Federated ML Application at Edge

Introduction
This document provides an overview of the Federated ML blueprint as well as an overview of key features and implementations of Federated ML in Akraino Release 5.

Overview
Akraino Blueprint: Federated ML
The purpose of Federated ML is to provide Federated Learning Platform for data stored locally, improves accuracy in the edge computing. The high-level relationships between the functional domains is shown in the figure below:

The Networking Domain: The Networking Domain uses a component called federated networking to create a safe cross-site networking among all the federated sits.

The Computing and storage Domain: The Computing and Storage Domain provides local/distributed computing and storage abilities to support the federated computing mission.

The Federated ML: The Domain called Federated ML is used to provide multi federated machine learning core components, such federated feature binning, federated feature selection, federated algorithm (LR, BOOST, NN) and etc.

The FATE-FLOW: The Domain called FATE-FLOW is used to schedule federated machine learning task, parse the flow DAG and manage the lifecycle of a federated machine learning task.

The FATE-SERVING: The Domain called FATE-SERVING is used to generate and provide an online interface for the federated model and manage the federated model.

The FATE-BOARD: The Domain called FATE-BOARD is used to visualize the federated machine learning task, monitor the federated model status and manage the logs.
The Federated ML Reference Architecture is shown in the figure below. For the full description of the Federated ML Reference Architecture please refer to the Federated Architecture Document.

Federated ML in Akraino R5
Key features and implementations in Akraino Release 5:

- Hetero SecureBoost: more efficient computation with GOSS, histogram subtraction, cipher compression, 2-4x faster
- Hetero GLM: improved communication efficiency, adjustable floating point precision, 2x faster
- Hetero NN: adjustable floating point precision, support SelectiveBackPropagation and dropOut on interaction layer, 2x faster
- Hetero Feature Binning: improved algorithm with cipher compression, 2x faster
- Intersect: add split calculation option and adjustable random base fraction, 30% faster
- Homo NN: restructure torch backend and enhanced grammar; train and predict with raw image data
- Intersect supports SM3 hashing method
- Hetero SecureBoost: L1 penalty & adjustable min_child_weight to prevent overfitting
- NEW SecureBoost Transformer: feature engineering module that encodes instances with leaf nodes from SecureBoost model
- Hetero Pearson: support local VIF computation
- Hetero Feature Selection: support selection based on VIF and Pearson
- NEW Homo Feature Binning: support virtual/recursive binning strategy
- NEW Sample Weight: set sample weights based on label or from feature column, Hetero GLM & Hetero SecureBoost support weighted training
- NEW Data Transformer: case-insensitive on data schema
- Local Baseline supports prediction task
- Cross Validation: output fold split history
- Evaluation: add multi-result-unfold option which unfolds multi-classification evaluation result to several binary evaluation results in a one-vs-rest manner

For more information:
https://wiki.akraino.org/display/AK/The+AI+Edge%3A+Federated+ML+application+at+edge

Akraino Edge Stack is an open source project under the LF Edge umbrella that creates edge software stacks that supports high-availability cloud services optimized for edge computing systems and applications. It offers users new levels of flexibility to scale edge cloud services quickly, to maximize the applications and functions supported at the edge, and to help ensure the reliability of systems that must be up at all times. The Akraino Edge Stack platform integrates multiple open source projects to supply a holistic Edge Platform, Edge Application, and Developer APIs ecosystem.
Akraino uses the “blueprint” concept to address specific Edge use cases to support an end-to-end solution.
A blueprint is a declarative configuration of the entire stack-- i.e., edge platform that can support edge workloads and edge APIs.
To address specific use cases, a blueprint architecture is developed by the community and a declarative configuration is used to define all the components used within that architecture such as hardware, software, tools to manage the entire stack, and method of deployment (Blueprints are maintained using full CI/CD integration and testing by the community for ready download and install).


Akraino is part of the LF Edge umbrella organization that establishes an open, interoperable framework for edge computing independent of hardware, silicon, cloud, or operating system. By bringing together industry leaders, LF Edge creates a common framework for hardware and software standards and best practices critical to sustaining current and future generations of IoT and edge devices.

LF Edge Projects address the challenge of industry fragmentation, and collaborates with end users, vendors, and developers to transform all aspects of the edge and accelerate open source developments.

[Insert Logos for: Akraino, Baetly, Fledge, EdgeX Foundry, Glossary of Edge Computing Home Edge, Project EVE]