

Applied Machine Learning in Hardware Software Lab Course for Interdisciplinary Graduate Students

Prof. John Raiti, University of Washington

Prof. John Raiti is an Associate Teaching Professor in Electrical & Computer Engineering at the University of Washington, and is the Technical Programs Advisor at the Global Innovation Exchange (GIX) where he teaches in the Interdisciplinary UW Master of Science in Technology Innovation degree program. He teaches UW graduate level courses in Sensors & Circuits, IoT and Connected Devices, Capstone-style Launch Projects, and Robotics (Mobility, Navigation, and Manipulation) with a focus on Human Robot Interaction (HRI).

Applied Machine Learning in Hardware Software Lab Course for Interdisciplinary Graduate Students

Luyao Niu, Kevin Arne, Linda E. Wagner and John G. Raiti
University of Washington

1 Abstract

This paper presents a redesigned graduate-level multidisciplinary engineering course that combines embedded machine learning (ML), edge–cloud architectures, and human-centered design. The course was implemented in the University of Washington’s Master of Science in Technology Innovation (MSTI) program and enrolled 40 graduate students from diverse academic backgrounds. Across five structured labs and a quarter-long team project, students progressed from ML model optimization and structured data classification on ESP32 microcontrollers to time-series gesture recognition using IMU sensors and hybrid edge–cloud inference using Microsoft Azure. A narrative-based thematic structure was used to unify technical concepts and promote engagement, while Generative AI tools were incorporated as reflective learning partners to support debugging, prototyping, and design exploration.

Evaluation included student perception data and direct learning-outcome measures. Students reported high engagement and intellectual challenge. Rubric-based assessments and project artifacts demonstrated successful deployment of embedded ML models, implementation of quantization and offloading strategies, and functional integration of hardware, software, and cloud services. The redesigned course demonstrates how experiential learning, narrative framing, and AI-augmented pedagogy can coexist with technical rigor. The results suggest that such an approach can support inclusion, systems-level reasoning, and practical ML deployment skills across multidisciplinary graduate populations.

2 Introduction and Background

The rapid integration of embedded systems, artificial intelligence (AI), and human–computer interaction (HCI) is reshaping how engineers design and interact with intelligent products. As engineering practice increasingly spans the physical and digital domains, there is a growing need for educational models that help students navigate this convergence through multidisciplinary learning experiences. Such experiences should not only cultivate technical proficiency but also nurture creativity, systems-level reasoning, and ethical awareness.

2.1 Existing Multidisciplinary Approaches

A substantial body of work has explored project-based and experiential approaches to teaching embedded systems, mechatronics, and human–robot interaction. Multidisciplinary project courses have been shown to foster teamwork and systems integration skills^{1,2,3,4,5,6}, while design-oriented programs have introduced creative prototyping and user-centered frameworks to improve engagement and retention⁷. Similarly, recent efforts in AI and data-driven engineering education have begun integrating machine learning into undergraduate and early graduate curricula^{8,9,10}.

However, existing implementations often fall into two extremes: (1) technically focused courses that emphasize programming and electronics but underemphasize user experience and design, or (2) design-driven courses that prioritize ideation and aesthetics without rigorous treatment of computation or data. Moreover, most courses addressing AI in engineering focus on high-level modeling or analytics rather than the deployment of AI on constrained embedded devices. The literature also reveals limited attention to Generative AI–assisted learning—particularly its potential to scaffold learning across students with diverse majors and technical backgrounds.

Within this landscape, the Hardware/Software Lab II (TECHIN 515) course offered at the University of Washington’s Master of Science in Technology Innovation program was identified as an opportunity to implement these concepts. Originally a technically focused course centered on embedded programming and system integration, TECHIN 515 has been revamped to address the growing need for multidisciplinary, AI-augmented learning. The redesigned course combines machine learning, embedded hardware design, and human-centered interaction within a thematic and narrative-based learning framework^{11,12}, encouraging students from engineering, design, and computing disciplines to collaborate on creative, technically rigorous projects. The course also intentionally integrates Generative AI tools to support reflection, code exploration, and design ideation—helping students critically engage with AI as both a technical resource.

2.2 Pedagogical Framework

The course design follows a directed experiential learning model, grounded in Kolb’s experiential cycle¹³ and project-based learning research¹⁴. Students move iteratively between experimentation, reflection, conceptualization, and application. The five individual labs progressively develop technical competencies in: communication networks, embedded ML, and cloud-edge computing, while cultivating design judgment and reflective practices.

Each lab emphasizes not only “how” to build a system, but “why” certain design trade-offs matter. Students assess hardware constraints, ML performance, and human-computer interaction (HCI) considerations, developing systems-level reasoning central to multidisciplinary engineering. Reflection prompts and discussion questions explicitly ask students to justify decisions from technical, user-experience, and ethical perspectives.

2.3 Thematic and Motivational Design

The course used a Harry Potter–inspired narrative as a thematic unifier to reveal real-world engineering principles. This narrative context supported engagement among diverse learners by placing abstract technical content within a familiar and imaginative frame. For example, students constructed a “Sorting Hat” that implemented decision-tree classification on the ESP32, and a “Magic Wand” that performed gesture recognition using an IMU and an Edge Impulse ML model. These projects allowed students to experience how hardware, software, and user experience design intersect in tangible and engaging ways.

In addition to the 5 individual labs, the course also incorporated a 10 week team project where interdisciplinary groups of four students propose, prototype, and test a novel intelligent device, applying both technical and human-centered principles learned throughout the labs. This progression addressed gaps in pre-existing curricula by connecting systems engineering, AI deployment, and design thinking, helping to advance the understanding of how to teach engineering concepts to both engineering and non-engineering students^{15,16}.

2.4 Contributions and Paper Organization

The contributions of this paper are threefold:

- a revised curricular model that blends technical rigor, creative engagement, and inclusivity within a multidisciplinary engineering program.
- a model for how Generative AI can be used as a learning and reflection partner in embedded systems education.
- a collection of insights into how narrative and thematic framing can enhance motivation and conceptual understanding across diverse student populations.

The remainder of this paper is organized as follows: Section 3 describes the course and its structure. Section 4 discusses implications. Section 5 presents broader discussion, and Section 6 concludes the paper.

3 Course Description and Structure

TECHIN 515 (Hardware Software Lab II) is a graduate-level course at the University of Washington that builds on TECHIN 513 (Managing Data and Signal Processing). Whereas TECHIN 513 focuses on foundations of data collection, signal processing, and machine learning, TECHIN 515 emphasizes end-to-end embedded product development. The course combines ESP32 microcontroller programming, machine learning deployment on constrained devices, human–computer interaction, and physical prototyping (PCB layout, CAD modeling, and enclosure fabrication). The course assumes prior exposure to Python programming, signal processing concepts through TECHIN 513, and prior experience with ESP32 and CAD modeling.

The data presented in this paper reflect the Spring Quarter 2025 offering of TECHIN 515, which represented the first full implementation of the redesigned, AI-integrated curriculum. Prior

iterations of the course focused primarily on embedded systems without structured machine learning or narrative theming.

The course design included two main components:

- Five individual labs (Labs 1–5) that incrementally introduced technical and design complexity, with Labs 3–5 intentionally framed around Harry Potter–themed projects to blend creative storytelling with rigorous technical learning.
- A quarter-long team-based project, where students formed interdisciplinary groups to apply their skills toward open-ended, self-defined product concepts.

To support students from diverse educational backgrounds, the course provided starter code, structured laboratory instructions, and optional enrichment resources such as tutorials and reference materials, plus, weekly lectures from engineering faculty. Students were encouraged to use AI-assisted tools as learning “tutors”. The instructional team met with students on a weekly basis to provide formative feedback, and students experiencing technical challenges had access to on-site technical staff for hardware and system-level support.

Learning outcomes were evaluated through student artifacts, live demonstrations, structured rubrics, and anonymous course evaluations. A longitudinal assessment is ongoing with the same cohort of students in their capstone project courses.

3.1 Instructor Background and Teaching Context

The course was led by two faculty members affiliated with the Department of Electrical and Computer Engineering, with a background in embedded systems, signal processing, robotics, and applied machine learning. The instructors have 7 years and 1 year of teaching experience in interdisciplinary engineering programs, respectively. The course is additionally supported by program director with background in Human Centered Design & Engineering, and technical staff with background in Civil Engineering.

3.2 Demographics of Students

The course enrolled 40 graduate students from the MS in Technology Innovation program, with a range of educational backgrounds such as:

- Engineering (Computer Science and Information Management, Computer Engineering, Digital Media Technology, Mechanical Engineering, Materials Science & Engineering, Network Engineering, Industrial Engineering) – 20.5%
- Business (Information Management, Information Systems and Operations Supply Chain Management, Marketing, International Tourism Management) – 10.3%
- Social Sciences & Humanities (Politics, Philosophy, Economics, History and Art History) – 10.3%
- Architecture (Architecture, Landscape Gardening, Landscape Architecture) - 7.7%

- Design (Industrial Design, Illustration, Communication Design, Interactive Media Arts, Product Design, Arts and Design, Art and Technology Integrated Design & Media, Visual Communication, Interior Design, Digital Media Art, Environment Design) - 51.3%

Among the enrolled students, 31 self identified as female and 9 self identified male. The range of student backgrounds created a unique interdisciplinary learning environment.

3.3 Team Formation and Size

While Labs 1–5 were completed individually, the 10 week (quarter-long) final project was conducted in teams. Students *self-formed into teams of four*, typically balancing technical and design expertise. This team size was chosen to ensure:

- Diversity of skill sets (programming, ML, hardware prototyping, design/UX).
- Manageable scope, allowing every member to contribute meaningfully.
- Authentic collaboration, reflecting interdisciplinary product development teams in industry.

Teams were responsible for proposing a project concept, developing a functioning prototype, and presenting their work at the end of the quarter. Each team met with faculty advisors weekly to report progress and discuss challenges to making progress.

3.4 Individual Lab Structure

The five individual labs were designed as exercises that gradually increased in technical and design complexity. Each lab required students to produce a working prototype, submit a report, and reflect on technical tradeoffs. Labs 3–5 introduced a Harry Potter–themed narrative to foster creativity and engagement while reinforcing rigorous technical skills.

3.4.1 Lab 1: LoRa and Networking

Students explored long-range wireless communication (LoRa) using ESP32 devices equipped with LoRa transceivers. They implemented a basic networking protocol to exchange messages between devices, experimenting with parameters that affect range and reliability.

Learning objectives: By completing this lab, students will:

- Implement long-range wireless communication using LoRa on an ESP32 platform and verify end-to-end data transmission.
- Analyze trade-offs between communication range, data rate, and power consumption in low-bandwidth IoT networks.

3.4.2 Lab 2: ML Model Pruning and Quantization

Students implemented pruning and quantization techniques on pretrained ML models to optimize them for resource-constrained devices. They compared the effects of pruning and 8-bit quantization on accuracy, model size, and inference speed.

Learning objectives:

- Apply model pruning and quantization techniques to reduce memory and compute demands for embedded ML deployment.
- Compare performance metrics (e.g., inference latency, accuracy loss, and model size) to understand the practical impact of compression strategies.
- Implement and benchmark optimized models on constrained hardware platforms such as ESP32.
- Justify design decisions by analyzing the trade-off between technical performance, system cost, and user experience.
- Discuss broader implications of lightweight AI—how energy efficiency, edge deployment, and accessibility reflect cross-disciplinary design and ethical priorities.

3.4.3 Lab 3: Magical Sorting Hat

Lab 3 introduces students to applied machine learning on embedded hardware using *structured categorical data*. Drawing inspiration from the Harry Potter universe, students design a “magical sorting hat” that classifies users into different houses based on survey responses. A demonstration sorting hat provided by the instructional team is shown in Figure 1.

Students first collected their own dataset of structured responses, ensuring balanced representation across all houses. They preprocessed the data, trained a decision tree classifier in Python, and then converted and deployed the model onto the ESP32 microcontroller. Input is collected through physical buttons, while feedback is presented via an OLED display, creating a playful yet interactive user experience. This lab emphasized the fundamentals of structured data classification—clean feature engineering, balanced dataset collection, and model deployment in a constrained environment. It also highlighted how simple models like decision trees can be effective on embedded platforms.

To encourage critical, multidisciplinary thinking, students are asked reflective questions:

- *Tech-UX tradeoff*: Are all ten survey questions necessary? If not, which could be removed to streamline user experience?
- *Model-Sensor Design Choice*: If you were to improve the sorting hat, what additional sensors (e.g., voice input, biometric signals) might you add, and what machine learning models would then be more appropriate than decision trees?

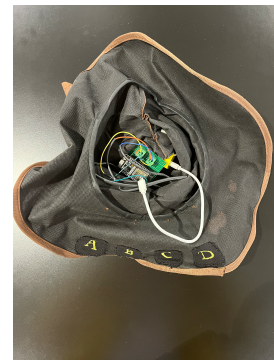


Figure 1: A demonstration sorting hat is used in class to increase engagement. A battery-powered ESP32 microcontroller makes sorting decisions by assigning students to houses based on student inputs (A, B, C, or D, as shown in the figure), which are sensed using conductive threads.

By working with structured tabular data in a tangible, creative project, students build confidence in end-to-end machine learning pipelines. This foundation prepares them for later labs, which expand into time-series sensor data and more complex models for gesture recognition and edge-cloud computing.

Learning objectives: By completing this lab, students will:

- Gain practical experience with structured data collection and preprocessing, including ensuring balanced datasets. Implement and deploy a decision tree classifier on a resource-constrained microcontroller (ESP32).
- Explore the tradeoff between user experience and technical accuracy by analyzing the necessity of all input questions.
- Critically reflect on model-sensor design choices, including potential improvements with additional sensors or alternative ML models.

3.4.4 Lab 4: Magic Wand

In Lab 4, students extended their learning from structured data classification to *time-series sensor data*. Using the ESP32 microcontroller paired with an MPU6050 inertial measurement unit (IMU), students built a “magic wand” capable of recognizing three spell gestures: Fire Bolt, Reflect Shield, and Healing Light. The project combined hardware prototyping, data collection, signal processing, and embedded machine learning in a playful Harry Potter-inspired theme. A demonstration magic wand is shown in Figure 2 (left).

Students began by designing a gesture capture pipeline, collecting multiple one-second IMU samples for each spell gesture. To emphasize generalization and avoid overfitting, they were required to collect and share datasets across multiple students. They then imported their data into Edge Impulse, where they trained gesture recognition models using built-in DSP blocks and experimented with hyperparameter tuning and neural network architectures. After achieving acceptable classification accuracy, the optimized quantized (Int8) model was deployed back to the ESP32.

On the hardware side, students implemented a button-triggered inference system and light-based feedback to indicate recognized gestures. Each student is provided with a starter perfboard, shown in Figure 2 (right), containing a soldered Seeed Studio XIAO ESP32, which serves as the basis for implementation and wand prototyping. This reinforced the importance of real-time processing and low-latency interaction for embedded HCI systems.

To encourage critical thinking, students were asked to reflect on design decisions such as:

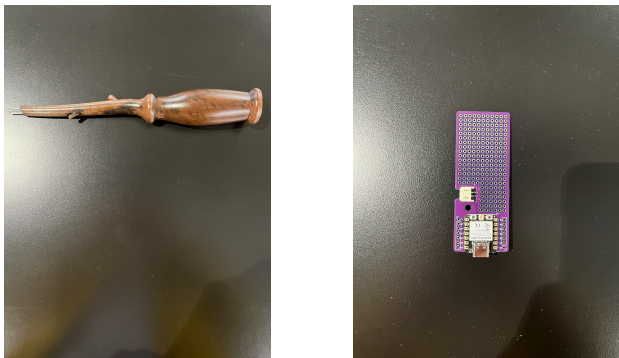


Figure 2: The left figure shows a demonstration magic wand used in class for engagement. The wand uses ESP32 to recognize spell gestures. A light on the tip indicates the recognition result. The right figure presents the perfboard provided to each student to prototype magic wand.

- *Data Diversity and Reliability*: Why is it important to collect gesture data from multiple participants rather than only from oneself?
- *Signal Processing vs. User Experience*: How do choices in DSP parameters and window size affect both model accuracy and user experience?
- *Robustness*: What strategies could further enhance performance if the current model struggles with consistency?

Lab 4 transitioned students from structured categorical data (Lab 3) to continuous sensor streams, highlighting challenges of noise, variability, and real-time feedback. It illustrated how interdisciplinary choices in signal processing, ML design, and hardware interaction directly shape system usability. Students gained experience building an end-to-end interactive system where technical tradeoffs (e.g., model complexity vs. latency) must be balanced with user-centered design considerations.

Learning Objectives: By completing this lab, students will:

- Understand the fundamentals of gesture recognition using IMU sensors. Collect, preprocess, and label time-series accelerometer data across multiple users.
- Apply digital signal processing (DSP) and lightweight neural networks using the Edge Impulse platform.
- Develop practical skills in model optimization and deployment on resource-constrained devices. Explore the role of hardware integration and user interaction in achieving reliable ML performance.

3.4.5 Lab 5: Edge-Cloud Hybrid Inference

In Lab 5, students extended their magic wand system from embedded gesture recognition to a *hybrid edge–cloud architecture*, exploring how computation can be dynamically distributed between the device and the cloud. Building on the model developed in Lab 4, they implemented a confidence-based offloading strategy that performs inference locally on the ESP32 when confidence is high, and transmits sensor data to the cloud (Microsoft Azure) for inference when uncertainty arises. This project integrated concepts from embedded systems, networking, and cloud computing, emphasizing the interplay between hardware constraints, connectivity, and system design.

Students began by training and hosting an upgraded gesture recognition model on Microsoft Azure, using cloud compute resources to handle larger, aggregated datasets. They deployed the trained model as a web service endpoint through Azure App Service, allowing their ESP32 devices to send requests and receive gesture predictions via HTTP communication. On the device side, students implemented JSON-based data serialization and an adaptive offloading mechanism that compared the model's confidence scores to a predefined threshold before deciding whether to rely on local inference or query the cloud.

Through this implementation, students encountered design tradeoffs among latency, bandwidth, privacy, and user experience. They also observed how cloud-based models could improve

accuracy but at the cost of real-time responsiveness and dependence on network stability, encouraging reflection on both the technical and human-centered implications of system architecture.

To encourage critical thinking, students were asked to reflect on design decisions such as:

- *Computation Placement and System Design*: When should inference be performed locally versus in the cloud? How do factors like latency, bandwidth, and reliability influence this decision?
- *Ethics and Privacy in Distributed Systems*: How do privacy, data security, and user trust affect design choices in hybrid intelligent systems?
- *Connectivity as a Design Constraint*: How might the system be redesigned to remain functional under intermittent or offline conditions?

Lab 5 guided students to think beyond embedded ML toward system-level integration and lifecycle design. By combining embedded sensing, cloud infrastructure, and communication protocols, students gained a holistic understanding of how distributed intelligence is implemented and experienced. This lab illustrated the interdisciplinary nature of modern engineering practice, where decisions in hardware, software, and networking must align with ethical, experiential, and performance considerations.

Learning Objectives: By completing this lab, students will:

- Understand principles of edge–cloud collaboration and computational offloading.
- Learn to train, host, and deploy ML models on Microsoft Azure, integrating web APIs with embedded devices.
- Implement decision logic for adaptive offloading based on confidence thresholds.
- Analyze latency, consistency, and privacy tradeoffs between local and cloud computation.
- Strengthen skills in network communication and system integration across hardware and software domains.

3.4.6 Magic Wand Duel Competition

Following the completion of Lab 5, students participated in an in-class *Wand Duel Competition* designed to validate their embedded and cloud-connected wand systems in an interactive, gamified setting. Although not a graded component, the activity served as an *experiential exercise* that reinforced the technical and interdisciplinary principles explored throughout the course.

During the competition, students paired their ESP32-based wands—each capable of recognizing three spell gestures (Fire Bolt, Reflect Shield, and Healing Light)—and connected to a central server that handled duel logic and state management. Each wand performed real-time gesture recognition using the model developed in Lab 5 and transmitted the detected spell and confidence score to the server through HTTP requests. The server determined outcomes based on predefined rules, applying house-specific “talents” such as enhanced damage or reduced mana cost to introduce variability and strategic depth.

This activity required students to integrate and troubleshoot multiple system layers under live conditions, including wireless communication, server connectivity, and embedded inference timing. It highlighted the importance of robustness and reliability in networked interactive systems, as real-time performance depended on accurate sensor input, stable Wi-Fi connections, and consistent latency between devices and the server.

To promote reflection and connection to broader learning goals, students discussed questions such as:

- *System Integration under Real Conditions*: What technical challenges emerged when multiple devices interacted through a shared network and server?
- *Designing for Engagement*: How did narrative, competition, and feedback elements enhance user experience and learning motivation?
- *Data Transmission Strategies*: What are the pros and cons of sending raw data versus predicted gestures to the server in terms of performance, privacy, and bandwidth?

The Wand Duel Competition provided an engaging culmination to the Harry Potter-themed lab series, allowing students to see their individual work function collectively within a shared interactive system. It emphasized not only technical proficiency but also creativity, teamwork, and experiential understanding—core attributes of multidisciplinary engineering education.

3.4.7 Synthesis and Learning Progression

Together, the themed labs and Wand Duel Competition created a coherent learning sequence that guided students from foundational embedded machine learning concepts toward system-level design and integration. Beginning with structured data classification in the Sorting Hat lab, students learned the basics of supervised learning and model deployment on microcontrollers. The Magic Wand lab then introduced time-series sensing, signal processing, and real-time feedback, expanding their understanding of data modality and human–computer interaction. The Edge–Cloud Offloading lab further extended this knowledge to distributed systems, prompting students to consider latency, privacy, and reliability in intelligent connected devices. Finally, the ungraded Wand Duel Competition brought these skills together in a live interactive environment, demonstrating the challenges of synchronization, communication, and multi-device collaboration. Across these experiences, students developed not only technical proficiency but also interdisciplinary reasoning—linking hardware design, software implementation, user experience, and ethical considerations within the broader context of emerging intelligent systems.

3.5 Team-Based Quarter-Long Projects

In parallel with the individual lab sequence, students worked in self-formed interdisciplinary teams of four on a quarter-long final project that integrated hardware prototyping, embedded programming, and applied machine learning addressing a real or perceived user need. The project served as a capstone-style experience, enabling students to extend the technical and conceptual foundations established in earlier labs—data acquisition, signal processing, model training, and human–computer interaction—into an open-ended, student-defined design challenge. Teams were

intentionally encouraged to blend diverse disciplinary expertise, drawing participants from electrical and computer engineering, human-centered design, and computer science backgrounds. This structure fostered authentic multidisciplinary collaboration, mirroring real-world engineering contexts where technical, aesthetic, and user-oriented considerations must converge. Teams were encouraged to balance innovation with feasibility, considering practical constraints such as budget, complexity, and timeline while making design decisions that integrated hardware performance, software functionality, and user experience.

Throughout the quarter, teams advanced through key phases of problem identification, scoping, prototype iteration, and final system integration. Regular milestone presentations required students to communicate progress, justify design choices, and provide peer feedback, reinforcing habits of professional engineering practice. Deliverables included a formal project proposal, midterm progress report, GitHub documentation, and a final public demonstration accompanied by a written technical report.

Evaluation criteria emphasized both technical and professional competencies:

- *System Design and Functionality*: Successful integration of hardware, software, and sensing elements.
- *Design Decision-Making*: Evidence of trade-off analysis among performance, usability, and resource constraints.
- *Project Management*: Effective time and task management, equitable contribution, and milestone tracking.
- *Communication and Reflection*: Clarity in written, oral, and visual documentation, and thoughtful articulation of lessons learned.

Projects reflected a broad range of student interests, including wearable health monitors and gesture-controlled smart devices. Many teams incorporated embedded machine learning, extending principles learned from earlier labs into larger system architectures. These projects not only demonstrated technical proficiency but also cultivated systems-level thinking, collaborative problem-solving, and entrepreneurial awareness, skills central to modern interdisciplinary engineering practice. Some example projects are summarized below:

- **SleepPose**: A smart pillow with pressure sensing array to estimate head orientation and sleep position. The pillow reshapes in real time for personalized support during sleep.
- **CareLink Pill**: A connected pill box to track medication adherence and support hypertension care. The pill consumption is verified by machine learning on the edge and synchronized with doctors in remote.
- **X Heal**: A wearable system to track rehabilitation exercises with clinical accuracy at home for ACL injuries. The system is connected to therapists portal to remotely monitor progress and adjust rehabilitation plans.

4 Implications

4.1 Feedback from Course Evaluations

The course evaluation results demonstrated that the redesigned TECHIN 515: Hardware/Software Lab II successfully achieved its interdisciplinary and experiential learning objectives. Students rated the course highly across summative dimensions, with a median overall rating of 4.7/5. The Challenge and Engagement Index (CEI) of 6.1/7 further suggests that students found the course both intellectually demanding and motivating.

4.1.1 Integration and Engagement

Formative items show particularly high ratings for course structure and engagement, with medians of 6.8, reflecting consistent opportunities for creative exploration and applied learning. Students explicitly highlighted the hands-on nature of the labs and the integration of hardware, software, and machine learning as the most impactful learning experiences. Many noted that the themed projects, particularly the Magic Wand and Sorting Hat, made technical concepts more approachable and memorable, enhancing both motivation and self-efficacy. Comments such as *“the labs give me a chance to cultivate my practical capabilities”* and *“the magic wand one is amazing”* underscored how themed, structured projects successfully bridged technical learning with creative engagement. The Harry Potter-themed labs effectively fostered curiosity, playfulness, and collaboration, leading students to perceive the course as uniquely stimulating and memorable.

4.1.2 Interdisciplinary Learning Outcomes

Students reported strong development in creativity (median 6.8), self-critique (6.8), and independent thinking (6.8), demonstrating growth in both technical and reflective domains. The interdisciplinary design—linking signal processing, embedded ML, networking, and user experience—helped students *“expand their knowledge in hardware and software”* and *“apply skills across different domains”* according to course evaluations. These results show that combining technical rigor with narrative-driven design encourages systems-level thinking, mirroring professional engineering contexts where innovation emerges from cross-disciplinary synthesis.

Direct measures of learning outcomes were assessed using structured rubrics for lab submissions and final projects. Across Labs 2–5, 34 out of 40 students successfully deployed functional ML models on embedded hardware and successfully demonstrated correct functionalities.

Performance assessment of final team projects showed that 90% of teams successfully integrated hardware, software, and enclosure and shipped the project for demonstrations.

4.1.3 Areas for Refinement

Despite the overall success, student feedback identified areas for continued improvement. Several students noted that lab instructions could be clearer or more robustly tested, particularly around debugging and hardware setup. A few also expressed a desire for earlier project ideation and

additional guidance for conceptualization. These comments suggest that while students valued open-ended exploration, additional structure, such as peer feedback or pre-quarter project ideation workshops, could further support diverse learning styles and reduce frustration.

4.2 Broader Educational Implications

4.2.1 Curricular Design Implications

The course sequence illustrates how thematic lab design can foster deep integration of concepts across hardware, software, and machine learning. By structuring the labs progressively from structured data classification to gesture-based sensing and finally to networked edge–cloud systems, students experience a gradual increase in complexity and autonomy. This approach aligns with the goals of systems-level engineering education, emphasizing not just isolated technical skills but also the interdependencies among sensing, computation, communication, and user experience. The course thus models a transferable template for embedding interdisciplinary, project-based learning in advanced engineering curricula, particularly in emerging areas such as intelligent devices, IoT, and human–AI interaction.

4.2.2 AI-Augmented Learning

An outcome of the course was students’ use of LLM-based tools such as ChatGPT to support debugging, documentation, and design exploration. This observation of AI usage highlighted both the potential and the pedagogical need to integrate AI literacy into engineering education. Students learned to critically evaluate AI-generated suggestions, refining them through empirical testing and domain reasoning. Such experiences encouraged a reflective stance toward AI tools: not as substitutes for understanding but as partners in problem-solving. Future iterations of the course will include explicit discussions on ethical use, prompt design, and validation of AI-generated outputs, contributing to the growing discourse on AI-augmented engineering learning.

4.2.3 Inclusive Learning through Thematic Design

The Harry Potter–themed labs demonstrated that *narrative and playfulness can serve as inclusive pedagogical strategies*. Students from diverse educational backgrounds including design, computer science, and electrical engineering, found the labs accessible and engaging. The story-driven framing lowered engagement barriers often associated with coding or circuit design, while shared datasets and modular hardware ensured equitable participation. This finding supports the idea that thematic design can democratize access to complex technical topics, particularly in courses that blend creative and analytical skills.

5 Discussion

5.1 Integration of Technical Rigor and Creative Engagement

The course redesign demonstrates that rigor and creativity need not be competing forces in multidisciplinary engineering education. By pairing rigorous hardware–software integration with

narrative, theme-based framing, students were able to engage deeply with complex technical systems while maintaining intrinsic motivation and curiosity. The sequential lab structure encouraged progressive autonomy: learners first practiced structured techniques such as quantization and networking before applying those principles in open-ended, imaginative contexts like the Sorting Hat and Magic Wand. This sequential approach aligned with experiential learning theory, in which iterative cycles of concrete experimentation and reflection lead to conceptual integration. Rather than diluting technical depth, the creative framing fostered ownership and persistence, allowing students from varied educational disciplines to find meaningful entry points into embedded machine-learning practice.

5.2 Reflections on the Role of Generative AI

Generative AI functioned simultaneously as a cognitive partner and an object of critical inquiry. Students reported using large-language models for code debugging, documentation, and design brainstorming, but they also confronted issues of reliability, bias, and authorship. This dual use reframed AI from a productivity aid to a reflective learning scaffold: one that revealed how technical fluency, ethical reasoning, and metacognitive awareness intersect. Importantly, the technology supported inclusion: students with weaker programming backgrounds could prototype and iterate more confidently, while advanced students used the same tools to explore optimization and system-level integration. However, these benefits required explicit guidance to prevent over-reliance and to ensure that AI outputs were interpreted, not accepted at a face value. Future iterations will formalize reflection prompts and rubrics on AI-assisted work to cultivate deliberate and transparent use.

5.3 Implications for Multidisciplinary Programs

The TECHIN 515 model offers a transferable framework for programs seeking to bridge engineering, design, and computing. Three design principles emerged as generalizable: (1) thematic framing that links creativity with technical challenge, (2) sequential progression from structured to open-ended inquiry, and (3) AI-augmented reflection that supports learners with diverse prior experience. Adopting such a model requires institutional support for cross-departmental collaboration and faculty comfort with both emerging tools and playful learning formats. When these conditions are met, courses can become sites for authentic multidisciplinary practice—where students learn to navigate the tradeoffs among algorithmic precision, user experience, and societal impact.

5.4 Impact of Student Background Diversity

The wide distribution of academic backgrounds has shaped the learning dynamics. Students with design and arts backgrounds often demonstrated strong strengths in prototyping, user experience, and creative ideation. Engineering-trained students contributed technical depth in debugging and optimization. This diversity created peer-learning opportunities but also required structured scaffolding, including starter code, guided labs, and AI-assisted debugging support. With AI-assisted learning, most students demonstrated confidence in taking on some development workload to speed up the development cycle.

6 Conclusion

This work illustrates how reimagining a hardware–software integration course through creative theming and Generative AI support can foster deeper technical learning and inclusion across disciplines. By blending experiential learning, playful narrative, and reflective AI use, the redesigned TECHIN 515 course demonstrates that engineering rigor and creative engagement can successfully coexist. The approach highlights the potential of AI-augmented pedagogy not as a shortcut to solutions but as a catalyst for growth, equity, and innovation. Future iterations will continue refining assessment frameworks and exploring how narrative-driven, AI-supported design can empower the next generation of engineers to think both critically and imaginatively about intelligent systems. In future offerings, the instructional team plans to integrate more structured AI-literacy assessment, longitudinal tracking of project outcomes, and cross-program collaborations with design and art disciplines.

References

- [1] B. kazem Ansaf and N. I. Jaksic, “Teaching mechanical design for mechatronics engineering students using a project-based sequential learning approach,” in *2018 ASEE Annual Conference & Exposition*, 2018.
- [2] F. S. Madani, A. C. Martins, A. H. Moreira, and A. H. Moreira, “Mechatronics engineering integrate project: An approach in project-based learning with the subjects of instrumentation, control systems, and microcontrollers,” in *2023 ASEE Annual Conference & Exposition*, 2023.
- [3] D. Davcev, B. Stojkoska, S. Kalajdziski, and K. Trivodaliev, “Project based learning of embedded systems,” *arXiv preprint arXiv:1606.07498*, 2016.
- [4] H. Üstünel, “A project based innovative approach to an embedded systems course laboratory in software engineering education,” *Computer applications in engineering education*, vol. 28, no. 1, pp. 160–166, 2020.
- [5] J. Murray, L. C. Paxson, S. Seo, and M. Beattie, “STEM-oriented alliance for research (soar): An educational model for interdisciplinary project-based learning,” in *2020 ASEE Virtual Annual Conference Content Access*, 2020.
- [6] G. Hoople, A. Choi-Fitzpatrick, and E. Reddy, “Drones for good: interdisciplinary project based learning between engineering and peace studies,” *International Journal of Engineering Education*, vol. 35, no. 5, pp. 1378–1391, 2019.
- [7] R. Lavi, C. Cong, Y. Lai, J. A. Lavalley, G. L. Long, N. Melenbrink, *et al.*, “The evolution of an interdisciplinary case-based learning first-year course,” in *2023 ASEE Annual Conference & Exposition*, 2023.
- [8] T. Pham, T. B. Nguyen, S. Ha, and N. T. N. Ngoc, “Digital transformation in engineering education: Exploring the potential of ai-assisted learning,” *Australasian Journal of Educational Technology*, vol. 39, no. 5, pp. 1–19, 2023.
- [9] J. Qadir, “Engineering education in the era of chatgpt: Promise and pitfalls of generative ai for education,” in *2023 IEEE Global Engineering Education conference (EDUCON)*, pp. 1–9, IEEE, 2023.
- [10] R. Baltà-Salvador, I. El-Madafri, E. Brasó-Vives, and M. Peña, “Empowering engineering students through artificial intelligence (ai): Blended human–ai creative ideation processes with chatgpt,” *Computer Applications in Engineering Education*, vol. 33, no. 1, p. e22817, 2025.

- [11] C. Freitas, “Narrative integration in engineering education through story-based pedagogy: Lessons from a first-year engineering case study,” in *2024 ASEE North Central Section Conference*, 2024.
- [12] V. Rao, G. Moore, O. A. Udekwu, and B. Hartmann, “Tracing stories across the design process: A study of engineering students’ engagement with storytelling in an undergraduate humancentered design course,” *International Journal of Engineering Education*, vol. 36, no. 2, pp. 762–772, 2020.
- [13] D. A. Kolb, R. E. Boyatzis, and C. Mainemelis, “Experiential learning theory: Previous research and new directions,” in *Perspectives on Thinking, Learning, and Cognitive Styles*, pp. 227–247, Routledge, 2014.
- [14] K. Kaipa, A. Purwar, O. M. Ayala, and S. Bawab, “Transforming engineering education: Project-based learning and technology integration in a senior-level mechanisms course,” in *2025 ASEE Annual Conference & Exposition*, 2025.
- [15] A. Pyster, D. H. Olwell, T. L. J. Ferris, N. Hutchison, S. Enck, J. F. Anthony, D. Henry, and A. Squires, “Graduate Reference Curriculum for Systems Engineering (GRCSE™) Version 1.1.” <http://www.bkcase.org/grcse-10/>, 2015. The Trustees of the Stevens Institute of Technology.
- [16] F. Lanubile, S. Martínez-Fernández, and L. Quaranta, “Teaching mlops in higher education through project-based learning,” in *2023 IEEE/ACM 45th International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET)*, pp. 95–100, 2023.