Integrated T&E of ML Capabilities to Improve Mission-Readiness of ML Models

AUGUST 8, 2024

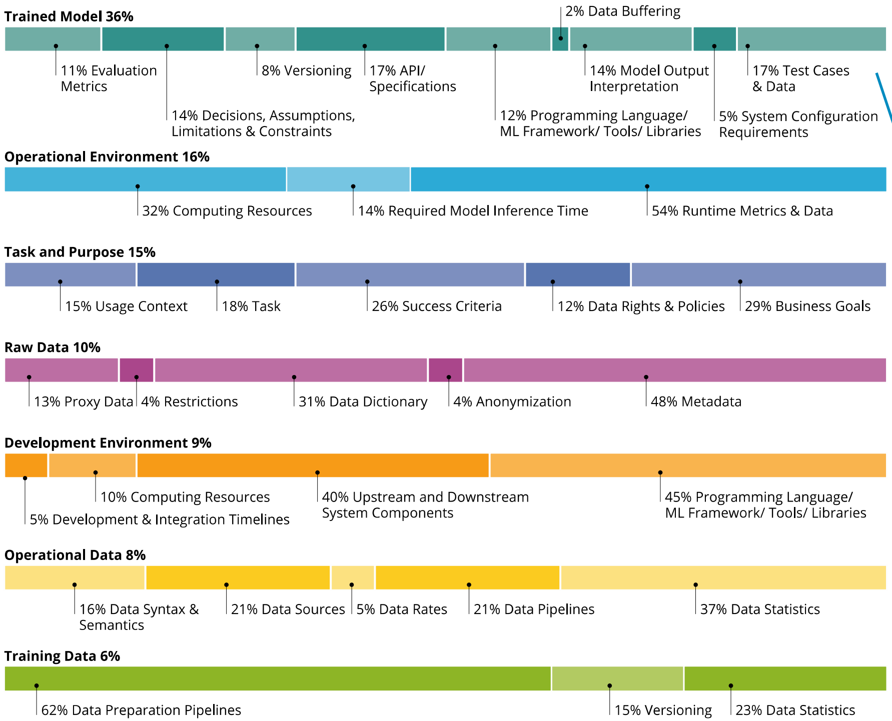
Grace A. Lewis
NDIA Emerging Technologies for Defense Conference

Carnegie Mellon University Software Engineering Institute

DM24-0929

Motivation

Conducted an empirical study to understand ML mismatch in DoD, government and industry: problem that occurs in the development, deployment, and operation of an ML-enabled system due to **incorrect assumptions** made about system elements by different stakeholders that results in a negative consequence.



Lack of test cases and data was one of the largest sources of mismatch.

Codified information into eight machine-readable descriptors: *System Context, Raw Data, Data Pipeline, Training Data, Trained Model, Development Environment, Production Environment, Production Data* — developed **TEC, a tool for input, sharing and detection (open source)**

Source: Lewis, Grace A., Stephany Bellomo, and Ipek Ozkaya. "Characterizing and Detecting Mismatch in Machine-Learning-Enabled Systems." In 2021 IEEE/ACM 1st Workshop on AI Engineering-Software Engineering for AI (WAIN), pp. 133-140. IEEE, 2021.

Reasons for Mismatch Related to Tests and Test Cases

Carnegie Mellon University Software Engineering Institute

Model Developer Perspective

(Data Scientist, ML Engineer)

- No requirements to guide development or test against
- No system context to guide design decisions
- No representative test data
- Unrealistic stakeholder expectations about machine learning capabilities

Model Integrator Perspective

(Software Engineer)

- No idea how to test ML models
- No data for testing
- No model documentation or specification to guide test case development
- ML model does not pass integration and/or system tests

Consequence: ML-enabled systems that do not meet requirements and fail during mission

MLTE (ML Test and Evaluation)

MLTE is a a system-centric, quality-attribute driven, semi-automated process and infrastructure that enables negotiation, specification, and testing of ML model and system qualities.

MLTE (ML Test and Evaluation) produces evidence of testing activities that can be shared with model acquirers and integrators to inform integration and T&E activities.

Co-developed and used by Army Artificial Intelligence Integration Center (AI2C) as part of their model development process with very positive feedback and results.

Available as open source.

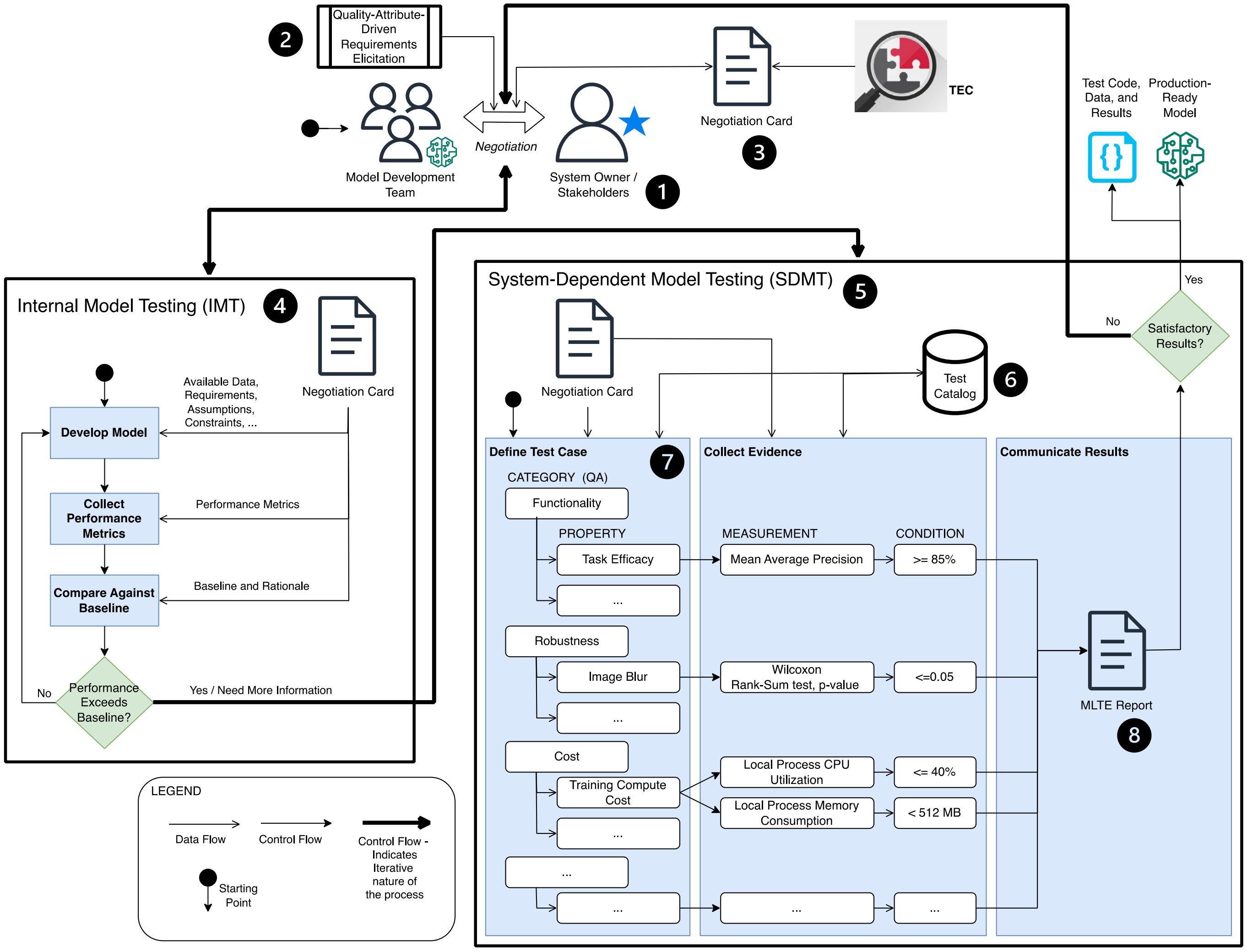
MLTE

1 **Continuous negotiation process** with stakeholders throughout model development

2 Model requirements elicitation driven by **quality attribute scenarios**

3 **Negotiation Card** records agreements and drives model development and testing

4 **IMT** recognizes the experimental nature of model development



5 **SDMT** focuses on obtaining evidence of satisfaction of model and system requirements

6 **Test Catalog** contains reusable (organizational) examples of test cases organized by quality attribute

7 **Test Cases** are first-class artifacts that define metrics, measurement methods, and passing conditions

8 **MLTE Report** records and communicates test results

MLTE Next Steps — Motivation

DoD T&E State of the Practice

- OT, DT, and CT organizations and activities are segregated and disconnected
- Limited by static, document-centric T&E processes
- Current push for Integrated T&E — more digital, tool-supported and iterative

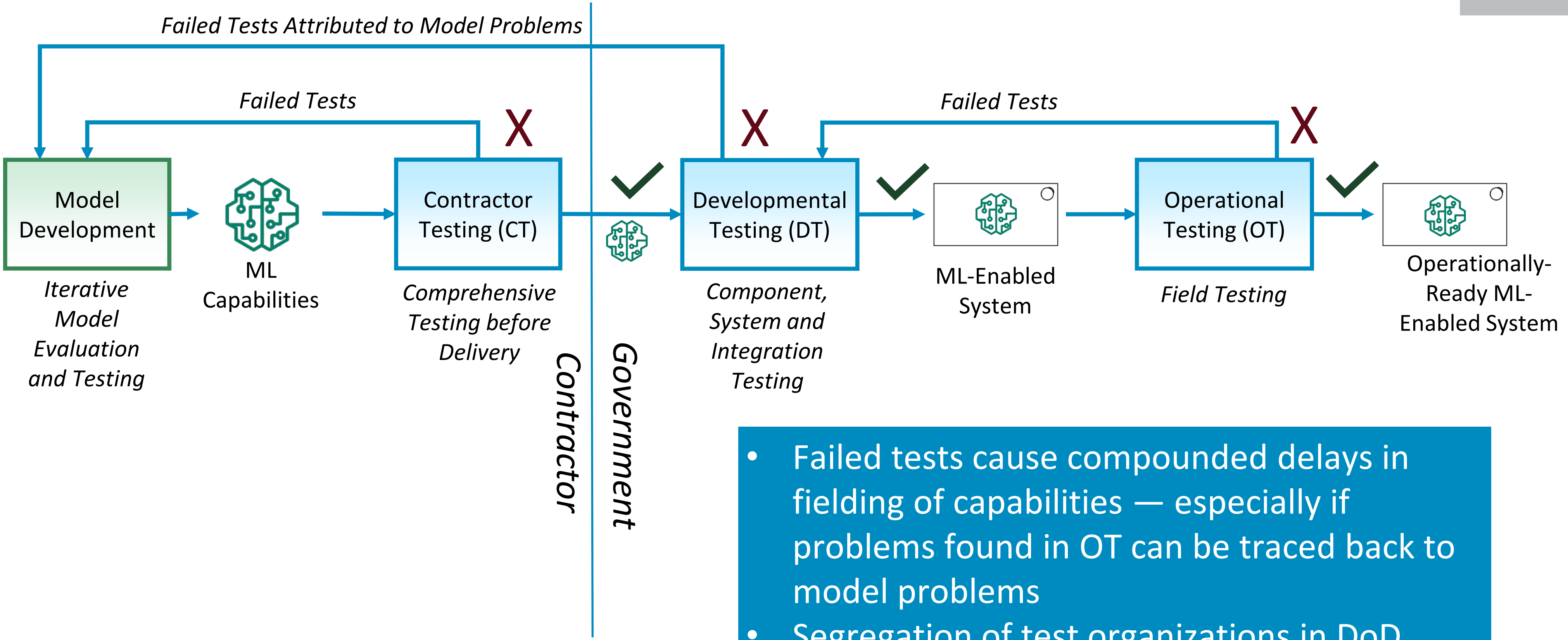
AI/ML T&E State of the Practice

- T&E mostly focuses on model performance
- System-relevant properties are ignored or pushed to discovery in operations
- Data quality evaluation criteria and tools are limited

Continuum

Extend MLTE to support faster fielding of ML-enabled capabilities by developing and transitioning a repeatable, system-centric, quality-attribute driven, semi-automated process and infrastructure for integrated testing of ML capabilities across CT, DT, and OT

Framing the Problem



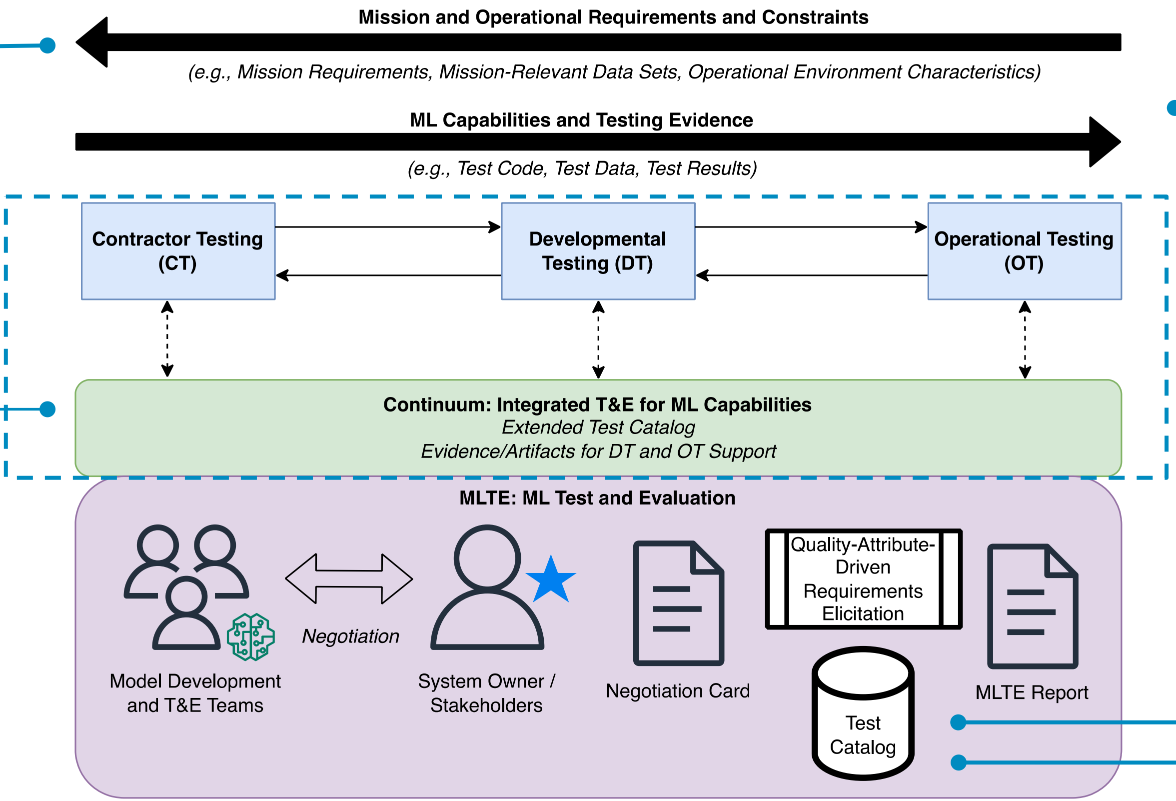
- Failed tests cause compounded delays in fielding of capabilities — especially if problems found in OT can be traced back to model problems
- Segregation of test organizations in DoD leads to uninformed and inefficient testing

Continuum — Establishing the Practice of ML Integrated T&E

Carnegie Mellon University Software Engineering Institute

Work with policy organizations to inform DoD-wide policies that will support the flow of information required for ML Integrated T&E

Work with DoD ML development and T&E organizations to transition and measure the practice of ML Integrated T&E



Extend MLTE outputs to produce testing evidence artifacts

Extend test catalog to include data quality properties

Extend test catalog to a larger set of DoD-relevant quality attributes

Integrated T&E Supported by Continuum

Model Development

Iterative Model Evaluation and Testing

ML Capabilities

Contractor Testing (CT)

Comprehensive Testing before Delivery

Contractor

Government

Developmental Testing (DT)

Component, System and Integration Testing

ML-Enabled System

Operational Testing (OT)

Field Testing

Operationally-Ready ML-Enabled System

Model and system requirements and constraints are discussed, documented, and negotiated from the beginning of the project

Test catalog enables test code reuse

CT of ML capabilities is done based on operationally-relevant scenarios, requirements, and data

Automated regression testing

Results of CT provide evidence of testing activities

Tests can be rerun and extended on the government side

Automated regression testing

Results of CT and DT inform and prioritize OT test planning

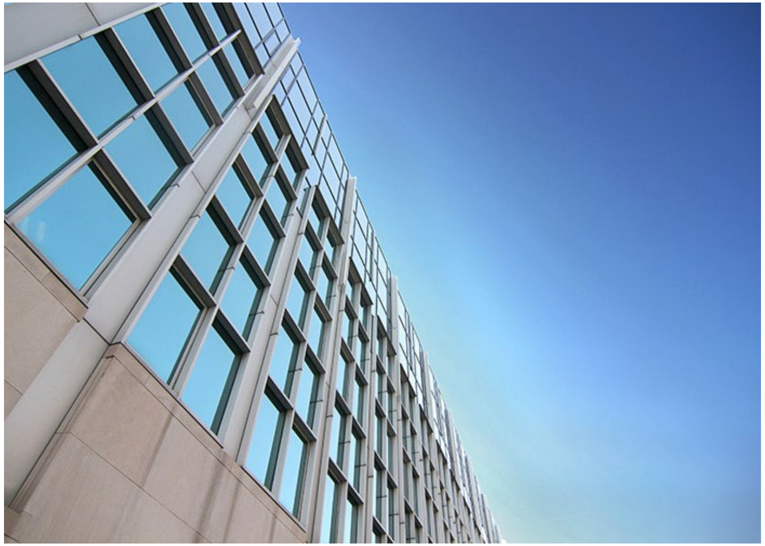
Goal: Reduce time to value (TTV) for ML-enabled capability delivery due to
1. **reducing unnecessary OT activities and**
2. **reducing remediation activities due to discovery of defects during OT.**

TTV is defined as time between end of DT (government determines system is ready) and end of OT (system passes all operational test events/iterations).

Contact Information

Dr. Grace A. Lewis (PI)
Principal Researcher
Tactical and AI-Enabled Systems (TAS) Initiative Lead

Telephone: +1 412.268.5851
Email: glewis@sei.cmu.edu
WWW: http://www.sei.cmu.edu/staff/glewis



Resources

Tools

TEC: ML Mismatch Analysis Tool

- Tool: https://github.com/cmu-sei/TEC
- Descriptors: https://github.com/cmu-sei/ml-mismatch-descriptors

Train-Test Leakage Detection Tool

- Command line version: https://github.com/malusamayo/leakage-analysis
- Jupyter Labs plug-in: https://github.com/Kurt-t/data-leakage-detection-extension

ML Test and Evaluation (Collaboration with AI2C)

- https://github.com/mlte-team/mlte

Papers 1

- Lewis, G. A., Bellomo, S., & Ozkaya, I. (2021, May). Characterizing and Detecting Mismatch in Machine-Learning-Enabled Systems. In 2021 IEEE/ACM 1st Workshop on AI Engineering-Software Engineering for AI (WAIN) (pp. 133-140). IEEE. https://arxiv.org/abs/2103.14101
- Lewis, G. A., Ozkaya, I., & Xu, X. (2021, September). Software Architecture Challenges for ML Systems. In 2021 IEEE International Conference on Software Maintenance and Evolution (ICSME) (pp. 634-638). IEEE. https://doi.org/10.1109/ICSME52107.2021.00071
- Collaboration Challenges Study: Nahar, N., Zhou, S., Lewis, G., & Kästner, C. (2022). Collaboration Challenges in Building ML-Enabled Systems: Communication, Documentation, Engineering, and Process. Proceedings of the 44th International Conference on Software Engineering (ICSE 2022). https://arxiv.org/abs/2110.10234 — **Distinguished Paper Award**
- Lewis, G. A., Echeverría, S., Pons, L., & Chrabaszcz, J. (2022, May). Augur: A Step Towards Realistic Drift Detection in Production ML Systems. In 2022 IEEE/ACM 1st International Workshop on Software Engineering for Responsible Artificial Intelligence (SE4RAI) (pp. 37-44). IEEE. https://doi.org/10.1145/3526073.3527590
- Yang, C., Brower-Sinning, R., Lewis, G. A., Kästner, C.(2022). Data Leakage in Notebooks: Static Detection and Better Processes. Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering (ASE 2022). https://arxiv.org/abs/2209.03345
- Yang, C., Brower-Sinning, R., Lewis, G. A., Kästner, C., & Wu, T. (2023). Capabilities for Better ML Engineering. In Proceedings of the AAAI's Workshop on Artificial Intelligence Safety (SafeAI 2023). https://arxiv.org/abs/2211.06409

Papers 2

- Maffey, K., Dotterer, K., Niemman, J., Crickshank, I., Lewis, G.A., Kästner, C. (2023). MLTEing Models: Negotiating, Evaluating, and Documenting Model and System Qualities. In Proceedings of the 45th International Conference on Software Engineering (ICSE 2023) – New Ideas and Emerging Results Track. http://arxiv.org/abs/2303.01998
- Nahar, N., Zhang, H., Lewis, G., Zhou, S., & Kästner, C. (2023). A Meta-Summary of Challenges in Building Products with ML Components — Collecting Experiences from 4758+ Practitioners. Proceedings of the 2nd Conference on AI Engineering — Software Engineering for AI (CAIN 2023). https://arxiv.org/abs/2304.00078 — **Early Career Researcher Award**
- Yang, C., Rustogi, R., Brower-Sinning, R., Lewis, G. A., Kästner, C., & Wu, T. (2023). Beyond Testers' Biases: Guiding Model Testing with Knowledge Bases using LLMs. Findings of the Association for Computational Linguistics: EMNLP 2023. https://aclanthology.org/2023.findings-emnlp.901.pdf
- Lewis, G.A., Muccini, H., Ozkaya, I., Vaidhyanathan, K., Weiss, R., and Zhu, L. (2024). Software Architecture and Machine Learning (Dagstuhl Seminar 23302). In Dagstuhl Reports, Volume 13, Issue 7, pp. 166-188, Schloss Dagstuhl – Leibniz-Zentrum für Informatik. https://doi.org/10.4230/DagRep.13.7.166
- Brower-Sinning, R., Lewis, G.A., Echeverría, S. & Ozkaya, I. (2024). Using Quality Attribute Scenarios for ML Model Test Case Generation. Proceedings of the 3rd International Workshop on Software Architecture and Machine Learning (SAML 2024) - Companion Proceedings of ICSA 2024, the 21st International Conference on Software Architecture. https://arxiv.org/abs/2406.08575