Détection d'intrusions fédérée et semi-supervisée pour l'IoT

Ons Aouedi, Kandaraj Piamrat, Guillaume Muller, Kamal Singh

Détection d'intrusions fédérée et semi-supervisée pour l'IoT

O. Aouedi[1], K. Piamrat[1], G. Muller[2], K. Singh[2]

[1]LS2N, Université de Nantes, BP 92208, 44322 Nantes CEDEX 3, France
[2]Université Jean Monnet, IOGS, CNRS, UMR 5516, LaHC, 42023 Saint-Étienne, France
guillaume.muller@univ-st-etienne.fr

1. Introduction

Contexte : IoT Industriel (IIoT)

L'« **Industrial Internet of Thing** » **(IIoT)** est un réseau de petits appareils (capteurs « intelligents »...) exécutant des applications industrielles. Il permet la création **services riches**, fondés sur les données : surveillance d'usines, prédiction de pannes, planification dynamique... Il est le socle fondateur de l'**industrie du futur**. À ce titre, il est en **plein essor**.

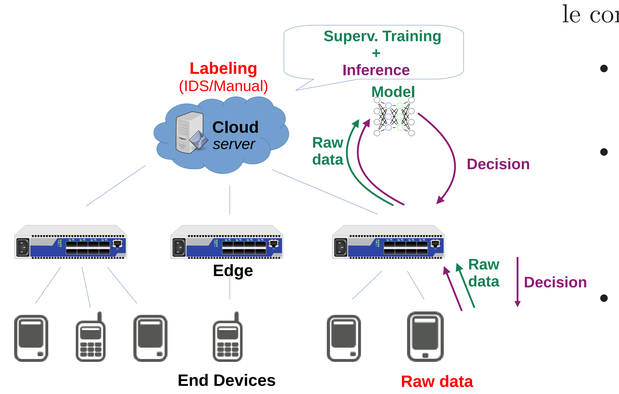
Problème : CyberSécurité

Du fait de la **criticité des installations** sur lesquelles elles sont installées, ces applications sont la cible de nombreuses **attaques**, visant soit à **paralyser l'infrastructure** (ex. : « Colonial Pipeline » en mai 2021), soit à **voler des données sensibles** (propriété intellectuelle, données personnelles...).

Solution : Système de Détection d'Intrusions (IDS)

Les systèmes de **détection d'intrusions** permettent de détecter et réagir rapidement pour **contrer ces attaques**. Les systèmes actuels s'appuient sur du **Machine Learning**, afin d'être en mesure de s'**adapter** aux nouveaux types d'attaques.

2. Limites de la détection d'intrusions dans l'IIoT



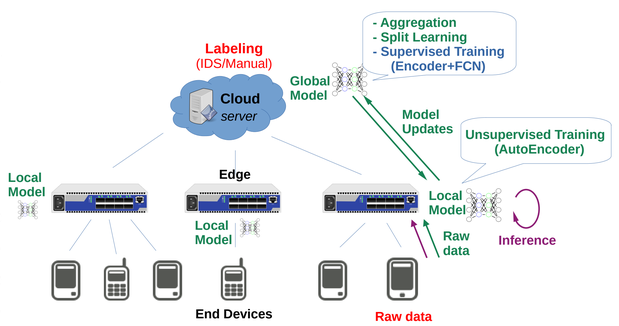
Particularités de la détection d'intrusion (IDS) dans le contexte IIoT :

- Les *données sont générées sur les petits appareils.*
- L'*étiquetage des données* (la caractérisation du trafic : attaque ou pas) nécessite une vision globale du trafic, donc doit avoir lieu *dans le « cloud »* et *est très coûteux* (🕒, 💾).
- Un processus de Machine Learning traditionnel (centralisé) supervisé nécessiterait la transmission les données, or :
  - Ces dernières peuvent être **sensibles** ;
  - Cela **alourdit la charge réseau** ;
  - Le **serveur** devient un **point de faiblesse** : il doit stocker et analyser une grande masse de données.

3. FLuIDS

Dans `FLuIDS` [1, 2] notre contribution consiste à combiner :

- un **apprentissage non-supervisé** au niveau des petits appareils, grâce à des auto-encodeurs.
- un **apprentissage supervisé** sur le serveur.
- pour relier les deux, un **apprentissage fédéré** [4] et une méthode de **Split-Learning** [5], qui permet à l'ensemble des entités d'apprendre l'Encodeur **conjointement** et en **respectant la sensibilité** des données.



4. Résultats

`FLuIDS` a été testé sur 2 jeux de données classiques de la détection d'intrusions, notamment IIoT : UNSW-NB15 et SCADA Gas Pipeline. Les résultats suivants portent sur le premier. Il a pu être montré que :

- `FLuIDS` **exploite les données non-étiquetées** : augmenter leur volume (à nb. étiquetées constant), augmente l'accuracy.
- `FLuIDS` **réduit** de 20% à 99% **le volume de communications**, en envoyant des mise-à-jour de modèles plutôt que des données brutes.
- `FLuIDS` est **au moins aussi performant** (F1-score) que les modèles **centralisés supervisés**.
- `FLuIDS` est **plus performant** quand il est entraîné **en mode fédéré** qu'en mode centralisé : il est peu sensible à une distribution non-IID des données.
- `FLuIDS` nécessite **plus temps pour converger** qu'un modèle centralisé, mais dans une limite ***raisonnable***.

5. Conclusions

- `FLuIDS` exploite **toutes** données (étiquetées ou non), grâce au Split Learning.
- `FLuIDS` **péserve la sensibilité** des données, puisqu'elles ne sont pas transmises.
- `FLuIDS` **réduit la charge du serveur**, puisque les calculs/stockages sont distribués.
- `FLuIDS` **réduit la charge du réseau**, puisque seuls des données « compressées » (modèles) sont transmises.
- `FLuIDS` **réduit la latence** lors de l'inférence (détection d'intrusion à proprement parler), puisque cette dernière s'effectue directement sur les petits appareils.

6. Références

- [1] O. Aouedi et al. "Federated Semi-Supervised Learning for Attack Detection in Industrial Internet of Things". In: *IEEE Transactions on Industrial Informatics, SS on Security and Privacy Issues in Industry 4.0 Applications* (2022)
- [2] O. Aouedi et al. "Intrusion detection for Softwarized Networks with Semi-supervised Federated Learning". In: *IEEE-ICC'22*. Seoul, South Korea, June 2022
- [4] Brendan McMahan et al. "Communication-efficient learning of deep networks from decentralized data". In: *Artificial Intelligence and Statistics*. PMLR. 2017, pp. 1273–1282
- [3] Peter Kairouz et al. "Advances and Open Problems in Federated Learning". In: *CoRR* abs/1912.04977 (2019). arXiv: 1912.04977. URL: http://arxiv.org/abs/1912.04977
- [5] P. Vepakomma et al. "Split Learning for health: Distributed deep learning without sharing raw patient data". In: *CoRR* abs/1812.00564 (2018). arXiv: 1812.00564. URL: http://arxiv.org/abs/1812.00564