Crafting an Undergraduate Course at the Intersection of Machine Learning and Software Engineering

Steve Chenoweth, Ph.D.
Computer Science & Software Engineering
Rose-Hulman Institute of Technology
Terra Haute, Indiana
Email: chenowet@rose-hulman.edu

Panagiotis K. Linos, Ph.D.
Computer Science & Software Engineering
Butler University
Indianapolis, Indiana
Email: linos@butler.edu

***Abstract*—This is a work-in-progress innovative practice short paper. It describes exploratory efforts to craft a new undergraduate course at the juncture of Machine Learning (ML) and Software Engineering (SE). Our motivation comes from recent industry practices and challenges to adopt and deploy ML as part of software development projects.**

We leverage reported industry experiences by ML practitioners, research findings from academia, as well as student feedback, in order to create a new undergraduate course where ML and SE cross paths. Toward this end, we conducted an empirical study -- student surveys and interviews of ML professionals. Questionnaires helped us measure students' interest in learning how to leverage SE topics to develop ML-based systems, and to gauge their expectations and concerns about the course. Our interviews with ML professionals helped us solicit and collect recommendations about the content of our course.

In summary, our paper presents qualitative results of an empirical study launched to help us craft a new course on ML and SE. We discuss a list of reported challenges from ML professionals and explain how we match them to related academic learning objectives and course content. We also describe how student feedback including their expectations and concerns helped us in this effort. We hope that such a course will not only prepare our students for the workforce but will provide a promising example for other institutions to follow.

Keywords—Machine Learning, Software Engineering

1. INTRODUCTION

During the past few years, the authors have been teaching an introductory course on ML. This course is offered to Computer Science and Software Engineering undergraduate students, and to other majors who wish to learn what ML is all about and why it is so important for their careers. More details about that course and the authors' experiences from teaching it can be found in [1]. While teaching this ML course, the authors have been following documented efforts and challenges from industry to adopt and deploy ML during production. As a recent example, such challenges are described in Paleyes, et al [2].

Artificial Intelligence (AI) systems are inherently more nebulous to develop and test because their specific behavior is less clear-cut. AI systems deployed in the "real world" are stirred-in with other systems – A client observes a need in the actions of existing enterprise systems, and this causes their request for some AI to improve those actions. Skills at building normal types of systems to serve some enterprise are needed for adding AI capabilities to those systems. We explore these new problems related to ML, via feedback we received from people working in this area.

The rest of this paper is organized as follows: In section 2, the key research questions are noted. Next, in section 3, we review the literature. Section 4 describes our efforts to solicit and collect data to help us craft a new course based on an empirical study. In section 5, we discuss the logistics of crafting our course. We conclude with section 6 summarizing the results of our study.

2. RESEARCH QUESTIONS

Some important areas of inquiry arise, including: *1) what skills are needed for ML systems building and production? and 2) how well do we prepare our students in dealing with such modern realistic challenges when they graduate and get a job?* As an early response to that, we can see some academic attempts to offer graduate courses on SE for ML (Christian Kastner and Eunsuk Kang from SEI/Carnegie Mellon University [3]). However, we couldn't find any undergraduate courses with content based in a traceable way on feedback from ML experts.

3. LITERATURE REVIEW

The application of ML to SE has been a research focus for some time, using ML to create or refine SE processes and tools. E.g., Du Zhang and Tsai, 2007 [4]. Our interest here is the reverse - building ML systems for other purposes, using sound SE principles.

The 2019 *Dotscience* report on the State of Development and Operations of AI Applications [5] surveyed 500 industry professionals about the state of these activities. They found challenges like duplicating work, rewriting models because a team member left, and justifying value because of the greater expense. Related to the last of these, the projects were taking "between 7 and 18 months to move ML and AI models from idea to production." The need for education and best practices is clear, to support this rapidly growing type of software engineering. Studies by *Dimensional Research* [6], *IDC* [7, 8], and *Algorithmia* [9] reported similar findings. Baier, et al study [10] was itself a literature review. Amershi, et al [11] investigated multiple projects at Microsoft.

In 2020 Paleyes, et al [2] summarize surveys of the deployment of ML systems. They describe the many skills required for successful ML projects. For example, Data Cleaning implies knowledge of what the data represents in a domain. Feature Engineering, on the other hand, requires understanding the model that the data will feed into, so that decisions can be made like converting and combining features for better performance by a particular algorithm. In addition, Paleyes, et al [2] introduce a table describing the ML deployment process in five stages. For each stage they identify considerations, issues and concerns cited in the papers they summarized. We used this list in analyzing the data for our study. Paleyes also recommends the survey method we use, for gathering data about Machine Learning deployment.

Serban and van der Blom [12] describe a trend toward adopting ML-specific SE techniques for developing these systems. An example is "Automated Model Deployment." Another is "Enforce Fairness and Privacy." They found some important practices still had a low adoption rate (in 2020), such as feature management and hyper-parameter optimization.

Kastner and Kang's 2020 paper [3] summarized their experience teaching a Software Engineering for ML course at Carnegie Mellon University. They note the prevalence of mostly informal reports on challenges to doing this in industry. They comment that "requirements, software architecture, and process seem rarely discussed in the academic literature when it comes to AI-enabled systems." In 2020 one of the authors of the present paper created a Software Engineering for ML course based on the same book by Hulten [13] that Kastner and Kang used. The latter describe the core problem as being that "data scientists may work with unversioned notebooks on static data sets and focus on optimizing model accuracy while ignoring scalability, robustness, update latency, and operating cost. Software engineers are trained to work with specifications and tend to focus on code, but may not be aware of the difficulties of working with data and unreliable models. They have a toolset for decision making, risk management, and quality assurance but it is not always obvious how to apply those to intelligent systems and their challenges." As examples they note that, "ML components are used for problems for which we cannot specify a solution…" And that, "ML components can have non-local and non-monotonic effects." Their methods of extracting issues in industry and applying them to their course was not made explicit. They did list colleagues and practitioners who reviewed the course content.

In addition, our references [14, 15, 16, 17, 18, 19] support the thesis that development processes for creating successful ML systems are significantly different from processes for normal business systems. ML tools used in industry have become more descriptive of process-related steps. Google's "Best practices for implementing machine learning on Google Cloud" includes a workflow for preprocessing datasets and for including ML into cloud-based systems [20]. However, such vendor guides to adding ML into commercial systems are tightly coupled to a vendor's specific services. It is not always clear from the guides how recommendations were formed, though some do describe the conditions under which they best hold. The underlying research is not shown.

Finally, in the field of teaching Artificial Intelligence generally, as with ML, the focus usually is on teaching what algorithms do. Much of the published research is about using AI to teach AI, e.g., "Teaching Introductory Artificial Intelligence with Pac-Man" [21]. Or, "A Robot Laboratory for Teaching Artificial Intelligence" [22]. Or, teaching AI using "agents" as a theme, a methodology pioneered by Russell and Norvig [23, 24]. Kastner and Kang's paper [3] is a step forward in trying to integrate AI generally with production systems, using software engineering principles.

4. EMPIRICAL STUDY

We launched an empirical study to collect qualitative data from both students and ML professionals. As a result, we gathered recommended topics and expected student skills for our planned SE for ML course. Collected responses are organized and described below in separate themes.

Students' Responses: We used an online questionnaire to gather data from former and current students who had taken our introductory ML courses. We wanted to gauge potential interest and expectations of students for our new proposed course. We sent the survey to 218 former students and 96 current students. We received 37 student responses.

Background: Students who responded were from Computer Science, Data Science, Mechanical Engineering, Mathematics and Software Engineering. The majority of these students had taken introductory courses on Machine Learning, Artificial Intelligence and Software Engineering. However, only a couple had taken advanced courses on ML or SE.

Interest: The majority of students indicated that there is a strong interest in offering such a course. For instance, one student states, "ML courses teach the theory of the machine learning algorithm well but do not necessarily cover implementation into software well." Another claims, "absolutely [I am interested in such a course], a big disconnect in these [ML] classes is that I am getting results and I am building cool models, but I don't know how to utilize, share, and deploy these models."

Expectations: When students were asked what they expected to learn from such a course they responded saying

- "To learn about applications of ML"
- "Gain a better understanding of implementation of ML systems into software"
- "To learn software engineering reference architectures, design patterns, and workflows centered around apps using ML"
- "A course that teaches me how to deploy and share my ML models, and other applications of my work."

Suggestions: Students provided some useful suggestions and ideas to help us craft such a course including "make a software engineering course a prerequisite to avoid reteaching the same information" and "Require SE and ML courses as prerequisites so that there is a strong base of knowledge before integrating the two topics".

Professionals' Responses: We used both an online survey and one-on-one interviews to collect data from 18 professional software engineers. Our goal here was twofold. First, to find out how difficult practitioners feel it is to deploy ML in the field and, second, if they have any recommendations about crafting a course that can help future graduates with using ML during software development.

Experience: Most of the professionals described themselves as Senior/Principal Machine Learning Engineers with 6-11+ years of experience at their current job, and having used ML during software development projects. We also asked practitioners if they had any formal training in ML or they were self-taught, and why they started learning about ML. Some selected responses include:

- "Formal training in graduate school classes"
- "It was covered in my AI courses, but much of what I've learned was self taught."
- "I had the benefit of some ML training in undergrad, and then taught myself from there"
- "Yes, I took a couple of courses in grad school on ML. I was interested in ML because of the idea that an algorithm can identify patterns in the data and work towards a desired outcome. Also that it could be used in a plethora of different domains. My interest in statistics and math brought me closer to ML"
- "Completed five courses at [some University] regarding machine learning. Currently taking additional ML classes in my Master's in Data Science/Machine Learning"
- "ML is a powerful technology used to optimize processes and decisions. I was excited about this concept and the idea of teaching computers to think for themselves without human supervision."

Software Development Process: The majority of the practitioners indicated that their organization follows an Agile software development/engineering paradigm with only one mentioning the Waterfall approach. Some responses include:

- "Depends on client. Currently using a Kanban or XP approach"
- "Agile, & Commitment Based Management"
- "Agile. Kind of all over the place, but the part I'm most involved in is SCRUM-based"
- "My team is small. We use a simple storyboard to organize and track tasks. We also have weekly ML meetings."

Challenges: When practitioners were asked if they have experienced any challenges during the development and deployment workflow of ML-based systems. Some responded saying:

- "They aren't easy to keep up-to-date as the data changes"
- "Yes, model convergence and performance with real world collected data"
- "Determining which parameters to use and train on is always a pain. Online model updating and

versioning is something I've seen as an issue, but not something that's come up on my projects”
- "The largest hurdle has been in the feasibility side of things. Most of the software we work on doesn't have to worry about the question – ‘Can this be done?’"
- “It's very important to choose a model that accurately describes the problem domain. Customers want to understand the reasoning of an automated system, so it's important to use simple, explainable models in those cases.”
- “ML doesn't follow normal software dev cycles. You don't have features to implement or nice clear work tickets to do. It tends to be experimental, and a good deal of trial and error. That can make it hard to work into normal software development processes.”

ML Training: Practitioners also explained how their organization trains developers in ML. Some responses include:
- “We take seminars and classes”
- “We don’t, it’s on the job training”
- “A combination of self-guided learning, training produced by in-house experts, training purchased from external vendors, and hands-on experience on projects”
- “Pairing with experts, internal training, bringing in external trainers”
- “Individual development plans”
- “Self-driven learning”

Some were unsure and others indicated that there is no ML training in their organization.

Expected Student Skills: They all endorse the idea of offering a supporting course and they recommend specific topics and skills they would expect to see from students taking it. Some responses are below.
- “[students should be] able to integrate ML with other systems”.
- “[students should understand] experimental design and product integration”.
- “[students should have] practical training in basic ML models and [additionally know] how to apply them in a realistic [simple/expandable] scenario”.

5. COURSE CONTENT

Based on our collected data, we have created a list of reported industry challenges, mostly taken from [2], and have made a first attempt to match them with related academic learning objectives, course content, tools, techniques, methods etc. (see Table I). We believe that our data supports responding to the ML industry challenges described in [2]. Our next steps include, among others, preparation of a course description, the syllabus, selection of a textbook and deciding on prerequisites. This leads toward more fully addressing our research questions, noted in section 2, above.

TABLE I: Matching industry challenges with academic learning objectives and course content

ML Industry Challenges [2]	**Learning Objectives**	**Course Topics & Activities**
Data discovery, curation, dispersion, labeling, visualization.	Students can manage big data.	Wrangle real data from industrial databases, applications and devices.
Model complexity and interpretation, managing resources.	Students can train and test models on realistic data.	Try different algorithms, hyperparameters, etc. on the same data and compare results.
Applying metrics, regulation and simulation to models.	Students can verify model predictions with users and for other considerations.	Interact with real users of the data to decide the value of ML results.
Operational support, deployment patterns, team dynamics, feedback and continuous delivery.	Students can deploy models in realistic environments with model updating.	Integrate an ML system with an existing system that is designed to enhance.
Biases, authorship, implications of decisions, user involvement, explainability, security challenges.	Students can manage security, trust, and ethics issues.	Establish related criteria ahead of time, then test the ML system against these.

6. CONCLUSION

Our experience from an effort to craft a new course at the intersection of Machine Learning and Software Engineering indicates that this will be a useful course for undergraduate students. We support our claim above, by means of qualitative results gathered from a survey conducted for interested students and software professionals. In addition, we considered a list of reported challenges from ML professionals and matched them with academic learning objectives and our course content. Moreover, we leveraged student feedback including their expectations and concerns while crafting our course.

REFERENCES

[1] Steve Chenoweth and Panagiotis K. Linos, "Qualitative Findings from an Online Course on Machine Learning". *2018 IEEE Frontiers in Education Conf. (FIE)*.
[2] *Challenges in Deploying Machine Learning: a Survey of Case Studies*, Andrei Paleyes, Raoul-Gabriel Urma and Neil D. Lawrence, NeurIPS 2020 Workshop: ML Retrospectives, Surveys & Meta-Analyses (ML-RSA).
[3] Teaching Software Engineering for AI-Enabled Systems, Christian Kastner and Eunsuk Kang, *Proceedings of 2020 IEEE/ACM 42nd Int. Conf. on Software Engineering: Software Engineering Education and Training (ICSESEET).*
[4] Du Zhang and Jeffrey J.P. Tsai, *Advances in Machine Learning Applications in Software Engineering*. Idea Group Publishing, 2007.
[5] Dotscience: The State of Development and Operations of AI Applications, 2019. Available at https://dotscience.com/assets/downloads/Dotscience_Survey-Report-2019.pdf
[6] Dimensional Research: "Artificial Intelligence and Machine Learning Projects Are Obstructed by Data Issues Global Survey of Data Scientists, AI Experts and Stakeholders." Alegion, May, 2019. Available at https://cdn2.hubspot.net/hubfs/3971219/Survey%20Assets%201905/Dimensional%20Research%20Machine%20Learning%20PPT%20Report%20FINAL.pdf
[7] Kyle Wiggers, "For 1 in 4 companies, half of all AI projects fail," *IDC* 8 July 2019. Available at https://venturebeat.com/2019/07/08/idc-for-1-in-4-companies-half-of-all-ai-projects-fail/
[8] *"*Investment in Artificial Intelligence Solutions Will Accelerate as Businesses Seek Insights, Efficiency, and Innovation, According to a New IDC Spending Guide." *IDC*, 30 August 2021. Available at https://www.idc.com/getdoc.jsp?containerId=prUS48191221
[9] Kyle Wiggers, "Algorithmia: 50% of companies spend between 8 and 90 days deploying a single AI model." *The Machine: Making Sense of AI.* Dec 11, 2019. Available at https://venturebeat.com/2019/12/11/algorithmia-50-of-companies-spend-upwards-of-three-months-deploying-a-single-ai-model/
[10] Lucas Baier, Fabian Jöhren, and Stefan Seebacher. "Challenges in the deployment and operation of machine learning in practice." *Twenty-Seventh European Conf. on Information Systems (ECIS2019),* Stockholm-Uppsala, Sweden, 2019.
[11] Saleema Amershi et. al., "Software Engineering for Machine Learning: A Case Study," *Microsoft Research*, accessed Oct 2021, https://docs.microsoft.com/en-us/azure/devops/learn/devops-at-microsoft/
[12] Alex Serban, Koen van der Blom, Holger Hoos and Joost Visser, "Adoption and Effects of Software Engineering Best Practices in Machine Learning," *ESEM '20*, October 8–9, 2020, Bari, Italy.
[13] Geoff Hulten, *Building Intelligent Systems: A Guide to Machine Learning Engineering.* Apress, 2019.
[14] Y. Simard et. al., "Machine Teaching--A New Paradigm for Building Machine Learning Systems Patrice", *Microsoft Research*, August 2017.
[15] Qianyu Guo, et. al., "An Empirical Study towards Characterizing Deep Learning Development and Deployment across Different Frameworks and Platforms," 2019 *34th IEEE/ACM Int. Conf. on Automated Software Engineering (ASE)*.
[16] Zhenpeng Chen, et. al., "An Empirical Study on Deployment Faults of Deep Learning Based Mobile Applications", *2021 IEEE/ACM 43rd Int. Conf. on Software Engineering (ICSE).*
[17] Luke Oakden-Rayner, et. al., "Hidden Stratification Causes Clinically Meaningful Failures in Machine Learning for Medical Imaging," *ACM CHIL '20*, April 2–4, 2020, Toronto, ON, Can.
[18] Luis Rivero et. al., *"*Deployment of a Machine Learning System for Predicting Lawsuits Against Power Companies: Lessons Learned from an Agile Testing Experience for Improving Software Quality," *BQS'20*, December 1–4, 2020, São Luís, Brazil. ACM ISBN 978-1-4503-8923-5/20/12.
[19] Tushar Chandra, "Regressing Towards Simpler Prediction Systems," *WSDM'15*, February 2–6, 2015, Shanghai, China.
[20] "Best practices for implementing machine learning on Google Cloud." https://cloud.google.com/architecture/ml-on-gcp-best-practices
[21] John DeNero and Dan Klein, "Teaching Introductory Artificial Intelligence with Pac-Man." *AAAI*, 2010.
[22] M. Pantic, et al, "Teaching introductory artificial intelligence using a simple agent framework." *IEEE Transactions on Education*, Vol. 48, Issue 3, Aug. 2005.
[23] Deepak Kumar and Lisa Meeden, "A Robot Laboratory for Teaching Artificial Intelligence." *SIGCSE 98*.
[24] Stuart Russell and Peter Novig, *Artificial Intelligence: A Modern Approach*, 4th Ed., Pearson, 2020.