Hewlett Packard Enterprise

Building Efficient AI Pipelines with Self-Learning Data Foundation for AI

Martin Foltin, Principal Engineer
AI Research Lab, Hewlett Packard Labs

[joint work with Annmary Justine, Aalap Tripathy, Revathy Venkataramanan, Sergey Serebryakov, Cong Xu, Suparna Bhattacharya, Paolo Faraboschi]

May 9, 2023

Outline

- Problem Statement and Goals of this work
- Self-Learning Data Foundation for AI
- Federated Common Metadata Framework (CMF)
- AI Model and Hyper-parameter Recommendation
- AI Pipeline Energy and Carbon Footprint Analysis and Optimization
- Summary

Problem Statement and Goals of this work

Problem Statement

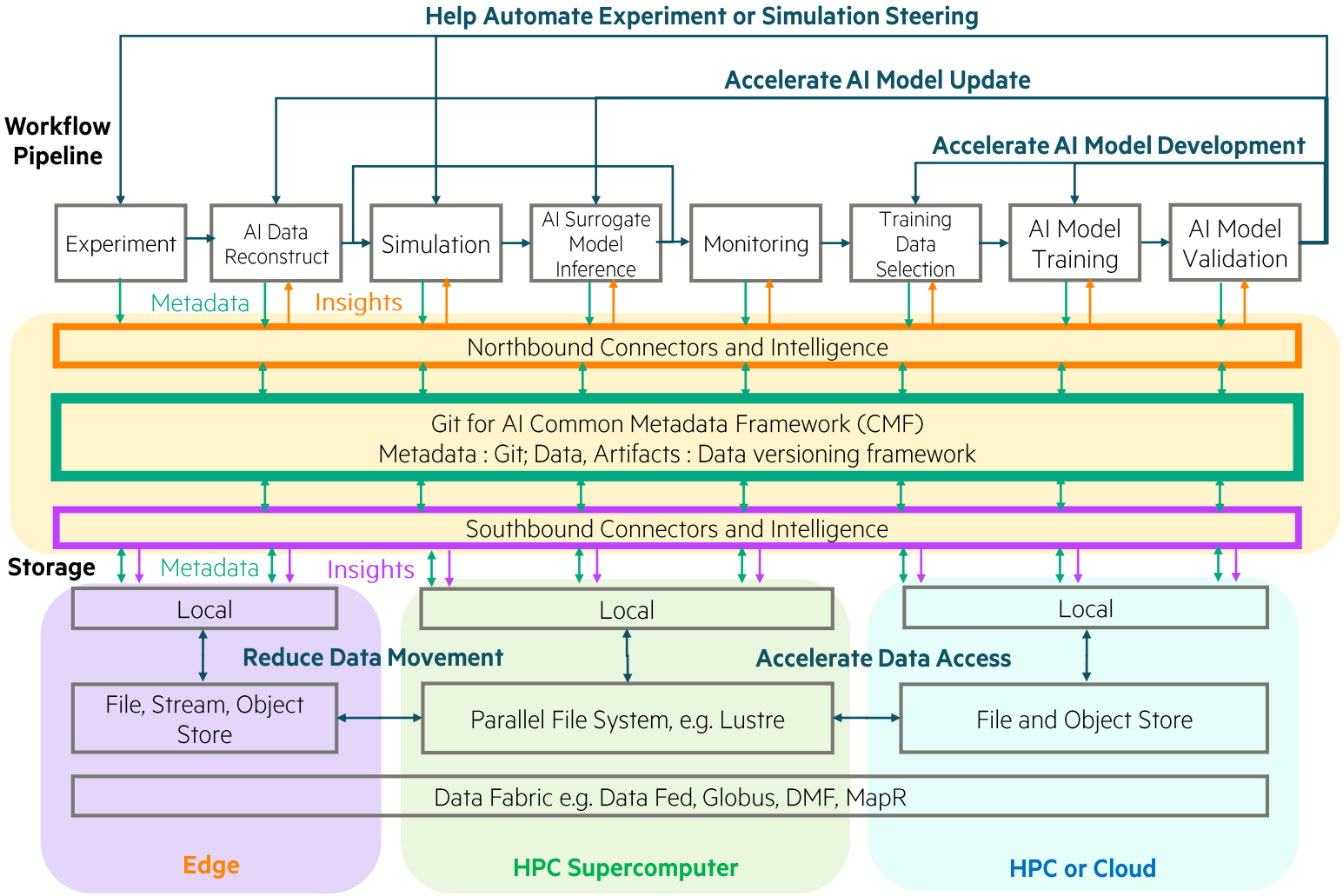
- Science workflows increasingly use AI model inference and training components
- Development of Trustworthy AI models is often laborious and compute intensive
- Deficiencies in data management for AI hurt model quality and workflow reproducibility

Goal: develop data management infrastructure that makes it easier to:

- Improve workflow reproducibility and portability to enable bootstrapping of new research
  - Including data management spanning Edge, HPC datacenter & Cloud
- Reduce human effort, compute cost and energy in developing Trustworthy AI models
  - Leverage historical experience to reduce exploration effort and compute time
  - Support collaborative development involving teams from different sites
- Account for energy consumption in end-to-end AI workflow optimization

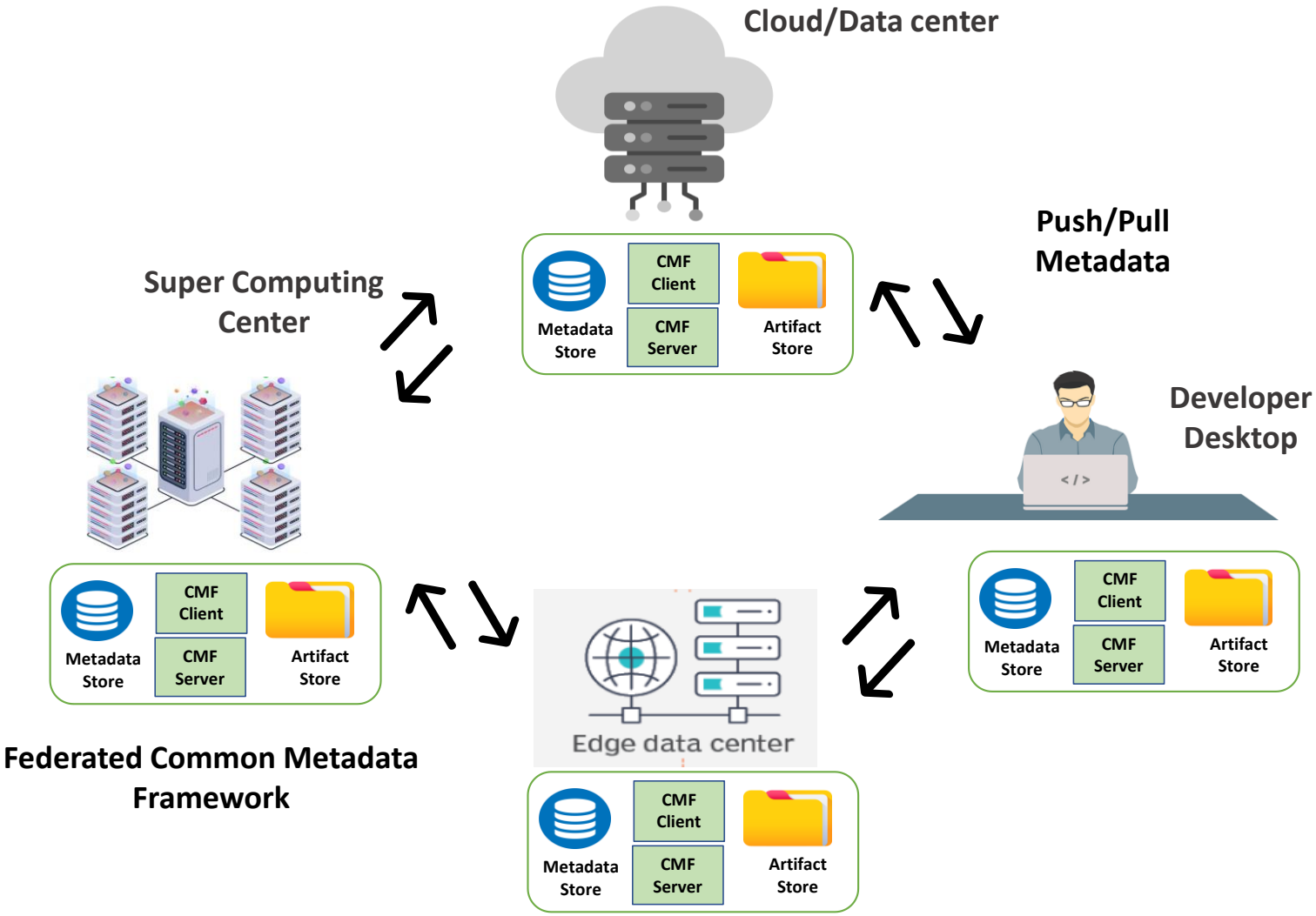
Self-Learning Data Foundation for AI

- A separate software layer agnostic of storage, frameworks and MLOps platforms
  - Spans both the training and inference flows
  - Meta-learning capabilities leverage historical correlations within and across flows
- Git for AI Common Metadata Framework
  - Manages and tracks lineage and metadata from AI pipelines (property graph)
- Northbound intelligence examples
  - Helps discover, select, utilize data of highest importance for model training
  - Assists AutoML tools in selecting optimal stages, model architectures, parameter seeds
- Southbound intelligence examples
  - Extracts insights from metadata to optimize data shaping and co-optimize data I/O and model training



Federated Common Metadata Framework (CMF)

Federated CMF Architecture and Characteristics



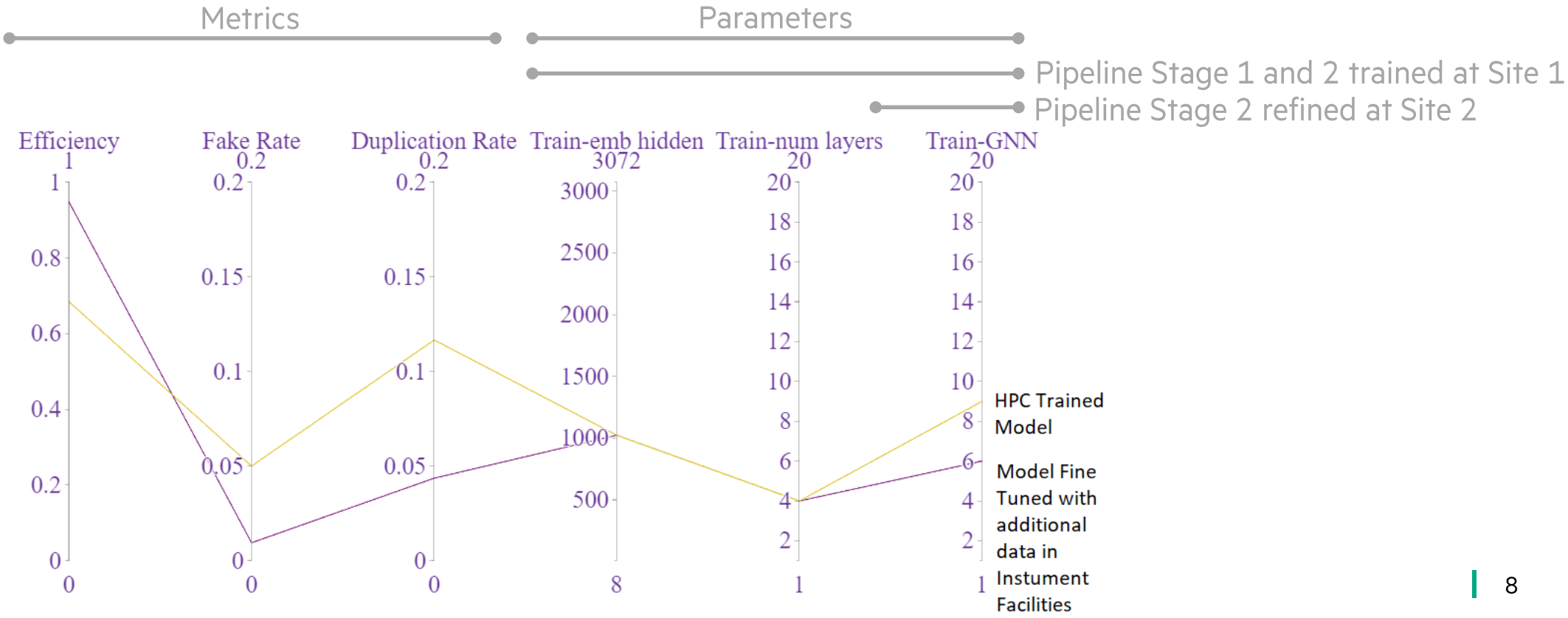
- End-to-end lineage & metadata tracking
- Integrated Git-like versioning
- Traceability of artifacts (models, intermediate data)
- Workflow audit trail leading to certifiability
- Federated across multiple sites, sharing of metadata and data selectively as needed
- Access to metadata independent of data reduces volume of transferred data
- Enables export of metadata in OpenLineage format to enable cross-platform sharing

Federated CMF Components

- Metadata Store
  - Explicit (through workflow APIs) and implicit (through hooks to workflow managers) metadata and lineage tracking
  - Pipeline → Context → Execution hierarchical abstraction for workflow metadata enables building lineages across disjoint executions and between different sites
- Artifact Store
  - Git-like artifact versioning through the unique content hash
  - Supports wide range of storage remotes (S3 object store, local and parallel file systems, etc.)
  - External artifacts not managed by CMF supported via uniform resource identifier (URI) mechanism
    - Derived artifact with URI encoding the location of source artifact
- CMF Server
  - Rest endpoint that can be accessed over HTTPS and used by other authorized clients to push or pull metadata
  - Merges metadata from different clients
    - The merge step identifies the branch in the pipeline tree under which an incoming execution fits or creates a new context branch or creates a new pipeline tree. Artifact metadata from different sites can be merged using its content hash as the joining key.
- CMF Client
  - Enables push or pull of subsets of metadata from a server to a local store
  - Enables pulling artifact from the remote store to the local store selectively. The design enables sharing of data when needed (reducing data movement), but each site can use local artifact store enabling data locality

Federated CMF Prototype Use Case

- AI model incremental refinement at a different site with new data
  - Exa.trkX high energy physics particle trajectory reconstruction multi-stage AI pipeline
    - Stage 1: Embedding Neural Network model, Stage 2: Graph Neural Network Model (GNN)
  - AI pipeline portable between sites
  - Pipelined trained at Site 1 and GNN model refined at Site 2 with CMF stitching disjoint executions between sites



Federated CMF Other Example Use Cases

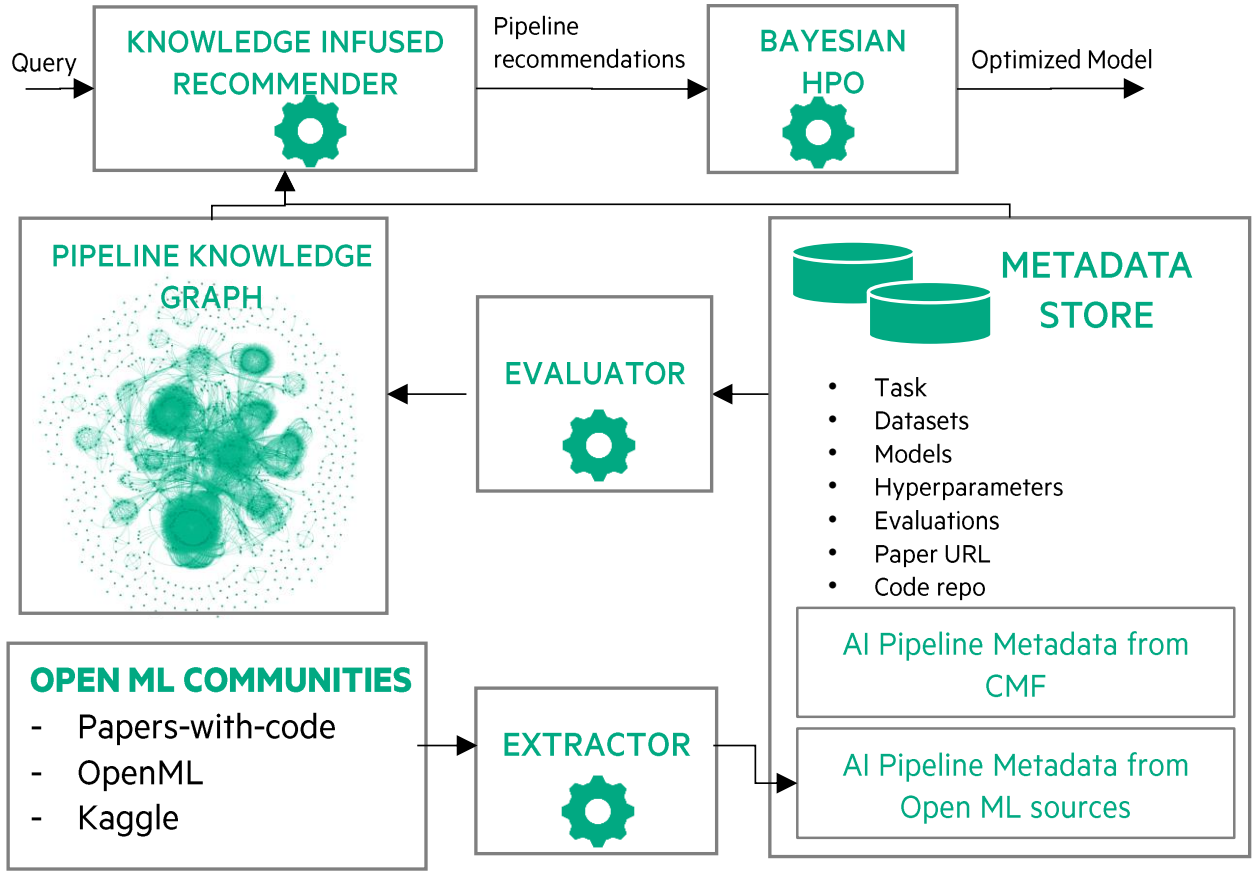
- Magnetic Confinement Fusion
  - Lifelong (continuous) learning of fusion control models trained at HPC center and deployed at the edge
  - Potential to accelerate integration of new research discoveries (plasma anomalies, etc.) to production flows in a certifiable manner
- Autonomous Electron Microscopy
  - Active learning of optimum scan positions for material structure and spectra investigations
  - Potential for tight coupling with HPC molecular dynamics or density functional energy simulations
- Real Time Wildfire Management
  - Fire detection and monitoring at the edge coupled to HPC modeling of fire spread
  - Potential to adjust fire spread model by real-time measurements from the field

AI Model and Hyper-parameter Recommendation

AI Model and Hyper-parameter Recommendation Objectives

- Utilize a-priori knowledge and metadata captured in CMF from executions of thousands of different pipelines to recommend a small set of models and hyper-parameters best fitting for a given task and dataset
- Use these recommendations as a seed (good known configurations) for AutoML methods such as neural architecture search (NAS) and Bayesian hyper-parameter search to accelerate model development - reduce human exploration effort, compute time and energy

AI Model and Hyper-parameter Recommendation Architecture



- Input query includes task name and dataset name
- Recommender knowledge graph is built based on thousands of AI tasks from Open ML communities imported to CMF
- AI task pairs assigned similarity metrics based on task categories, modalities and dataset characteristics
- Knowledge graph maintains task similarity distances
- Input query task matched to closest reference tasks from the graph. Recommendations ordered based on similarity score
- Pipelines, models, and hyper-parameters (when available in Open ML communities) returned with each recommendation

AI Model and Hyper-parameter Recommendation Example Usage and Results

- Example output for a query with “3d Anomaly Detection” task name and “MVTEC” dataset name:

INPUT CONFIG
Task: 3d Anomaly Detection
Dataset: {
Name: MVTEC
Type: Image
min_datapoints: 1000
}
Other: {
Run_time: 72hrs
Min_accuracy: 85
Inference time: 60s }

OUTPUT CONFIG
Requested Task: '3d Anomaly Detection'
Recommended Task: '3D Anomaly detection and segmentation’
Similarity score: '0.625’
Requested Task category: Segmentation
Recommended Task category: Detection, Segmentation
Requested Task Modality: Image
Recommended Task Modality: Image

RECOMMENDED PIPELINE
Pid: 'the-mvtec-3d-ad-dataset-for-unsupervised-3d’
Datasets: [{ 'full_name': 'THE MVTEC 3D ANOMALY DETECTION DATASET’, 'name': 'MVTEC 3D-AD', 'url': 'https://www.mvtec.com/company/research/datasets/mvtec-3d-ad'}],
Git repo: [{ 'description': 'Awesome-3D-Anomaly-Detection-and-Localization/Segmentation',
'framework': 'none’, 'name': 'Awesome-3D-Anomaly-Detection',
'url': 'https://github.com/JerryX1110/Awesome-3D-Anomaly-Detection'}],
Paper Title: 'The MVTec 3D-AD Dataset for Unsupervised 3D Anomaly Detection and Localization’,
Paper url: 'https://arxiv.org/pdf/2112.09045v1.pdf’,}

- Example Bayesian Hyper-parameter Optimization speed-up for several binary and multi-class classification problems with tabular data mapped to gradient boosted tree models (XGBoost), when starting from hyper-parameters recommended by our tool:

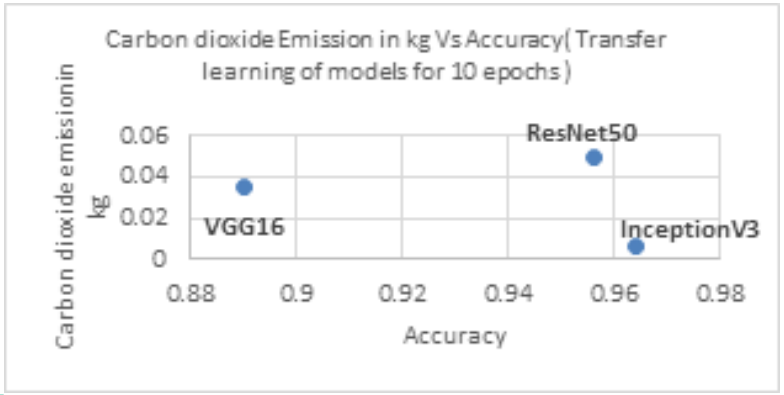
Churn Modelling		TelcoCustomerChurn		ForectCoverType	
SpeedUp	LossDiff	SpeedUp	LossDiff	SpeedUp	LossDiff
8.60	-0.11%	11.84	-0.63%	1.47	-2.02%

AI Pipeline Energy and Carbon Footprint Analysis and Optimization

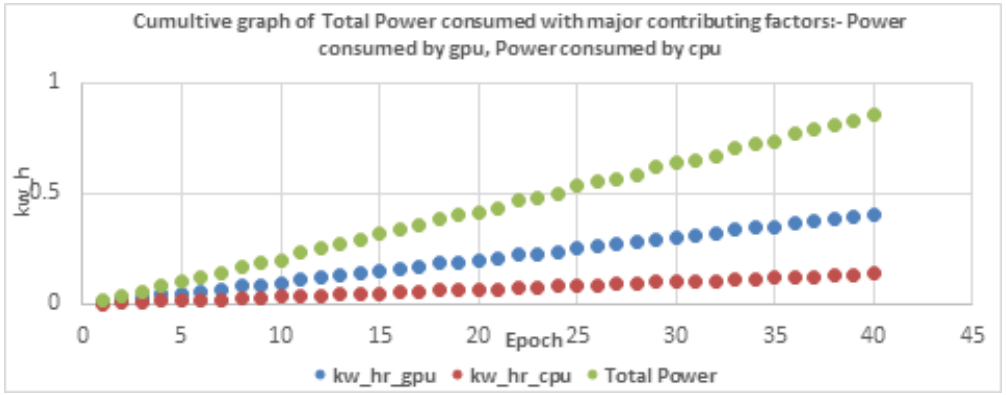
AI Pipeline Energy and Carbon Footprint Analysis

- Objective
  - Accurate reporting of energy consumption and carbon footprint in distributed AI pipelines (including data preparation, model training and inference) enabling to include energy consumption in AI model optimization and evaluation of various trade-offs (e.g., accuracy versus energy)
- Architecture
  - Utilize Federated CMF to record energy consumption in each AI workflow stage execution as metadata
  - Compute energy consumption during the execution rather than post fitting
  - The current POC uses experiment-impact-tracker open source to monitor CPU and GPU energy usage on commodity server. Work in progress to develop client-server architecture for HPC with user clients polling system level process for energy usage
- Example preliminary results:

CO_2 consumption in 2 layer transfer learning vs. accuracy:



GPU, CPU and total power consumption for ClimateNet:



Hardware - ProLiant DL380 Gen10
Memory- 96GiB, CPU – 2 Intel(R) Xeon(R) Silver 4110 CPU @ 2.10GHz

Hardware - ProLiant DL380 Gen10
Memory- 48 GiB, Intel(R) Xeon(R) Gold 6248 CPU @ 2.50GHz, 1GPU: Tesla V100-PCIE-16GB

Summary

- We developed Federated Common Metadata Framework (CMF) to:
  - Manage workflow lineage and metadata across multiple systems or sites
  - Enable workflow reproducibility and portability among different sites
  - Provide community-wide visibility to data, workflows and metadata, enabling to bootstrap new research
  - Support workflows spanning Edge, HPC datacenter and Cloud
- We also developed AI model and Hyper-parameter recommender tool to:
  - Recommend best pipeline, model and hyper-parameters from given task and dataset
  - Reduce human exploration and compute time and energy by accelerating Network Architecture Search and Hyper-parameter optimization
  - By using a-priori knowledge captured in CMF from executions of thousands of different pipelines
- Finally, we are developing AI Pipeline Energy and Carbon Footprint Analysis tool to:
  - Estimate end-to-end energy consumption of distributed AI pipelines, including data processing, model training and inference
  - Include energy consumption in AI model optimization and evaluation of various trade-offs like energy efficiency at training vs energy efficiency at inference and cost of retraining

Thank you

martin.foltin@hpe.com
annmary.roy@hpe.com
aalap.tripathy@hpe.com
revathy.venkataramanan@hpe.com
sergey.serebryakov@hpe.com
cong.xu@hpe.com
suparna.bhattacharya@hpe.com
paolo.faraboschi@hpe.com