Continuous Quality Improvement of AI-based Systems: the QualAI Project

Nicole Novielli[1], Rocco Oliveto[2], Fabio Palomba[3], Fabio Calefato[1], Giuseppe Colavito[1], Vincenzo De Martino[3], Antonio Della Porta[3], Giammaria Giordano[3], Emanuela Guglielmi[2], Filippo Lanubile[1], Luigi Quaranta[1], Gilberto Recupito[3], Simone Scalabrino[2], Angelica Spina[2], Antonio Vitale[2,4]
[1]University of Bari, Italy, [2]University of Molise, Italy, [3]University of Salerno, Italy, [4]Politecnico di Torino, Italy

ABSTRACT

QualAI is a two-year project aimed at defining a set of recommenders to continuously monitor, assess, and improve the quality of AI-based systems, with a particular focus on machine learning (ML) applications. We will develop recommenders for the quality assurance of both data and ML models to enable practitioners to mitigate technical debt. Special attention will be paid to communication challenges that may arise in hybrid teams comprising data scientists and software developers. This paper presents the project outline, provides an executive summary of the research activities, outlines the expected project outcomes, and reports the results obtained to date.

CCS CONCEPTS

• **Software and its engineering → Software evolution**; **Maintaining software**; • **Computing methodologies → Artificial intelligence**;

KEYWORDS

Software Engineering, Machine Learning, Quality Assurance, Recommender Systems

ACM Reference Format:
Nicole Novielli[1], Rocco Oliveto[2], Fabio Palomba[3], Fabio Calefato[1], Giuseppe Colavito[1], Vincenzo De Martino[3], Antonio Della Porta[3], Giammaria Giordano[3], Emanuela Guglielmi[2], Filippo Lanubile[1], Luigi Quaranta[1], Gilberto Recupito[3], Simone Scalabrino[2], Angelica Spina[2], Antonio Vitale[2,4]. 2024. Continuous Quality Improvement of AI-based Systems: the QualAI Project. In *Proceedings of the 18th ACM / IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM '24), October 24–25, 2024, Barcelona, Spain.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3674805.3695393

1 INTRODUCTION

Machine learning (ML)-based systems present unique challenges in development and quality assurance. Unlike traditional software, where developers specify behavior based on requirements, ML-based systems rely on data scientists to define construct operationalization, build training sets, and select ML techniques. These systems require special attention to data quality, model performance, and integration issues. Furthermore, a successful laboratory performance alone does not guarantee the success of an AI system after deployment, as several crucial factors might determine the successful adoption of ML-based systems in real-world scenarios [1].

In a typical software system, given the requirements, the behavior is always specified by the developers. In contrast, in AI-enabled software, data scientists define the operationalization of constructs that are relevant to the addressed problems. In addition, they build training sets and identify the appropriate ML technique, which will eventually define the system behavior. Such systems require maintenance and quality assurance like any other system, but special attention should be devoted to the typical issues that affect the quality of data and ML models [15]. As such, assessing and improving the quality of ML-based systems presents unique challenges.

Key quality concerns for ML-based systems include *model* and *data quality*, as training data may become outdated due to concept drift, necessitating regular updates. Further concerns are associated with the effectiveness of team *communication* due to potential technological gaps and communication issues that heterogeneous teams could experience, including data scientists and software developers. In complex ML-based systems, models serve various purposes and integrate into different system components [9]. The experience of Google Health's AI-based diagnostic software underscores that while unit-level testing of each ML model is necessary, it is not sufficient. Quality assurance should also be performed at *integration* and system levels. Furthermore, dependency libraries require specific attention, as model accuracy may degrade when migrated to a different ML framework due to compatibility issues [7, 10]. Finally, further issues might arise at the level of *deployment* and *operations* as automated build processes of some modules of an ML-based system and the construction of the relative container images often require training one or more ML models, with quality issues potentially occurring in this phase.

New methods and strategies are needed to keep ML-based systems responsive, monitored, and dependent on reliable variables, aligning with the **MLOps**[1] engineering culture and practice. To support this vision and address the issues mentioned above, the QualAI project aims to define a set of recommenders for continuous monitoring, assessment, and improvement of AI-based systems, with a focus on ML-based systems.

The main expected outcome of the project is the definition of quality assurance recommenders for both data and ML models. Also, we aim at providing empirically-driven guidelines for practitioners on how to mitigate technical debt [15]. Emphasis will be placed on

ESEM '24, October 24–25, 2024, Barcelona, Spain

ACM ISBN 979-8-4007-1047-6/24/10
https://doi.org/10.1145/3674805.3695393

[1]MLOps, https://ml-ops.org, 2020

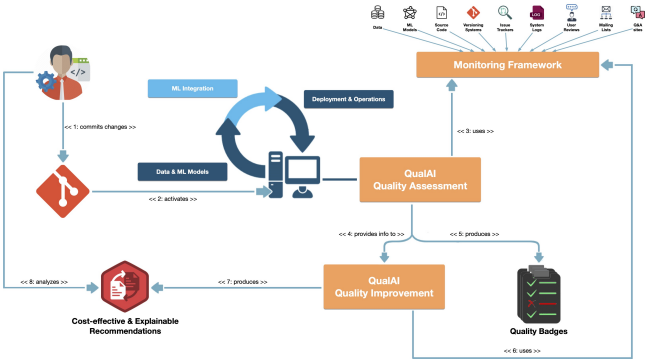

Figure 1: The Workflow of the QualAI framework.

communication issues that could arise between data scientists and software developers. Finally, we will define approaches to (i) identify quality issues in the CI/CD pipeline and (ii) monitor the quality of the system during the operations phase. QualAI will facilitate both the analysis of the recommendations and the planning of the corrective operations suggested using a cost-effective analysis.

QualAI [8] is a two-year project that was funded in July 2023 by the European Union - NextGenerationEU through the PRIN 2022 call for projects of the Italian Ministry of University and Research and started in September 2023. In the following, we describe the project research goals, provide an executive summary, and report the results obtained to date.

2 GOALS AND EXPECTED OUTCOMES

2.1 Project Goal and Final Outcome

QualAI aims to design and implement tools and methodologies to improve the quality of AI-based systems, with a particular focus on ML-based systems. The framework addresses quality from multiple perspectives: data and ML models, integration, deployment, and operations. QualAI monitors the quality of ML-based systems throughout their entire life cycle, collecting information to automatically assess quality. When issues are identified, the framework suggests corrective actions to remove technical debt and improve the overall quality of the system (see Figure 1). A key feature of QualAI is its ability to provide cost-effective and explainable recommendations. The framework ranks identified issues and associated corrective actions based on the potential cost-benefit ratio for developers. These recommendations include human-readable explanations, either textual or visual, to increase the confidence of practitioners and support informed decision-making [2].

QualAI recommenders can be integrated into existing continuous integration and deployment pipelines for ongoing quality assurance. The Quality Assessment component analyzes new versions of ML-based systems from various perspectives when changes in data or code are committed. Meanwhile, the Quality Improvement component identifies refactoring operations to address issues and improve overall system quality. Each recommendation includes a human-comprehensible description and cost-benefit analysis, which facilitates prioritization and scheduling of improvements. The framework relies on a continuously updated and shared knowledge base that contains both internal resources (such as source code, ML models, training data, issues, and logs) and external resources (such as other ML-based projects or Q&A sites) related to the system under analysis.

The effectiveness of QualAI will be empirically evaluated through mixed-method research. This approach combines the mining of data science projects to establish recommender accuracy with survey- and interview-based studies involving developers to gather feedback on recommender effectiveness. As part of this research, we will define guidelines for conducting empirical studies and creating replication packages. In addition, we plan to apply QualAI recommenders to industrial software systems in collaboration with our industry partners.

2.2 Objectives and Expected Results

Objective 1: Definition of a monitoring framework for knowledge management. As a shared preliminary objective, the work plan involves determining appropriate and representative data sources. These include analyzing developers' communication, user feedback, source code, notebooks, build logs, and application logs. We'll review and synthesize common ML process models to ensure our approaches support realistic application scenarios. The result will be a shared knowledge base providing data for all quality assessment and improvement approaches in the subsequent steps.

Objective 2: Definition of approaches to assess and improve the quality of data and ML models. We will examine the factors leading to the degradation of ML systems' properties such as robustness, efficiency, privacy, interpretability, fairness, and reproducibility. Our goal is to create a comprehensive catalog of the issues affecting the properties mentioned above and strategies to mitigate them. The first expected result is the definition of novel approaches to identify issues in the data used for training the models, in the machine learning techniques used to build them, and in their configuration, based on both static and dynamic analysis. The second expected result is a set of cost-effective recommendation techniques that can automatically improve data and model quality. Specifically, we will devise approaches to help users understand and address the root causes of quality degradation, such as recommending the removal of gender-based features that may compromise privacy and fairness. All recommendations will be explainable to facilitate the identification of further critical issues to address. Moreover, the proposed recommenders will balance the cost needed to addressed the identified issues and their effectiveness in improving the ML-based system.

Objective 3: Definition of approaches to assess and improve the quality of integration between the underlying ML models and the rest of the system. We will establish methods for assessing and improving the integration of ML models within the larger system. We will focus on several relevant aspects, including team communication, technical gaps, and system security. The first expected result is a set of cost-effective techniques that can automatically detect quality issues at the integration and system level. To achieve this goal, we will define novel approaches for detecting quality issues both in the integration (process-oriented) and in the resulting system (product-oriented). Such approaches will be mostly based on static analysis techniques (e.g., detection of community and code smells). Finally,

novel approaches will be defined for automatically improving the quality of the integration (e.g., techniques for automatically adapting the technologies used by data scientists to production-ready code) and of the resulting system (e.g., ML-based system-specific refactoring operations). We also plan to devise approaches based on data-driven techniques that will still follow an explainable and cost-effective philosophy. The second expected result is a set of cost-effective approaches to recommend operations to resolve quality problems at the ML integration level.

Objective 4: Definition of approaches to assess and improve the quality of deployment and operation of ML-based systems. We will formulate strategies to assess and improve the deployment and operation of ML-based systems. Specifically, we will focus on the CI/CD philosophy, pipeline configuration, the use of virtualization/containerization techniques, and the quality of software logs. We aim to define and validate techniques to automatically detect quality issues in deployment and operation using both static analysis (e.g., detecting Dockerfile configuration smells) and dynamic analysis (e.g., analyzing execution logs). Furthermore, we plan to devise cost-effective approaches to recommend and implement fixes for quality issues in the deployment and operation of ML-based systems. Hence, the first expected result is a set of techniques that can automatically detect quality issues at the deployment and operation levels. The second expected result is a set of cost-effective approaches that can recommend operations to fix quality issues at the deployment and operation levels. In particular, new approaches will be defined to automatically improve the quality of deployment and operation.

3 INTERMEDIATE RESULTS

QualAI is a two-year research project that started in September 2023 and is organized into six Work Packages (WPs). WP1 focuses on the conceptualization of QualAI. WP2, WP3, and WP4 aim to define recommenders for the quality assurance of data and ML model (WP2), ML integration (WP3), and deployment and operations (WP4). WP5 is dedicated to open science dissemination, while WP6 is concerned with the management of the QualAI project. As far as the technical WPs (WP1-WP4) are concerned, the project has already achieved significant intermediate results during the first ten months of research activities, which we briefly report in the following.

Understanding Developer Practices and Code Smells Diffusion in AI-Enabled Software. In response to continuous change requests and strict time-to-market pressures, developers frequently update their software systems to meet user requirements. This practice often leads to the release of immature products, as developers may neglect best practices to reduce delivery times, thus causing the accumulation of technical debt. We have conducted a preliminary analysis that addresses this gap by exploring the diffusion of Python-specific code smells, — i.e., sub-optimal design decisions identifiable through software metrics and usually associated with technical debt — and the activities that lead to their introduction. We examined 200 AI-enabled systems, extracting 10,611 release information points. Our findings reveal that code smells related to object-oriented principles are rare in Python, with Complex List Comprehension being the most prevalent and long-lasting smell. Furthermore, we found that evolutionary activities are the primary contributors to the introduction of code smells [6].

The Impact Data Quality on the Performance and Maintenance of AI-based systems. The development of AI-based systems involves addressing challenges due to data quality assurance. Data collection from diverse sources can introduce quality issues and the evolving nature of phenomena under study might introduce threats due to data drift, which might eventually impact the performance of AI models. To address these issues, we have investigated how the quality of data affects the performance of classifiers. Specifically, we have defined data quality filters and investigated their impact on pre-trained models for the automatic classification of issues [3]. In addition, we have created a catalog of existing data smells and the tools to detect them. Furthermore, we have assessed the prevalence of data smells and their correlation with data quality metrics. The empirically-driven findings provide insights into the challenges of maintaining AI-enabled systems [11].

A catalog of configuration smells for deployment and operations. QualAI aims to define new approaches for quality assurance of the deployment and operation of ML-based software systems, with focus on CI/CD pipelines, containerization, and operation. To this end, we conducted empirical studies to define a catalog of issues that may affect deployment and operation. Specifically, we have investigated Dockerfiles, as Docker is the *de facto* standard for software containerization. Even if best practices exist for writing Dockerfiles, developers do not always comply with them. Violations of such practices and poor design choices, known as Dockerfile smells, can negatively impact the reliability and performance of Docker images. Based on the results of mixed-method empirical studies, including an analysis of Dockerfile histories maintained by experts and a survey in which we asked professional developers to evaluate Dockerfiles, we have created a taxonomy of Dockerfile smells. This taxonomy can be used as a guide for novices to understand which aspects to focus on for writing high-quality Dockerfiles [12, 13]. Furthermore, we investigated what Dockerfile smells receive more attention from developers, and to what extent they are willing to accept automated suggestions for addressing these smells. The results of our empirical study demonstrated that most developers pay more attention to changes aimed at improving Dockerfile performance and are willing to accept fixes for the most common smells [13].

Non-functional Requirements of AI-enabled Systems. Among the most critical issues affecting the reliability of ML systems concern, non-functional requirements are also investigated in the scope of the QualAI project. In particular, fairness has recently emerged as a crucial property to ensure when developing and testing AI-enabled systems. We first devised a novel context-aware requirements engineering framework addressing fairness concerns in machine learning systems (ReFair). ReFair enables early fairness analytics by leveraging natural language processing, specifically word embeddings, to classify sensitive features within User Stories. It is capable of recommending context-specific sensitive features to consider during implementation. ReFair was evaluated using a synthetic dataset of about 12k user stories [5].

Large Language Models (LLMs) for Task Automation in Software Development. LLMs have been shown to be effective for a wide variety of software engineering tasks, including code generation and labeling. In particular, we have investigated the case of automated

issue labeling, as it is a crucial task for the effective management of software projects. To date, several approaches have been proposed to solve this task, many of which leverage machine learning and fine-tuning of BERT-like language models, achieving state-of-the-art performance. More recently, decoder-only models such as GPT have become prominent in SE research because of their surprising capabilities to achieve state-of-the-art performance even for tasks for which they have not been trained. We investigated to what extent we can leverage GPT-like LLMs to automate the issue labeling task. Our results demonstrate the ability of GPT-like models to correctly classify issue reports in the absence of labeled data that would be required to fine-tune BERT-like LLMs [4].

As far as code generation is concerned, commercial tools, such as GitHub Copilot, are nowadays available to support developers in their coding tasks. However, these models may be trained on proprietary data and source code, introducing problems related to intellectual property leaks. We have performed an empirical study focusing on black-box attack (reconstruction attack) on an LLM designed for a specific coding task: code summarization. The results reveal that, while the attack is generally unsuccessful, it can succeed in reconstructing versions of the code that closely resemble the original [14].

4 CONCLUSION

In this paper, we have described QualAI, a two-year project aimed at defining a set of recommenders to continuously monitor, assess, and improve the quality of AI-enabled systems. This paper reports the project outline and an executive summary of the research activities. The project started on September 2023 and, to date, it has already achieved significant intermediate results, which we discussed in this paper.

ACKNOWLEDGMENTS

This research was partially funded by the European Union - NextGenerationEU through the Italian Ministry of University and Research, Projects PRIN 2022 ("QualAI: Continuous Quality Improvement of AI-based Systems", grant n. 2022B3BP5S, CUP: H53D23003510006) and by the project PNRR-NGEU which has received funding from the MUR – DM 118/2023.

REFERENCES

[1] Emma Beede, Elizabeth Baylor, Fred Hersch, Anna Iurchenko, Lauren Wilcox, Paisan Ruamviboonsuk, and Laura M. Vardoulakis. 2020. A Human-Centered Evaluation of a Deep Learning System Deployed in Clinics for the Detection of Diabetic Retinopathy. In *Proc. of the 2020 CHI Conf. on Human Factors in Computing Systems (CHI '20)*. ACM, 1–12. https://doi.org/10.1145/3313831.3376718

[2] Vito Bellini, Angelo Schiavone, Tommaso Di Noia, Azzurra Ragone, and Eugenio Di Sciascio. 2018. Knowledge-aware Autoencoders for Explainable Recommender Systems. In *Proc. of the 3rd Workshop on Deep Learning for Recommender Systems* (Vancouver, BC, Canada) *(DLRS 2018)*. ACM, 24–31. https://doi.org/10.1145/3270323.3270327

[3] Giuseppe Colavito, Filippo Lanubile, Nicole Novielli, and Luigi Quaranta. 2024. Impact of data quality for automatic issue classification using pre-trained language models. *Journal of Systems and Software* 210 (2024), 111838. https://doi.org/10.1016/j.jss.2023.111838

[4] Giuseppe Colavito, Filippo Lanubile, Nicole Novielli, and Luigi Quaranta. 2024. Leveraging GPT-like LLMs to Automate Issue Labeling. In *Proceedings of the 21st International Conference on Mining Software Repositories* (Lisbon, Portugal) *(MSR '24)*. Association for Computing Machinery, New York, NY, USA, 469–480. https://doi.org/10.1145/3643991.3644903

[5] Carmine Ferrara, Francesco Casillo, Carmine Gravino, Andrea De Lucia, and Fabio Palomba. 2024. ReFAIR: Toward a Context-Aware Recommender for Fairness Requirements Engineering. In *Proc. of the IEEE/ACM 46th International Conference on Software Engineering* (Lisbon, Portugal) *(ICSE '24)*. Association for Computing Machinery, New York, NY, USA, Article 213, 12 pages. https://doi.org/10.1145/3597503.3639185

[6] Giammaria Giordano, Giusy Annunziata, Andrea De Lucia, and Fabio Palomba. 2023. Understanding Developer Practices and Code Smells Diffusion in AI-Enabled Software: A Preliminary Study. In *Proc. of IWSM/MENSURA 23, September 14–15, 2023, Rome, Italy*. CEUR. https://ceur-ws.org/Vol-3543/paper18.pdf

[7] Q. Guo, S. Chen, X. Xie, L. Ma, Q. Hu, H. Liu, Y. Liu, J. Zhao, and X. Li. 2019. An Empirical Study Towards Characterizing Deep Learning Development and Deployment Across Different Frameworks and Platforms. In *2019 34th IEEE/ACM Int'l Conf. on Automated Software Engineering (ASE)*. IEEE Computer Society, 810–822. https://doi.org/10.1109/ASE.2019.00080

[8] Nicole Novielli, Rocco Oliveto, Fabio Palomba, Fabio Calefato, Giuseppe Coalvito, Vincenzo De Martino, Antonio Della Porta, Giammaria Giordano, Emanuela Guglielmi, Filippo Lanubile, Gilberto Recupito, Simone Scalabrino, Angelica Spina, and Antonio Vitale. 2024. QualAI: Continuous Quality Improvement of AI-based Systems. In *Joint Proc. of RCIS 2024 Workshops and Research Projects Track*. CEUR. https://ceur-ws.org/Vol-3674/RP-paper3.pdf

[9] Zi Peng, Jinqiu Yang, Tse-Hsun (Peter) Chen, and Lei Ma. 2020. A first look at the integration of machine learning models in complex autonomous driving systems: a case study on Apollo. In *Proc. of the 28th ACM Joint Meeting on European Software Engineering Conf. and Symposium on the Foundations of Software Engineering* (Virtual Event, USA) *(ESEC/FSE 2020)*. ACM, 1240–1250. https://doi.org/10.1145/3368089.3417063

[10] Hung Viet Pham, Thibaud Lutellier, Weizhen Qi, and Lin Tan. 2019. CRADLE: Cross-Backend Validation to Detect and Localize Bugs in Deep Learning Libraries. In *2019 IEEE/ACM 41st Int'l Conf. on Software Engineering (ICSE)*. 1027–1038. https://doi.org/10.1109/ICSE.2019.00107

[11] Gilberto Recupito, Raimondo Rapacciuolo, Dario Di Nucci, and Fabio Palomba. 2024. Unmasking Data Secrets: An Empirical Investigation into Data Smells and Their Impact on Data Quality. In *Proc. of the IEEE/ACM 3rd International Conference on AI Engineering - Software Engineering for AI* (Lisbon, Portugal) *(CAIN '24)*. Association for Computing Machinery, New York, NY, USA, 53–63. https://doi.org/10.1145/3644815.3644960

[12] Giovanni Rosa, Simone Scalabrino, Gregorio Robles, and Rocco Oliveto. 2024. Not all Dockerfile Smells are the Same: An Empirical Evaluation of Hadolint Writing Practices by Experts. In *Proceedings of the 21st International Conference on Mining Software Repositories* (Lisbon, Portugal) *(MSR '24)*. ACM, New York, NY, USA, 231–241. https://doi.org/10.1145/3643991.3644905

[13] Giovanni Rosa, Federico Zappone, Simone Scalabrino, and Rocco Oliveto. 2024. Fixing Dockerfile smells: an empirical study. *Empirical Software Engineering* 29, 5 (2024), 108. https://doi.org/10.1007/s10664-024-10471-7

[14] Marco Russodivito, Angelica Spina, Simone Scalabrino, and Rocco Oliveto. 2024. Black-Box Reconstruction Attacks on LLMs: A Preliminary Study in Code Summarization. In *Proc. of 17th International Conference on the Quality of Information and Communications Technology (QUATIC)*.

[15] D. Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips, Dietmar Ebner, Vinay Chaudhary, Michael Young, Jean-Francois Crespo, and Dan Dennison. 2015. Hidden technical debt in Machine learning systems. In *Proc. of the 28th Int'l Conf. on Neural Information Processing Systems - Volume 2* (Montreal, Canada) *(NIPS'15)*. MIT Press, 2503–2511.