding the application of AI/ML in engineering rather than the intricacies of model development. The hands-on nature of the project, combined with the collaborative team environment, provided students with practical experience in applying cutting-edge technology to solve real-world challenges in the wind energy sector.

<u>Project #3: Voice-Controlled Robotic Arm with NLP</u>

This project explored the integration of voice recognition and natural NLP technologies to enable intuitive, voice-based control of a Universal Robots arm, making it a compelling demonstration of AI's potential in robotic applications. The project was implemented in three progressive phases, each building on the capabilities developed in the previous one to achieve increasingly sophisticated levels of control.

In the initial phase, the students utilized the Google Cloud Speech API [14] to implement a basic voice-command interface. This setup enabled the robot to interpret simple spoken instructions, such as directional movements, and translate them into precise robotic actions. This foundational step not only showcased the effectiveness of voice recognition but also set the stage for more advanced natural language capabilities.

The second phase introduced NLP through the integration of ChatGPT through OpenAI's API [15] to process more complex voice commands. For example, a command such as "move closer by five centimeters" required the system to interpret both the direction and the distance while converting units into the robot's operational framework. An example prompt designed to send to ChatGPT is shown in Figure 5. ChatGPT's ability to parse and contextualize natural language allowed for seamless communication between the user and the robot, significantly expanding the range of instructions that the system could handle. This phase highlighted the system's ability to bridge natural language understanding with precise robotic control.

```
prompt = f"""
A robot arm is ready for operation. The robot is waiting for commands so it can move the end effector tool in
a specific direction and by a specific distance. The possible directions are:

'up': move the robot's end effector up, which is the positive z direction of the base coordinate.
'down': move the robot's end effector down, which is the negative z direction of the base coordinate.
'left': move the robot's end effector left, which is the negative x direction of the base coordinate.
'right': move the robot's end effector right, which is the positive x direction of the base coordinate.
'forward': move the robot's end effector closer to the operator, which is the negative y direction of the
base coordinate.
'backward': move the robot's end effector further from the operator, which is the positive y direction of the
base coordinate.

The following text was transcribed from a speech command: '{text}'. Please determine which direction the
command is most likely to move the robot and also determine the distance the command wants the robot to move.
If the direction cannot be determined, the direction will be set to "NULL". If the distance cannot be
determined, the distance will be set to 0.

Please only return the response of the "direction" and "distance" determined from the text command in
stringified JSON format without any extra characters or spaces. Note that if the distance has a value rather
than "DEFAULT", please convert the distance in the units of meters and only return the value.
"""
```

Figure 5. Project #3 - Example of the prompt designed to be sent to ChatGPT

In the final phase, the project demonstrated the use of ChatGPT to implement a context-aware dynamic block-stacking task. The system maintained a digital twin of the workspace, enabling it to track the real-time configuration of blocks and compute the appropriate pick-and-place coordinates based on user commands, such as "stack the left block on the right." The robot executed these commands accurately, considering the current arrangement of blocks. This innovative use of NLP ensured that even vague or context-dependent instructions were effectively interpreted and executed, showcasing the potential for advanced automation applications.

Throughout the project, Python was extensively used to integrate and implement the various functionalities, including speech recognition and natural language processing. This work demonstrated not only the feasibility of voice-enabled robotic systems but also the significant role that NLP can play in creating user-friendly robotic interactions. By successfully combining voice recognition and advanced NLP tools, the project provided an accessible and practical example of how emerging technologies can enhance the usability and functionality of robotic systems.

The projects above are designed to align with the unique capabilities and learning needs of MET students. Recognizing the challenges posed by traditional AI/ML education, the projects are carefully developed to leverage accessible, easy-to-use AI tools such as Google Teachable Machine and RoboFlow. The successful completion of these projects by the students demonstrates the feasibility of integrating AI/ML into MET senior projects, confirming that students are capable of engaging with and applying these technologies effectively.

Assessment Strategy

The assessment strategy for this course combines quantitative and qualitative methods to evaluate student learning outcomes comprehensively [16]. Each method is tailored to align with the course objectives and pedagogical framework, providing a multifaceted understanding of students' engagement and skill development.

1. Pre- and Post-Course Quiz – A 50-question multiple-choice quiz was administered at the beginning and end of the course to assess overall learning gains. The quiz covered fundamental AI/ML concepts, ensuring a consistent benchmark for evaluating knowledge acquisition. By comparing pre- and post-course performance, we quantified the improvement in students' understanding of AI/ML principles and their ability to apply them to engineering problems.
2. Topic-Specific Quizzes – At the conclusion of each lecture topic, students completed short quizzes ranging from 8 to 14 multiple-choice questions. These quizzes focused on the specific concepts taught during each topic. They served as formative assessments, providing immediate feedback on students' grasp of the material and identifying areas requiring further clarification.
3. Self-Evaluation and Reflection Survey – At the end of the course, students participated in a self-evaluation and reflection survey. This survey aimed to capture their perceptions of learning, challenges encountered, and suggestions for improving the course. It provided valuable qualitative data, complementing the quantitative assessments, and offered insights into the students' experiences and engagement with the AI/ML content.
4. Senior Project Evaluation – The AI/ML-focused senior projects were assessed through their successful completion, as well as presentations and reports. Each project was evaluated based on its technical implementation, innovation, teamwork, and the application of AI/ML tools to solve engineering problems. The presentations were attended by other faculty members who also provided additional feedback and grade suggestions. This comprehensive evaluation ensured that the projects not only tested students' understanding of AI/ML concepts but also demonstrated their ability to apply these skills in practical, real-world scenarios.

The effectiveness of this teaching approach was evaluated using the outlined assessment methods. The next section presents the results, analyzing their impact on student learning and engagement.

Results and Analysis

A total of 10 students participated in the Fall 2024 semester of the course, providing the primary dataset for this analysis. Since the quizzes and surveys were voluntary and not mandatory, most students submitted their responses, but there are instances of missing data for one or two students for certain assessments. The analysis below is based solely on the data collected.

It is important to note the limitation of the low sample size, which is inherently tied to the nature of a senior project course. These courses typically involve smaller class sizes to facilitate personalized mentorship and project supervision, making the sample size a reflection of the program's structure rather than a broader representation of MET students.

A) Pre- and Post-Course Quiz

The pre- and post-course quiz results reveal a significant improvement in students' understanding of fundamental AI/ML concepts. The average pre-course score was 26.7 out of 50, reflecting limited prior knowledge of the subject. After completing the lecture series, the average score increased to 47.9 out of 50, indicating a substantial gain in comprehension. This improvement is particularly notable given that the quiz primarily assessed foundational AI/ML concepts, which are not overly advanced but essential for practical application. The results, as illustrated in Figure 6, demonstrate the effectiveness of the lecture series in addressing knowledge gaps and equipping MET students with the necessary AI/ML competencies for their senior projects.

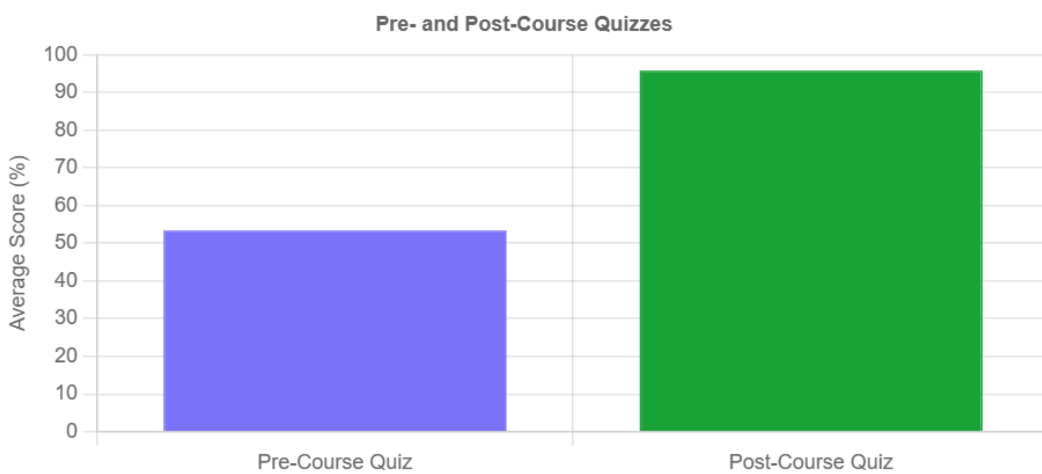

Figure 6. Pre- and post-course quiz results

B) Topic-Specific Quizzes

The topic-specific quizzes provide valuable insights into student learning outcomes across the eight lecture topics. The average scores for each quiz are presented in Table 1. Beginning with a 76.6% average for Quiz 1, which tested foundational machine learning concepts, the scores fluctuated slightly in the early topics. Quiz 3, focusing on data preparation and model evaluation, recorded a lower average of 61.1%, reflecting the complexity of this topic, as it delved into advanced concepts like feature selection, data splitting, and overfitting. Notably, these quizzes were intentionally designed to be more challenging, testing students' grasp of advanced AI/ML concepts with greater depth than the pre- and post-course quizzes.

A notable upward trend is evident in later quizzes, with scores rising to 83.3% for Quiz 5 and culminating at 90.4% for Quiz 8. This improvement is partly attributed to the parallel progress of the capstone projects, where students directly applied AI/ML concepts. The advanced topics of computer vision and natural language processing covered in the later lectures are key elements of the capstone projects, and students' hands-on experience likely reinforced their understanding, contributing to higher quiz performance.

Table 1 Average Scores for Topic-Specific Quizzes

Topic-Specific Quizzes	*#1*	*#2*	*#3*	*#4*	*#5*	*#6*	*#7*	*#8*
Average Score	76.6%	66.7%	61.1%	71.9%	83.3%	86.7%	87.3	90.4%

Additionally, the improvement suggests that students were well engaged with the lecture topics. Many appeared to put extra effort into studying the material beyond the classroom, motivated by both the lectures and the practical applications in their projects. The alignment between the lectures and the capstone projects encouraged deeper exploration and application of concepts, reinforcing the effectiveness of this integrated teaching approach.

C) Self-Evaluation and Reflection Survey

The student self-evaluation survey was divided into two parts to comprehensively assess the learning experience: Part 1 focused on evaluating the students' understanding of AI/ML concepts covered during the lectures, while Part 2 centered on their reflections and self-assessment of the capstone project experience.

Part 1 consisted of 16 questions designed to evaluate students' understanding and confidence in core AI/ML topics taught during the lectures. The questions ranged from foundational concepts like supervised learning, classification, and regression to advanced topics like neural networks, convolutional neural networks (CNNs), natural language processing (NLP), and reinforcement learning (RL). Students rated their confidence on a four-point scale: Excellent (4), Good (3), Average (2), and Poor (1). The average scores for each question are presented in Table 2.

From the data, it is evident that students expressed confidence in their understanding of most topics, with average scores consistently falling between 3.0 (Good) and 3.75 (Excellent-Good). Notably, students rated their understanding of practical tools and libraries (e.g., TensorFlow, OpenCV) highly, with an average score of 3.625, reflecting the hands-on approach of the lectures. Similarly, confidence in fundamental machine learning concepts such as supervised learning (3.625), CNNs (3.625), and data preprocessing (3.5) was strong, indicating the effectiveness of the foundational lectures in building core competencies.

However, slightly lower scores were observed in more abstract and challenging areas, such as differentiating model-based and model-free reinforcement learning (3.0) and applying Q-learning (3.125). These results suggest that while the hands-on examples provided some clarity, reinforcement learning remains inherently a complex and challenging topic. This aligns with the nature of MET students' prior preparation, which often lacks a strong mathematical foundation, further contributing to the difficulty of mastering advanced topics like RL.

Table 2 Self-Evaluation Scores for Machine Learning Topics

	Self-Evaluation Questions	Average Scores (Excellent = 4; Good =3, Average = 2, Poor = 1)
1	Understanding the three main types of ML (Supervised, Unsupervised, RL)	3.625
2	Identifying practical applications of ML in engineering	3.75
3	Differentiating between classification and regression tasks	3.25
4	Explaining the perceptron algorithm	3.125
5	Importance of data preprocessing in ML pipeline	3.5
6	Evaluating ML models (e.g., accuracy, precision, recall)	3.75
7	Understanding the structure and function of neural networks	3.25
8	Role of activation functions in deep learning	3.375
9	Understanding principles of CNNs	3.625
10	Applications of CNN in image recognition	3.5
11	Key concepts of NLP (e.g., tokenization, text generation)	3.25
12	Using tools like NLTK or OpenAI's API for NLP tasks	3.25
13	Difference between model-based and model-free RL	3.0
14	Applying Q-learning to grid-world navigation	3.125
15	Identifying tools and libraries for ML/RL tasks	3.625
16	Selecting the right type of ML or RL algorithm	3.5

The second part of the self-evaluation section provided insightful feedback on how students perceived their learning experience during the capstone projects. This part begins with three core questions that assess students' understanding of project-specific technical concepts, their perception of how effectively the projects demonstrated AI/ML applications in real-world scenarios, and their confidence in explaining their projects to others. The average scores for each question are presented in Table 3.

These results suggest that students felt confident in their understanding of technical concepts and how AI/ML can be applied to real-world engineering problems. Notably, the highest score for confidence in explaining their projects indicates that the capstone experience effectively prepared students to articulate and share their technical work with others, an essential skill for their future careers.

Table 3 Self-Evaluation Scores for the Capstone Projects

	Self-Evaluation Questions	Average Scores (Excellent = 4; Good =3, Average = 2, Poor = 1)
1	Understanding of technical concepts involved in the project	3.625
2	Effectiveness of the project in understanding real-world AI/ML applications	3.75
3	Confidence in explaining the project to others	4

This section also includes a "Challenge Faced" question that aimed to identify the most significant difficulties students encountered during their capstone projects. Students were provided with a set of options and could select all that applied, covering various aspects of the project process, including understanding project requirements, coding and implementation, using tools and libraries, debugging and troubleshooting, teamwork, managing time and deadlines, presentation and report writing, and other challenges. The results are shown in Figure 7.

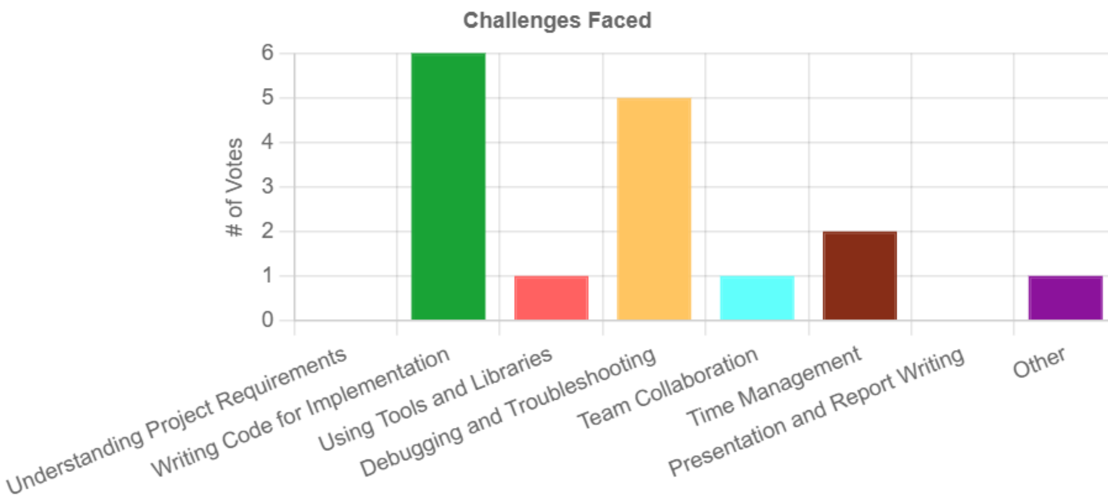

Figure 7. Results to the "Challenge Faced" question

The results reveal that coding challenges, encompassing both writing code for implementation and debugging, were the most significant hurdles faced by students, with 6 and 5 votes, respectively. These findings align with the fact that many MET students have limited prior programming experience, making the technical aspects of the projects particularly challenging.

To gain deeper insights into students' learning experiences and gather constructive feedback for course improvement, three reflection questions were designed. The first question is, "How has your senior project helped you apply the AI/ML concepts learned in class?" The responses, as shown in Table 4, demonstrate that the senior projects effectively bridged theoretical lectures with practical applications. Note that only selected student responses are listed, as not all students answered this question, and a few incomplete or off-topic responses were excluded. Students generally appreciated how the projects reinforced their understanding of AI/ML concepts through hands-on experience. Their reflections commonly emphasized that the senior projects helped bridge theory and application, deepened their curiosity, and motivated self-directed exploration. Several students noted that working on real-world problems enhanced their comprehension of machine learning principles and gave them a clearer view of AI's role in engineering practice. These insights demonstrate how the capstone projects successfully supported both conceptual learning and practical skill development.

Table 4 Selected Student Responses to Reflection Question #1

Reflection Question 1: How has your senior project helped you apply the AI/ML concepts learned in class?	
Student #1	*"Yes, Senior project was the practical part of the concepts studied at AI/ML. Both topics were well aligned and helped to understand the big picture of AI/ML"*
Student #2	*"The project introduced us to AI/ML through using software such as Python and how AI is programmed"*
Student #3	*"It made we know machine learning through building a project that make me curious how to move forward in the project so I research completely to help me figure it out"*
Student #4	*"The project has provided with knowledge about various AI/ML concepts like supervised learning and data processing techniques"*
Student #5	*"We understand the concept of supervised learning, coding using Python and how AI is programmed"*

The second question asked students how the skills and knowledge gained from the course would benefit their future careers. As reflected in Table 5, students generally expressed that the course helped them build relevant AI/ML skills aligned with current engineering industry demands. Several noted that gaining exposure to machine learning, programming, and automation tools gave them a competitive edge and increased confidence as they enter the workforce. Others emphasized that this course served as their first meaningful experience with AI, providing a valuable foundation for future study or professional development. Collectively, these responses affirm the course's role in preparing MET students for emerging technologies in their career paths.

Table 5 Selected Student Responses to Reflection Question #2

Reflection Question 2: How do you think the skills and knowledge gained from this course will benefit your future career?	
Student #1	*"As I look for a career in the engineering field, and the way AI is being implemented in most corporations, the knowledge I have will give me a headstart to succeed"*
Student #2	*"In future, I am planning to get a BS degree on Artificial Intelligence, with this project, I learned the basic startup about coding"*
Student #3	*"100% agreed! I understand how the concepts and applied learning about robotics, machine learning, AI, computer vision, python from the big picture and how they interact and how to put them together to solve engineering problems"*
Student #4	*"I had no background in any type of coding before, and now it gives me experience in the future"*
Student #5	*"Now that I am familiar with coding, ML, and AI. This is something I can say is a skill I have gained"*

The third question asked for suggestions to improve the course. As summarized in Table 6, students provided several actionable recommendations. A common subject was the suggestion to expand the course credit hours, reflecting the time commitment required for both lectures and project work. Some students advocated for developing a dedicated AI/ML course within the MET curriculum, while others recommended offering preparatory training, particularly in Python programming and the use of robotic systems like Universal Robots. These suggestions align well with the broader goals of enhancing support resources, managing workload expectations, and expanding access to AI/ML learning opportunities for MET students.

Table 6 Selected Student Responses to Reflection Question #3

Question 3: Do you have any suggestions to improve this course?	
Student #1	*"(1) 3 credits for senior project assigned. However, lecture class and lab took more time. It should be at least 4 credits for this course. (2) Machine learning/AI applied to mechanical engineer career should be a course by itself"*
Student #2	*"The only suggestion I have is maybe for the college to have a class for coding and python"*
Student #3	*"Student should have previous knowledge of using Universal Robot with python script"*
Student #4	*"Not really, I really enjoyed the class"*

D) Senior Project Evaluation

The evaluation of senior projects is based on three components: project design and implementation (50%), presentation (20%), and report writing (30%). These criteria assess both group and individual contributions. Project design and implementation, evaluated at the group level, focus on problem definition, technical content, complexity, resource utilization, time management, and alignment with stakeholder needs. Presentations are assessed for organization, clarity, professionalism, visual aids, and the ability to address questions from faculty and peers. Report writing, evaluated individually, considers clarity, logical flow, professional language, completeness of data and analysis, quality of conclusions, use of visuals, and proper citations. A 5% bonus for quiz submission was excluded from this assessment.

The senior project evaluation results reflect the success of integrating AI/ML concepts into capstone projects, with scores ranging from 75% to 92.5% and a class average of 84.72%. These strong performances indicate that the projects and supporting lectures effectively equipped students with the necessary technical and practical skills. The consistently high results validate the approach and confirm the feasibility of incorporating AI/ML into the MET curriculum through senior projects.

Discussion and Conclusion

This study highlights the importance of equipping MET students with foundational AI/ML skills to meet the evolving demands of the engineering industry. By integrating AI/ML into senior projects, students were able to bridge classroom concepts with practical applications, aligning perfectly with the applied learning focus of MET programs. Rather than relying on traditional computer science materials, this tailored approach emphasizes real-world problem-solving, making AI/ML both accessible and relevant to MET students. The assessment data confirms the success of this strategy, demonstrating its effectiveness in preparing students to confidently address modern engineering challenges. While the implementation of this design has proven effective, it has also highlighted significant challenges for both students and educators. Addressing these challenges is essential to ensuring the continued success of integrating AI/ML learning into the MET curriculum.

For students, the biggest challenges come from their limited math and coding backgrounds, which are common in traditional MET programs. To help them overcome these hurdles, the course adopts a practical, hands-on approach that ties AI/ML learning directly to solving engineering problems. This makes the concepts more relatable and easier to understand. Instead of focusing on complex mathematical details, the curriculum emphasizes high-level AI/ML principles and simpler models that are sufficient for most engineering applications without overwhelming the students. Tools like Google Teachable Machine and Roboflow have been game-changers, allowing students to train models without needing extensive coding skills. Similarly, tools like ChatGPT are introduced to assist with coding and debugging, making these tasks more approachable and helping students produce clean, understandable Python code. Collaborative group work also plays a key role, encouraging teamwork and shared problem-solving to address individual challenges. Perhaps most importantly, the students' genuine motivation to learn AI/ML has been a driving force, helping them push through difficulties and fully engage with the material.

For educators, integrating AI/ML into MET programs comes with its own unique challenges. A key difficulty is finding room in an already packed curriculum to introduce AI/ML concepts. Embedding AI/ML modules into senior-level courses, such as robotics or senior projects, has emerged as a practical short-term solution. Another challenge is striking the right balance between theory and application. It's important to give students enough foundational knowledge to grasp AI/ML principles, but diving too deeply into theory can quickly overwhelm MET students. Achieving this balance is an evolving process, requiring continual practice and refinement as AI/ML education in MET is still in its early stages. Faculty expertise and resource limitations add to the complexity, as many MET educators may not have specialized training in AI/ML or programming, and IT resources can be limited. Fortunately, modern AI/ML tools are increasingly accessible and user-friendly, and there is a wealth of online resources available to support both teaching and learning. Finally, the rapid pace of advancements in AI/ML means educators need to continually invest time in staying updated with the latest tools, models, and applications to keep their teaching relevant and impactful.

In conclusion, this study demonstrates the effectiveness of integrating AI/ML learning into MET education through senior projects. By bridging theoretical concepts with hands-on applications, this approach makes AI/ML more accessible and relevant to engineering students. It not only helps students build essential AI/ML skills but also aligns seamlessly with the practical, applied focus of MET programs. The assessment results underscore the success of this strategy, showing that students can effectively grasp and apply AI/ML concepts to solve real-world problems. Future efforts could expand AI/ML integration into other senior-level MET courses, such as robotics. The lecture content developed can also serve as optional modules to enrich other senior-level courses, such as HVAC, fluid systems, or product design, allowing students and instructors to incorporate AI elements without necessitating a full transformation of the course's original focus. There is also a strong case for developing dedicated applied AI/ML courses tailored specifically for engineering students. Additionally, we plan to explore opportunities to introduce foundational AI/ML concepts into lower-level engineering courses, like introductory programming and systems modeling, to provide students with early exposure to these technologies.

References

1 Naixin, Z., Wai Yie, L., Tong, Z., & Changqing, W., "Artificial Intelligence in Engineering Education: A Review of Pedagogical Innovations." INTI JOURNAL 2024, no. 46 (2024): 1-10.
2 Ellingrud, K., Sanghvi, S., Madgavkar, A., Dandona, G. S., Chui, M., White, O., & Hasebe, P., "Generative AI and the future of work in America." (2023). Retrieved from https://www.mckinsey.com/mgi/our-research/generative-ai-and-the-future-of-work-in-america

3 Boggs, A. S., Kerrianne Buchanan, Heather Evans, David Griffith, Dimitrios Meritis, Lisa Ng, and M. Stephens. "National institute of standards and technology environmental scan." Societa l and technology landscape to inform science and technology research (2023): 1-79.

4 Meers, D. and Illingworth, D., "Manufacturers embrace AI: Study reveals familiarity and investment in automation," (2025). Retrieved from https://www.cohnreznick.com/insights/manufacturing-checkup-artificial-intelligence

5 Mills, J. E., & Treagust, D. F., "Engineering education—Is problem-based or project-based learning the answer." Australasian journal of engineering education 3, no. 2 (2003): 2-16.

6 OpenCV (https://opencv.org/)

7 TensorFlow (https://www.tensorflow.org/)

8 ChatGPT (https://chatgpt.com/)

9 Universal Robots UR3e (https://intelitek.com/scorbot-er-4u-educational-robot/)

10 Logitech C920 (https://www.logitech.com/en-eu/shop/p/c920-pro-hd-webcam.960-001055)

11 Google Teachable Machine (https://teachablemachine.withgoogle.com/)

12 *ur_rtde* - Universal Robots RTDE Interface (https://sdurobotics.gitlab.io/ur_rtde/)

13 Roboflow (https://roboflow.com/)

14 Google Cloud Speech API (https://cloud.google.com/speech-to-text/)

15 OpenAI API (https://platform.openai.com/)

16 Olds, B. M., Moskal, B. M., & Miller, R. L., "Assessment in engineering education: Evolution, approaches and future collaborations." Journal of Engineering Education 94, no. 1 (2005): 13-25.