Tapis Machine Learning Hub Service for Science Gateways

Dhanny Indrakusuma, Joe Stubbs, Nathan Freeman and Anagha Jamthe
Texas Advanced Computing Center
The University of Texas at Austin
Austin, TX, USA
Email: dhannywi@utexas.edu, (jstubbs, nfreeman, ajamthe)@tacc.utexas.edu

***Abstract*—The adaptation of machine learning (ML) in scientific and medical research in recent years has heralded a new era of innovation, catalyzing breakthroughs that were once deemed unattainable. In this paper, we present the Machine Learning Hub (ML Hub) – a web application offering a single point of access to pre-trained ML models and datasets, catering to users across varying expertise levels. Built upon the NSF-funded Tapis v3 Application Programming Interface (API) and Tapis User Interface (TapisUI), the platform offers a user-friendly interface for model discovery, dataset exploration, and inference server deployment.**

***Index Terms*—machine learning, tapis, open-source, science gateways**

I. Introduction

In recent years, the integration of machine learning methodologies into scientific and medical research has propelled the boundaries of innovation, leading to remarkable breakthroughs. As highlighted in the 2024 Artificial Intelligence Index Report by Stanford University, this transformative adoption has ushered in a new era of possibilities [1].

In response to the evolving landscape of research, we designed Machine Learning Hub [2] as a user-friendly Tapis service [3] to help users with varying machine learning expertise discover and interact with pre-trained machine learning models and datasets. In addition to the details described below, we are working with the team at Intelligent Cyberinfrastructure with Computational Learning in the Environment (ICICLE) AI Institute [4] to create a federated models repository by including the machine learning models developed by researchers on the ML Commons platform into the ML Hub.

II. System Design

The Machine Learning Hub aims to streamline Tapis users' machine learning workflows. The system is divided into two distinct components: the user interface (UI) component and the server-side components. Communication between the server and the UI client is facilitated by the Flask-CORS extension [5] that handles Cross-Origin Resource Sharing (CORS), enabling cross-domain queries, and the authentication middleware uses Tapis API [6] to generate JSON Web Token (JWT) to authenticate a user's session. The server-side components consist of a set of RESTful APIs served over HTTP with Hugging Face Hub API [7] integration - consisting of the hub and the inference server, and persistent data storage. The server-side components are deployed to a Kubernetes cluster using Tapis Pods service [8]. We describe below the key functionalities of each component. An illustration of the ML Hub system can be seen in Fig. 1.

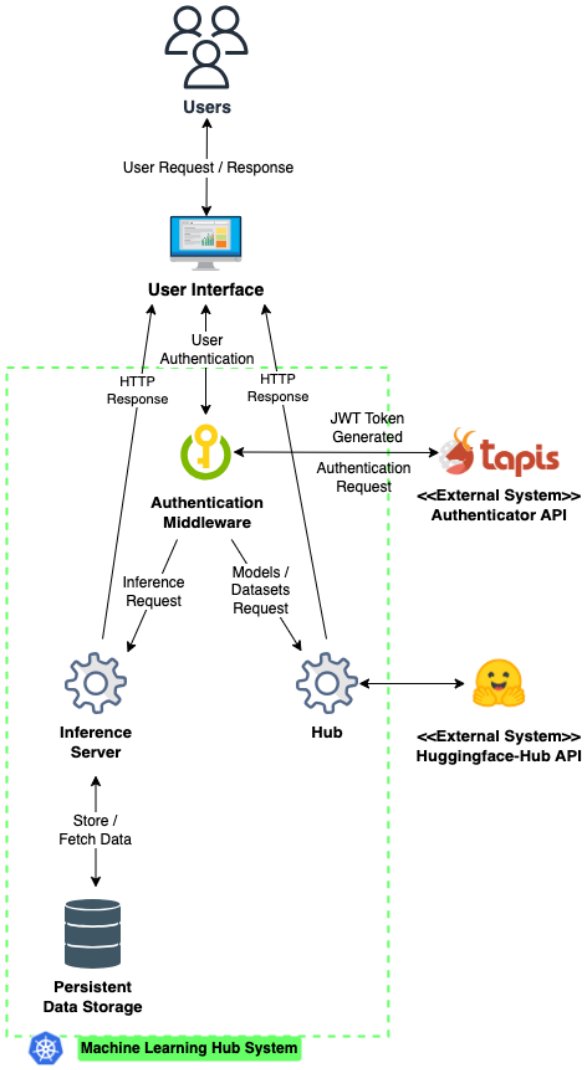

Fig. 1. Overview of ML Hub Service

A. Hub: Models and Datasets Discovery

The Hub consists of two REST APIs developed in Python Flask [9] - models hub and datasets hub. Each API has several endpoints with functionalities allowing users to view detailed popular models or datasets, filter them by specified parameters, view detailed information, and download models or datasets. We describe these functionalities in more detail below.

1) Models Hub: The models hub showcases the most downloaded machine learning models from Hugging Face [10]. In addition, it has additional functionalities such as (i) filtering models by author, task, trained dataset, query keyword, foundational libraries, and languages; (ii) fetching detailed information for a particular model and its associated model card; (iii) model download; (iv) checking the availability of inference server for a given model.

2) Datasets Hub: Like the models hub, the datasets hub allows users access to the top open-source datasets from Hugging Face and filter them by author, query keyword, task, language, size category, and official benchmark. In addition to fetching detailed information for a specified dataset and its dataset card, users also have the option to download the dataset.

B. Inference Server

Implemented using FastAPI [11], the inference server currently supports PyTorch-based [12] pre-trained models based on the FLAN-T5 (Text-to-Text Transfer Transformer) architecture [13]. The Flan-T5 is a fine-tuned T5 model that can perform various language tasks such as keyword generation and editing an English text based on the given instruction.

When running an inference, the inference server validates the user's request and then loads the user-specified model from the data storage. After processing the user's input, the server returns a JSON response containing the model's output. Users can also request that an inference server be provisioned if a model currently does not have an active inference server.

C. Persistent Data Storage

The persistent data storage uses a Network File System (NFS) volume to cache the machine learning models used by the inference server and the configuration file containing metadata of the associated models.

D. User Interface

The user interface (UI) for ML Hub is implemented using React [14] and TypeScript [15] and is currently under active development. The TypeScript types and fetch bindings for ML Hub are part of the @tapis/tapis-typescript NPM package [16]. The ML Hub module within the tapis-typescript is generated automatically from the Hub API's OpenAPI specifications [17], and the fetch bindings are then used to make API calls for the UI.

The UI for ML Hub is a service within TapisUI [18], a serverless web application built on the Tapis API. TapisUI runs entirely on GitHub pages and provides a user-friendly platform to interact with the models, datasets, and inference server. Screenshots and descriptions of the user interface can be seen in Fig. 2.

III. Target Users

The target users for this demo fall into two categories:

- Researchers with domain expertise that utilize national, campus, and local cyberinfrastructure resources who want to leverage machine learning to improve their research outcome but do not have machine learning expertise.
- Researchers with moderate machine learning expertise who are looking for a simpler way to host their model without worrying about user authentication and network issues.

IV. Session

We presented the Machine Learning Hub as a dynamic platform designed to help researchers with different levels of machine learning expertise discover and explore pre-trained machine learning models and datasets. For the session, we will begin by giving the audience a high-level overview of ML Hub functionalities and proceed with a live demo.

During the live demo, we will show how a user can utilize the ML Hub service within TapisUI to discover models and datasets, as well as interact with the inference server to run inference and initiate a model's inference server deployment. In case of a poor internet connection, we will present a pre-recorded version of the demo.

Acknowledgment

The Machine Learning Hub project is supported by the National Science Foundation Division of Advanced CyberInfrastructure Awards: 1931439, 1931575, and 2112606.

References

[1] R. Perrault and J. Clark, *Artificial Intelligence Index Report 2024*. Stanford University's Institute for Human-Centered Artificial Intelligence, 2024. [Online]. Available: https://aiindex.stanford.edu/wp-content/uploads/2024/04/HAI_2024_AI-Index-Report.pdf

[2] D. Indrakusuma, N. Freeman, and J. Stubbs, "Machine learning hub for tapis poster presentation," in *Science Gateways 2023 Annual Conference*. Zenodo, 2023. [Online]. Available: https://doi.org/10.5281/zenodo.10055681

[3] J. Stubbs, R. Cardone, M. Packard, A. Jamthe, S. Padhy, S. Terry, J. Looney, J. Meiring, S. Black, M. Dahan, S. Cleveland, and G. Jacobs, "Tapis: An api platform for reproducible, distributed computational research," in *Advances in Information and Communication*, K. Arai, Ed. Cham: Springer International Publishing, 2021, pp. 878–900.

[4] (2021) Icicle. Last access: 2024-05-31. [Online]. Available: https://icicle.osu.edu/

[5] (2013) Flask-cors. Last access: 2024-05-31. [Online]. Available: https://github.com/corydolphin/flask-cors

[6] (2020) Tapipy. Last access: 2024-03-31. [Online]. Available: https://github.com/tapis-project/tapipy

[7] (2020) Hugging face hub. Last access: 2024-03-08. [Online]. Available: https://github.com/huggingface/huggingface_hub

[8] R. C. Christian Garcia, Joe Stubbs and N. Freeman, "Tapis pods service exploration and initial performance analysis," in *Science Gateways 2023 Annual Conference*. Zenodo, 2023. [Online]. Available: https://doi.org/10.5281/zenodo.10034631

[9] (2023) Flask. Last access: 2023-4-19. [Online]. Available: flask.palletsprojects.com/en/2.3.x/

[10] (2017) Hugging face. Last access: 2024-03-08. [Online]. Available: https://huggingface.co

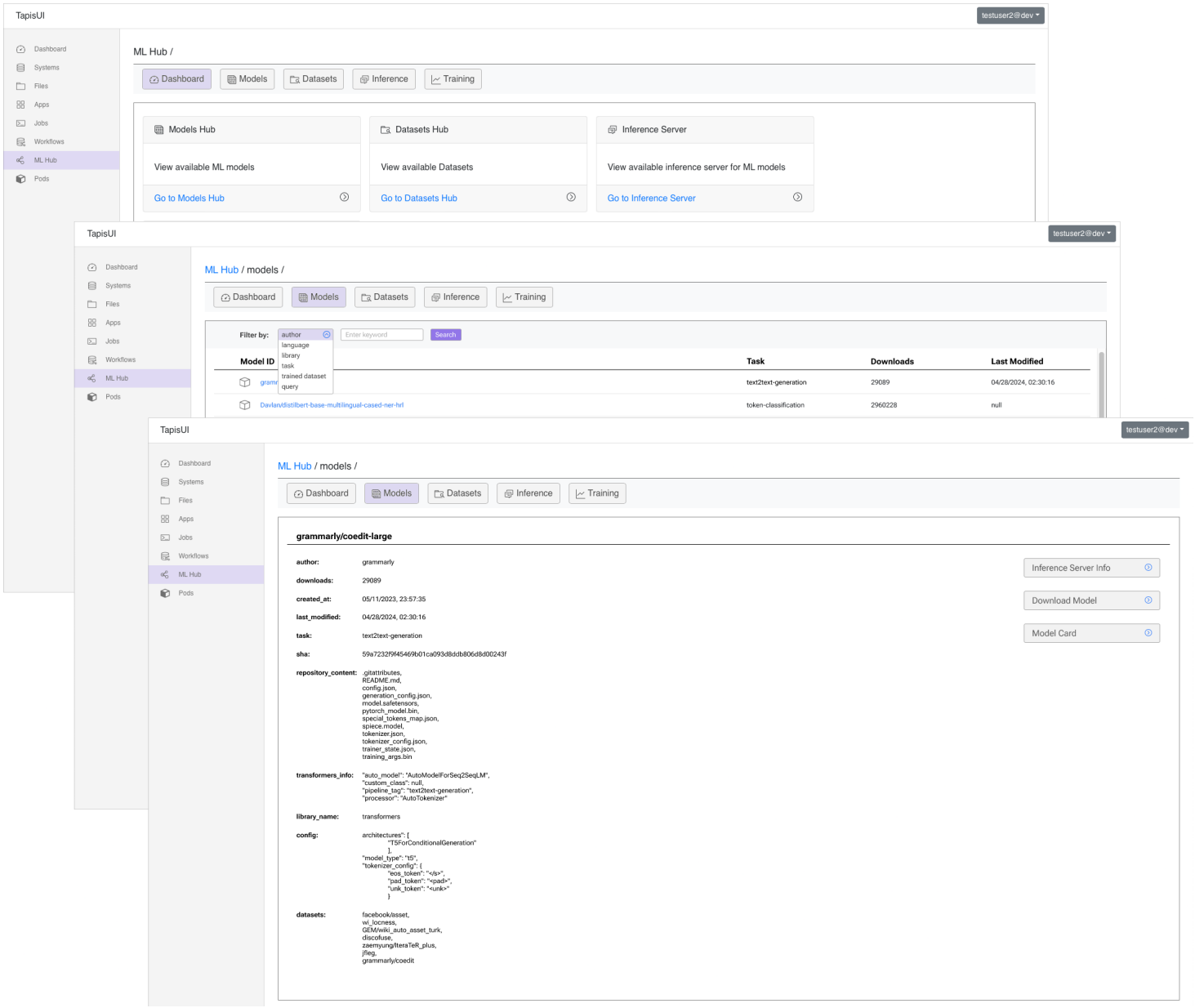

Fig. 2. User Interface for ML Hub, from top to bottom: (1) ML Hub Dashboard, (2) Top 100 models with filtering functionalities, and (3) A single model details page

[11] (2018) Fastapi. Last access: 2024-03-08. [Online]. Available: https://fastapi.tiangolo.com

[12] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[13] H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, E. Li, X. Wang, M. Dehghani, S. Brahma, A. Webson, S. S. Gu, Z. Dai, M. Suzgun, X. Chen, A. Chowdhery, S. Narang, G. Mishra, A. Yu, V. Zhao, Y. Huang, A. Dai, H. Yu, S. Petrov, E. H. Chi, J. Dean, J. Devlin, A. Roberts, D. Zhou, Q. V. Le, and J. Wei, "Scaling instruction-finetuned language models," 2022. [Online]. Available: https://arxiv.org/abs/2210.11416

[14] (2013) React. Last access: 2024-05-31. [Online]. Available: https://react.dev/

[15] (2012) Typescript. Last access: 2024-05-31. [Online]. Available: https://www.typescriptlang.org/

[16] (2021) Tapis-typescript. Last access: 2024-05-31. [Online]. Available: https://github.com/tapis-project/tapis-typescript

[17] (2017) Openapi specification. Last access: 2023-02-13. [Online]. Available: https://swagger.io/specification/

[18] J. Y. Chuah, J. Rosenberg, K. Strmiska, J. Stubbs, S. Cleveland, and J. McLean, "Tapis ui - a rapid deployment serverless science gateway built on the tapis api," in *Gateways 2021*. Zenodo, 2021. [Online]. Available: https://zenodo.org/record/5570569