Applications of MLOps in the Cognitive Cloud Continuum

Sergio Moreschini[1][0000−0002−5582−9487]*

Tampere University, Tampere, Finland
sergio.moreschini@tuni.fi

Abstract. Background. Since the rise of Machine Learning, the automation of software development has been a desired feature. MLOps is targeted to have the same impact on software development as DevOps had in the last decade.
Objectives. The goal of the research is threefold: (RQ1) to analyze which MLOps tools and platforms can be used in the Cognitive Cloud Continuum, (RQ2) to investigate which combination of such tools and platforms is more beneficial, and (RQ3) to define how to distribute MLOps to nodes across the Cognitive Cloud Continuum.
Methods. The work can be divided into three main blocks: analysis, proposal and identification, and application. The first part builds the foundations of the work, the second proposes a vision on the evolution of MLOps then identifies the key concepts while the third validates the previous steps through practical applications.
Contribution. The thesis's contribution is a set of MLOps pipelines that practitioners could adopt in different contexts and a practical implementation of an MLOps system in the Cognitive Cloud Continuum.

Keywords: Software Engineering · Machine Learning · MLOps.

1 Introduction

DevOps [2] is defined as a set of practices to encourage collaboration between application development and IT operations teams. The main purpose of DevOps is to ensure fast release of quality software changes and operating resilient systems. DevOps methodology has become a core concept of the software development lifecycle for practitioners and with the increasing adoption of Machine Learning (ML)-based software the methodology needs to be extended to include the ML development steps that differ from the original software development. The process of including an ML pipeline when developing software needs to be addressed so that the new software system will ensure both long-term maintainability and adaptable nature. These requirements are due to the hybrid nature of such ML-based as the long term maintainability is inherited from the DevOps practices,

* Supervisors:
David Hästbacka, Tampere University, david.hastbacka@tuni.fi
Davide Taibi, University of Oulu, davide.taibi@oulu.fi

while the adaptable nature is achieved through continuous training of new data continuously provided to the ML algorithm. For this reason, such extension is categorized as an evolution of the classical DevOps and denominated MLOps [7].

With the increasing availability of devices connected to the Internet and the ability to generate data, MLOps has the potential to become the reference model to develop software capable of detecting anomalies, projecting future trends, augmenting intelligence and so much more. However, as most of these devices composing the environment have limited computational power it is important to investigate also how to develop applications along the so-called COgnitive CLoud CONtinuum (COCLCON).

The main goal of this thesis is to study the most common approaches when developing ML-based software in the COCLCON. In this work I will attempt to answer the following research questions:

RQ$_1$ Which MLOps tools and platforms can be used in the COCLCON?
RQ$_2$ What combination of MLOps tools and platforms can be used in the COCLCON optimized pipeline?
RQ$_3$ How to distribute MLOps across the COCLCON?

2 Background

The concept of MLOps is a new hype in academic literature [3]. Even if the problem of automating ML applications was firstly addressed in 2015 [14], the first mentions of the term MLOps itself are from 2018. In the last 4 years, the engagement with the topic grew exponentially, so that at the time of writing there are more than 200 million projects adopting ML on GitHub [15]. Consequently multiple works, both in white and grey literature, tried to define their vision on the concept of MLOps, but many of them differed on multiple aspects mostly related to the pipeline [15] [6] [10]. One of the main goals of this work is to propose a pipeline which has strong literature foundations, takes into account common practices and the state-of-the-art of MLOps projects and, most important, validates it through practical applications.

2.1 MLOps

Software development has seen its last revolution with the introduction of DevOps. The methodologies proposed by DevOps helped companies to improve results and create a culture based on two fundamental factors: the increased frequency of software releases and the reliability of the produced software. These two factors that once seemed to be opposite of each other started not only to coexist but also to grow together following the dynamic nature of DevOps practices. Such dynamic nature has been represented through the iconic DevOps pipeline which aims to portray the division of application Developers (Dev) and IT Operations (Ops) tasks in teams as an infinite loop.

The increased adoption of ML-based software has created a new figure in the corporate organizational environment: the ML developer. Such a figure actively

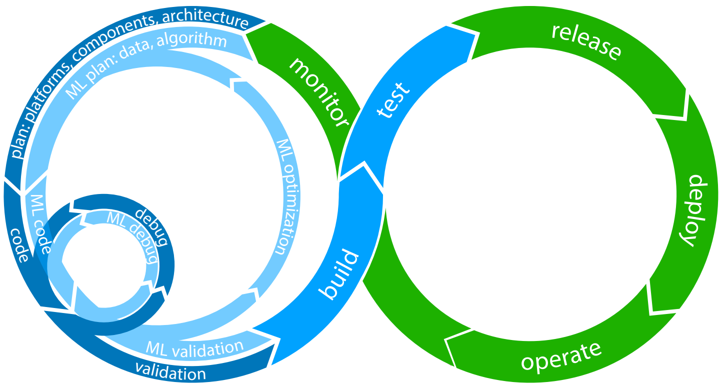

Fig. 1. Proposed MLOps pipeline [10]

participates in the development of the software, performing tasks that are parallel to the Dev engineer. The natural evolution of the development cycle for agile software, and therefore of the DevOps pipeline, which includes the development of ML-based software has been defined as MLOps.

The graphical representation for MLOps proposed in [10] is depicted in Figure 1. Such representation aims at highlight the diversification yet affinity when developing the ML-based software from the software developer and the ML developer perspectives. The main differences between the proposed MLOps pipeline and the original DevOps lie in the Plan and Code phases, moreover, a subsequent phase has been added and defined as Validation.

2.2 Cognitive Cloud Continuum

Another important aspect to take into account is where to deploy the ML-based software. One of the most recent hypes in the cloud computing domain is the concept of Cloud Continuum, which together with the concept of Cognitive Cloud has raised interests of funding agencies [1].

The first definitions of Cloud Continuum were presented in 2016 [5], [4]; while the first one presented it as a *"continuum of resources available from the edge to the cloud"* the second focused on computationally related aspects. Since then, more than 30 definitions have been proposed for Cloud Continuum. The definitions have focused on the distribution of resources both from the point of view of the entity responsible for the computation and of the computational power.

The term cognitive was originally used in computer science to refer to the behavior of computers analogous or comparable to the human brain. In the 2010s, “cognitive computing” became the new research trend aiming to develop novel systems relying on extensive amounts of data from many sources. With the advent of the era of big data, the increasing amount of unstructured data caused

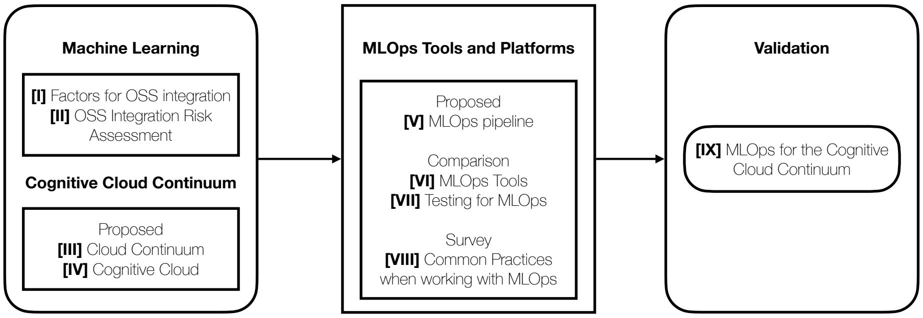
Machine Learning
[I] Factors for OSS integration
[II] OSS Integration Risk Assessment
Cognitive Cloud Continuum
Proposed
[III] Cloud Continuum
[IV] Cognitive Cloud
MLOps Tools and Platforms
Proposed
[V] MLOps pipeline
Comparison
[VI] MLOps Tools
[VII] Testing for MLOps
Survey
[VIII] Common Practices when working with MLOps
Validation
[IX] MLOps for the Cognitive Cloud Continuum

Fig. 2. PhD structure

problems in information analysis and processing; in this scenario, cognitive computing provided solutions by imitating the human way of thinking.

Analyzing the evolution of both terms, the Cognitive Cloud Continuum is moving towards an extension of the traditional Cloud towards multiple entities. Such entities not only are capable of providing data, store it and processing it, but they are also capable of sensing the environment and, by learning from it, they can adapt the computational load.

3 The Proposed Approach

The structure of the PhD is depicted in Figure 2. It is composed by 3 main steps, divided in 9 sub-steps, that might be submitted as individual publication:

- Step 1: **Analysis of the literature** for ML in the COCLCON
- Step 2: **MLOps: platforms, tools, methods and processes**
  - Proposed MLOps pipeline
  - Comparison of testing and tools
  - Survey on the impact of MLOps in the industry domain
- Step 3: **Applications of MLOps in the COCLCON**
  - Investigation of MLOps tools usage in the COCLCON
  - ML distribution to the different nodes of the COCLCON

The research method is based on both empirical methodology and practical applications. The empirical methodology includes systematic literature reviews, case studies, surveys, and interviews. Starting from the analysis of the literature I aim at answering RQ1 in the first step. To answer RQ2 I make use of the aforementioned empirical studies; the goal is to provide clear pipeline proposals by finding the optimal combination of tools used at each step of the MLOps pipeline. RQ3 revolves around the concept of Cognitive Cloud Continuum, therefore a fundamental part is the definition of the two concepts composing it. Following this, I aim at investigating MLOps tools and their usage in this particular environment to practically develop an MLOps system in the third step.

Step 1: Analysis of the literature The implementation of ML models strongly relies on the capability of importing Open Source Libraries in the same way that Open Source Software (OSS) has been integrated into commercial products. When talking about OSS it is important to estimate factors and metrics to evaluate its reliability of it before embedding it [9]. Some key points desired when integrating OSS are continuous updates and maintainability and based on these it is possible to calculate the risk of abandonment [8]. Once the properly available libraries have been selected the development of the software can begin.

Another important aspect to take into account is the device on which the software needs to be developed, how the calculation needs to be carried out, and in which environment such device is [12] [11]. The analysis of the literature focuses on these aspects which are the foundation on which the development of the software lies.

Step 2: Identification of methods and tools. Developing software that relies on ML techniques necessitates a different approach when compared to normal DevOps. Among the various reasons, there is one based on the need to include the figure of the ML developer who needs to develop the system in parallel with the software engineer. For this reason, an extension of the DevOps pipeline is required [10].

Once the pipeline is clearly stated, it is important to analyze the state of the art of the current tools for ML-based projects and how they are used [13]. Particular attention needs to be placed on those tools used for testing the overall systems [15]. Furthermore, it is also critical to investigate practitioners' common practices when working with such ML-based systems.

Step 3: MLOps in the COCLCON. In the last step, I aim at using the knowledge acquired in the previous steps to provide an MLOps system capable of delivering applications along the COCLCON.

4 Current Status

The Research work started in January 2021. During this period I investigated Step 1, and Step 2.1. Step 1 consisted of four different works [9] [8] [12] [11] aiming at answer RQ1.

Step 2.1 is the first result towards answering RQ2 and has been achieved through the proposition of an MLOps pipeline published in [10]. The work, not only envisions a pipeline for MLOps but also produces a meaningful comparison to classical DevOps. The publication is the first part of a roadmap for the development of MLOps practices. At the stage of writing, I am currently developing the second part of this step which will result in two publications.

As for the remaining steps I am currently collaborating with different partners to investigate how to properly address problems related to RQ3.

5 Expected Contribution

The main contribution of this thesis is to analyze the evolution of the development process of MLOps, particularly applied in the Cognitive Cloud Continuum. The main contribution of this thesis will be a validated set of MLOps pipelines that companies can adopt in different contexts, together with their pros and cons.

References

1. Cognitive cloud: Ai-enabled computing continuum from cloud to edge (ria) (2022), https://ec.europa.eu/info/funding-tenders/opportunities/portal/screen/opportunties/topic-details/horizon-cl4-2022-data-01-02, accessed: 2022-07-07
2. Bass, L., Weber, I., Zhu, L.: DevOps: A software architect's perspective. Addison-Wesley Professional (2015)
3. Calefato, F., Lanubile, F., Quaranta, L.: A preliminary investigation of mlops practices in github. In: IEEE ESEM '22. p. 283–288 (2022)
4. Chiang, M., Zhang, T.: Fog and iot: An overview of research opportunities. IEEE Internet of things journal **3**(6), 854–864 (2016)
5. Gupta, H., Nath, S.B., Chakraborty, S., Ghosh, S.K.: Sdfog: A software defined computing architecture for qos aware service orchestration over edge devices. arXiv preprint arXiv:1609.01190 (2016)
6. Gupta, S.C.: Mlops: Machine learning lifecycle. https://towardsdatascience.com/machine-learning-lifecycle-in-mlops-era-5b45284c0e34 (2022)
7. John, M.M., Olsson, H.H., Bosch, J.: Towards MLOps: A framework and maturity model. In: Euromicro / SEAA (2021)
8. Li, X., Moreschini, S., Pecorelli, F., Taibi, D.: Ossara: Abandonment risk assessment for embedded open source components. IEEE Software **39**(4), 48–53 (2022)
9. Li, X., Moreschini, S., Zhang, Z., Taibi, D.: Exploring factors and metrics to select open source software components for integration: An empirical study. Journal of Systems and Software **188**, 111255 (2022)
10. Moreschini, S., Lomio, F., Hästbacka, D., Taibi, D.: Mlops for evolvable ai intensive software systems. In: SQ4AI@SANER (2022)
11. Moreschini, S., Pecorelli, F., Li, X., Naz, S., Albano, M., Hästbacka, D., Taibi, D.: Cognitive cloud: The definition. In: DCAI (2022)
12. Moreschini, S., Pecorelli, F., Li, X., Naz, S., Hästbacka, D., Taibi, D.: Cloud continuum: The definition. In: Under Review (2022)
13. Recupito, G., Pecorelli, F., Catolino, G., Moreschini, S., Nucci, D.D., Palomba, F., Tamburri, D.: A multivocal literature review of mlops tools and features. In: SEAA (2022)
14. Sculley, D., Holt, G., Golovin, D., Davydov, E., Phillips, T., Ebner, D., Chaudhary, V., Young, M., Crespo, J.F., Dennison, D.: Hidden technical debt in machine learning systems. In: NIPS. vol. 28. Curran Associates, Inc. (2015)
15. Symeonidis, G., Nerantzis, E., Kazakis, A., Papakostas, G.A.: Mlops - definitions, tools and challenges. In: 2022 IEEE CCWC. pp. 0453–0460 (2022)