Towards Easy and Scalable Deployment of Machine Learning Models for Medical Data Analysis

Christian Steinmeyer*, Dr. Lena Wiese†
Research Group Bioinformatics
Fraunhofer Institute of Toxicology and Experimental Medicine
Hanover, Germany
{*christian.steinmeyer, †lena.wiese}@item.fraunhofer.de

I. Introduction

Machine Learning (ML) plays an increasing role in digital health with an increasing number of applications relying on Artificial Intelligence (AI) as part of their functionality [1]. Recent years brought many improvements for data science workflows, especially for the development phase. With an increased level of sophistication in those toolsets it has become significantly easier to incorporate ML models into applications. Especially in the cloud computing domain, extensive tools like *Amazon SageMaker*[1] allow for drastic reduction in time to market through abstraction and flexibility. In the context of digital health, however, due to the sensitive nature of data and concerns towards privacy and security, service providers might not be able to use these public cloud computing services. Instead, they rely on infrastructure available only on-premise. This raises the question how ML models processing sensitive information can be deployed on-premise in an easy and scalable way.

II. Use Case Bronchoconstriction

Our current use case scenario is that of bronchoconstriction. We assess the effectiveness of bronchodilators with visual analysis in *Precision Cut Lung Slices*. As part of an optimization of the image analysis workflow, we transition from multiple manual steps to an end-to-end approach, consisting of a fully automated pipeline based on (Artificial) Neural Networks (NNs). Because the underlying data are sensitive in nature, they must stay on-premise.

III. Requirements

We identified the following requirements for an on-premise deployment workflow: First, it should be integrable with local architectures. Instead of integrating a ML model into one complex application, deploying it independently aids with separation of concern and encourages change. That is, if more data becomes available and the model needs an update, the rest of the application should not need interference. Second, it should be flexible. By minimally changing the workflow's configuration, one should be able to target different deployment systems. This allows easy transitions from development to test or production systems and more. Third, it should be able to scale out. If demand for the model service changes, it should ideally adjust automatically, but at least be adjustable through manual configuration (e.g. creating multiple instances of the same service). Fourth, it should be user friendly. As the workflow in question targets data scientists and ML researchers (vs. software engineers or deployment managers), extended knowledge about infrastructure should be helpful but not required. Finally, it should be easily maintainable, enabling smaller teams with limited resources (e. g., in a research context) without the need for a whole operations unit.

[1]Amazon SageMaker is a tool that provides help "*to build, train, and deploy [ML] models quickly*" in the cloud using Amazon Web Services (https://aws.amazon.com/sagemaker/).

IV. Approach

In order to fulfill the above requirements, we deploy trained ML models on local infrastructure as microservices offering access via a small *RESTful API*. That means we define a standardized interface through which some end user application(s) can access the model, using secure web protocols. For our use case of bronchoconstriction, we use *Flask*[2] and a NN implemented in *Python*. This microservice containing the API and model itself is deployed in a virtualized environment using *Docker* containers [2]. Thus, we can guarantee stable behavior and remain flexible.

V. Conclusion and Outlook

We described challenges in deploying ML models that process sensitive information, as well as requirements for a workflow to overcome these challenges. Our approach fulfills some, but not all of these requirements. In future work we will evaluate extensions to our setup that further help us reach the goal of easy and scalable deployment.

References

[1] A. L. Fogel and J. C. Kvedar, "Artificial intelligence powers digital medicine," *NPJ digital medicine*, vol. 1, no. 1, pp. 1–4, 2018.

[2] D. Merkel, "Docker: lightweight linux containers for consistent development and deployment," *Linux journal*, vol. 2014, no. 239, p. 2, 2014.

[2]Flask is a micro web framework written in Python (https://flask.palletsprojects.com/).