2026 ASEE ANNUAL CONFERENCE June 21–24, 2026 | Charlotte Convention Center, NC

Host Sponsor UNC Charlotte

Paper ID #50111

A Dual-Focus Course on Biomedical Signal Processing and Machine Learning for Electrical and Computer Engineering Students

Dr. Sylmarie Davila-Montero, The Citadel

Sylmarie Dávila-Montero serves as an assistant professor in the Electrical and Computer Engineering Department at The Citadel. Her teaching experience spans across various undergraduate levels, including freshmen, sophomores, juniors, and seniors. She received a B.S. degree in electrical engineering and a minor in applied mathematics (with high honors) from the University of Puerto Rico at Mayagüez, Puerto Rico, in 2015 and a Ph.D. degree in electrical engineering from Michigan State University (MSU), East Lansing, Michigan, in 2022. Sylmarie worked as a Systems Engineer for the MITRE Corporation in 2015 and as a Data Analyst/Contractor for the Environmental Protection Agency in 2018 and 2019. Her research interests include real-time processing of biomedical and social signals, efficient implementation of machine learning algorithms, design of wearable social behavior and health monitoring systems, and engineering education.

©American Society for Engineering Education, 2026

A Dual-Focus Course on Biomedical Signal Processing and Machine Learning for Electrical and Computer Engineering Students

Introduction

With the increase in data availability and wearable technology, future engineers must be equipped to meet data processing needs as they enter the workforce. With the rise of artificial intelligence (AI), data processing engineers should be able to understand the capabilities, limitations, and implications that AI tools have on processing and extracting information from data of interest.

In electrical and computer engineering curricula, courses such as "Signals and Systems" and "Digital Signal Processing" introduce students to the analysis and processing of signals, such as removing noise, understanding frequency content, and implementing algorithms for such processing. Other courses, such as "Pattern Recognition" or machine learning and AI classes, provide tools for extracting and recognizing patterns from data. However, these concepts are rarely presented from a comprehensive perspective, where signals in the physical world are collected by sensors connected to processing units that convert those signals from analog to digital form, and further integrated with signal processing techniques combined with machine learning, especially in real-time scenarios.

To equip students with a comprehensive perspective on the world of sensors and data processing, a new signal processing and machine learning course focused on biomedical applications was developed for senior electrical and computer engineering students. Wearable sensor technology and their biomedical applications are continuing to form a key part of the well-being of our communities and the future of engineering. To provide students with real-word scenarios, this course, designed using the backward design method, had the primary objective of teaching students how to apply signal processing and machine learning techniques to analyze physiological signals in real time. This course was not designed as a substitution for classes like "Signals and Systems" or "Pattern Recognition," but as an advance level class that helps connect key concepts taught in both of these classes and provides hands-on experience in the development of biomedical signal analysis pipelines.

This paper describes the approach that was taken to present students with a comprehensive perspective on the processing and analysis of physiological signals and to evaluate developed skills among students. Overall, students participated in a well-rounded learning experience that blended theoretical concepts with practical applications. This paper includes course learning objectives, a description of the course modules, major assessments (exams and final project description), minor assessments (labs, homeworks, and quizzes), description of grading rubrics, and students' reflections and feedback. Students' reflections, collected through minute papers at the end of every class, demonstrate student engagement and understanding of the class material. Additional feedback was collected through the use of class evaluations and a post-class questionnaire.

Design and Objectives

Course Context and Background

The course was developed at The Citadel, a primarily undergraduate institution located in South Carolina. The development of this course aligns with the America's AI Action Plan [1] to educate the future workforce on topics related to artificial intelligence and emerging technologies, especially in Senior Military Colleges. Offered as a 3-credit hour course, the curriculum was designed with flexibility to accommodate both undergraduate and graduate students. For graduate-level credit, students incorporate additional research components throughout the semester, allowing for deeper investigation of topics and more sophisticated project outcomes.

The course enrolls a maximum of 18 students per section, a deliberate constraint that enables the instructor to provide individualized attention during hands-on laboratory activities and project work. While the course primarily attracts electrical and computer engineering (ECE) majors, it has also drawn students from mechanical engineering backgrounds. This diversity enriches classroom discussions and necessitates careful attention to varying levels of prerequisite knowledge among enrolled students.

Prerequisites and Student Preparation

The recommended prerequisites for the course include signals and systems, microcontrollers, and programming fundamentals. However, recognizing that rigid prerequisite enforcement might limit access to this interdisciplinary content, these requirements are not strictly enforced. Instead, at the course outset, students are explicitly advised about the foundational knowledge expected in these areas. Those lacking specific prerequisite exposure are encouraged to engage in self-directed learning to bridge knowledge gaps as needed throughout the semester.

This flexible approach to prerequisites accommodates the diverse student population. ECE students typically possess strong backgrounds in signals and systems and microcontroller programming, while mechanical engineering students often bring complementary knowledge from mechatronics and control system courses, familiarizing them with terminology related to sensors, data acquisition, and real-time systems. This variation in student backgrounds, while presenting instructional challenges, ultimately contributes to richer peer learning opportunities.

Pedagogical Rationale and Design Philosophy

The course design stems from a pedagogical philosophy centered on making abstract signal processing concepts tangible and motivating through real-world biomedical applications. Traditional courses in signal processing often intimidate students due to their mathematical rigor and perceived disconnect from tangible applications. By focusing the content in physiological signals, a concept students can relate to through their own bodies and experiences, the course reduces this intimidation factor and increases student motivation and engagement.

The dual-focus structure serves multiple pedagogical purposes. First, it demonstrates the synergy between signal processing and machine learning, two domains increasingly connected in modern engineering practice. Students learn not only how to process biomedical signals but also how processed signals serve as inputs to machine learning algorithms for clustering, classification, prediction, and decision-making. This integrated approach mirrors real-world workflows in the biomedical device industry and healthcare technology sectors.

The course design follows backward design principles, beginning with clearly defined learning outcomes and then developing assessments and instructional activities aligned with those outcomes [2]. Additionally, Bloom's taxonomy guided the scaffolding of learning objectives, progressing from lower-order cognitive skills (describing, identifying, listing) to higher-order skills (applying, analyzing, implementing, demonstrating) [3]. This intentional progression ensures students build foundational knowledge before tackling complex applications and design challenges.

Course Learning Objectives (CLOs)

Questions like "What knowledge do we want students to attained in this class?" and "What skills do we want them to further develop or strengthen?" were asked to create course-level learning objectives/outcomes. As a result of this exercise, seven course learning objectives were established:

1) *Describe* major physiological signals based on their characterization.
2) *Identify* signal processing and machine learning techniques that can be used to perform real-time signal analysis.
3) *List* limitations of processing techniques to perform real-time signal processing.
4) *Identify* the appropriate processing unit (microcontroller, DSP, FPGA, etc.) to implement the processing algorithm.
5) *Apply* signal processing techniques to characterize a given/recorded physiological signal.
6) *Implement* the design of a real-time signal processing algorithm in MATLAB for the processing of a given physiological signal.
7) In the form of a written report, *demonstrate* ability to document the design and implementation process of a signal processing algorithm.

These objectives reflect both technical competencies and professional skills. CLOs 1-5 emphasize conceptual understanding and analytical thinking, requiring students to understand signal characteristics, processing methods, implementation constraints, and hardware selection. CLO 6 focuses on practical implementation skills, requiring hands-on coding and algorithm development. CLO 7 addresses professional communication skills, essential for engineering practice, by requiring students to document their work in a clear, structured manner suitable for technical audiences. Table I shows the relationship of the course learning outcomes with ABET student outcomes [4] and Bloom's taxonomy levels.

Table I. Course learning outcomes and their relationship to ABET student outcomes and Bloom's taxonomy.

	ABET student outcomes							
Course learning objective (CLO)	**Outcome 1**	**Outcome 2**	**Outcome 3**	**Outcome 4**	**Outcome 5**	**Outcome 6**	**Outcome 7**	**Bloom's Taxonomy**
1) *Describe* major physiological signals based on their characterization	1	1				2		Comprehension
2) *Identify* signal processing and machine learning techniques that can be used to do real-time signal analysis	3							Knowledge
3) *List* limitations of processing techniques to perform real-time signal processing	1							Knowledge
4) *Identify* the appropriate processing unit (microcontroller, DSP, FPGA, etc.) to implement the processing algorithm	1							Knowledge
5) *Apply* signal processing techniques to characterize a given/recorded physiological signal	3							Application
6) *Implement* the design of a real-time signal processing algorithm in MATLAB for the processing of a given physiological signal	2	1			3	3	1	Create
7) In the form of a written report, *demonstrate* ability to document the design and implementation process of a signal processing algorithm			2					Create

1 = objective addresses outcome slightly, 2 = moderately, 3 = substantively

The emphasis on real-time processing (CLOs 2-4) distinguishes this course from traditional signal processing courses that focus primarily on offline analysis. Understanding the constraints of real-time implementation (computational complexity, latency requirements, hardware limitations) prepares students for the realities of deploying biomedical signal processing systems in clinical or consumer device contexts.

Course Content

The course content is organized into two distinct but interconnected parts spanning 16 weeks. Part I, "Introduction to Digital Signal Analysis Including Real-Time Considerations," establishes foundational knowledge in signal processing and machine learning techniques with explicit attention to real-time implementation constraints. Part II, "Processing of Biomedical Signals," applies these techniques to specific physiological signals through case studies and in-depth analysis. This progression from general methods to specific applications allows students to first build a toolkit of techniques and then practice selecting and applying appropriate methods to authentic biomedical signal processing challenges.

Part I: Introduction to Digital Signal Analysis Including Real-Time Considerations

Part I comprises the first 13 weeks of the course and is further divided into two subsections: Signal Processing Techniques (Modules 1-2) and Machine Learning Techniques (Modules 3-4). The first part of the course is focused on helping students meet CLOs 1-7 by establishing the fundamental concepts and methods necessary for biomedical signal analysis. Throughout this portion of the course, real-time processing considerations are explicitly addressed, distinguishing this curriculum from traditional signal processing courses that focus exclusively on offline analysis.

Module 1: Signals and Biomedical Signal Processing (Weeks 1-2) - The course begins by establishing a common understanding of physiological signals and their unique characteristics. Students are introduced to the concept of a "physiological signal" as measurable electrical, mechanical, or chemical phenomena generated by living systems. The module covers the distinctions between analog, discrete, and digital signals, providing the foundation for understanding how continuous physiological phenomena are captured and represented digitally. The fundamental concepts of sampling theory and the Nyquist rate are covered, ensuring students understand the relationship between continuous physiological phenomena and their digital representations. A central framework introduced in this module is the general model of signal processing for biomedical signals, which encompasses signal acquisition principles, signal conditioning requirements, signal digitization, pre-processing and signal processing techniques including filtering, time analysis, frequency analysis, time-frequency analysis, feature extraction, and the use of machine learning techniques. Students learn about the complete pipeline from sensor placement through analog-to-digital conversion to computational analysis. Signal conditioning techniques, including amplification, anti-aliasing filtering, and artifact removal, are presented in the context of preparing physiological signals for analysis. The concept of signal characterization – extracting meaningful parameters and features that describe signal properties – is introduced as the ultimate goal of the first part of the processing pipeline. Concepts like periodicity and deterministic versus stochastic signals are introduced as part of understanding signals characterization.

This foundational module addresses multiple course learning objectives by helping students understand how to describe major physiological signals (CLO #1), how to apply signals processing concepts to characterize recorded biomedical signals (CLO #5), and get familiarized with the processing framework to eventually implemented in MATLAB (CLO #6).

Module 2: Signal Processing Techniques and Real-Time Considerations (Weeks 3-10) - Module 2 represents the most extensive portion of the course, spanning eight weeks and covering core signal processing methods essential for biomedical applications. The module is divided into two parts, separated by the first examination, allowing students to consolidate learning at the midpoint.

Part I (Weeks 3-6) begins with temporal characteristics of signals, examining how physiological signals evolve over time and how time-domain features can be extracted. Students then explore processing and transformation techniques, with particular emphasis on the Fourier transform and its discrete implementation. The one-dimensional discrete Fourier transform (DFT) is presented both theoretically and practically, with students implementing DFT algorithms and interpreting frequency-domain representations of physiological signals.

Filter design constitutes a major component of this section. Students learn about finite impulse response (FIR) and infinite impulse response (IIR) filters, their design principles, and their application to biomedical signals. Practical considerations such as filter order, cutoff frequencies, and passband/stopband characteristics are explored through hands-on design exercises. Students apply filters to remove powerline interference, baseline wander, and other common artifacts from physiological recordings.

Following the first examination, Part II (Weeks 7-10) introduces more advanced time-frequency analysis methods. The short-time Fourier transform (STFT) is presented as a solution to analyzing non-stationary signals (those whose frequency content changes over time) which is characteristic of many physiological signals. Students learn to create and interpret spectrograms, visualizing how signal frequency content evolves. Wavelet transforms are introduced as a powerful alternative to STFT, offering variable time-frequency resolution. Students explore different wavelet families (e.g., Daubechies, Symlets, Morlet) and their applications to physiological signal decomposition and denoising. Additional signal processing methods, including cosine transforms, correlation functions, and power spectral density estimation, round out the signal processing toolkit.

A critical component of Module 2 is feature extraction, which bridges signal processing and machine learning. Students learn to extract biomedical and biological features (e.g., heart rate, QRS duration, spectral edge frequency) as well as signal processing features derived from transformations. These include physical features and statistical measurements (mean, variance, skewness, kurtosis), signal power in specific frequency bands (e.g., delta, theta, alpha, beta, gamma bands for electroencephalography (EEG)), and wavelet-based measures (e.g., wavelet coefficients, wavelet entropy).

Throughout Module 2, real-time processing considerations are incorporated into the content. Students examine the computational complexity of different algorithms, comparing fast Fourier transform (FFT) efficiency to DFT, evaluating the memory requirements of different filter structures, and considering latency constraints in clinical monitoring applications. These discussions prepare students to make informed decisions when implementing algorithms on resource-constrained embedded systems.

Two laboratory activities during Module 2 are structured to reinforce each technique as it is introduced. Laboratory activity #1 asks students to study EEG signals, their frequency content by applying the FFT, designing filters to remove artifacts and extract signals at specific frequency bands, determine an adequate window size to process data in a real-time fashion, and extract power spectral density features. Laboratory activity #2 asks students to apply the FFT, the STFT, and the Wavelet transform to previously studied EEG signals. Each lab requires students to write

MATLAB code, document their approach, analyze results, and reflect on the advantages and limitations of each technique.

Module 2 addresses six course learning objectives by providing students with the tools to justify how they are describing major physiological signals (CLO #1), the tools to perform signal processing (CLO #2), providing the limitations and considerations of signal processing tools when performing real-time analysis (CLO #3), how to apply signal processing concepts to characterize recorded biomedical signals and extract features (CLO #5), how to implement half of the processing framework in MATLAB (CLO #6), and how to document design decisions (CLO #7).

Module 3: Machine Learning Techniques and Real-Time Considerations (Weeks 11-13) - Module 3 introduces machine learning as the logical extension of signal processing for biomedical applications. While signal processing enhances signal characteristics and extracts features from raw physiological data, machine learning enables automated pattern recognition, classification, and decision-making based on those features. The module begins with an introduction to machine learning fundamentals, establishing key terminology and concepts. The distinction between clustering (unsupervised learning) and classification (supervised learning) is emphasized, helping students understand when each approach is appropriate. Feature extraction is revisited in the context of preparing data for machine learning algorithms, with particular attention to feature transformation techniques such as Principal Component Analysis (PCA). Students learn how PCA can reduce dimensionality, eliminate redundancy, and improve computational efficiency – all critical considerations for real-time biomedical applications. Clustering methods covered include k-means clustering and hierarchical clustering, with examples drawn from biomedical contexts such as grouping similar physical activities from inertial motion units (IMUs). Students implement clustering algorithms and learn to evaluate cluster quality using metrics such as silhouette coefficients and within-cluster sum of squares. Classification models form the core of Module 3. Students are introduced to several classification algorithms, progressing from simpler to more complex methods:

- k-Nearest Neighbors (k-NN): A straightforward distance-based classifier used to introduce classification concepts
- Decision Trees: Rule-based classifiers that provide interpretable decision boundaries
- Support Vector Machines (SVM): More sophisticated classifiers that find optimal decision boundaries, particularly useful for biomedical data with complex separability
- Naive Bayes: Probabilistic classifiers useful when features can be assumed independent
- Neural Networks: Introduction to basic feedforward networks and their training through backpropagation

For each classification method, students learn the underlying principles, implement the algorithm in MATLAB (using built-in functions from the Statistics and Machine Learning Toolbox), and apply it to biomedical datasets. Model evaluation is heavily emphasized, with students learning to partition data into training and testing sets, perform cross-validation, and compute performance metrics including accuracy, sensitivity, specificity, precision, F1-score, and area under the receiver operating characteristic (ROC) curve. Real-time considerations are explicitly addressed in the machine learning context. Students examine the computational requirements of training versus inference, discussing why models are typically trained offline but must perform inference in real-time.

One laboratory activity during Module 3 integrates signal processing from earlier modules with newly learned machine learning techniques. The lab involves: (1) loading a dataset of pre-recorded physiological signals, (2) extracting features using techniques from Module 2, (3) training multiple classification models, (4) evaluating and comparing model performance, and (5) selecting the most appropriate model based on both accuracy and real-time feasibility. These integrated labs reinforce the complete workflow from raw signal to automated decision-making.

Module 3 addresses four course learning objectives by providing students with machine learning tools to perform signal analysis (CLO #2), providing the limitations and considerations of machine learning tools when performing real-time analysis (CLO #3), how to implement the complete processing framework in MATLAB (CLO #6), and how to document design decisions (CLO #7).

Module 4: Processing Units (Week 13) - Module 4 provides a practical perspective on hardware implementation, addressing the question: "Where will my signal processing and machine learning algorithms actually run?" Students are introduced to various processing units commonly used in biomedical devices, such as microcontrollers, digital signal processors (DSPs), field-programmable gate arrays (FPGAs), and other platforms (graphical processing units (GPUs), application-specific integrated circuits (ASICs), and cloud-based processing).

Rather than teaching hardware design in depth, this module provides students with a framework for selecting appropriate processing units for specific applications. Guiding questions include: What are the computational requirements (operations per second)? What are the latency constraints (maximum allowable delay)? What are the power constraints (battery-operated device versus wall-powered)? What is the production volume (prototype versus mass-market device)? By working through case studies, students practice matching processing units to application requirements.

Module 4 addresses two course learning objectives by providing students with the limitations and considerations of real-time analysis perform in hardware devices (CLO #3) and the background information and techniques to identify the appropriate processing unit based on the type of analysis to perform and the product end goals. A second examination, administered at the end of Week 13, assesses student understanding of signal processing techniques (Module 2), machine learning methods (Module 3), and hardware considerations (Module 4).

Part II: Processing of Biomedical Signals

The second part transitions to application-oriented learning through specific case studies that integrate signal processing with machine learning approaches. Students explore complete workflows for problems such as arrhythmia detection from ECG and identifying the number of neurons firing from local field potential recordings. This second part of the course is focused on helping students meet CLOs #6 and #7. This portion of the course demonstrates how the techniques from Part I are selected and combined to address real-world biomedical signal processing challenges.

Module 5: Electric Activities of the Cell (Week 15) - Module 5 provides biological context by examining the origin of bioelectrical signals at the cellular level. Students learn about the electrical characteristics of cell membranes, including resting potential, ion channels, and the action potential generation mechanism. Understanding these fundamental processes helps students appreciate why physiological signals have particular characteristics and what information they convey about biological function.

The module covers electric data acquisition techniques specific to cellular recordings, including patch-clamp electrophysiology and microelectrode arrays. Students examine typical features extracted from action potential recordings, such as spike amplitude, duration, firing rate, and interspike intervals. Analysis techniques applicable to cellular electrical activity, including spike sorting, burst detection, and synchrony analysis, are presented.

Module 6: Electrocardiogram (Week 16) - The final instructional week focuses on the electrocardiogram (ECG), one of the most widely used physiological signals in clinical practice. Students learn about the cardiovascular system's electrical activity and how it manifests in surface ECG recordings. The characteristic waveforms (P wave, QRS complex, T wave) are examined in detail, along with the physiological events they represent. Processing and feature extraction techniques specific to ECG are emphasized. Students are presented with QRS detection algorithms (e.g., Pan-Tompkins algorithm), how to calculate heart rate and heart rate variability metrics, and extract intervals (RR, PR, QT) with clinical significance. The module explores how ECG morphology changes in various cardiovascular diseases, including arrhythmias (atrial fibrillation, ventricular tachycardia), myocardial infarction, and conduction abnormalities.

Tools, Software, and Datasets

The course relies primarily on MATLAB as the computational environment, chosen for its extensive signal processing and machine learning capabilities, its widespread use in both academia and industry, and its relatively gentle learning curve for students with prior programming experience. Students utilize several MATLAB toolboxes throughout the course:

- Signal Processing Toolbox: For filter design, spectral analysis, and time-frequency transforms
- Wavelet Toolbox: For wavelet decomposition, denoising, and feature extraction
- Statistics and Machine Learning Toolbox: For implementing classification and clustering algorithms

Students are encouraged to explore Python as an alternative, and to complete assignments using Python libraries such as NumPy, SciPy, and scikit-learn. This flexibility accommodates students who enter the course with Python experience and wish to deepen those skills. Regardless of which programming environment students select, it is emphasized that these are primarily prototyping environments and that full deployment of designed algorithms require other development environments and hardware devices discussed in Module 4.

Datasets used throughout the course come primarily from publicly available repositories, ensuring students work with authentic physiological signals. The data sources is PhysioNet [5] – a comprehensive resource providing access to ECG, EEG, EMG, and other physiological recordings, including the MIT-BIH Arrhythmia Database, the European ST-T Database, the Sleep-EDF Database, and the Auditory evoked potential EEG-Biometric dataset. Working with real physiological data introduces students to practical challenges such as missing data, variable signal quality, annotation inconsistencies, and class imbalance, challenges rarely encountered with synthetic or idealized datasets but ubiquitous in real-world applications.

Instructional Methods

The instructional approach employs a blend of traditional and active learning pedagogies designed to accommodate diverse student backgrounds while fostering deep understanding of both theoretical concepts and practical implementation skills. Class sessions follow a primarily traditional lecture format enhanced with interactive elements. Each session begins with a 5-10 minute review addressing questions from students' "minute papers" [6] – brief written reflections submitted at the end of every class where students summarize key points, pose questions about the material, or answer specific questions provided by the class instructor. This feedback loop ensures continuity between sessions and clarifies misconceptions before introducing new content.

Lectures utilize PowerPoint presentations with extensive visual representations of signals, algorithms, and processing outputs. A distinguishing feature is the integration of live coding demonstrations where the instructor writes and executes MATLAB code during class, narrating the thought process and modeling debugging strategies.

Active learning techniques are integrated throughout each session. Think-pair-share activities require students to consider questions individually, discuss with a neighbor, and share insights with the class. In-class problems and discussions prompt analytical thinking through questions like "Which classifier would you choose for this application and why?" These strategies maintain engagement and ensure students actively process material rather than passively receive it. “Weekly” quizzes (5-10 minutes) assess knowledge retention from the past two to three class sessions, combining conceptual questions and brief computational problems. These low-stakes assessments maintain accountability and encourage regular study habits while providing formative feedback to both students and instructor.

Laboratory Structure

Laboratory sessions provide hands-on implementation experience. Labs are structured for independent work, each student completes and submits their own assignment, but collaboration is explicitly encouraged. Students frequently form informal working groups to discuss approaches and troubleshoot problems, balancing individual accountability with collaborative learning.

The instructor circulates during lab sessions, providing guidance through Socratic questioning rather than direct solutions. For example, when a student's filter produces unexpected results, the instructor might ask: "What were you expecting? What does your output suggest about the filter's behavior? Have you visualized the frequency response?" This approach develops debugging skills and analytical thinking.

No pre-lab requirements are imposed, recognizing students' diverse backgrounds and varying prerequisite knowledge. Instead, lab time itself provides opportunity for exploration and learning through doing, with lecture materials available via Canvas for students who wish to prepare in advance.

Final Project: Support

The final project, centered on processing local field potentials (LFPs) for undergraduate students, represents the culminating demonstration of student learning. LFPs were chosen because they incorporate many biomedical signal processing challenges: non-stationarity, noise contamination, complex frequency content, and sophisticated feature extraction requirements.

The final three weeks transition from new content to project preparation. Class sessions cover LFP background, neurophysiological origins, recording techniques, and analysis approaches. The instructor presents published research dissecting signal processing and machine learning pipelines, and students examine LFP datasets to discuss domain-specific challenges.

A portion of class time during these weeks is dedicated to in-class project work, providing protected time with instructor support immediately available. This allows the instructor to monitor progress, provide timely intervention, and foster idea exchange among students. Students receive iterative feedback throughout development during these sessions and office hours, mirroring professional engineering practice where designs evolve through implementation, evaluation, and refinement cycles.

Assessments and Evaluation

The assessment strategy employs multiple evaluation methods designed to measure student achievement across all seven course learning objectives. This multi-faceted approach provides formative feedback throughout the semester while culminating in summative assessments that evaluate comprehensive understanding. Table II describes the grade distribution, which emphasizes summative assessments (65% combined for exams and project) while maintaining meaningful weight for formative assessments (35%) that support learning throughout the semester.

Formative Assessments

Minute Papers (5%) are submitted at the end of every class, graded on completion and thoughtfulness rather than correctness. They provide valuable insight into student understanding and guide the instructor's opening reviews in subsequent sessions.

Table II. Grade Distribution.

Evaluation method	Percentage of final grade
Homeworks and Labs	15%
Quizzes	15%
Exam 1	20%
Exam 2	20%
Final project	25%
Class participation (Minute papers)	5%
Total	100%

"Weekly" Quizzes (15%) lasting 5-10 minutes cover material from the previous two to three sessions, combining conceptual questions and brief computational problems. Graded on correctness with partial credit for demonstrated understanding, these low-stakes assessments maintain accountability while providing immediate feedback. Approximately 8 quizzes distribute this weight, making each individual quiz relatively low-pressure. Quizzes start the third week of class.

Homework Assignments (part of 15% with labs) include three assignments serving distinct purposes: (1) MATLAB Signal Processing Onramp Tutorial [7] – graded on completion to establish baseline programming proficiency; (2) Conceptual preparation for Exam 1 – graded on quality and correctness of written responses to practice questions; (3) Concept Map of Part I material – graded on completeness of visual integration of signal processing topics.

Laboratory Assignments (part of 15% with homework) provide hands-on practice aligned with Modules 2 and 3. Three labs require MATLAB implementation of signal processing and machine learning techniques, with each including design questions and reflection components. Labs are graded based on rationale and justification – rewarding students who demonstrate understanding through evidence-based reasoning rather than merely obtaining correct results. Evaluation considers code functionality, documentation quality, design justification using learned material, and depth of analysis in reflections.

Summative Assessments

Two examinations assess mastery of theoretical concepts and practical skills at different course stages. The first exam (20%), administered after Module 2 (Week 7), covers foundational signal processing. The exam structure includes two parts. The first part focuses on concepts and definitions (25 points), short-answer and multiple-choice questions similar to quizzes. The second part focusses on signal preprocessing techniques (75 points) and application problems covering filter design, Fourier Transform, convolution, temporal feature extraction, and distinguishing LTI/non-LTI systems. Partial credit rewards critical thinking and rational.

The second exam (20%), administered after Module 4 (Week 13), assesses advanced techniques and machine learning. Structure mirrors Exam 1 with two parts: "Concepts and Definitions (25 points)" and "Design and Application (75 points)". The second part requires comprehensive problems integrating signal processing and machine learning, such as designing complete processing pipelines, selecting appropriate transforms (FT, STFT, wavelets), constructing feature matrices, and addressing supervised learning, overfitting, and performance evaluation.

The final project requires independent execution of the complete signal processing and machine learning workflow on local field potential recordings. Students process extracellular signals and train models to recognize neural spikes, producing two deliverables: (1) a 4-8 page technical report following IEEE paper structure, and (2) complete MATLAB code. Report Requirements include: introduction to extracellular recordings and biomedical significance; literature review of applicable techniques from peer-reviewed sources; adaptation of the general signal processing model; detailed pipeline documentation; dataset description; preprocessing methods; at least two feature extraction techniques; at least two machine learning models with training/testing phases and performance metrics; results and interpretation; hardware processing unit selection and justification; conclusion; disclosure of resources; and IEEE-formatted references (minimum three).

The evaluation rubric (80 points maximum) assesses four categories:

- Technical Content (55 pts): Ten dimensions including background content, literature review, model adaptation, signal processing framework, dataset description, preprocessing, feature extraction, machine learning models, analysis completeness (≥80% model accuracy expected), and hardware selection – each scored on 5-point scale (Excellent/Satisfactory/Minimal/Unsatisfactory)
- Disclosure (2 pts): Description of resources used for project completion
- Code (5 pts): Complete, functional MATLAB files
- Organization (9 pts): Adequate sections, good transitions, proper IEEE citations
- Presentation (9 pts): Grammar and style, quality of graphics, appropriate length

The rubric emphasizes process and documentation alongside results. Thorough justification and thoughtful analysis may outscore higher accuracy achieved without understanding, aligning with CLO #7 (demonstrate ability to document design processes) and professional engineering communication expectations.

Assessment Alignment

All seven CLOs are evaluated through multiple methods, ensuring validity and reliability. The grading philosophy emphasizes understanding over correctness, awarding partial credit generously for sound reasoning. Feedback is provided promptly (within one week for quizzes/homework, two weeks for labs/exams) with detailed comments explaining errors and suggesting improvements. This learning-oriented approach supports student development rather than merely sorting performance, acknowledging that early struggles can lead to later mastery through the iterative assessment structure.

Student Reflections and Feedback

Student feedback was collected through multiple mechanisms to assess learning experiences and inform course improvements. Data sources included minute papers submitted at the end of each class session (n=16, 100% participation rate), official course evaluations (n=7, 44% response rate), and a post-class questionnaire (n=5, 31% response rate). The course enrolled 16 students during a summer semester offering on a compressed schedule.

Data Collection Methods

Minute Papers provided real-time feedback throughout the semester, with all 16 students consistently submitting reflections summarizing key points learned and posing questions. Course Evaluations captured both quantitative ratings and qualitative comments about course organization and learning value. Post-Class Questionnaire gathered reflective feedback on the most impactful course components after completion.

Appreciation for Real-World Applications and Hands-On Learning

The dominant theme across all feedback mechanisms was student appreciation for authentic biomedical applications. Students consistently expressed that connecting signal processing theory to tangible physiological signals made abstract concepts more accessible and increased motivation. In course evaluations, students highlighted "The application of signal knowledge," "I enjoyed the fact that we applied electrical engineering skills to a different field. It was an enlightening and fresh feeling," and "The topic was interesting and the assignments and labs were useful for gaining experience processing signals."

The impact of working with real physiological data was particularly strong. One student noted: "The most impactful material I encountered were the real brain activity signals that were used in the labs and final project. In one of the labs we used seizure data from real patients to practice isolating and processing. Using real signals helped me understand not only the importance of the concepts we were learning but also the important applications." Another connected course content to personal experience: "Being able to create the algorithm based on a signal of my choosing was impactful due to my wife being a paramedic and having to use these systems for work constantly."

Minute papers consistently emphasized the value of hands-on MATLAB implementation. Following the first lab activity, students reported: "Today's activity has tremendously helped me improve my understanding of signal processing. I am a hands on learner, and getting some grind time in MATLAB really helped me put all the pieces together," and "Today's lab gives some hands-on experience to reaffirm what we've been told. It was helpful to go through and do the math ourselves that was explained in class earlier, mainly because I am not great at internalizing things just told to me." Students explicitly identified themselves as hands-on learners who benefited from translating theoretical concepts into working code.

Machine Learning Integration and Synthesis

As the course transitioned to machine learning applications, minute papers revealed student engagement with the integrated pipeline. Students identified memorable concepts including: "Being able to confidently condition and pre-process a signal seems paramount to efficient machine learning," and "Feature selection for machine learning. Making sure you have the model looking at the right features from the start can for sure help increase the accuracy of the model."

Students particularly appreciated counterintuitive concepts. Multiple students noted interest in the idea that 100% training accuracy indicates overfitting: "The 70% 30% rule for training a Machine learning algorithm and the idea that a 100% accurate model is a bad thing stands out as interesting. They are concepts that make sense in retrospect, but I would not have thought of it without being told."

Exposure to diverse machine learning algorithms expanded student horizons: "Learning about the different machine learning models was very interesting. Some models are better used for some data types or even if the data is labeled or not. I've heard about neural network models but never knew about like decision trees, or k-means, or support vectors."

Impact of the Final Project

The final project emerged as the most frequently cited impactful component. One student provided detailed reflection: "The final project allowed for an application of all of the subject matter taught to be utilized in a real data analytic application. The project was a culmination of all knowledge gained from projects, quizzes, and tests. Upon completing the project, I had an increased confidence as well as understanding of a complicated subject. I believe that the project structure allowed for me to clarify areas I was weak and strengthen an understanding that would allow me to implement ML in my professional life. I do not believe a project that simply tested the sections of the class would have had a similar impact."

This validates the comprehensive project design requiring integration of signal processing, feature extraction, machine learning, and hardware selection. Students recognized the project as more than assessment – it functioned as a synthesizing learning experience that consolidated fragmented knowledge into coherent understanding.

Renewed Interest and Interdisciplinary Perspectives

Several students expressed that the course sparked interest in machine learning and biomedical applications. One wrote: "Being exposed to machine learning in general was by far the best part. It renewed my eagerness of trying to learn something new, and although it's not something I could pursue at this school, it did give me something exciting to learn on my own time." Students from diverse backgrounds appreciated the interdisciplinary nature, with comments indicating the course broadened perspectives beyond traditional ECE domains.

Challenges and Areas for Improvement

Despite positive feedback, students identified challenges informing future improvements. The most consistent concern related to pacing due to the compressed summer schedule. Multiple students indicated they wished the course were taught during a full semester to allow more time to grasp material. The accelerated pace, covering 16 weeks of content in approximately 10 weeks, created intensity that limited opportunities for deeper exploration.

Minute papers revealed specific confusion points. Following the first lab, students reported: "I didn't get as far as I wanted to today, a small mistake upfront threw off all the next steps," and "Still a little overwhelmed but this type of activity is helping bridge gaps." These highlight the cascading effect of early coding errors and the need for additional debugging support.

Students sought more guidance on decision-making criteria, with questions including: "How to know which [feature selection method] to choose for the best results," "Which methods are more or less effective for the various biomedical signals," and "How are you supposed to know how [PCA is] organizing the data?" These suggest students desired more explicit frameworks for making design choices – moving beyond "what" techniques exist to "when" and "why" to apply them.

Some students requested clearer expectations: "The only thing I wished is that we had extra time and a better idea of how some of the signals should have looked." The diversity of student backgrounds created differential challenge levels, with some needing to build foundational skills concurrently with learning new content.

Conclusion

Student feedback indicates a challenging but rewarding course that successfully achieved learning objectives while maintaining high engagement. Students valued authentic applications, hands-on experiences, and the comprehensive final project. They appreciated the interdisciplinary nature and the integration of signal processing with machine learning as valuable preparation for emerging technologies. Areas for improvement center on pacing (addressed by offering in regular semesters), providing additional decision-making frameworks, and enhanced support for varied prerequisite backgrounds.

Overall, students participated in a well-rounded learning experience that blended theoretical concepts with practical applications. The course content was divided into two parts: the first focused on digital signal processing techniques and the second on machine learning techniques. For both sections, the implications of these algorithms for real-time processing were discussed. Specific physiological signals and their common processing analyses were also covered. Two exams assessed students' understanding of theoretical concepts and design implications, while a final project evaluated their ability to apply what they learned to process a physiological signal. Three in-class labs, which served as stepping stones for the final project, were also part of the course evaluation. Throughout the course, students practiced design, programming, research, and communication skills.

References

[1] The White House, "*America's AI Action Plan*," Executive Office of the President, Washington, DC, USA, Jul. 2025. [Online]. Available: https://www.whitehouse.gov/wp-content/uploads/2025/07/Americas-AI-Action-Plan.pdf. Accessed: Aug. 19, 2025.

[2] G. Wiggins and J. McTighe, *Understanding by Design*, 2nd ed. Alexandria, VA, USA: Association for Supervision and Curriculum Development (ASCD), 2005.

[3] L. W. Anderson and D. R. Krathwohl, Eds., *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives*. New York, NY, USA: Longman, 2001.

[4] ABET, "*Criteria for Accrediting Engineering Programs, 2025–2026*," ABET, USA, 2025. [Online]. Available: https://www.abet.org/accreditation/accreditation-criteria/criteria-for-accrediting-engineering-programs-2025-2026/. Accessed: Jan. 16, 2026.

[5] "*The Research Resource for Complex Physiologic Signals*," PhysioNet. [Online]. Available: https://www.physionet.org/. Accessed: Jan. 16, 2026.

[6] M. C. Lutterodt, "Qualitative formative feedback to the teacher benefits both student and teacher: The use of 'LEARN evaluation,' a modified 'one minute paper,'" *Improving University Science Teaching and Learning*, vol. 10, no. 1, 2017.

[7] MathWorks, *Signal Processing Onramp*, MATLAB Academy. [Online]. Available: https://matlabacademy.mathworks.com/details/signal-processing-onramp/signalprocessing. Accessed: Jan. 16, 2026.