Scoping of Non-Functional Requirements for Machine Learning Systems

Scoping of Non-Functional Requirements for Machine Learning Systems

Khan Mohammad Habibullah, Juan Garcia Diaz, Gregory Gay, Jennifer Horkoff
Chalmers and University of Gothenburg
Gothenburg, Sweden
khan.mohammad.habibullah@gu.se, gusgarjub@student.gu.se, greg@greggay.com, jennifer.horkoff@gu.se

I. Context

Machine Learning (ML) systems increasingly perform complex decision-making and prediction tasks—e.g., in autonomous driving—based on patterns inferred from large quantities of data. The inclusion of ML increases the capabilities of software systems, but also introduces or exacerbates challenges. ML systems can be more complex, time-consuming and expensive to specify, develop, and test than traditional systems, and can suffer from issues related to safety, lack of explainability, limited maintainability, and bias [1], [2]. As in other domains, ML systems must satisfy certain quality requirements—known as non-functional requirements (NFRs)—to be considered fit for purpose [1].

II. RE Challenges and Motivation

Past research has revealed challenges related to the definition and measurement of the attainment of NFRs for ML systems, including uncertainty, non-deterministic behavior, domain dependence, requirement interdependence, lack of measurement techniques, scoping and breakdown, documentation, and lack of awareness among practitioners and users [1], [2].

We propose that addressing these challenges requires establishing clear *scoping* for NFR definition and measurement for ML systems [3]. ML systems contain many distinct *components*—some related specifically to ML and others that do not directly interact with ML-based elements—including structured code (code encapsulated in classes or services), unstructured code scripts (support code, generally in a standalone file—e.g., to perform data pre-processing), data and resource files, and executable programs (e.g., a trained ML model). These components interact to deliver system functionality.

NFRs can be defined and measured over not just the full ML system, but over individual components and groups of components [3], [4]. We propose that understanding of how to define and measure attainment of NFRs for ML systems requires understanding (a) what these components are, and (b), how their interactions affect the resulting system quality. We are interested in addressing both aspects, with a focus on quantitative measurement of NFR attainment.

To illustrate, consider "maintainability"—a challenging and crucial NFR for ML systems [1]. How maintainability is defined and measured may vary based between components. For example, assessments of the maintainability of a dataset used for training a ML model may differ from assessing the maintainability of the code that uses the trained model. The assessment of maintainability for the full system, then, depends on the data, code, and other components.

We previously performed an initial scoping of NFRs over ML system components [3]. In this extended abstract, we present an updated and clarified scoping as a starting point for future research on NFR definition and measurement.

III. Related Work

Past research has recognized that quality of ML systems is influenced by the interaction of different components. For example, Sculley et al. noted that ML aspects are part of a larger software system, which may include components for configuration, data collection, feature extraction, analysis, monitoring, and glue code [4]. Sibert et al. also note that different components of ML systems each influence overall system quality [5]. Vogelsang and Borg emphasized that requirements engineering for ML should focus on requirements over data along with requirements for the system [6]. Shivashankar and Martini discussed maintainability challenges in different components of a ML workflow [7]. As previously noted, we also previously proposed an initial scoping for NFRs [3].

IV. Methodology

We are interested in exploring how *scoping* of NFRs over different components and groups of components within ML systems affects their definition and measurement, addressing the following research questions:

- **RQ1:** Over what components and groups of components of a ML system can different types of NFRs (e.g., maintainability) be scoped?
- **RQ2:** How should NFRs be defined over different scopes and across scopes?
- **RQ3:** How can attainment of specific NFRs be measured over different scopes or across scopes?

We performed an exploratory study to address RQ1 by identifying the primary components of ML systems that can be considered as targets for NFR scoping.

To identify components, we performed a non-systematic literature review. We selected publications from Scopus that discuss the development lifecycle of ML systems and ML training pipelines, along with definition, identification, measurement, and metrics related to NFRs for ML systems.

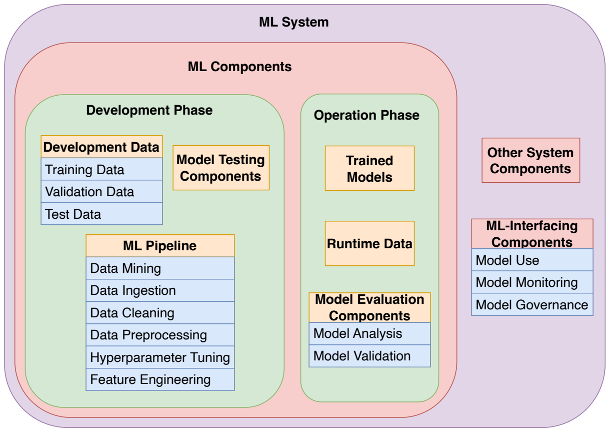

Fig. 1. Components (rectangles), separated into groupings (rounded).

From the examined studies, we synthesized a list of components. We then grouped components based on their semantic purpose—first, based on whether they were specifically related to performing ML (e.g., related to data processing or model training) or delivering system functionality. Within these groupings, we then performed further sub-grouping—e.g., grouping all components related to training or evaluating models. In parallel, we also grouped components based on their syntactic purpose—i.e., the type of information they represented (i.e., structured code, script, data, or model).

In a series of meetings, we discussed each component and the groupings and came to a shared understanding, resolving cases where we disagreed.

V. ML System Components for NFR Scoping

Fig. 1 presents our proposed division of an ML system into different relevant components. We break systems components first into "ML components"—those that are responsible for training and performing ML—and components that interact either directly or indirectly with those ML components to deliver services. In this figure, components are represented as rectangles. Some components represent a collection of closely-integrated sub-components—e.g., the ML pipeline consists of multiple data processing and model training steps that each may affect the ML system.

We further group ML components into two sub-groupings. First, we group components related to the *development* of models—including the data and training pipeline and components used to test the model before it is integrated. Second, we group components related to the use of the model in *operation*—including models, data collected in the field, and components responsible for runtime evaluation.

Fig. 2 presents a second way of grouping components, based on the type of information represented. We propose that each component fits one of four "types" of information—structured code (reusable code encapsulated as, e.g., classes or services, generally used to deliver system functionality), scripted code (unstructured code encapsulated in a file, generally used to provide support services—e.g., data processing or model training), data resources, and executable models. These types will likely affect how NFRs are measured.

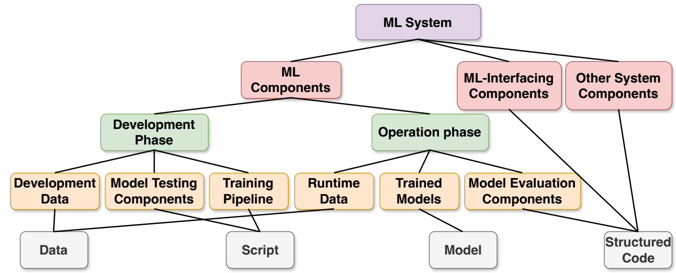

Fig. 2. Components, grouped by type of information represented.

Consider maintainability—the degree to which a system can be modified. In traditional systems, maintainability is often measured based on "coupling" between components. Coupling between code elements is well-understood. However, new or revised coupling measurements may be needed at different scopes in an ML system, and such measurements may vary depending on the types of information represented by these components. For example, coupling between or within code, model, and data is potentially quite different in its meaning and measurement than coupling between multiple components consisting of structured code.

VI. Conclusion

We present a proposed set of components and different groupings of components within ML systems for the purpose of scoping NFR definition and measurement. We offer this component breakdown to aid researchers and practitioners working with NFRs for these systems.

Currently, we are identifying measurements for select NFR types, such as maintainability, with a focus on automated quantitative measurement. We intend to evaluate definition and measurement scopes with industrial partners.

References

[1] K. M. Habibullah, G. Gay, and J. Horkoff, "Non-functional requirements for machine learning: Understanding current use and challenges among practitioners," *Requirements Engineering*, vol. 28, no. 2, pp. 283–316, 2023.

[2] K. M. Habibullah, H.-M. Heyn, G. Gay, J. Horkoff, E. Knauss, M. Borg, A. Knauss, H. Sivencrona, and P. J. Li, "Requirements and software engineering for automotive perception systems: an interview study," *Requirements Engineering*, pp. 1–24, 2024.

[3] K. M. Habibullah, G. Gay, and J. Horkoff, "Non-functional requirements for machine learning: An exploration of system scope and interest," *SE4RAI workshop, 44th Int. Conf. on Software Engineering (ICSE 2022)*, 2022.

[4] D. Sculley, G. Holt, D. Golovin, E. Davydov, T. Phillips, D. Ebner, V. Chaudhary, M. Young, J.-F. Crespo, and D. Dennison, "Hidden technical debt in machine learning systems," *Advances in neural information processing systems*, vol. 28, 2015.

[5] J. Siebert, L. Joeckel, J. Heidrich, A. Trendowicz, K. Nakamichi, K. Ohashi, I. Namba, R. Yamamoto, and M. Aoyama, "Construction of a quality model for machine learning systems," *Software Quality Journal*, vol. 30, no. 2, pp. 307–335, 2022.

[6] A. Vogelsang and M. Borg, "Requirements engineering for machine learning: Perspectives from data scientists," in *2019 IEEE 27th Int. Requirements Engineering Conf. Workshops (REW)*. IEEE, 2019, pp. 245–251.

[7] K. Shivashankar and A. Martini, "Maintainability challenges in ml: A systematic literature review," in *2022 48th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*. IEEE, 2022, pp. 60–67.