# An Architecture for Automatic ML/AI Workflow management and supervision

Peini Liu*†, Jordi Guitart*†
*Barcelona Supercomputing Center, Barcelona, Spain
†Universitat Politècnica de Catalunya, Barcelona, Spain
E-mail: {peini.liu, jordi.guitart}@bsc.es

***Keywords*—*Architecture, Automatic, Machine Learning Workflows.***

## I. Extended Abstract

Scientific computation problems have been faced with the need to analyze increasing amounts of data as part of their application workflows, and the science-based model is being combined with big data and machine learning models to solve complex problems and phenomena [1][2]. The machine learning workflow is composed of some reproducible steps that can be executed as a pipeline to build a model efficiently by saving iteration time, helping in debugging and detecting [3]. Currently, businesses and researchers are investigating and improving the methodology of developing and deploying machine learning workflows in both training and inference phases, which helps the data science team focus on their requirements and the data engineer team deploy and operate machine learning workflows efficiently and automatically [4].

This work presents an architecture for automatic machine learning workflows, which provides capabilities of monitoring and automatic management on the end-to-end life-cycle of machine learning workflows, including tracking and observing at the training stage, and releasing, monitoring, deployment, auto-detecting and infrastructure management at the inference stage. To validate feasibility, we have conducted a case study based on our architecture and deployed it in the cloud, and showed its automation.

### A. *Architecture for Automatic Machine Learning Workflows*

While working on machine learning workflows, different teams have their own focuses. The Data science team faces challenges regarding model training, especially on building and improving the model. However, the data engineer team usually works on integrating and deploying the model into production, and operating the model to provide reliable, robust, and efficient service.

The architecture for the end-to-end automatic machine learning workflows is shown in Figure 1. There are many phases and steps required to make the machine learning model in production to provide values. The top describes the steps for the data science team before a model into production. Normally, the data science team will first discover the use case and data, and then develop a machine learning workflow that contains data preparation, validation and preprocessing, as well as model training, validation and testing. Workflow manager (e.g., Scanflow) can track the metadata such as metrics and

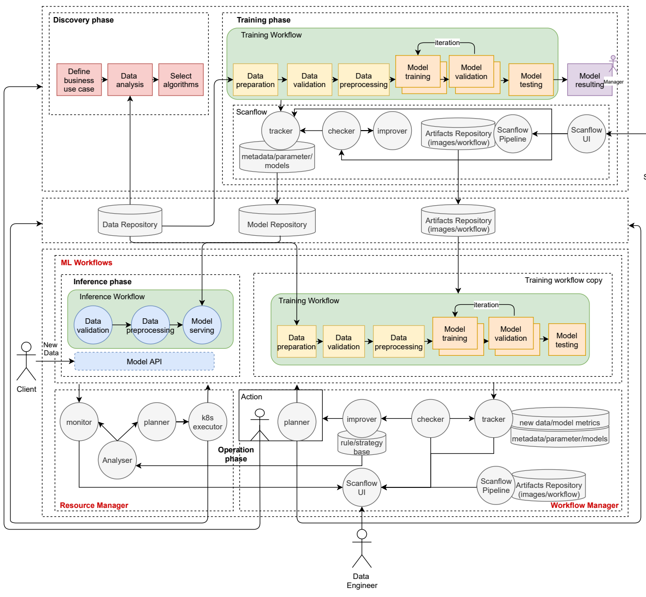

Fig. 1. Architecture for Automatic Machine Learning Workflows

scores and the artifacts during the training phase, analyze them and automatically tune the hyper-parameters, early stopping and do neural architecture search for improving the model [5].

The bottom describes the model in production, including the model inference workflow deployment and the operation phase that automatically manages the machine learning workflow from both the application layer (e.g., workflow manager Scanflow) and the infrastructure layer (e.g., resource manager Kubernetes). For deploying and managing the machine learning workflow at scale, the data engineer team should also build a workflow managed by the workflow manager but wrap and deploy the model as a service. From the application layer controlled view, the workflow manager could log the model metrics(such as scores) and artifacts(such as new data) to detect outliers, adversarial or drift and provide model explanations[6] and finally trigger the machine learning workflow to be re-trained or the model to be updated. From the infrastructure layer controlled view, allowing the model as a service helps it to be released, updated and rollouted independently, and can monitor the latency and failure rate of its predicted invocations at inference time [7][8]. With these observations, the resource manager can automatically scale the service to achieve the reliability and efficiency of the model.

### *B. A Case Study*

To illustrate the feasibility of our proposed architecture for automatic machine learning workflows, we used a leaf classification problem as a running example [9]. The automation scenarios for both training and inference phases are presented in Table I.

TABLE I. AUTOMATION SCENARIOS

| Stages | Type of events | Actions |
|---|---|---|
| training | model with hyper-parameters | different hyper-parameters tuning |
| | multiple models | model selecting |
| inference | predicting service failure rate over 90% | scale the service with more replicas |
| | the rate of the number of requests per second to the predicting service in 5 minutes over 2 | scale the service with more replicas |
| | outliers or model drift when new data comes | retain and update the model |

During this machine learning project, firstly we create a training workflow that contains data gathering, data preprocessing and model training as shown in Figure 2. The model training step uses the random-forest algorithm with different hyper-parameters tuning running in parallel. Then we end up with the best model selection based on accuracy.

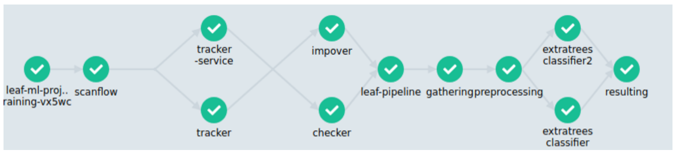

Fig. 2. Leaf training workflow

After the model training, the inference workflow is ready to be deployed in production. Figure 3 shows the inference workflow, data preprocessing and model serving are services and exposed through the cluster for clients to do invocations for predictions.

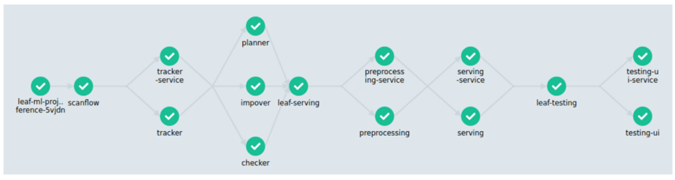

Fig. 3. Leaf inference workflow

The workflow manager(e.g., Scanflow) is started for each workflow for logging the metrics and artifacts of the workflow and debugging and managing the robustness of the model. As for the infrastructure management, we use Kubernetes as the basic container orchestration platform and Istio as a service mesh to gather all the traffic through each service and trace each invocation. Finally we use Keda event to drive autoscaling. Corresponding results are shown in Figure 4.

### *C. Conclusion*

This work presents an architecture for automatic machine learning workflows, which provides capabilities of monitoring and automatic managing on end-to-end life-cycle of machine learning workflows. The technique behind each component will continuously evolve, but we believe the architecture could be a blueprint for improving the efficiency of the machine learning model into production. The next step we will improve the management policy of the planner in each stage.

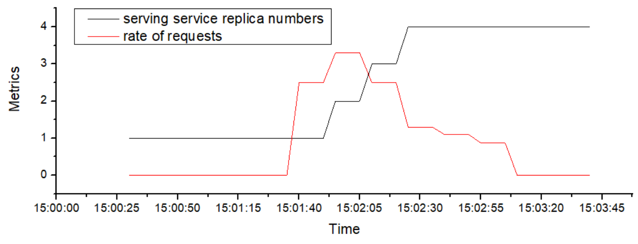

Fig. 4. Auto-scaling: The rate of the number of requests per second to the serving service over 2, system scale the service with more replicas.

## II. ACKNOWLEDGMENT

We thank Lenovo for providing the technical infrastructure to run the experiments in this paper. This work was partially supported by Lenovo as part of Lenovo-BSC collaboration agreement, by the Spanish Government under contract PID2019-107255GB-C22, and by the Generalitat de Catalunya under contract 2017-SGR-1414 and under grant 2020 FI-B 00257.

## REFERENCES

[1] NITRD Group, "The Convergence of High Performance Computing, Big Data, and Machine Learning," September 2019.

[2] M. Asch *et al.*, *Big Data and Extreme-scale Computing: Pathways to Convergence-Toward a shaping strategy for a future software and data ecosystem for scientific inquiry*, 2018, vol. 32, no. 4.

[3] kubeflow, "The machine learning toolkit for kubernetes," 2021. [Online]. Available: https://www.kubeflow.org/

[4] Seldon, "Machine learning deployment for enterprise," 2021. [Online]. Available: https://www.seldon.io/

[5] "Katib: a kubernetes-native project for automated machine learning (automl)," 2021. [Online]. Available: https://github.com/kubeflow/katib

[6] J. Klaise *et al.*, "Alibi: Algorithms for monitoring and explaining machine learning models," 2019. [Online]. Available: https://github.com/SeldonIO/alibi

[7] Kubernetes, "Production-grade container orchestration - automated container deployment, scaling, and management," 2021. [Online]. Available: https://kubernetes.io/

[8] Istio, "Connect, secure, control, and observe services," 2021. [Online]. Available: https://istio.io/

[9] Kaggle, "Leaf classification," 2021. [Online]. Available: https://www.kaggle.com/c/leaf-classification

**Peini Liu** received her M.S. degree in College of Computer at National University of Defense Technology (NUDT), in 2018. She is currently a Ph.D. student in Computer Architecture Department of the Universitat Politècnica de Catalunya (UPC) and collaborating with Emerging Technologies for Artificial Intelligence group of Barcelona Supercomputing Center (BSC). Her research interests include virtualization/containerization technologies, cloud native, resource management and the convergence of HPC, Big Data and AI.