AI Model Factory : Scaling AI for Industry 4.0 Applications

Dhaval Patel, Shuxin Lin, Dhruv Shah, Srideepika Jayaraman,
Joern Ploennigs, Anuradha Bhamidipaty, Jayant Kalagnanam
IBM Research
{pateldha@us., shuxin.lin@, dhruv.shah@, j.srideepika@, joern.ploennigs@ie, anubham@us., jayant@us.}ibm.com

Abstract

This demo paper discusses a scalable platform for emerging Data-Driven AI Applications targeted toward predictive maintenance solutions. We propose a common AI software architecture stack for building diverse AI Applications such as Anomaly Detection, Failure Pattern Analysis, Asset Health Forecasting, etc. for more than a 100K industrial assets of similar class. As a part of the AI system demonstration, we have identified the following three key topics for discussion: Scaling model training across multiple assets, Joint execution of multiple AI applications; and Bridge the gap between current open source software tools and the emerging need for AI Applications. To demonstrate the benefits, AI Model Factory has been tested to build the models for various industrial assets such as Wind turbines, Oil wells, etc. The system is deployed on API Hub for demonstration.

Introduction

The field of Automated Data Science (AutoDS) and Automated Machine Learning (AutoML) has significantly helped to increase the adoption of AI-based solutions in various areas such as healthcare, human resources, manufacturing industries, etc. Given a dataset (Tabular, Time Series, Images, etc.), there exists a plethora of general-purpose AI tools that turn data into meaningful AI artifacts (i.e., trained AI Model) (Patel et al. 2020a). Such AI artifact helps monitor the data in real-time and generates valuable insight such as data anomalies (Patel et al. 2022a; Zerveas et al. 2021; Patel et al. 2020b; Shrivastava et al. 2019) or upcoming failures (Patel et al. 2020c). These insights subsequently drive decision-making within business facilities and throughout their operations. So far, current work focuses on developing tools and techniques for analyzing a single dataset that originates from a single asset. However, the Industrial sector has multiple types of assets, and each asset has multiple instances. With the broader adoption of AI solutions in Industry 4.0, there is an emerging need for mass production of AI models fairly automatedly.

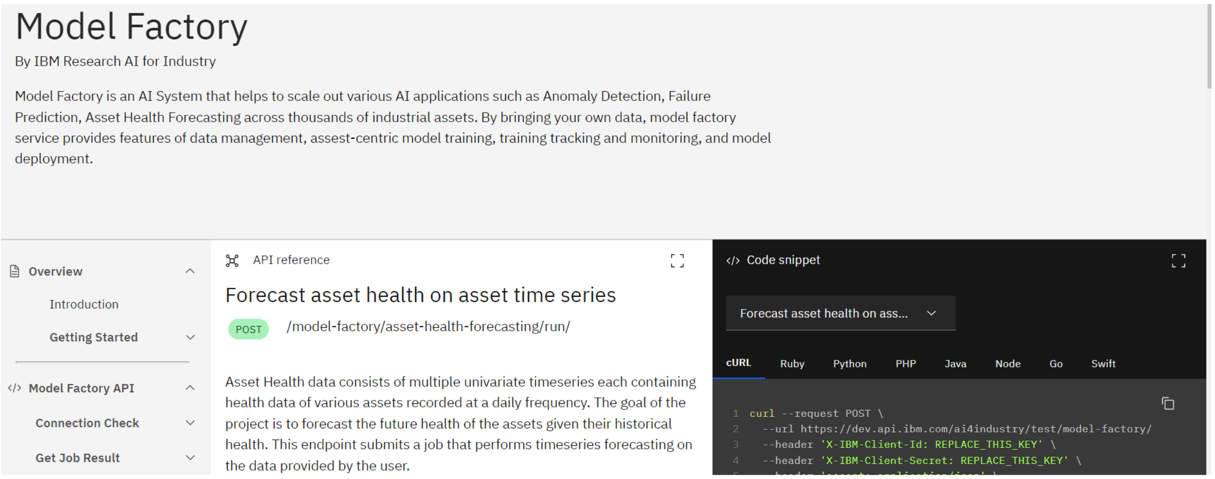
Example 1 : Automated Robot Diagnostic. The automobile company has more than 2000 robots working on their shop floor, performing different functions. Robotic actions are monitored, and anomalous actions are reported. In particular, robotic torque sensor data is used to create a digital signature given a defined trajectory and load combination to perform a particular task. The signature of each robot is later used to diagnose mechanical deterioration.

The Automated Robot Diagnostic discussed in Example 1 demonstrates a typical training AI workload in Industry 4.0. To emphasize the need for AI scaling in our working example, each robot needs a multi-variate anomaly model (i.e., 2000 models). Ideally, the anomaly model discovery process explores around 100s of different machine learning models (including their parameters)(Patel et al. 2022b; Patel, Phan, and Mueller 2022). In summary, a Data Scientist or AI practitioner working on providing an AI solution needs to explore 2000×100 for Example 1. In a very ad-hoc manner, one can use a popular MLFlow library (MLFlow 2022) for machine learning modeling and manually generate 2000 experiments to be conducted with 100 runs each. However, such an ad-hoc approach misses specific optimization, assets of similar types may share a similarity, and thus their joint exploration may benefit the model discovery process.

On top of cross-asset learning, we have also witnessed the requirement of building different AI applications on the same asset dataset. For example, in the IBM Maximo product, the output of Anomaly Detection (an unsupervised approach) and Failure Forecasting Model (a semi-supervised approach) are displayed for each asset. Currently, training of the Anomaly and failure prediction models are isolated. However, provided data is shared across different AI applications, and it is an inherent expectation to conduct a common data prepossessing across multiple AI applications and coordinate across multiple assets. The AI Model factory proposed in this paper is designed to deal with both the emerging needs: Scale across multiple assets as well as multiple AI applications.

AI Model Factory : System Overview

Figure 2 gives an overview of AI Model Factory's layered architecture. The tool extends existing open source libraries such as MLFlow and Ray (the bottommost layer) to enable scalable model training across multiple assets and AI applications. Our crucial contribution comes from designing and developing the middle two layers (Model Factory runtime and core). The top layer is how a data scientist or end AI



Model Factory

By IBM Research AI for Industry

Model Factory is an AI System that helps to scale out various AI applications such as Anomaly Detection, Failure Prediction, Asset Health Forecasting across thousands of industrial assets. By bringing your own data, model factory service provides features of data management, assest-centric model training, training tracking and monitoring, and model deployment.

Forecast asset health on asset time series

POST /model-factory/asset-health-forecasting/run/

Asset Health data consists of multiple univariate timeseries each containing health data of various assets recorded at a daily frequency. The goal of the project is to forecast the future health of the assets given their historical health. This endpoint submits a job that performs timeseries forecasting on the data provided by the user.

```
curl --request POST \
  --url https://dev.api.ibm.com/ai4industry/test/model-factory/
  --header 'X-IBM-Client-Id: REPLACE_THIS_KEY' \
  --header 'X-IBM-Client-Secret: REPLACE_THIS_KEY' \
```

Figure 1: Model Factory Preview Service @ IBM Developer API Hub

user interacts with the Model Factory. We explain the system's working with the help of an Asset Health Forecasting Application. This model factory application builds time series forecasting model for predicting the future health of an asset using its historical information. The number of assets ranges from 2000-3000 per asset group.

At the start of the process, the end user needs to provide two inputs to initiate the applications:

1. **Data** The input data should be organized in S3-compatible COS buckets or a local file system. At present, Model Factory prescribes three commonly used data organizations. These three prescriptions are Asset Specific, Asset Agnostic, and Simplified. Figure 3 pictorially displayed then.
2. **Parameter**. User also provides application-specific parameters such as *asset id*, *timestamp*, *input feature*, *output target*, *prediction window size* which denotes the forecast horizon, the learning parameter which can be "single-task" where a different model is trained for each asset or "multi-task" where a model is trained on multiple assets and the level parameter which can be basic, advanced and comprehensive. The basic level does model exploration among statistical forecasting methods like ARIMA, the advanced level does additional exploration with ML models, and the comprehensive level does even more exploration by including Deep Learning models.

Once the application is executed, the end user can track the execution of the process using various tracking APIs such as `/track` and `/report`. The `/track` method shows the progress of the running project with the number of assets that have finished training and the total number of assets to be trained. The `/report` method gives an extensive summary of the finished project, including pipeline exploration time, validation metrics, test metrics, and more.

Once the entire process is finished, the project's status is updated as Done. Then the users have the choice to deploy the trained models to ML model serve platforms using `/deploy` endpoint. Currently, the service supports IBM Watson Machine Learning, enabling the model to serve in production. More cloud platform support is yet to come.

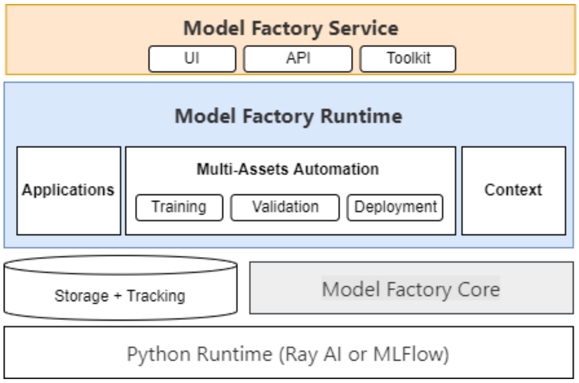
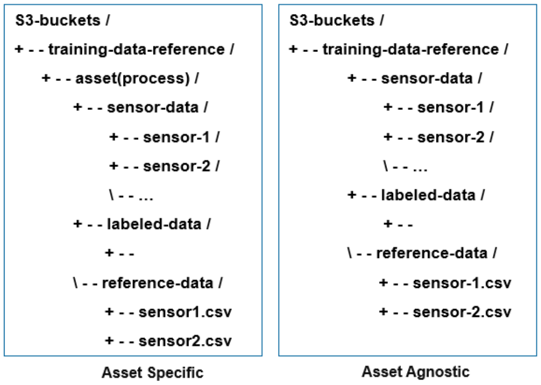


Figure 2: Layered architecture of Model Factory



```
S3-buckets /
+ - - training-data-reference /
    + - - asset(process) /
        + - - sensor-data /
            + - - sensor-1 /
            + - - sensor-2 /
            \ - - ...
        + - - labeled-data /
            + - -
        \ - - reference-data /
            + - - sensor1.csv
            + - - sensor2.csv
```

```
S3-buckets /
+ - - training-data-reference /
        + - - sensor-data /
            + - - sensor-1 /
            + - - sensor-2 /
            \ - - ...
        + - - labeled-data /
            + - -
        \ - - reference-data /
            + - - sensor-1.csv
            + - - sensor-2.csv
```

Figure 3: Dataset Organization

References

MLFlow. 2022. A platform for the machine learning lifecycle. https://mlflow.org/. Accessed: 2022-12-07.

Patel, D.; Ganapavarapu, G.; Jayaraman, S.; Lin, S.; Bhamidipaty, A.; and Kalagnanam, J. 2022a. AnomalyKiTS: Anomaly Detection Toolkit for Time Series. In *AAAI*, 13209–13211. AAAI Press.

Patel, D.; Phan, D.; and Mueller, M. 2022. Time Series Anomaly Detection Toolkit for Data Scientist. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, 3202–3204.

Patel, D.; Phan, D.; Mueller, M.; and Rajasekharan, A. 2022b. Toolkit for Time Series Anomaly Detection. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22, 4812–4813. New York, NY, USA: Association for Computing Machinery. ISBN 9781450393850.

Patel, D.; Shrivastava, S.; Gifford, W.; Siegel, S.; Kalagnanam, J.; and Reddy, C. 2020a. Smart-ML: A System for Machine Learning Model Exploration using Pipeline Graph. In *2020 IEEE International Conference on Big Data (Big Data)*, 1604–1613.

Patel, D.; Yousaf Shah, S.; Zhou, N.; Shrivastava, S.; Iyengar, A.; Bhamidipaty, A.; and Kalagnanam, J. 2020b. FLOps: On Learning Important Time Series Features for Real-Valued Prediction. In *2020 IEEE International Conference on Big Data (Big Data)*, 1624–1633.

Patel, D.; Zhou, N.; Shrivastava, S.; and Kalagnanam, J. 2020c. Doctor for Machines: A Failure Pattern Analysis Solution for Industry 4.0. In *2020 IEEE International Conference on Big Data (Big Data)*, 1614–1623.

Shrivastava, S.; Patel, D.; Gifford, W. M.; Siegel, S.; and Kalagnanam, J. 2019. ThunderML: A Toolkit for Enabling AI/ML Models on Cloud for Industry 4.0. In Miller, J.; Stroulia, E.; Lee, K.; and Zhang, L.-J., eds., *Web Services – ICWS 2019*, 163–180. Cham: Springer International Publishing. ISBN 978-3-030-23499-7.

Zerveas, G.; Jayaraman, S.; Patel, D.; Bhamidipaty, A.; and Eickhoff, C. 2021. A Transformer-Based Framework for Multivariate Time Series Representation Learning. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD, 2114–2124.