Improving Machine Learning Test and Evaluation with MLTE

featuring Alex Derr, Sebastián Echeverría, and Kate Maffey as Interviewed by Grace Lewis

Welcome to the SEI Podcast Series, a production of the Carnegie Mellon University Software Engineering Institute. The SEI is a federally funded research and development center sponsored by the U.S. Department of Defense. A transcript of today's podcast is posted on the SEI website at sei.cmu.edu/podcasts.

Grace Lewis: Hi, and welcome to the SEI Podcast Series. My name is Grace Lewis, and I am a principal researcher and lead of the SEI's Tactical and AI-Enabled Systems initiative. Today, I am joined by my colleagues in the Software Solutions Division, Alex Derr and Sebastián Echeverría, and by Kate Maffey, an Army data scientist, to discuss MLTE [Machine Learning Test and Evaluation], a new tool that provides a process and infrastructure for machine learning test and evaluation. Welcome to the three of you. Let's start off by having you tell us a little bit about yourself, the work that you do here at the SEI, and the coolest part of your job. Sebastian, why don't you start us off?

Sebastian Echeverria: Sure. Hello, I am Sebastian Echeverria. I come originally from Chile, South America, where I worked for several years in industry working with ML [machine learning] models and LIDAR systems to measure different types of loads on trucks. I came to CMU to get a master's degree in software engineering, and after that the opportunity appeared to

work at the SEI. That is how I got here. I have worked on different types of projects over the years, starting mostly on edge devices and different methods to help [people] working at the edge. Then, I moved mostly to IoT [Internet of Things] security also focused on the edge, but a bigger focus on how to secure different connections when you have hostile environments. Most recently, I have been working on software engineering for AI [artificial intelligence] on how to apply the rigor of software engineering practices to systems that have AI components. That includes projects like Portend where we are developing tools that help people developing models to detect data drift in their models early. That also includes MLTE, which is the project that we're going to be talking about in a bit.

Grace: What about the coolest part of your job?

Sebastian: Yes, the coolest part of my job I think is…There are many, but one that I always keep in mind is the fact that I am always learning because we have to use new technologies for the different projects that we have to complete. We need the newest technologies to actually be able to rise to the challenges that we are trying to deal with. The fact that we are always learning and always on the bleeding edge of many new technologies is one of the coolest things of my job.

Grace: That is indeed very cool. What about you, Alex, tell us a little bit about yourself.

Alex Derr: Yes. Hi, my name is Alex Derr. I have worked at the SEI for just over three years now. I am originally from Sioux Falls, South Dakota. I went to school at Dakota State University, which is in Madison, South Dakota. While I have been here, similar to Sebastian, I have been working on projects that focus on software engineering for AI, which has been both this MLTE project, but also a project called TEC, which was focused on developing a tool to collect information from different stakeholders and identify system mismatch in order to prevent long-term issues with machine learning problems. Another project I am working on, and I have worked on previously, is the Vessel project, which is focused on container reproducibility. I think my favorite part about working at the SEI is having a very diverse team. We all have different backgrounds and different expertise, so there is always someone to talk to when you have different kinds of issues. It is not like we are all working on the exact same thing, which makes the work really interesting.

Grace: It is a very diverse team, Alex, you are absolutely right. Now Kate, you

are an Army data scientist who previously worked at the AI2C, the Army AI Integration Center. You are also a cocreator and codeveloper of MLTE. Tell us a little bit about yourself, how this collaboration with the SEI has been, and what is also the coolest part of your job.

Kate Maffey: Yes, thank you, Grace. It is great to be here. Hi, everyone. I am Kate Maffey. As Grace said, I am a data scientist for the Army. And I have a little bit of an unconventional background academically. I started out studying Middle Eastern studies in undergrad and spent a couple of years working in Europe before moving to Pittsburgh in 2020 to do a master's degree at Carnegie Mellon University. It was actually during the master's degree that I started collaborating with SEI. I worked on a capstone project with a couple of peers, and our advisor, Christian Kästner, recommended that we bring someone named Grace Lewis into our project. That is when my collaboration started. Grace was kind enough to help advise us throughout our capstone. That capstone project is actually what turned into MLTE, which is the tool we are talking about here today. I have continued to collaborate with SEI over the last couple of years since that. I think that was 2021, and it has been great. The SEI is full of really knowledgeable people who have lots of experience across industry and academia. I think that has been an incredible part of working with all of them. The coolest part of my job is probably just that, frankly, it is getting to work with lots of brilliant and interesting people and really to have those interesting conversations where you explore a technology that is new and you get people's input, and you hear stories. I think that is really something that brings a lot of joy and fascination to my job.

Grace: Thank you very much. I am glad you are getting to learn a lot more about our people. But we are here to talk about the MLTE tool, which was just recently released. We will include a link to the tool in our transcript. Sebastian, can you tell us a bit more about why the tool was developed and some of the problems that the MLTE tool solves?

Sebastian: Sure. The starting point for the development of MLTE comes from the increase in the use of ML [machine learning]-enabled components in different types of systems. Since the development of the current generation of AI models is very new in comparison to other disciplines, what has been seen and we have also we have seen in different surveys is that there tends to be a mismatch between the expectations of the people developing these machine learning models and the people developing the actual system that is going to be using those models. It is common for some of these ML models

to work properly in isolation, but when they try to integrate it into the system and field them and deploy them and actually make them work in production, there are issues that people were not really expecting. The approach that we are taking with MLTE, which is both a process and a tool, is to try to help people when they are having this [trouble], to avoid having this mismatch, basically, and to help them out and give them some guidance and some help to try to avoid these pitfalls and to try to actually make it easier for ML-enabled components to work in integration with the final system that is going to be deployed. The overall way that we do this with MLTE is that the process helps you take a system-level perspective of where the model that you are developing is going to be. We start from the idea of quality attributes that the model and the component— the software component that is going to be using the model—has to achieve. The process has a whole discussion, a cyclic, iterative discussion, between the stakeholders and the model developers first. After that, there is a lot of iteration also in terms of how to actually ensure that the model is meeting these requirements once you have defined them. The process helps you guide that whole life cycle of development, and the tool helps you define tests that can actually validate that your model component is meeting those system-wide requirements or quality attributes. The goal of the tool is to make it easy to automate and to gather that information so that at the end you can see if your model components are good enough for system integration or if you need more iteration in terms of meeting those requirements.

Grace: Yes, that is great. Basically, what MLTE does is it really takes a system perspective to testing of models, where you are not just testing just the model, but you are testing it in the context in which it will be used. Kate, MLTE started as a capstone project like you said before. Obviously, this was proposed by you and your team because this is something that you were seeing in practice, that this practice of testing and evaluation of machine learning models did not exist. From your experience, what are some of the reasons why so many ML models fail in production?

Kate: Yes, there are a lot, as we all on this team know, and as I am sure many listeners know. But I think there are two that really stick out that I would like to highlight. The first is sort of what one of the things Sebastian alluded to, which is really not understanding the system requirements when you are creating a model. An example of this would be, someone is creating an image detection model, but they don't realize that the system it is going to be employed on is a handheld device, like a small camera, for instance. That

really affects how the model is going to be able to perform. If you go into it as a model developer, and you don't know what that larger system is going to be, you could really run into big problems that cause delays or just cause the project not to work, have to go back to the drawing board. The second one is lack of communication. This is really well catalogued in the literature, but it is also pretty easy to see in real life, as again, I am sure many people have experienced. That is that there is a lack of understanding either in terms of the specifics that a model needs to do, or there is just one person who has a lot of the operational knowledge, perhaps that person is a domain expert, and they don't spend a lot of time talking to the model developer, for instance, who actually has to write the code. This could lead to something like—again to go with the image detection example—maybe there is an image detection model, and it has only been tested and trained on blue sky, green grass background, and it needs to work in an environment where there is a desert, or it is snowy or some other biome difference. That might seem like an obvious thing to work towards, but if you are a model developer, and you haven't been having consistent communication with the people who know more about the use case, it is easy to see how that might break down and it might cause some issues in terms of having a system fail, and especially having a machine learning component that doesn't fit well into the system.

Grace: Yes, the communication is really a big problem because what happens then is that, as a data scientist that perhaps hasn't been trained in software engineering and quality attributes and designing and things like that, does what they know better, which is basically just to test for accuracy, right? Then you see all these problems because they're not looking at the system. Okay, so Sebastian told us on a very high level what MLTE is. Kate talked about some of the [maybe] motivation for why MLTE exists. Alex, can you tell us a little bit about how MLTE works and how it solves some of these problems that have been mentioned today?

Alex: Some of the issues Kate mentioned might sound simple on a surface level, but as you really get into solving them, there is a lot more there that needs to happen in order for these requirements to be elicited during the design time. The MLTE process, along with the negotiation card, are focused mainly on some of the things that she mentioned. Our research is focused around identifying things that need to be found and defined during design time. Then they can be discussed with the model developers and also the stakeholders in the project. This will be things, like she mentioned, there are things like how much CPU is available, how much memory is available, what kind of environment is this going to be used in? Because there are cases

where a stakeholder will come in and say, *I need a model that can detect X things in this image*, and the developer will make that. Then there are so many other things that need to be accounted for that don't get accounted for. MLTE focuses around structuring a discussion with a negotiation card that will help you to identify all of these pain points that could show up later in the process. Then MLTE takes it one step further and going past actually defining these requirements during this negotiation phase. It helps you also define tests that will allow you to ensure that these cases are being met. This happens through defining certain pieces of the model during the negotiation card discussion. But then past that, it helps you to define test cases using the MLTE library and also running and implementing these tests through the test catalog. One piece that we have of MLTE now is the test catalog that I mentioned. This focuses around having a resource for developers that are looking to test their models and then allowing them to look at other examples of how people have tested their models in the past for different things. For example, if you need your image or your model needs to be robust –to blur, you can go into the test catalog and you can see, hey, how have people tested for robustness to blur in the past. You can look at some examples, take inspiration from those, and then use that to test your own model and provide feedback and provide validation that your model is going to work as expected. Obviously, this is a very large source of knowledge, and it is not going to be prepopulated.

We have a couple of examples now that come preloaded with the MLTE library. We plan to expand these. We have general and widely applicable examples to ship with MLTE, but the idea here is that each organization will run their own instance of MLTE. As the organization uses MLTE, and their developers make new tests for different scenarios that are applicable to the scenarios that they work in, they will upload those into the test catalog that is only available within their organization. Then other developers in the future can look at those previous examples, look at the test catalog, and then there can be more cohesion and more just ease of use to actually get the tests started and structure them similarly. That is one big piece of MLTE is that test catalog.

Then the library, again, going back to the actual library portion. Once you have the test written, you have the test implemented, the MLTE library also provides functionality to help you run these tests and provide a report and an easy way to export your results. Once you define all your tests, you can create certain pieces in the backend that are going to be redesigned slightly. So, it was a spec previously, it is going to be changed. That is still currently in progress. But the idea is you can define your list of test cases, your list of

tests that you want to perform, and then once you have them all set up, you can have them all run. It will give one big output at the end, where you will have the results of each test, the pieces that were part of the negotiation card or that were important to these tests. And you will get one PDF export at the end of a report to say, *Hey, these are the things that I defined as my requirements. These are the ways that I decided to test them, and these are the results*. And so, the goal with this is that you could have your contractor say, *Hey, this is my report. These are the tests I ran on the model, and this is what I got as my output*. Then, if you are then transitioning that to a partner or to a customer, the customer can then look at those tests, look at the report, and then in theory they could be able to re-run all those tests and confirm that they are all matching and they have similar results and they meet the expectations that they set for the model. That, as a whole, is the goal of MLTE. There is a lot there, and there is still a lot to go, but that is the goal and that is the structure that we have set up.

Grace: To summarize, Alex, it seems like there is some key parts of MLTE that are important for solving some of these challenges that were mentioned earlier. One is the negotiation card, which records all these discussions that are happening between the stakeholders and that model developers use then to develop a model. The test catalog is also a big piece, because, as we said before, a lot of data scientists probably don't know how to do all these tests. Being able to have examples really helps with the testing process. And also, this idea that test cases are like integral to MLTE. That is like the essence of MLTE. And also, that you can run a report that basically is going to show you whether the test failed or not. That seems to address every one of the challenges that we mentioned before. We heard the word *negotiation* a lot. Kate, why is it important to bring in stakeholders beyond developers to this process of ML testing and evaluation? How do you see MLTE facilitating this collaboration, this coordination?

Kate: We talked earlier about system requirements being a challenge to capture effectively and also about communication. Really those are two things that are core to making sure that stakeholders and domain experts are involved in the project. I think there is a tendency to want to isolate different skill sets within a project. But really what ends up happening is there is a lack of understanding across the team in terms of what the system can and can't do and even in terms of what means success for a system. So having stakeholders involved, and especially—when we say stakeholders, we could mean like a domain expert, we could mean a system owner, but we also could mean like a user, for instance. When you think about different systems and the way they interact with the world and their environments,

you can see why it would be really important to have a user give input and be involved in the process of evaluating a project, but especially in defining the requirements for a project. Because what means success could be important for our user to articulate not just someone who, for instance, is funding the project or something like that. That is why it is so important to have everyone involved in a project be part of the process. Especially at the beginning, when we talk about system requirements and about how do we define success for this project, that is really critical. What MLTE does to facilitate this is as Alex described is this negotiation card, which is an important artifact. But I think it is important to highlight that by having the negotiation card as part of a larger process, MLTE really tries to guide teams into having these formalized touch points where everyone that I just mentioned actually sits down, ideally together, and has a conversation. That might be something a lot of teams do. But what we have seen and found in our collective experience is that is actually quite hard to get teams to do [this] organically. By having this structured process and giving them not just the opportunity and encouragement to sit down at the beginning of a project and then throughout it, but also to give that infrastructure. When you sit down, it is not that you are just sitting down without an agenda, you have the negotiation card in front of you, and that really guides the conversation. It offers an agenda for discussion and allows for questions and discussion points that maybe wouldn't have come up organically but are really important for both the developers and then the stakeholders and others involved in the project to think about and have a shared understanding both at the beginning of the project and then as the project progresses and things change.

Grace: Yes, so you mentioned something, Kate, which is touch points, which I think is very important because the negotiation card is not something that is one and done. It is something that you revisit after going through the process that Alex said. You come back to the negotiation with the results, and you can say, *You know what, I know we talked about this requirement, but it is impossible for this model to be this accurate or for this model to be this fast. Let's have a conversation again and let's redefine these points*. I definitely can see how that stakeholder involvement is important not just once but throughout the process. The SEI is a federally funded research and development center. A big part of our work as an FFRDC is to transition a lot of the research, the ideas that we have to the public and to stakeholders in the government. Sebastian, how can people learn more about MLTE and about ML test and evaluation in general?

Sebastian: As we mentioned before, we recently released version 1.0 of the MLTE tool. The MLTE tool is available as open source in GitHub currently. And

the repository you have there, it doesn't only contain the code for the MLTE tool, it also contains a lot of documentation related to it, including the description of the process that is supposed to go along with this tool. Also, a substantial amount of tutorials on how to use the tool itself and several demos that showcase how a person would actually be using the tool for a specific context that we are defining in those. There is a lot of material there to not just get the tool but to understand how to use it and how to apply it to the specific context that you want to use it in. We have also published a couple of papers that explain in more detail the thought process behind this and how this MLTE process and tool relate to the existing body of work. Some of those papers are also linked in the GitHub repository. Currently, the best option is to start there, and you can find a lot of different resources to learn more about how and when to use the MLTE process and tool.

Grace: What I am hearing is that everybody has to go to our GitHub website and download our tool. Obviously, there is a big section, though, that says, *We welcome all feedback*. What is next for the three of you? What projects are you working on now? Let's start with you, Alex.

Alex: Yes, so building on MLTE, we actually have the Continuum project, which is a three-year LTP [Late TRL project] that I am currently a part of. It focuses on both improving MLTE, improving the MLTE process, but also largely on the transition of MLTE to the DoD and Integrated T&E. We want to take what we have built, take what we have learned, and show that to the world, show that to DoD, help them improve their practices so that they can learn from what we have done and learn from each other. That is one of my big projects right now. My other project is, like I mentioned before, is Vessel, which is focused on container reproducibility. That one is also super interesting. You can learn more. That one has had a press release recently put out by the SEI. We also recently published our Diff tool, for container differences, open source, also available on GitHub through the SEI's organization. Those are the main things for me right now at the SEI.

Grace: Okay, super interesting. How about you, Sebastian?

Sebastian: Yes, as it turns out, I am mostly working on the same projects that Alex is. In terms of the first project, Continuum, my current focus is to try to improve the user-friendliness and the simplicity of the tool so that it can be more easily integrated into the workflows of people who want to use it. We are doing a lot of refactoring to have the same functionality but for it to be more approachable, easier to understand, easier to start to use, and make it simpler, while still being flexible enough to work in the different

contexts that people have where they may want to use the tool. In terms of the Vessel project, I am currently focusing on everything that is related to making sure that the transition of the work there is going to be smooth. So I am working on both open sourcing the different parts that we are developing, ensuring that the quality of the tool is up to a level where it can actually be robust for people that are using it, as well as interacting with different open source developers of different libraries that we are integrating and working with them so that the resulting tools that we are developing there can work properly when embedded in different workflows as well.

Grace: Super interesting as well. Now, Kate, what is your collaboration with the SEI going to look like going forward? How do you expect AI2C and other organizations to benefit from this collaboration and the results of this work?

Kate: I am excited to keep collaborating with SEI moving forward on the Continuum project which, as Alex and Sebastian both mentioned, is three years of long-term funding to really mature and broaden the reach of MLTE, which is super exciting. I think there are lots of organizations across the DoD that struggle with a lot of the same things that we are describing today. Especially as more and more people are looking to integrate AI and also data-driven practices, MLTE has a lot of tools that are beneficial to those organizations and to the people within them. I am super excited to continue being a part of that with the SEI. As for AI2C and broader organizations, it is really important to have a number of tools for the different issues that come up and to ensure the facilitation of projects that are challenging. Because many of them involve technologies that weren't previously used, or even processes that weren't previously used. For instance, going from something that was a little bit of a manual process to trying to automate it with data, all those things are hard to do. So having a tool like MLTE that really helps to facilitate and give a set of guidelines for teams as they sit down to tackle those problems, I think it is super beneficial both for AI2C and, as we said, with Continuum, trying to broaden the reach of MLTE and bring these tools and this process to more people. I think that is really important for people to have something to lean on and for organizations to have some guidance and infrastructure to help them throughout those processes.

Grace: Great. I want to thank you all for taking the time to talk with us today about this work. To our listeners, thanks for joining us today. We will include links in our transcript to all the resources mentioned in this podcast. The SEI Podcast Series is available in all places you can find podcasts: Apple podcasts, SoundCloud, Spotify, and SEI's YouTube channel. And as always, if you have any questions, please don't hesitate to email us at

info@sei.cmu.edu. Thank you.

Thanks for joining us, this episode is available where you download podcasts. Including SoundCloud, TuneIn radio, and Apple podcasts. It is also available on the SEI website at sei.cmu.edu/podcasts and the SEI's YouTube channel. This copyrighted work is made available through the Software Engineering Institute, a federally funded research and development center sponsored by the U.S. Department of Defense. For more information about the SEI and this work, please visit www.sei.cmu.edu. As always, if you have any questions, please don't hesitate to e-mail us at info@sei.cmu.edu. Thank you.