Software Engineering for Machine-Learning Applications

The Road Ahead

Foutse Khomh, Bram Adams, Jinghui Cheng, Marios Fokaefs, and Giuliano Antoniol

THE NEED AND desire for more automation and intelligence have led to breakthroughs in machine learning (ML) and artificial intelligence (AI), yet we still experience failures and shortcomings in the resulting software systems. The main reason is the shift in the development paradigm induced by ML and AI. Traditionally, software systems are constructed deductively, by writing down the rules that govern the system behaviors as program code. However, with ML techniques, these rules are inferred from training data (from which the requirements are generated inductively). This paradigm shift makes reasoning about the behavior of software systems with ML components difficult, resulting in software systems that are intrinsically challenging to test and verify.

Given the critical and increasing role of ML- and AI-based systems in our society, it's imperative for both the software engineering (SE) and ML communities to research and develop innovative approaches to address these challenges. In fact, the learned behavior of an ML-based system might be incorrect, even if the learning algorithm is implemented correctly, a situation in which traditional testing techniques are ineffective. A critical problem is how to effectively develop, test, and evolve such systems, given that they don't have (complete) specifications or even source code corresponding to some of their critical behaviors.

Motivated by these challenges, we organized the First Symposium on Software Engineering for Machine Learning Applications (SEMLA) at Polytechnique Montréal on 12 and 13 June 2018, with the kind support of Polytechnique Montréal’s Department of Computer Engineering and Software Engineering, the Institute for Data Valorization (IVADO), SAP, and Red Hat. The event attracted around 160 participants from all over the world, including students, academics, and industrial practitioners.

SEMLA’s main objective was to create a space in which SE and ML experts could come together to discuss challenges, new insights, and practical ideas regarding the engineering of ML- and AI-based systems. The program included talks and panels presented by renowned academic researchers and industrial practitioners, including keynote speakers David Parnas, Lionel Briand, and Yoshua Bengio. The full program is at http://semla.polymtl.ca.

Here, we summarize some key challenges these experts identified.

**System Accuracy**

The first topic concerned the accuracy of systems built using ML and AI models, and the responsibilities of engineers building them. For example, one keynote speaker mentioned three categories of AI research:

- building programs that imitate human behavior to better understand human thinking (used in psychology research),
- building programs that play games well (challenging and fun), and
• demonstrating that practical computerized products can use the same methods that humans use (risky and often naive).

He stressed that researchers should be very concerned about AI systems in the third category because they can't guarantee 100 percent accuracy or correct answers in all cases. He also raised concerns that people are using the Turing test to falsely claim intelligence in systems. He commented, “Turing did not claim that his test was a test for artificial intelligence!”

In response, a leading AI expert stated that AI’s goal is not to achieve 100 percent accuracy because

• humans are also far from 100 percent accuracy in their daily tasks, and
• AI technology’s strength comes from the ability to abstract up from different factors of variation between environments, to obtain models that can generalize and transfer to situations that weren’t encountered before.

He further explained that AI technologies’ main challenge is the curse of dimensionality—that is, the need for sufficient, labeled data to cover all important factors (features) of a given problem. AI, in fact, needs more training data than humans do!

Whereas the key properties of techniques such as deep learning (for example, compositionality, encoding into a simpler domain, and conditional computation) aim to reduce dimensionality’s impact, applications of AI still risk being limited to domains in which labeled data is cheap. Although labeled data is somehow abundant in some SE domains (such as defect prediction), other domains (such as requirement elicitation) are more challenging. Overall, AI’s full impact on SE is still unclear.

Because of AI and ML systems’ intrinsic imperfection, one panelist argued, only harmless AI technology or applications should be released to the public, since the responsibility of every engineer is to protect the public. He also mentioned that the public should be informed accurately of the AI technology it’s being exposed to. For example, instead of touting a “100 percent self-driving car,” automotive companies should advertise their products as “AI-assisted cars,” with a clear list of the ways in which AI is assisting.

Another panelist emphasized that AI isn’t a panacea. He illustrated how simple techniques could give the illusion of AI, or how the blind application of AI wouldn’t improve the workflow of workers. For example, in principle, an intelligent robot could easily replace a human worker to hand another worker the right tool for a given job, but not if the worker afterward throws the tool back on a pile. (The robot will have a hard time retrieving the right tool from an unordered pile.) However, using an intelligent robot to return tools in an ordered fashion (which is a different problem) could allow other robots later on to be deployed to hand over tools to workers. If a traditional computer science algorithm can solve a problem, we should just use that.

System Testing

The second hot topic our experts discussed was the difficulty of testing ML and AI systems. Our panelists debated whether we should tackle the testing of those systems the same way we do the testing of traditional systems, since an AI system’s behavior might be incorrect even if the learning algorithms are implemented correctly. One keynote speaker explained how in complex cyber-physical systems (CPSs), when no clear specifications of the intended systems exist (that is, humans have a lot of knowledge but can’t formalize it), only AI can approximate the system’s intended behavior by learning models from the available data.

This is a clear improvement over the manual design of models and controllers. However, it pushes most of the risk toward the trained models’ quality. So, how can we perform adequate quality assurance (QA) of AI models, given that the number of environments in which the models will be deployed is unlimited and that the human operator will require a detailed explanation of any failures?

Fortunately, we can use AI technology to reduce the search space of the environments to be tested, nudging QA techniques to those environments most likely to have failures or violate important safety constraints. Such an approach could even work in the system-of-systems context of CPSs, where each sensor and actuator must be validated not only in isolation but also in close integration with each other.

However, this QA doesn’t guard against hardware failure. So, hardware systems should incorporate fault-tolerance mechanisms to cope with such failures. One audience participant also observed that hardware could incorporate fault-tolerance mechanisms to mitigate the effect of AI model errors, improving AI systems’ robustness.

Another major challenge is that humans, once they’ve started trusting AI in their daily tasks, could
begin adapting their behavior to the AI assistance. For example, a study in Munich showed how assisted braking initially reduced the number of accidents, until drivers relied too much on the assisted braking and drove more aggressively. So, when is an AI-enabled product ready for release to the public? Although four million miles of test drives can’t prevent a serious accident in the next mile, how much information in the test drives can be used to debug and fix the corresponding fault?

An additional important question discussed at SEMLA is humans’ role in an AI-driven world. Are humans obsolete once AI technologies become mainstream? The panelists unanimously disagreed because humans are essential for putting the decisions of AI into context. Although the outcome (and potential failures) of the AI impacts the humans’ recommendations, those recommendations are also a human filter for AI failures. Further sociological research is necessary to study how AI technology affects human behavior.

Industrial Applications
SEMLA’s second day was devoted to industrial applications of AI. These industrial speakers discussed the current state of AI in industry and the challenges they face when applying AI models. They also discussed some open problems and what they consider to be their biggest needs and top priorities.

For example, a presenter from Google Brain pinpointed several programming-language issues involved in ML libraries, models, and frameworks. Different approaches in the current libraries have different advantages and disadvantages. Creating an efficient syntax for automatic differentiation that can deliver ease of implementation, performance, usability, and flexibility is important but difficult. Testing and debugging these implementations are also salient challenges. Industrial practitioners further mentioned that collaboration among experts from...
different fields is important for developing ML applications.

**Healing the Rift**

From these two days of intensive discussions, two key questions emerged:

- How should software development teams integrate the AI model lifecycle (training, testing, deploying, evolving, and so on) into their software process?
- What new roles, artifacts, and activities come into play, and how do they tie into existing agile or DevOps processes?

Answering these questions requires combined knowledge in SE and ML.

Unfortunately, a rift exists between these two communities, which we tried to understand through SEMLA. One reason for this rift is that stakeholders in the AI community focus on algorithms and their performance characteristics, whereas stakeholders in the SE community focus on implementing and deploying those algorithms.

So far, no real venue has integrated both fields, yet intersections exist between them, one of which is testing. The notion of coming up with ways to break a system is integral to ML, and the scale of test sets (thousands to millions of instances) is huge compared to the number of test cases in software systems. On the other hand, missing test cases or test cases that are known to fail (but are rare) are a larger issue for ML than for regular software systems.

We believe that the SE and ML communities should work together to solve the critical challenges of assuring the quality of AI and software systems in general. We have a lot to benefit from each other!