

Système d'Apprentissage Fédéré (Federated Learning) sur IoT

Cheikhani Lemrabet

1 Architecture du Système

Le flux de données suit une topologie *Edge-to-Cloud* :

$$\text{Sensors (Edge)} \xrightarrow{\text{Data}} \text{Kafka} \xrightarrow{\text{Processing}} \text{Spark (Fog)} \xrightarrow{\text{Weights}} \text{Dashboard (Cloud)} \quad (1)$$

L'innovation réside dans l'utilisation du **Federated Learning**. Au lieu de centraliser les données brutes (risqué pour la vie privée et coûteux en bande passante), nous ne transférons que les paramètres du modèle (poids et biais).

2 Implémentation et Méthodologie

2.1 Génération de Données (Producers)

Les capteurs simulent des données linéaires avec un bruit gaussien ϵ :

$$y = \beta x + \alpha + \epsilon \quad (2)$$

Cela permet de valider si l'algorithme de régression peut converger vers les valeurs réelles de β et α .

2.2 Entraînement Distribué (Spark Fog Nodes)

Chaque nœud Spark consomme un flux Kafka et applique une régression linéaire sur des micro-batches. L'utilisation de `foreachBatch` permet d'intégrer **Scikit-Learn** au sein de l'écosystème distribué de Spark, offrant un compromis idéal entre performance et flexibilité.

2.3 Agrégation Globale (FedAvg)

Le Dashboard agit comme l'agrégateur central. Il calcule la moyenne des poids reçus :

$$W_{global} = \frac{1}{N} \sum_{i=1}^N W_i \quad (3)$$

Où W_i représente les coefficients envoyés par chaque nœud local.

3 Analyse des Résultats

La capture d'écran fournie démontre une convergence réussie. Le modèle global affiche :

- **Pente (Slope)** : 0.9339
- **Interception** : 30.4341

Le graphique montre que malgré les variations locales de **node-1** et **node-2**, le modèle global reste stable et représentatif de l'ensemble du réseau IoT.