A Hierarchical Framework of the Impact of End-User Machine Learning Tools on Applied Machine Learning Education

Jonathan Niemirowski, David Hall, Krystal Cruse and Mary Fendley

DOI: 10.18260/jet.43.2.5

Abstract

Development of end-user tools for machine learning (ML) has increased accessibility to ML projects, but current literature lacks pedagogical guidance on using these tools to assess or develop course content in higher education. This exploratory study presents an abstraction hierarchy model as a preliminary course design framework for assessing the information and functions associated with applied ML classroom projects from engineering and technology courses. These projects were designed for students without a robust computer science or software development background. To investigate how end-user ML tools impact the required knowledge and skills to complete an applied ML project, and how these observations might inform future course design, three applied ML projects were qualitatively evaluated and organized into the abstraction hierarchy model based on instructor observation. Each component was assessed for whether it required less instructional time due to the inclusion of end-user ML tools. Results indicated that contemporary tools streamlined the teaching of the creating, training, testing, and deploying ML models, allowing for increased focus on the initial design of an ML project, the creation and processing of data, and the evaluation of the final design in a real-world context, as opposed to typical training and testing metrics. These preliminary observations suggest that a barrier of entry allows non-experts to more readily approach and utilize ML technologies, though further validation across additional courses and instructors is needed to assess how broadly this framework applies.

1. Introduction

Artificial intelligence (AI) and machine learning (ML) are fast-growing technologies in STEM industries. The past decade has seen the introduction of the fourth industrial revolution, characterized by an explosive connection of devices, information, and automated processes (Pedota & Piscitello, 2022). Machine learning stands at the forefront of this advance, thanks in part to advances in processing and data transfer/storage speeds (Jordan & Mitchell, 2015). The rapid adoption of AI has led many to consider this decade to be the start of the fifth industrial revolution, characterized by more mindful human-machine interaction and concerns of ethical AI usage (Ali et al., 2022).

At Louisiana Tech University, one of many initiatives to keep pace with this rapidly changing landscape has been to incorporate machine learning content within the Instrumentation and Control Systems Engineering Technology (ICET) Program. This four-year bachelor's program employs project-based learning to develop students' practical skillsets in electrical and mechanical systems. Previously, machine learning education was dependent on a strong background in computer science (CS) and mathematics. During the development of course content for this program, a need emerged for ML tinkerers with the skills to effectively apply machine learning concepts to industrial problems; students should have experience in preparing and managing data for training as well as implementation of solutions within real-world contexts. With recent advances in machine learning tools, engineering students can utilize the technology without rigorous mathematics and computer science backgrounds (Sundberg & Holmström, 2024). While these new tools allow for increased complexity and depth in course content, they also allow for significant knowledge gaps to form within students' understanding of the material. These gaps in content knowledge show a clear need for better guidance on maximizing student learning and confidence in applied ML courses (Niemirowski, 2023; Sundberg & Holmström, 2024).

This article reflects on selected applied ML projects from engineering courses, exploring how end-user tools impacted the project's scope and difficulty for the instructor and the students. The background literature will outline the increasing need for ML education in non-computer science programs; the current challenges in teaching ML to non-CS students; the development of current end-user ML tools

that enable a new population of ML tinkerers; and abstraction hierarchy models. Several sample classroom ML projects will be described, modeled using abstraction hierarchies, and commented on for the implications, limitations, contributions, and recommendations for future studies.

2. Materials and Methods

2.1 End-User Tools for Machine Learning Applications

The advance of machine learning has expanded hardware and software tools that support practical ML applications (Huyen, 2020; Giraldo & Laso, 2025). Of note are end-user tools designed to lower the technical barrier to developing ML projects. Low-code and no-code AI now enable users to build ML systems with limited programming experience (Sundberg & Holmström, 2024). For example, Edge Impulse provides simplified visual interfaces and an MLOps wizard for data processing, model creation, and training, with minimal computer science knowledge required for model creation (Hymel et al., 2022).

These innovations have broadened access to ML in numerous fields (Aggarwal et al., 2022; Jordan & Mitchell, 2015; Khalajzadeh et al., 2018; Tai, 2020). Domain experts can now develop ML projects without the extensive technical expertise once required, further accelerating the adoption of MLOps in diverse disciplines (Sundberg & Holmström, 2023). Engineering is one such field, applying ML in areas such as computer vision, predictive maintenance, and control systems. This rapid development of ML necessitates equally rapid developments in ML education.

2.2. The State of Machine Learning Education

Current ML courses can generally fall into two categories: ML fluency courses aim to educate students on the theory and mechanisms of ML technologies for the purpose of application, research, and development, while ML literacy focuses on providing context and fundamental knowledge so that students can participate in conversations surrounding the technology (Long & Magerko, 2020). ML literacy courses also equip students for the application of developed AI platforms, such as generative AI (Neumann et al., 2024; Shailendra et al., 2024). While ML literacy courses are common across K-12 and higher education, ML fluency courses tend to be restricted to computer science education.

It has been observed that the teaching experience for a computer science machine learning course is distinctive from teaching non-CS students. Machine learning content will often focus on theory, demonstrating the technology on benchmark datasets to demonstrate how machine learning functions. These courses can be offered as advanced courses and electives, with a foundation of mathematics and programming pre-requisites completed before accessing them. As such, pedagogical content knowledge in these traditional machine learning courses can assume a familiarity with programming and statistics before exploring machine learning content.

In contrast, non-CS students approach machine learning with a far less robust background. To be approachable, many non-CS courses must either cover the requisite math and programming or deliver content on machine learning that does not require these skills. Significant pre-requisite content will create a steep learning curve for students to climb before they can engage in machine learning content, which also limits the scope of the ML content that the course may cover. A common method of translating CS content to non-CS students is to produce a lightweight version of the same content (Chilana et al., 2015). However, this may result in learning outcomes where students are only familiar with the content and are unable to develop or apply it. This is counterintuitive, as non-CS students are not expecting to develop deep theoretical knowledge of ML, but hands-on skills and experience that can be applied to their respective fields (Wilson & DeJong, 2024).

Lao (2020) explored methods to provide ML education to "tinkerers", recognizing that ML tools are increasing accessibility to the technology. Lao (2020) noted three audiences of ML education: high-end technical users, general adults seeking to implement ML in their work, and K-12 students. Technical ML courses assume an existing background in computer science with a high barrier of entry, requiring CS competence, statistics, calculus, and linear algebra. General adult courses are far more accessible but do not provide sufficient depth for implementation. Instead, many of these courses focus on providing a context for conversations with others who will implement ML. Finally, K-12 courses provide smaller activities and projects that develop skills related to model implementation but are simple and abstract enough to bypass math and programming requirements.

Though resources exist for introducing non-CS majors to ML, these workshops and programs are often fleeting glimpses into the field and technology. These benchmark applications offer students opportunities to implement ML models in curated sandboxes and prepared datasets, such as MNIST (Deng, 2012; Wagstaff, 2012). Students are often not required to perform data analysis or processing to implement their models, and evaluation may involve a cursory analysis of training and testing metrics. This

focused approach is excellent for teaching theoretical frameworks for ML and model design. While CS students will often continue in courses that provide data science methods and more in-depth projects, non-CS students may only receive one course and thus require more cohesive coverage. Course content needs to include the context and challenges of ML implementation in industrial settings where small, noisy, and incomplete data sets must be processed to identify underlying patterns and determine whether ML methods are applicable.

A review of papers covering machine learning courses for engineers were evaluated (Dixit et al., 2018; Kim, 2022; Muqri et al., 2022; Zhang, 2020). These courses demonstrate a preference toward teaching ML methods with benchmark applications. Datasets are preprocessed, and model development is simplified to a unique ML sandbox that is not representative of commercial tools or real-world applications. Dixit et al. (2018) showed a step away from this approach, incorporating pre-processing and feature generation in their learning modules.

Focusing on theory and benchmark applications is not a problem unique to engineering students. Academic research sees a similar gap in ML knowledge for computer science; Cadavid et al. (2020) found that half of their investigated papers reached implementation of a model. Likewise, Wagstaff (2012) found that only 1 of 148 ML algorithm papers presented at ICML 2011 interpreted their results in the context of the problem's domain.

Additional concerns were discovered within the literature. One concern of machine learning education is the development of responsible knowledge and skillsets (Weerts & Pechenizkiy, 2022). Ethical concerns include overcoming the common perception that ML is a one-size-fits-all solution to any problem, given sufficient model complexity (Narayanan, 2019). Students should develop an awareness for the scope and limitations of ML, not simply the tools required to apply them (Smith et al., 2022).

Sulmont et al. (2019) explored the methods and content knowledge required to teach ML to non-CS majors classifying learning outcomes from surveyed instructors into the Structure of Observed Learning Outcome (SOLO) taxonomy (Biggs & Collis, 2014), where learning outcomes are categorized by the student's expected stage of learning. Sulmont et al. (2019) also noted three preconceptions that non-CS students had for machine learning. First, that ML is important and that students are motivated to develop their understanding. Second, students have mostly heard of ML technology through popular media sources, which can risk warped perspectives of the technology and require that instructors debunk myths perpetuated by sensational sources. Third, that ML implementation is inaccessible to them, and that the lack of a solid math and programming background prevents them from utilizing ML technology.

2.3. Abstraction Hierarchy Model

Abstraction hierarchy models are used in human factors to describe complex tasks or processes, particularly when they involve human-machine interaction (Rasmussen, 1985). In an abstraction hierarchy, the purpose and functions of a process are classified into five layers of decreasing abstraction. This begins with the general purpose – the primary objective(s) of the system. Abstract functions are the constituent tasks that accomplish the general purpose(s), and general functions are the further abstracted tasks that are connected to each abstract function. This continues downward into physical functions and physical forms, the tangible steps and components of the system. Each element is connected through means-ends links to other elements. The "why" of an element is answered by the elements above it, while the "how" of an element is answered by the elements below it. These elements can be further decomposed on a secondary axis. Part-whole relations are a common example, decomposing from the total system to individual components.

While initially developed for supervisory control applications, abstraction hierarchy models have also been extensively applied in other domains to outline and assess complex systems. It has been used to structure library resources (Jacob et al., 2006), to examine road transport systems (Salmon et al., 2019), and to organize classroom content (Vongalis-Macrow, 2010). Abstraction hierarchy modeling was selected for this study due to its versatility and ability to categorize functions, tasks, and resources in a form that identifies why and how a function is achieved.

2.4. Model Development

To visualize how end-user ML tools assist ML education in non-CS courses, course content from two engineering courses was reviewed by the authors. Abstraction hierarchy models were developed by the course instructor through direct observation of instructional time allocated to each physical function across the quarter, informed by comparison against the instructor's experience teaching equivalent content without the selected end-user tools. A generalized form of the abstraction hierarchy model is presented in Figure 1. This general design encompasses every step from initial data collection to model deployment and monitoring. The primary objectives of the project are defined in the general purpose. The objective is decomposed into three broad categories

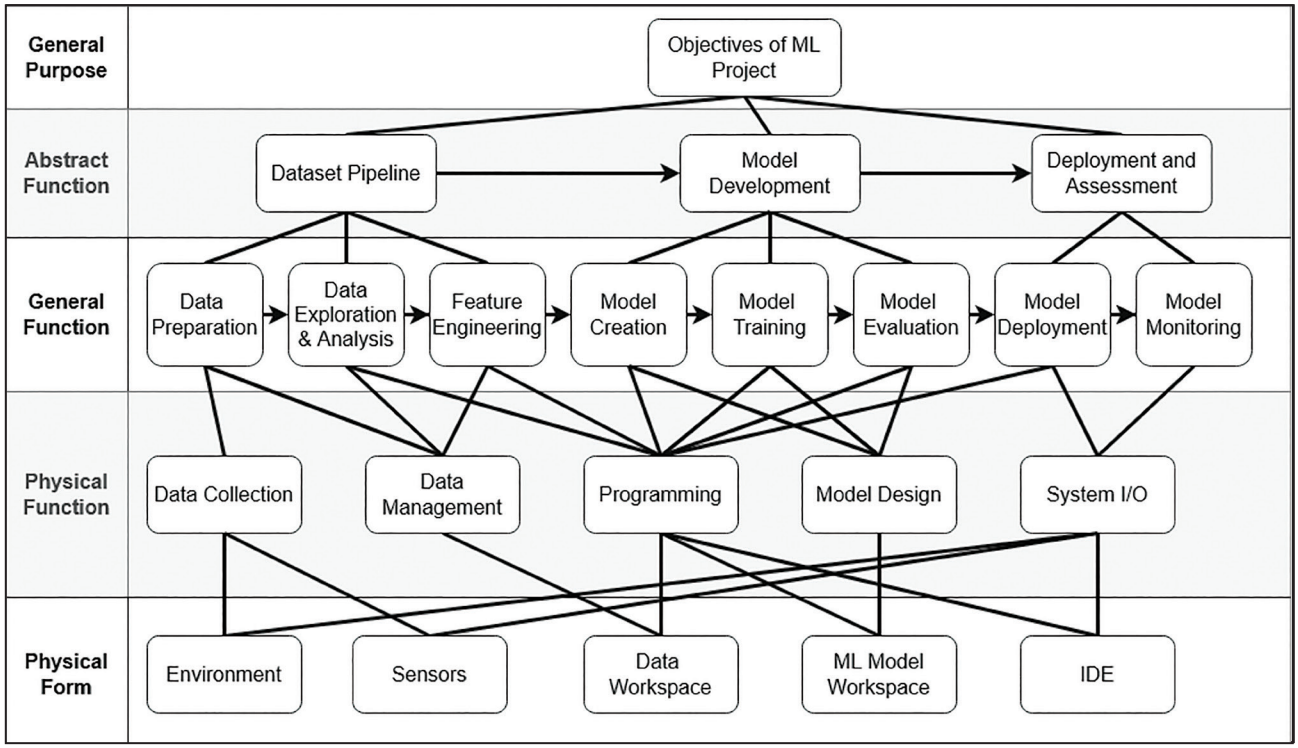

Figure 1. Abstraction hierarchy of a generic ML project.

and nine general functions that are linked to MLOps terms provided by the literature (John et al., 2021; Kreuzberger et al., 2023; Wazir et al., 2023). Physical functions are identified as skillsets that users are expected to develop and perform to accomplish the respective functions. Each of these skillsets is supported and/or enabled by selected tools and resources, identified under physical forms.

A component was marked as streamlined when the instructor observed a reduction in instructional time devoted to that function. For example, an ML model software reduces the time spent learning how to create, train, and test models, and allows for additional time to be spent on best practices in model creation and development.

2.5. Handwriting Classification Project: Our Baseline ML Project

MNIST is a dataset that comprises 70,000 images of handwritten digits from 0 to 9 (Deng, 2012). Each sample is a black and white image normalized into a 28x28 pixel space. The dataset is frequently used for an ML tinkerer's first computer vision project by classifying each digit. A similar dataset is Fashion MNIST, containing images of various clothing articles (Xiao et al., 2017). This dataset is well-explored, relatively small, and requires little to no processing. These factors allow a tinkerer to focus on understanding the fundamentals and mechanics of the ML implementation, and as such demonstrates the quintessential ML project.

A traditional introductory ML course may instruct students to create a neural network model in a selected programming language. Such a project may take a day to develop with ML-focused libraries such as TensorFlow, or weeks if students are expected to develop their model "from scratch." Conversely, MNIST-based projects are commonly used in introductory workshops as an initial demonstration of an ML tool's functionality and are often brief enough to perform multiple times within a single workshop. This dataset also serves as a platform from which to evaluate more advanced topics. Student projects can apply and compare various ML algorithms to assess their performance. Noise can also be added to the data when teaching data processing techniques (Snehith et al., 2023).

2.6. Predictive Maintenance Project

One of the engineering technology courses explored machine learning applications in predictive maintenance. Most of the course was spent on a dataset of simulated turbofan operation and failure (Saxena et al., 2008). This dataset was selected due to the need to filter, normalize, and process the data before use. Students were tasked with applying various data processing methods and ML algorithms to the dataset. Google Colaboratory was used as a Python

environment. The platform includes numerous modules, like Pandas, TensorFlow, scikit-learn, for processing the data and managing ML artifacts (Abadi et al., 2016; McKinney, 2010; Pedregosa et al., 2011).

The predictive maintenance course was selected for this study due to the emphasis on dataset generation and feature extraction over ML model development. Students were expected to perform multiple rounds of ML training and testing to compare data processing methods. The data processing initially only required a conversion of data to arrays that could be utilized by the regression models, but would culminate in creating time-dependent samples for use with recurrent neural network models.

2.7. Image Classification Project

Another course introduced engineering technology students to various machine learning applications, with a focus on computer vision. Students were supplied with OpenMV camera microcontrollers and access to Edge Impulse, a cloud-based ML development platform (Abdelkader et al., 2017; Hymel et al., 2022). A common project was to develop an image classification program that could distinguish an individual's face from others. To complete this project, students collected sample images, which were compiled into a single dataset that was distributed and uploaded to Edge Impulse. There, students explored different designs of convolutional neural networks and transfer learning models to implement and evaluate through training, testing, and implementation on the OpenMV platform.

This project was selected for evaluation as the instructor observed that the chosen end-user tools enabled a course project that spanned the full pipeline from data collection to deployment. OpenMV and Edge Impulse enabled the creation and management of a dataset, model, and initial deployment on the OpenMV platform, all of which required little to no programming to complete the tasks. This allowed students to focus on the quality of the dataset they were creating as well as testing and evaluating their project with a real-world implementation, providing an opportunity to understand various challenges within the scope of the project. Students engaged in various challenges in dataset creation: defining the scope of the project and data to collect; identifying valuable features within their data; as well as reducing various types of bias in their data collection.

3. Results

The following abstraction hierarchy models are produced from observations of the three ML projects described above. The following examples will provide specific combinations of project tasks and tools. Elements that were positively affected by the incorporation of selected ML tools are highlighted in green within each model.

Figure 2 outlines the abstraction hierarchy model for a simple ML project to classify handwritten digits. There is no initial data collection or real-world implementation observed with most example projects; the objective of such a project is to create and observe the function of ML models against a well-known and refined dataset. As such, the general purpose and abstract function are reduced to model development and a limited pipeline for preparing the dataset.

First, features must be identified and extracted from the dataset; in this example the features are simply the normalized pixel values of each image sample, often stored as a single array. These features are then passed to an ML model for training and testing. Introductory MNIST projects often do not include any significant modification of the data before use in the selected ML model, which further reduces the scope of the project. Such data augmentation could be included as a third general function, as will be shown in later examples of ML projects. Finally, these physical functions are supported by three physical forms: a workspace for data management; a similar workspace for ML model development; and an IDE to write code within. Because Google Colaboratory provided a workspace for the dataset, model design, and programming, it is treated as a single object within the hierarchy.

Note that this model does not highlight any elements in green to indicate positive effects from the selected tools. It is observed that many MNIST projects use one single workspace for all three components. This is also true of the predictive maintenance and computer vision projects discussed later in this section. The choice of workspace(s) is based on the objectives and expected skill level of the learner. An MNIST project may use a fundamental programming language without any additional tools and expect students to write the code for their model and training algorithms independently. Alternatively, a no-code ML tool may function as the entire physical form and even automatically complete the physical functions for the user.

As outlined in the methodology, this project is intended to demonstrate a typical baseline ML project. It will be demonstrated in later examples that the inclusion of select ML tools can streamline this core development process and provide room to focus on the initial design and final deployment of the model.

Figure 3 displays the abstraction hierarchy for the Predictive Maintenance projects. The abstract function now includes additional dataset exploration, analysis, and feature engineering. This is due to the

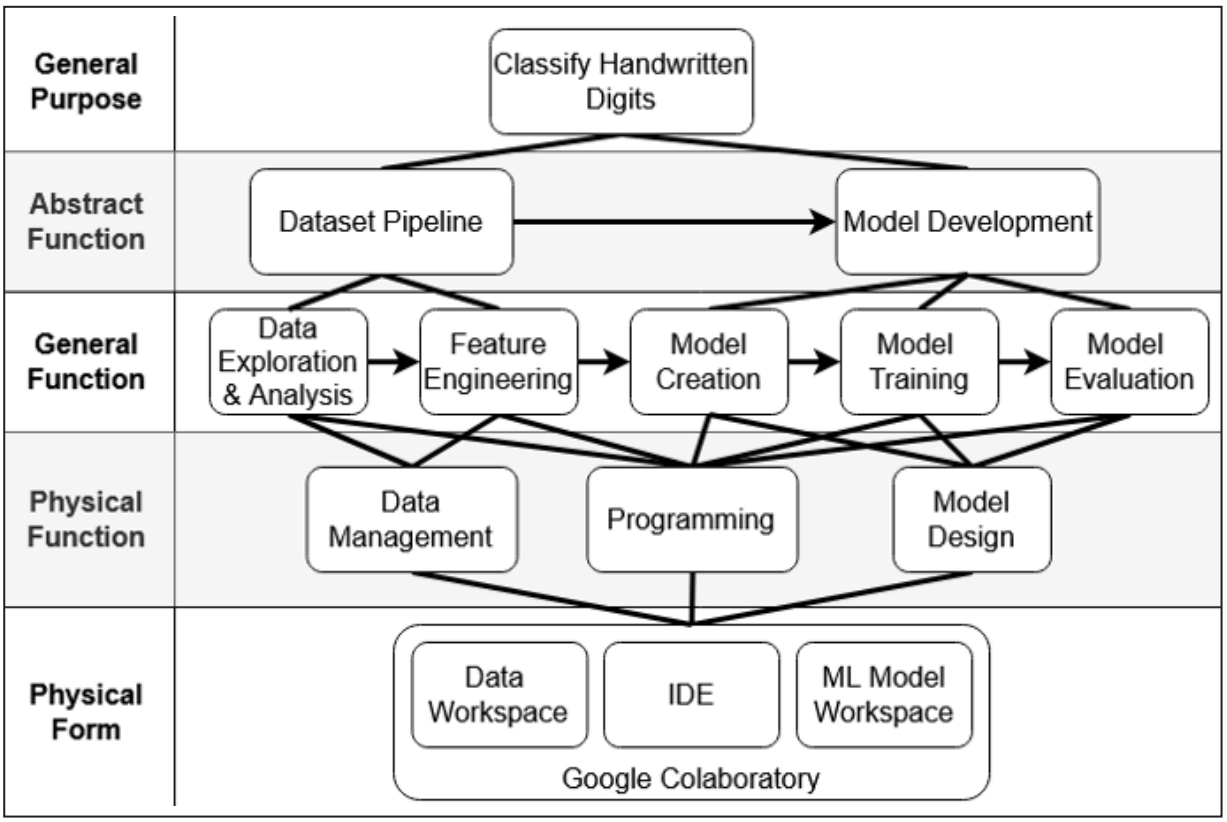

Figure 2. Abstraction hierarchy of a handwriting classification project.

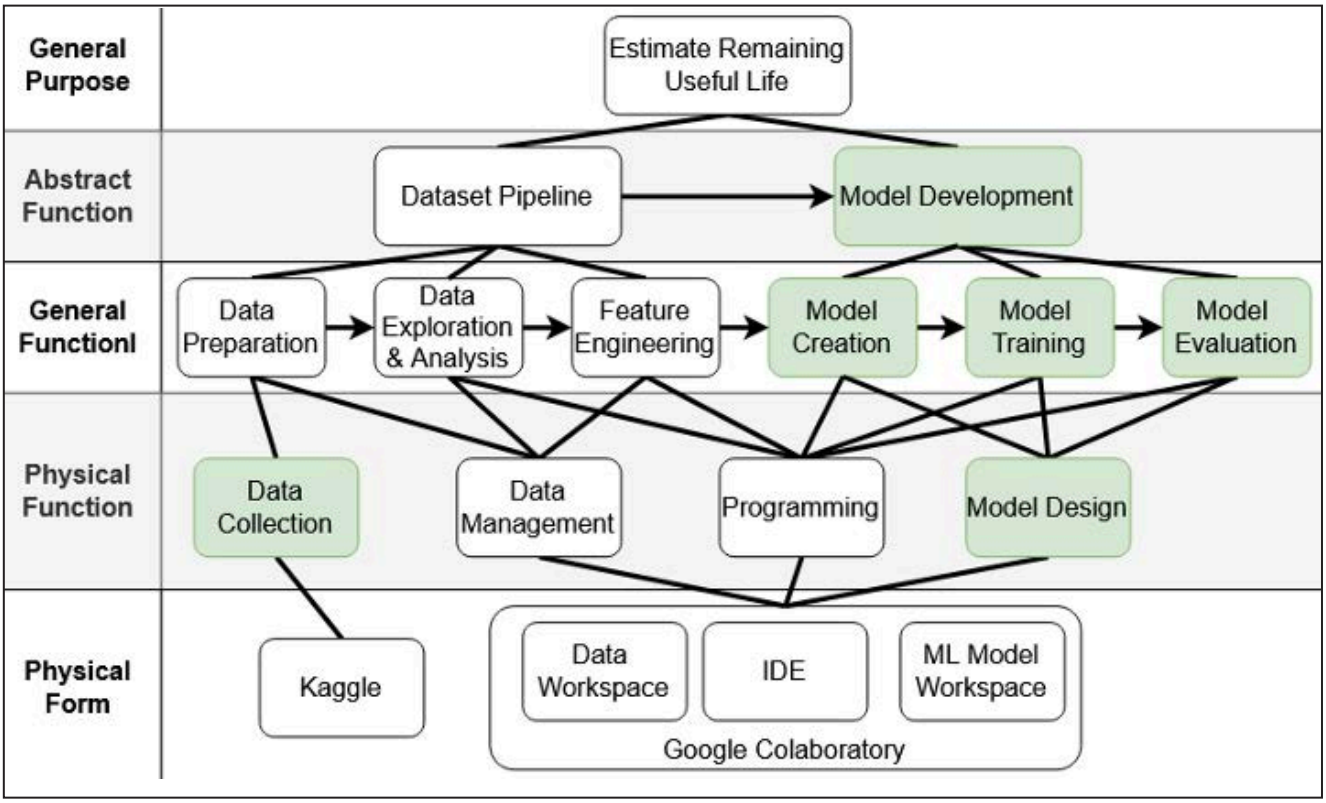

Figure 3. Abstraction hierarchy of a predictive maintenance project.

need for initial data processing and filtering before the data can be effectively used by a ML model; such data processing has been added as a general function to the project. The increased dataset pipeline was a primary focus of the course project. The NASA CMAPPS dataset was acquired through Kaggle, which satisfied the requirement for data collection within the project.

It was observed throughout the course that the development of the ML models was somewhat streamlined by the selection of Python libraries. The creation, training, and testing of ML algorithms was enabled using the scikit-learn and TensorFlow modules. Pandas enabled students to perform their data processing techniques but did not significantly reduce the technical skills required to do so. As such, the data processing and management were not identified as streamlined components. After three weeks of introductory topics on Python programming and ML, students were able to employ new data processing techniques and ML algorithms within a single class and were able to run multiple iterations to assess different combinations for each assignment. This was particularly useful within the final project, where students were instructed to create three unique models and compare their performance.

Figure 4 shows the computer vision project, which expands the project scope to include model deployment and review. Dataset generation is now produced through both data collection and data management, both of which were streamlined by the selected tools. OpenMV enabled students to capture images that were stored in a format ready to upload to Edge Impulse, which then provided a workspace to evaluate the dataset at large and prune their data with various tools and visualization features.

Edge Impulse's ML workspace also allowed for the rapid creation, training, and testing of ML models. These models were also ready to deploy with sample implementation code back into the OpenMV platform, which provided students with a springboard to integrate their trained ML model into their system response. Combined, these streamlined components allowed for the creation of image datasets containing thousands of pictures in a single week (crowdsourced by students), and for the creation and implementation of their ML model in as much time. The only component of this project not streamlined by the ML tools was the system response.

4. Discussion

The utilization of end-user and no-code ML tools in curricula may reduce the need for pre-requisite knowledge and skills before students can engage ML content, potentially allowing non-CS majors to more readily approach ML technologies within a single course (Sulmont et al., 2019; Sundberg & Holmström, 2023). The importance of technical skills in programming also shifts under this framework. Courses may be able to cover the creation and train-

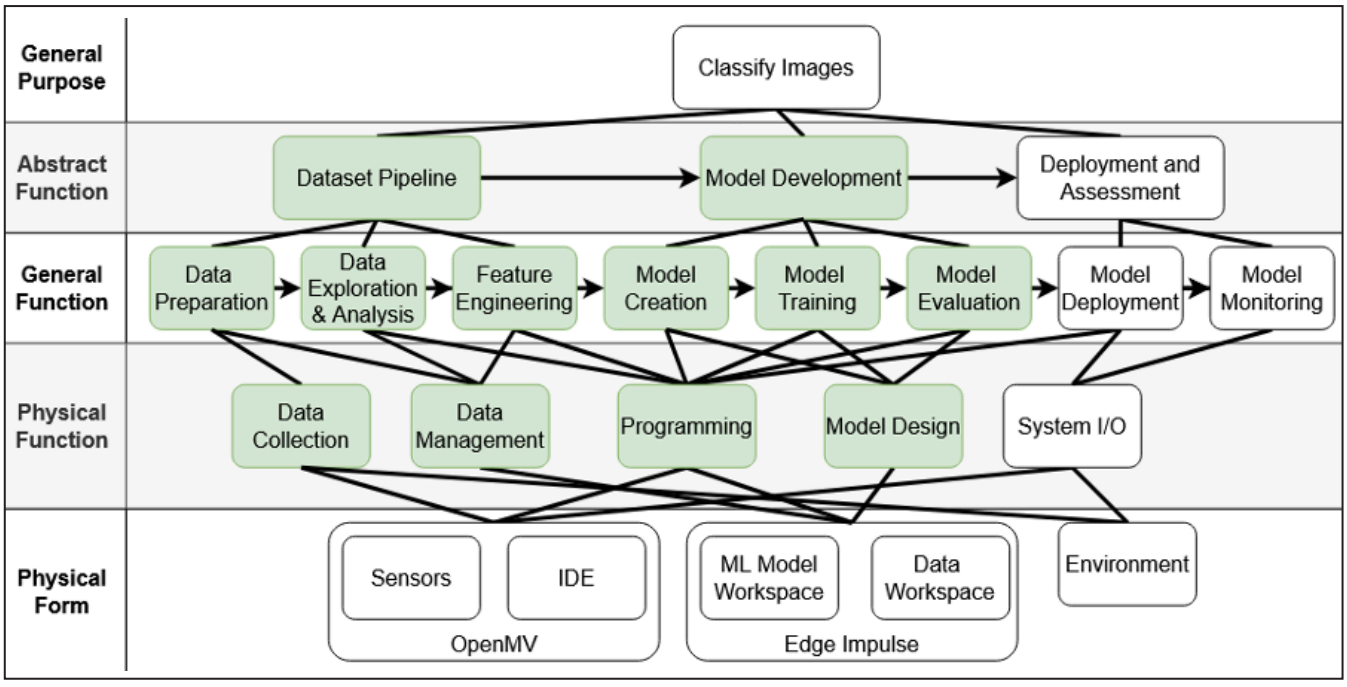

Figure 4. Abstraction hierarchy of a computer vision project.

ing of ML models more quickly, focusing instead on data science, model deployment, and high-level design considerations. As an exploratory study based on qualitative observation of three courses, these findings should be read as preliminary indications rather than established outcomes. It should be mentioned that the intent is not to bypass or skip fundamental course content. Without a solid foundation of math and computer science, students' abilities to explore and utilize machine learning will be limited. However, the depth of the required math and computer science content can be variable from course to course. A careful selection of ML tools can provide an accessible ML project that can be tailored to the time, effort, and resources available to the course.

4.1. Applications of Model

The purpose of this framework is to serve as an illustrative tool for instructors preparing ML projects within their course. They can identify which elements of MLOps should be included in the scope of the project. From here, they can identify the skillsets required of students to complete those tasks. These skillsets can be identified as pre-requisites for the project; topics to include in the course design; or elements to streamline through the selection of end-user tools. Student capabilities can then be assessed to determine where they would benefit from support, and tools selected and assessed for their degree of support. The framework can also provide reflections on the performance of a project and the tools selected.

4.2. Limitations and Future Studies

While this paper serves to highlight the need for new educational material, more insight is required to effectively assess and design courses with end-user ML tools in mind. Each abstraction hierarchy is a qualitative assessment of a limited selection of applied ML courses, developed as an illustrative rather than empirically validated tool. It is expected that this framework may be transferable to ML curricula in other contexts, but further validation is necessary. Future studies would expand this work to assess additional courses across K-12 and higher education. Furthermore, insight would be sought from ML instructors in various backgrounds to validate the performance and flexibility of the model.

The three classroom projects examined in this study varied considerably in tools and scope, from an introductory benchmark to a multi-week predictive maintenance analysis and full-pipeline image classification. The variety limits the extent to which any single hierarchy can be assumed to generalize to any specific ML project. However, a common structural pattern was observed within each project. End-user tools reduced instructional time devoted to core model creation, training, and testing; that freed time was redirected toward the broader functions of data management and real-world deployment and evaluation. Instructors adapting the framework to a new project should expect similar categories of end-user tools to affect similar general functions, while validating that expectation against their own course context.

In addition, the model would benefit from a quantitative element to measure the amount of time and effort that is originally required in each component, or how much is saved by the inclusion of a tool. Future studies would identify measurable educational outcomes and assessment tools to better quantify the impact of ML tools on ML projects, the requisite course material, and instructor/student time and effort within the classroom.

Generative AI tools are a separate and deliberate consideration in project or course design. Its integration into applied ML coursework is a new and rapidly evolving matter, substantial enough to warrant dedicated study beyond the scope of this paper (Yelamarthi et al., 2024). Future work could directly examine how generative AI tools affect the instructional time and skillsets required across the hierarchy.

4.3. Contributions

The primary contribution of this study is to provide a pilot study on the impact of end-user ML tools on applied ML project design. By outlining course projects with the abstraction hierarchy model, an instructor may identify elements that can be pared down or require additional scaffolding, as well as achieve a larger scope that can provide much needed real-world context to their course projects (Webster & Turner, 2024).

The need for ML literacy continues to grow as industries adopt new ML technologies into their operations. The abstraction hierarchy models presented here emphasize how end-user tools can broaden the scope of ML instruction, enabling educators to expand classroom content beyond core concepts in ML and engage learners at higher levels of understanding. This also provides time to address important issues such as the ethical and responsible application of ML. Such skills are currently lacking within ML education content and will prove crucial as ML adoption continues to rapidly expand.

Declaration of Interest

The authors declare that they have no competing interests.

References

Abadi, Martín, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. 2016. "TensorFlow: A System for Large-Scale Machine Learning." In *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation*, 265–83. Berkeley, CA: USENIX Association.

Abdelkader, Ibrahim, Youssef El-Sonbaty, and Mohamed El-Habrouk. 2017. "OpenMV: A Python Powered, Extensible Machine Vision Camera." arXiv. https://doi.org/10.48550/ARXIV.1711.10464.

Aggarwal, K., Mamoun M. Mijwil, Aws-Hasan Al-Mistarehi, Sinan Alomari, Müjde Gök, Anas M. Z. Alaabdin, Souzan Haydar Abdulrhman, et al. 2022. "Has the Future Started? The Current Growth of Artificial Intelligence, Machine Learning, and Deep Learning." *Iraqi Journal for Computer Science and Mathematics* 3, no. 1: 115–23.

Ali, Syed Hammad, Hussam A. Al-Sultan, and Mahmood T. Al Rubaie. 2022. "Fifth Industrial Revolution: New Perspectives." *International Journal of Business, Management and Economics* 3, no. 3: 196–212.

Biggs, John B., and Kevin F. Collis. 2014. *Evaluating the Quality of Learning: The SOLO Taxonomy (Structure of the Observed Learning Outcome)*. New York: Academic Press.

Cadavid, Juan Pablo Usuga, Samir Lamouri, Bernard Grabot, Robert Pellerin, and André Fortin. 2020. "Machine Learning Applied in Production Planning and Control: A State-of-the-Art in the Era of Industry 4.0." *Journal of Intelligent Manufacturing* 31, no. 6: 1531–58. https://doi.org/10.1007/s10845-019-01531-7.

Chilana, Parmit K., Caroline Alcock, Shruti Dembla, Anne Ho, Amy Hurst, Brice Armstrong, and Philip J. Guo. 2015. "Perceptions of Non-CS Majors in Intro Programming: The Rise of the Conversational Programmer." In *2015 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, 251–59. Washington, DC: IEEE.

Deng, Li. 2012. "The MNIST Database of Handwritten Digit Images for Machine Learning Research [Best of the Web]." *IEEE Signal Processing Magazine* 29, no. 6: 141–42. https://doi.org/10.1109/MSP.2012.2211477.

Dixit, Ayan, Uday Shankar Shanthamallu, Andreas Spanias, Sathwik Rao, Sandeep Katoch, Mahesh K. Banavar, Gautam Muniraju, et al. 2018. "Multidisciplinary Modules on Sensors and Machine Learning." In *Proceedings of the 125th ASEE Annual Conference and Exposition*. Washington, DC: American Society for Engineering Education.

Giraldo, Luis, & Sergio Laso. 2025. "Democratizing Machine Learning: A Practical Comparison of Low-Code and No-Code Platforms." *Machine Learning and Knowledge Extraction*, 7(4), 141. https://doi.org/10.3390/make7040141

Huyen, Chip. 2020. "Machine Learning Tools Landscape v2 (+84 New Tools)." December 30, 2020. https://huyenchip.com/2020/12/30/mlops-v2.html.

Hymel, Shawn, Colby Banbury, Daniel Situnayake, Alex Elium, Cart Ward, Matthew Kelcey, Mathijs Baaijens, et al. 2022. "Edge Impulse: An MLOps Platform for Tiny Machine Learning." arXiv. https://doi.org/10.48550/ARXIV.2212.03332.

Jacob, Elin K., Hanne Albrechtsen, and Nayana George. 2006. "Empirical Analysis and Evaluation of a Metadata Scheme for Representing Pedagogical Resources in a Digital Library for Educators." *Knowledge Organization for a Global Learning Society*. 205 – 212.

John, Mercy Malu, Helena Holmström Olsson, and Jan Bosch. 2021. "Towards MLOps: A Framework and Maturity Model." In 2021 *47th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, 1–8. Washington, DC: IEEE.

Jordan, Michael I., and Tom M. Mitchell. 2015. "Machine Learning: Trends, Perspectives, and Prospects." Science 349, no. 6245: 255–60. https://doi.org/10.1126/science.aaa8415.

Khalajzadeh, Hourieh, Mohamed Abdelrazek, John Grundy, John Hosking, and Qiang He. 2018. "A Survey of Current End-User Data Analytics Tool Support." In *Proceedings - 2018 IEEE International Congress on Big Data, BigData Congress 2018*, 41–48. Washington, DC: IEEE. https://doi.org/10.1109/BigDataCongress.2018.00013.

Kim, Hyunju. 2022. "Lessons Learned from Collaborative Initialization of Machine Learning Class and STEM Contest with University and

Industry Partnership." Paper presented at the 2022 ASEE Annual Conference & Exposition, Minneapolis, MN, June 2022.

Kreuzberger, Dominik, Niklas Kühl, and Sebastian Hirschl. 2023. "Machine Learning Operations (MLOps): Overview, Definition, and Architecture." *IEEE Access* 11: 31866–79.

Lao, Natalie. 2020. "Reorienting Machine Learning Education Towards Tinkerers and ML-Engaged Citizens." PhD diss., Massachusetts Institute of Technology. http://appinventor.mit.edu/assets/files/NatalieLao_PhD_Dissertation.pdf.

Long, Duri, and Brian Magerko. 2020. "What Is AI Literacy? Competencies and Design Considerations." In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, 1–16. New York: Association for Computing Machinery.

McKinney, Wes. 2010. "Data Structures for Statistical Computing in Python." In *Proceedings of the 9th Python in Science Conference*, edited by Stéfan van der Walt and Jarrod Millman, 51–56. Austin, TX: SciPy.

Muqri, Mustafa, Sheila Boghikian-Whitby, Mushtaq Muqri, Zaid Muqri, and Saba Muqri. 2022. "Leveraging the Power of Python, Octave and Matlab for Machine Learning." Paper presented at the 2022 ASEE Annual Conference & Exposition, Minneapolis, MN, June 2022.

Narayanan, Arvind. 2019. "How to Recognize AI Snake Oil." Arthur Miller Lecture on Science and Ethics, Massachusetts Institute of Technology, Cambridge, MA, November 21, 2019.

Neumann, Alexander T., Yu Yin, Suleman Sowe, Stefan Decker, and Matthias Jarke. 2024. "An LLM-Driven Chatbot in Higher Education for Databases and Information Systems." *IEEE Transactions on Education*.

Niemirowski, Jonathan. 2023. "Threads, Buckets, and Impact: A Framework for Tool Accelerated Machine Learning Courses." PhD diss., Louisiana Tech University.

Pedota, Michele, and Lucia Piscitello. 2022. "A New Perspective on Technology-Driven Creativity Enhancement in the Fourth Industrial Revolution." *Creativity and Innovation Management* 31, no. 1: 109–22. https://doi.org/10.1111/caim.12468.

Pedregosa, Fabian, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, et al. 2011. "Scikit-Learn: Machine Learning in Python." *Journal of Machine Learning Research* 12: 2825–30.

Rasmussen, Jens. 1985. "The Role of Hierarchical Knowledge Representation in Decision Making and System Management." *IEEE Transactions on Systems, Man and Cybernetics* SMC-15, no. 2: 234–43. https://doi.org/10.1109/TSMC.1985.6313353.

Salmon, Paul M., Gemma J. M. Read, Nicholas Stevens, Guy H. Walker, Vanessa Beanland, Roderick McClure, Bryan Hughes, Ian R. Johnston, and Neville A. Stanton. 2019. "Using the Abstraction Hierarchy to Identify How the Purpose and Structure of Road Transport Systems Contributes to Road Trauma." *Transportation Research Interdisciplinary Perspectives* 3: 100067.

Saxena, Abhinav, Kai Goebel, Don Simon, and Neil Eklund. 2008. "Damage Propagation Modeling for Aircraft Engine Run-to-Failure Simulation." In *2008 International Conference on Prognostics and Health Management, PHM 2008.* Washington, DC: IEEE. https://doi.org/10.1109/PHM.2008.4711414.

Shailendra, Shalini, Ramhari Kadel, and Akhilesh Sharma. 2024. "Framework for Adoption of Generative Artificial Intelligence (GenAI) in Education." *IEEE Transactions on Education* 67, no. 5: 777–85.

Smith, J. J., Saleema Amershi, Solon Barocas, Hanna Wallach, and Jennifer Wortman Vaughan. 2022. "REAL ML: Recognizing, Exploring, and Articulating Limitations of Machine Learning Research." In *2022 ACM Conference on Fairness, Accountability, and Transparency*, 587–97. New York: Association for Computing Machinery. https://doi.org/10.1145/3531146.3533122.

Snehith, N., M. T. N. D. S. Harsha, Srilatha Bodempudi, L. R. L. Bailundo, S. Shameem, and J. V. Namgiri. 2023. "Reconstructing Noised Images of Fashion-MNIST Dataset Using Autoencoders." In *2023 International Conference on Ambient Intelligence, Knowledge Informatics and Industrial Electronics (AIKIIE)*, 1–6. Washington, DC: IEEE. https://doi.org/10.1109/AIKIIE60097.2023.10390180.

Sulmont, Elisa, Elizabeth Patitsas, and Jeremy R. Cooperstock. 2019. "What Is Hard about Teaching Machine Learning to Non-

Majors? Insights from Classifying Instructors' Learning Goals." *ACM Transactions on Computing Education* 19, no. 4. https://doi.org/10.1145/3336124.

Sundberg, Linus, and Jonny Holmström. 2023. "Democratizing Artificial Intelligence: How No-Code AI Can Leverage Machine Learning Operations." *Business Horizons 66*, no. 6: 777–88. https://doi.org/10.1016/j.bushor.2023.04.003.

Sundberg, Linus, & Jonny Holmström. 2024. "Teaching Tip: Using No-Code AI to Teach Machine Learning in Higher Education." *Journal of Information Systems Education* 35(1), 56-66. https://doi.org/10.62273/CYPL2902

Tai, Ming-Tsan. 2020. "The Impact of Artificial Intelligence on Human Society and Bioethics." *Tzu Chi Medical Journal* 32, no. 4: 339–43. https://doi.org/10.4103/tcmj.tcmj_71_20.

Vongalis-Macrow, Athena. 2010. "Reconceptualising Access in Education Policy: Method and Mindset." *Policy Futures in Education* 8, no. 5: 513–27.

Wagstaff, Kiri L. 2012. "Machine Learning That Matters." In *Proceedings of the 29th International Conference on Machine Learning*, 1851–56. Madison, WI: Omnipress. https://doi.org/10.48550/arXiv.1206.4656.

Wazir, Saddam, Gaurav Saxena Kashyap, and Pushpanjali Saxena. 2023. "MLOps: A Review." arXiv preprint arXiv:2308.10908.

Webster, Rustin, and Matthew Turner. 2024. "Engineering Technology Capstone: Exploring the Current State." *Journal of Engineering Technology* 41(2), 34–39.

Weerts, Hilde J. P., and Mykola Pechenizkiy. 2022. "Teaching Responsible Machine Learning to Engineers." In *Proceedings of the Second Teaching Machine Learning and Artificial Intelligence Workshop*. https://github.com/hildeweerts/.

Wilson, D.M. and DeJong, B.P., 2024. "Why Students Select Engineering Technology Majors." *Journal of Engineering Technology,* 41(2), 42-53.

Xiao, Han, Kashif Rasul, and Roland Vollgraf. 2017. "Fashion-MNIST: A Novel Image Dataset for Benchmarking Machine Learning Algorithms." arXiv preprint arXiv:1708.07747.

Yelamarthi, Kumar, Raju Dandu, Mohan Rao, Venkata Prasanth Yanambaka, and Satish Mahajan. 2024. "Exploring the Potential of Generative AI in Shaping Engineering Education: Opportunities and Challenges." *Journal of Engineering Education Transformations* 37, no. 2: 439–45. https://doi.org/10.16920/jeet/2024/v37is2/24072

Zhang, Yajun. 2020. "Teach Machine Learning with Excel." Paper presented at the 2020 ASEE Virtual Annual Conference, Virtual Online, June 2020. https://doi.org/10.18260/1-2--35268.

Jonathan Niemirowski

Jonathan Niemirowski received his degrees from Louisiana Tech University. He holds a PhD in Engineering with the Engineering Education track and received an MS in Molecular Sciences and Nanotechnology and a BS in Nanosystems Engineering in 2018 and 2015, respectively. His research interests are course design frameworks for machine learning education within engineering and engineering technology. Dr. Niemirowski is a member of the International Society of Automation and the American Society for Engineering Education.

David Hall

David Hall earned his Ph.D. in mechanical engineering from Georgia Institute of Technology and his B.S. degree in mechanical engineering from Louisiana Tech University. His research has focused on engineering education, trenchless technology, fracture mechanics, and computer graphics. He is currently an Associate Professor of Mechanical Engineering at Louisiana Tech University. Dr. Hall is a member of the American Society for Engineering Education.

Krystal Cruse

Krystal Cruse received her PhD in Engineering within the track of Engineering Education at Louisiana Tech University in 2012. She received an MS in Mathematics, an MS in Mechanical Engineering, and a BS in Mechanical Engineering from Louisiana Tech University in 2012, 2010, and 2008, respectively. She works as an Assistant Professor in Mechanical Engineering at Louisiana Tech University in Ruston, Louisiana. Her research interests are student engagement and retention. Dr. Cruse is a member of the American Society for Engineering Education.

Mary Fendley

Mary Fendley holds degrees from Indiana University and Wright State University. She is currently an Associate Professor in Industrial Engineering at Louisiana Tech University, and has been teaching both in-class and virtually for the past 12 years. Her research interests are in the areas of cognitive engineering, modeling, decision analysis, training, and human factors.