End-to-End Automation of ML Model Lifecycle Management using Machine Learning Operations Platforms

Chung-Chian Hsu[1], Pin-Han Chen[2], I-Zhen Wu[1]
Department of Information Management[1], International Graduate School of Artificial Intelligence[2]
National Yunlin University of Science and Technology, Yunlin, Taiwan
{hsucc, m11263004, m11123005}@yuntech.edu.tw

2024 International Conference on Consumer Electronics - Taiwan (ICCE-Taiwan) | 979-8-3503-8684-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICCE-Taiwan62264.2024.10674445

***Abstract*— In machine learning research, automation presents numerous challenges, particularly in areas such as environment setup, model deployment, and maintenance, which can lead to significant time consumption and tedious tasks. To address these challenges, Machine Learning Operations (MLOps) has emerged as a solution. In this study, we propose to use the open-source Kubeflow to tackle the challenges. Kubeflow provides tools like Pipeline, Katib, and Kserve, which support tasks such as training, hyperparameter tuning, model comparison, deployment, and maintenance.The modular design of Pipeline is highly beneficial for debugging and ensuring environment consistency. The tight integration between Katib and Kubeflow enables highly automated hyperparameter tuning. Kserve addresses the issue of manual deployment, reducing the potential for human error. Our experimental result demonstrates that leveraging Kubeflow and its associated tools allows for a more streamlined and automated approach to machine learning operations, mitigating many of the challenges and labors associated with manual processes.**

***Keywords*— MLOps, machine learning, water quality dataset.**

I. Introduction

The rise of smart cities is a strategic response to the increasingly complex challenges faced by cities worldwide. With the growing demand for machine learning or deep learning model training and deployment automation in smart cities, Machine Learning Operations (MLOps) has emerged as a nascent field. MLOps is a methodology for managing the lifecycle of machine learning models, providing a unified framework covering processes from model development to deployment, maintenance, and monitoring. There are several MLOps platforms available, including Kubeflow, MLflow, Metaflow, Flyte, and MLReef [1], [2], [3]. This study opts for Kubeflow as the primary operational platform due to its open-source nature, integration, scalability, and Kubernetes foundation.

The primary advantage of MLOps lies in automating the development and deployment process, particularly in ensuring environmental consistency. This aids in addressing challenges in subsequent research stages, such as version control, data pipeline management, and model updates. This research aims to leverage the advantages of deployment on the platform to further analyze potential issues and improvement areas. The study primarily utilizes Kubeflow's internal functionalities from model development to deployment. The main objective is to understand how to train and deploy models using Kubeflow, with the ultimate goal of achieving a highly automated deployment process. Developing reliable models enables precise data analysis and prediction to be achieved. Ensuring consistent results across different environments through Kubeflow's technology contributes to sustainable smart cities.

II. Method

Fig. 1 illustrates the proposed framework for automating the development and deployment of a machine learning model in which we use to predict water quality as an illustrative example. The framework consists of 6 steps: (1) Loading the dataset. (2) Preprocessing through standardization, handling missing values, and feature selection. (3) Applying Katib for hyperparameter optimization. (4) Training with optimized hyperparameters. (5) Obtaining training results, comparing, and saving models. (6) Deployment for unseen data prediction.

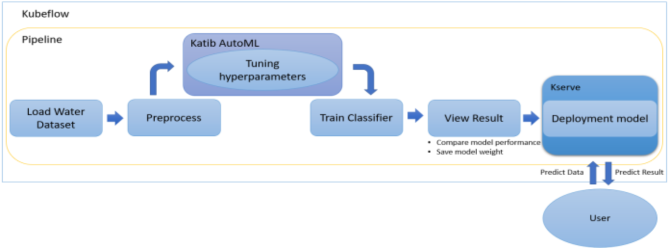

Fig. 1. Proposed framework for automating ML development and deployment.

A. Pipeline

We leverage Kubeflow's Pipeline to handle data loading, preprocessing, model training, and deployment. Each component within the Pipeline is independent, allowing us to modularize the code, making it easier to debug and update. Additionally, when code migration is necessary, packaging the project into YAML files ensures consistency across environments, addressing challenges such as environmental consistency during migration. While Pipelines may not offer significant time advantages compared to conventional training methods, their superiority lies in ensuring environmental consistency. This advantage becomes crucial for subsequent research in smart city applications.

B. Katib AutoML

This study utilizes Katib for optimizing hyperparameters, thereby automating the tuning process, saving labors and enhancing model performance. Katib's integration with

Kubeflow Pipeline facilitates seamless hyperparameter optimization and subsequent model training within the Pipeline. Katib supports various AutoML algorithms. In this research, we employ a combination of Random Search (RS) and Tree-structured Parzen Estimator (TPE) [4] for hyperparameter optimization. TPE selects promising parameters based on model predictions, while RS employs random sampling in the exploration phase to identify promising performance combinations. We switch to TPE when exploring regions of better performance. The evaluation metrics used in this study include Precision, Recall, and F1-Score.

C. *Deployment model*

Deploying models can indeed be a time-consuming process. After training, models need to be deployed to production environments. Manual deployment requires setting up environments, and discrepancies in versions may affect performance. Selecting orchestration tools to deploy containers onto chosen architectures is also necessary. However, manual deployment means infrequent model updates and decreased deployment efficiency. Automated deployment addresses these issues by swiftly deploying models. It ensures consistency in environments, reducing the likelihood of human errors.

D. *Kserve Auto deployment model*

Kserve offers efficient and systematic advantages in automated deployment. When combined with Kubeflow Pipeline, we can access performance metrics and parameters during training, and manage model versions through Kserve, enabling easy version tracking and switching. Kserve addresses drawbacks of many automated and manual deployment methods, including version control, model deployment risks, and potential human errors. Given the significant deployment requirements in smart cities, Kserve is favored for its ability to achieve large-scale deployment and facilitate model version tracking more easily.

III. EXPERIMENTAL RESULTS

In this study, we used the Water Quality dataset[1] relevant to the environmental domain from Kaggle to verify the proposed framework. This dataset comprises 10 features and 3276 records and is divided to a training and a test set with 70% and 30%, respectively.

A. *Establish the pipeline for ML model development*

A Pipeline as shown in Fig. 2 based on the proposed framework was established to simultaneously train multiple models including Decision Tree, KNN, SVM, CatBoost, and XGBoost, and Katib's AutoML was used to adjust parameters before training. The performance of the models with their respective adjusted hyperparameters is presented in Table 1. The weights of the best-performing model were stored for deployment purpose. From Table 1, it is evident that XGBoost performed the best on this dataset. Therefore, we proceeded with deploying the trained XGBoost for further applications.

After deployment, we successfully tested the model on predicting unseen data records through a URL and compared them with the ground truth.

TABLE I Prediction performance of various machine learning models

Model	F1-score	Precision	Recall
Decision Tree	0.709	0.629	0.667
KNN	0.499	0.542	0.461
SVM	0.667	0.709	0.629
CatBoost	0.720	**0.782**	0.668
XGBoost	**0.731**	0.773	**0.693**

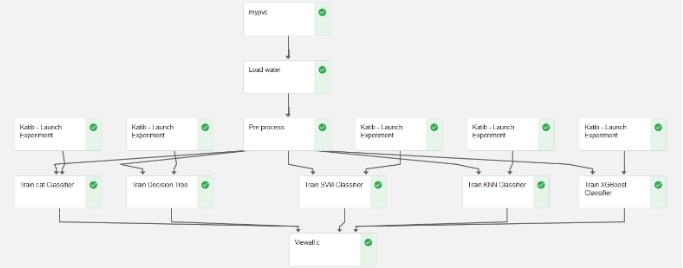

Fig. 2. The constructed pipeline for training and selection of five ML models.

B. *Automating model retraining*

In Kubeflow, we can set up scheduled jobs to perform model retraining when new data are available. Leveraging this functionality, we can integrate Docker with Kubeflow to update both the code and the dataset. During the time interval between two training sessions, code or dataset updates can be implemented, thereby facilitating code maintenance or updates.

IV. CONCLUSION

This study successfully utilized Kubeflow to automate the model development and deployment process and employed Katib for hyperparameter optimization to enhance model performance. Leveraging the functionalities provided by Kubeflow, we were able to easily deploy and utilize models, while automating model training and performance evaluation, and determining stored models for subsequent deployment. This not only increased the efficiency of model training but also reduced human errors, which are crucial for realizing smart cities. However, native Kubeflow faces challenges in updating code, requiring integration with Docker to address this issue.

REFERENCES

[1] Ruf, P., Madan, M., Reich, C., & Ould-Abdeslam, D. (2021). Demystifying MLOps and Presenting a Recipe for the Selection of Open-Source Tools. *Applied Sciences*, *11*(19), 8861. https://www.mdpi.com/2076-3417/11/19/8861.

[2] Sun, P., Wen, Y., Nguyen Binh Duong, T., & Xie, H. (2016). MetaFlow: a Scalable Metadata Lookup Service for Distributed File Systems in Data Centers. arXiv:1611.01594. Retrieved November 01, 2016, from https://ui.adsabs.harvard.edu/abs/2016arXiv161101594S.

[3] Zaharia, M. A., Chen, A., Davidson, A., Ghodsi, A., Hong, S. A., Konwinski, A., Murching, S., Nykodym, T., Ogilvie, P., Parkhe, M., Xie, F., & Zumar, C. (2018). Accelerating the Machine Learning Lifecycle with MLflow. *IEEE Data Eng. Bull.*, *41*, 39-45.

[4] Bergstra, J., Bardenet, R., Bengio, Y., & Kégl, B. (2011). *Algorithms for hyper-parameter optimization* Proceedings of the 24th International Conference on Neural Information Processing Systems, Granada, Spain.

[1] https://www.kaggle.com/datasets/adityakadiwal/water-potability