Abstract

Continuous integration lays the foundations for building software applications. The CI system aims to use an automated process for validating the software updates to ensure a smooth transition for software production. Machine learning applies training on different datasets to achieve a specific result. CI system aims to use the commit history on a repository to help the programmers to learn what's causing the changes to the code. While researchers have done some work on CI systems in the past, there were significant areas for improving the process of using CI to build ML code project repositories.

1 Introduction

- Describe CI

  Karlaš, B., Interlandi, M., Renggli, C., Wu, W., Zhang, C., Mukunthu Iyappan Babu, D., Edwards, J., Lauren, C., Xu, A., & Weimer, M. (2020). Building Continuous Integration Services for machine learning. *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. https://doi.org/10.1145/3394486.3403290

  POTGIETER, T. R. E. N. T. O. N. (2022). *Automated Machine Learning on Aws: Fast-track the development of your production-ready machine... learning applications the Aws way*. PACKT PUBLISHING LIMITED.

- Describe Machine Learning
- Why integrate ML in CI?
- Papers' existing work on ML and CI

- ☑ ~~Give an outline for the problem of building an ML CI system.~~
- ☐ Identify why I'm building the project by finding out what's interesting about the project.
- ☐ Explain the challenges facing building an ML CI system.
- ☐ Provide a context for building an ML CI system.
- ☑ ~~List the challenges for an ML CI system.~~

1.1 Project Description

My project aims to automate the ML model testing to validate new updates. If the ML training model passes the benchmark assessments, then the application will execute downstream

processes. In addition to the technical aspect of the project, I will conduct an extensive literature review using multiple sources to lay the foundations for extending the current work on the ML CI system. Machine Learning is becoming a more popular application as it's becoming more crucial for developers to simplify their development and regularly check through their progress.

The paper aims to address some of the problems facing ML development from a lack of depth explored into building ML CI systems to the limitations in the system's test cases.

The first paper on building an ML CI system is the paper "Building CI Services for ML (2020") providing the initial attempts to develop a "CI for ML" service.

Why is the project an interesting topic to consider?
It's an exciting prospect to expand upon the new field of research for building ML CI systems. ML has plenty of areas for future applications in predictions and improving existing systems, therefore it is vital to have a system acting as a quality maintenance software to constantly check and restructure the software.

Should I talk about my experiences working with ML?

The biggest challenges facing building an ML CI system:

- There are a limited number of CI Services for building ML applications.
- It's more difficult to recycle the test cases for the ML CI systems compared with the traditional CI systems. ML testing releases information about the test data, which must be kept from the user to prevent possible leads to overfitting.

Introducing the CI/CD methodology

The CI/CD pattern has become a very popular methodology to automate the development and release of software into production. The main idea behind this practice is to make incremental, reliable, and frequent software code changes, and then deploy these changes automatically and seamlessly into production.

Although researchers have done plenty of prior works on the CI system, it's important to apply CI in automating the ML deployment processes. The goal of the CI System is to allow a programmer to deploy the application into the market after continuous verification processes for each update.
My project aims to automate the testing to help to build new models. The project seeks to use a website platform where you can enter data and make predictions. Using the website, the user can check whether the code is correct. Whenever the user makes new changes to the data, the program will reset the machine learning development cycle by rebuilding the application. Each build will train and test the model and determine whether the tests passed. A continuous

integration system is a sandbox environment for testing the model before moving the program onto the deployment stage. There are many areas for the ML continuous integration system: to improve the models in the financial forecast sector.

Currently, there are areas for improvement in the research for building pipelines to automate the continuous ML system [**On Continuous Integration / Continuous Delivery for Automated Deployment of Machine Learning Models using MLOps**]. My project aims to apply an automatic pipeline process to configure the ML training and testing steps.

Context: present project web servers run several ML methods. We need to create a server to automate the running of the ML methods.
The project breaks down into several key concepts:

1. ML Life Cycle: highlights the development cycle for improving an ML model. The developer will use the training and testing ML data files for evaluating the model. Understanding the ML life cycle helps you with enhancing the model validation process.
2. ML Continuous Integration system: provides a platform to take the ML Code and datasets to run automated training. If the tests pass, the platform will execute user-defined downstream processes.

1.2. Goals

The project focuses on validating the models with an external source control system (GitHub Actions). After the validation, a user can deploy the model on external sources,e.g., a website, app store or cloud service.

1. Review and configure the ML pipelines for training.
2. Integrate GitHub to monitor the code changes.
3. Implement a tracking method on the datasets.
4. Any changes to the code or dataset will run the ML training pipeline.
5. Optimize for the best ML model by comparing new scores with historical ML model scores.
6. Run automated steps (such as sending an email or pushing code to servers) when ML benchmarks pass or fail targets.

1.3 Project overview

My initial focus aims to produce a literature review to determine how to improve software development. Before January 2023, the main focus is on researching different existing ML CI papers to identify what is needed for expanding the current functionalities and provide the structure for writing the final report. After January 2023, the focus will shift to coding and testing the application for compiling the results into the reports.

1.4 Report Structure

The report will begin with contexts provided on “Machine Learning” and “Continuous Integration”. Afterwards, I outline the CI development process. The requirements section will follow the context section to plan for the subsequent design and implementation of the ML continuous integration system. Implementing the system will help to establish a conclusion and the evaluations to identify the areas for future work.

2. Background

2.1 Purpose

This section aims to define the prior knowledge from the implementation of CI systems in Software Development and Machine Learning areas. Using the background knowledge to identify the areas for improving the existing software for a more successful ML CI system. The organisations increased the funding for AI and ML systems according to a study done by algorithma (Oppenheimer 2021 enterprise trends in ML), therefore applying an ML CI system has plenty of potentials.

Oppenheimer, D. (2021). *2021 enterprise trends in ML - Algorithmia*. Info Algorithmia. Retrieved December 11, 2022, from

2021 enterprise trends in machine learning

2.2 Machine Learning

More and more developers are applying ML for processing information and ML is a constantly growing industry. ML’s most significant limitation is that there are a lot of models that fail to make it to the deployment stage. Ranawana and Karunananda estimated that 87% of the ML projects failed to release in the market. It is critical to apply a method to successfully implement an application to automate the verification of the ML model structure. Ranawana’s report stated the ML CI system will help to shorten the Machine Learning development process from 6 months. Shortening the process is attributed to addressing the issues of deploying, scaling, versioning, and data collection.

Ranawana, R., & Karunananda, A. S. (2021). An agile software development life cycle model for Machine Learning Application Development. *2021 5th SLAAI International*

Conference on Artificial Intelligence (SLAAI-ICAI). https://doi.org/10.1109/slaai-icai54477.2021.9664736

Link the project build with the ML life cycle.

2.3 Continuous Integration

Define continuous integration system: a continuous integration system is an application that automates the testing whenever a change has been made to the input code or code datasets. The integration process is ongoing during the build stages for the application and rebuilds whenever a programmer makes a commit to the repository.

Continuous integration builds the software whenever a developer makes an update (Continuous Integration (Improving software quality and reducing risk) Chapter 1). The goal is to run the CI server regularly and to provide a stage for presenting the results. CI server aims to reduce the custom scripts for running different components.

2.4 Pipeline

A pipeline aims to configure and automate the continuous integration processes, which are split into different stages. A pipeline can help us with tracking through the different stages of the application. It becomes easier for the user to identify the stages, which the pipeline is causing an error. Garg, Pundir, et al's research found that pipeline development is helpful for the developer to reuse ML code for the pipeline stages.

ML life cycle

The ML life cycle helps to provide guidance on how to build the best ML models. The application aims to automate the iteration process in the ML CI system to select the model with the maximised training and testing scores.

Continuous integration builds the software whenever a developer makes an update [3]. The goal is to run the CI server regularly and to provide a stage for presenting the results. CI server aims to reduce the custom scripts for running different components. A build script is an example of a custom script to help to compile and build new applications. The developer can apply the build script for more generalized scenarios, e.g., outside a continuous integration system. Continuous integration requires testing data as an essential component. In traditional models, the program feedback on the test accuracy scores to the developers to improve the model. Testing scores

may cause overfitting because they may mislead the user towards a skewed result. A possible solution is to allow the CI system to select independent testing data for each training process. In my project, the developer will only receive the pass or fail statements as feedback. CI system reruns whenever there's an edit. Tracking the ML software and dataset edits help the customer identify the point to roll back when making mistakes during the development process. A developer should focus on testing and releasing stages of the ML development cycle for designing the ML continuous integration system.

I will provide a section that provides a timeline for continuous integration. The timeline will examine how continuous integration has shifted from software development to machine learning. In 2012, a research paper applied continuous integration to help to build new software applications. It wasn't until 2020 that researchers investigated building a CI system for Machine Learning applications.

2.1 Automated Workflow

External platforms identify the context for building a CI system and reduce the workload for designing the application [4]. GitHub Action is fantastic for applying to repositories by analyzing the users' repository history:

1. Only a small proportion of the users (0.7% of the 416,266 repositories [4]) used GitHub Actions in their projects. There's plenty of potential for exploring integrating GitHub Action with a CI system because GitHub Action isn't a standard tool for automating. The developers can use the GitHub Action history to understand the most typical workflow for planning the projects.

2.2 CI Server

A CI server will help to automate the tasks and takes the structure as follows:

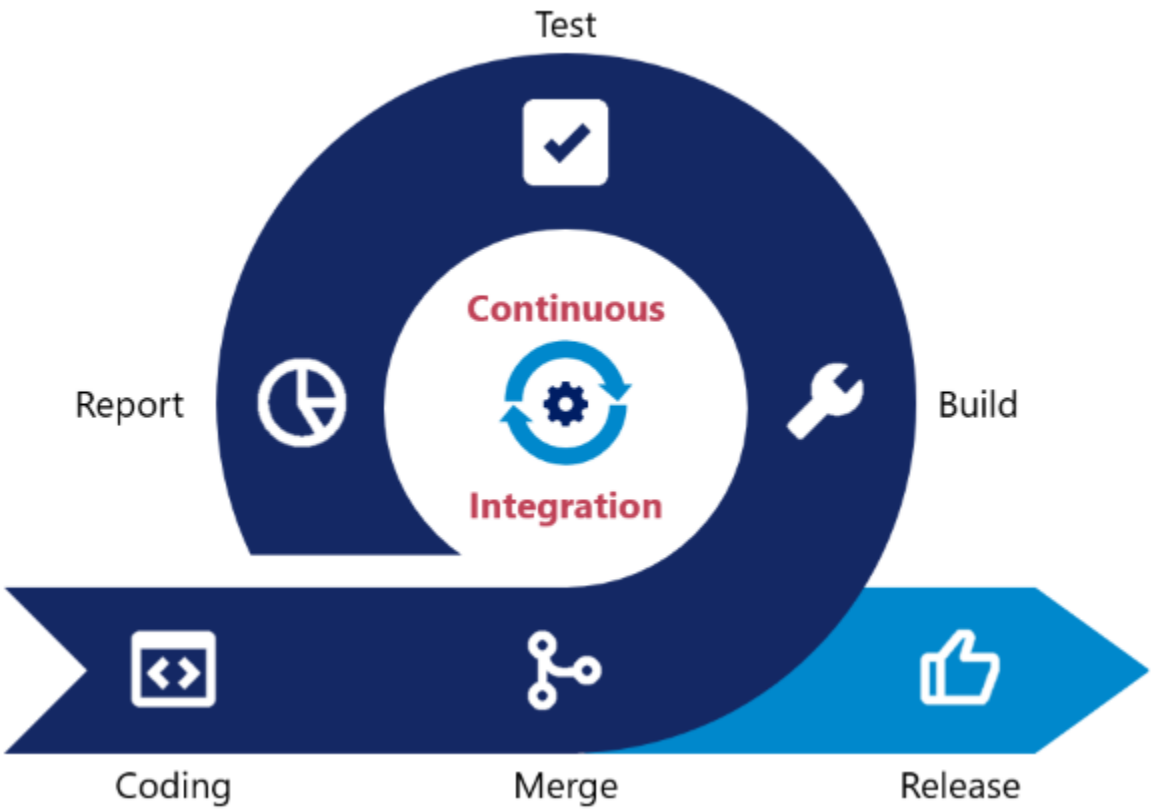
- Continuous Change Impact Analysis Process [5] will help to identify the links between the different components.

2.3 A developer's work cycle.

A developer's work cycle provides the framework for the continuous integration system's purposes. When working on a project, a developer reviews any changes made to a branch before committing the latest changes. A continuous integration system automates the commit process to determine whether the application will accept the model.

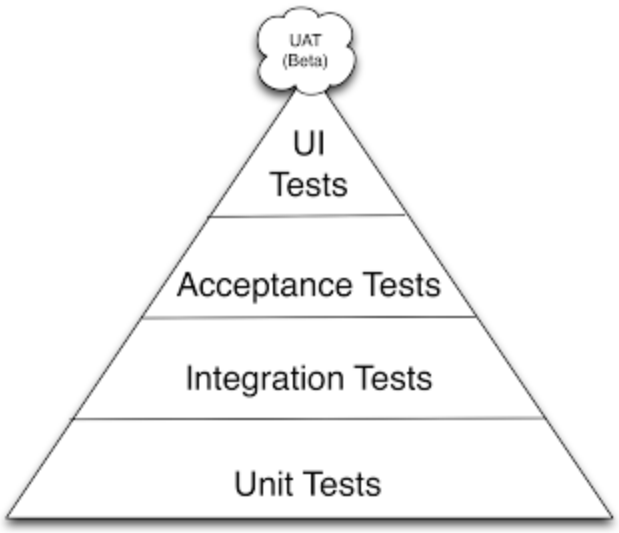
3. Notes

28-11-2022



The integration looks at the **CI cycle**:

1. The developer commits code or updates data on the repository. The CI examines the repository for any changes.
2. The CI identifies the changes and rebuilds the program for automated testing. The test will determine whether the changes will be updated.
3. The CI generates a feedback report from the test results and shares it with the project participants.



Using the iterative CI scenario, we can include the testing component in the continuous integration system. You can apply a CI service to configure the testing components into the CI system. An alternative method would be to design your tests for the critical components. Identifying the code coverage helps with determining the untested code sections. For every new function, writing new tests is the best approach to improve test coverage.

4. Next steps

Research other papers to increase the variety of references.

1. Academic content for determining whether anyone conducted prior research on the topic.
2. Look for existing software packages on the ML continuous integration tool.

5. Sources

1. https://github.com/iterative/cml
2. Building Continuous Integration Services for Machine Learning
3. Continuous Integration (Improving software quality and reducing risk) Chapter 1
4. How Do Software Developers Use GitHub Actions to Automate Their Workflows?
5. Communicating continuous integration servers for increasing the effectiveness of automated testing [2012]
6. Pilato, C. Michael, Ben Collins-Sussman, and Brian W. Fitzpatrick. Version Control with Subversion: Next Generation Open Source Version Control. O'Reilly Media, Inc., 2008.

Search site: https://www.engineeringvillage.com/search/quick.url?SEARCHID=206b621c86f64a33a06ec50204a5635d&COUNT=1&usageOrigin=&usageZone=

5.1 Core functionalities

Dissect the articles using the core functionalities.

- Ability for the user to configure ML training pipelines (steps of code that train a Learner)

Building Continuous Integration Services for Machine Learning (*P2: 1.1 Production Requirements - ML Life Cycle*): encompasses the Machine Learning Life Cycle to determine the users' requirements. Using the requirements, it is possible to set the criteria for the application on how to evaluate the ML models. The paper explores the options for the user to configure the ML input for the CI system. The study has the problem of overfitting where the testing errors may incorrectly redirect the user towards a specific result.

While the user can configure the ML training pipelines, the paper states that the pipelines are run in an automatic process.

[**Advances, challenges and opportunities in creating data for trustworthy AI (P3)** and **Building Continuous Integration Services for Machine Learning (*P2: 1.1 Production Requirements - ML Life Cycle*)**] differs in approaches on pipeline configuration:

- **Advances, challenges and opportunities in creating data for trustworthy AI** uses performance and datasets as criteria to determine the training pipeline

On Continuous Integration / Continuous Delivery for Automated Deployment of Machine Learning Models using MLOps : linking artificial intelligence to help with building a continuous integration system. The paper acknowledges a limited number of studies conducted into building a CI system.

- Ability to monitor one or more code repositories for changes (i.e. GitHub, GitLab, etc...)
  [Building Continuous Integration Services for Machine Learning (*P3*)]

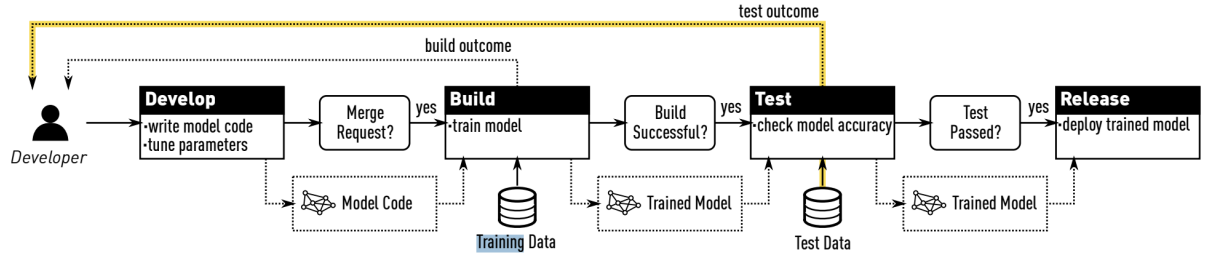


Figure 3: The development lifecycle of a machine learning model in the framework of traditional software development. The shaded yellow line depicts the information leakage pathway that our method tries to solve.

The study focuses on monitoring the code from one repository: an area for improvement would be to integrate the code for monitoring the changes across the different repositories. I would consider implementing an application to track the changes across multiple repositories: tracking changes on different repositories has helpful potential in large-scale projects. In a large scale project, the users work on each functionality in each repository.

Notes: systems can be scalable and run the code from the different machines on a larger scale.

- Ability to monitor a data repository for changes to datasets

[Building Continuous Integration Services for Machine Learning (*P4*)] applies dynamic testing sets to ensure that the testing cases were representative and aims to reduce the degree of overfitting in the model.

On Continuous Integration / Continuous Delivery for Automated Deployment of Machine Learning Models using MLOps :the ML model will monitor the data continuously to maintain the model quality because when the machine runs more data, the model quality deteriorates over time. Implement code to check the quality for the CI system: I will implement a function to reset the CI system or included multiple frameworks to run the different applications and datasets

- Ability to run an ML training pipeline in response to changes in either code or datasets

[Building Continuous Integration Services for Machine Learning (*P9*) - Related work] Generate an automated process for running the ML pipeline.

- Ability to store/compare ML results with the previous run

[Building Continuous Integration Services for Machine Learning (*P2*) - Figure 2]

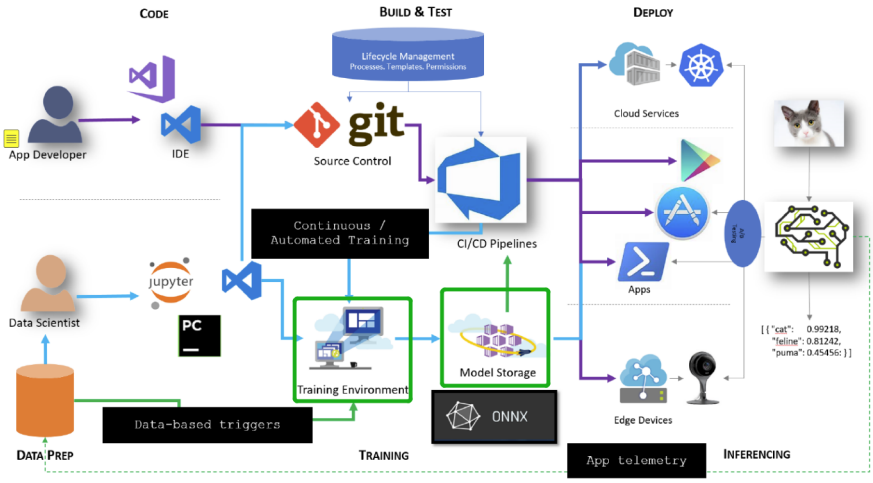


Figure 2: A macroscopic view of where a CI/CD service stands in modern ML application development lifecycle.

Implement an additional stage between the training and the pipeline to store the training scores.

- Ability to trigger automated actions based on the training results.

  The training results trigger a feedback system for the users (via email). In addition to the pass/fail detail, the user receives visualized results format.

On Continuous Integration / Continuous Delivery for Automated Deployment of Machine Learning Models using MLOps: implement a container for the dataset.

Problem: the ML model will deteriorate when the model encounters more datasets. Implement simulation methods for a timeframe to determine whether the user's model will worsen over time.