A Cross-evaluation approach for Reputation-aware Model Weighting

Filtering Contributions in Federated Learning for Intrusion Detection

Journées non-thématiques du GDR RSD, Lyon, 27/01/2023

Authors:
Léo Lavaur
Pierre-Marie Lechevalier

Supervisors:
Fabien Autrel
Yann Busnel
Hélène Le Bouder
Romaric Ludinard
Marc-Oliver Pahl
Géraldine Texier
Context: Intrusion Detection

- Different families: misuse detection, anomaly detection, specification-based...

- Machine learning (ML) and deep learning (DL) often used for their performance;
  - Eg., auto-encoder (AE) can be used for anomaly detection.

- DL need a lot of data to be efficient, training them locally is a challenge;
  - Eg., for AE, anything not known is an anomaly → higher false-positive rate.

Fig 1: Typical AE workflow for IDS
Collaborative Intrusion Detection

1. Data collection
2. Centralized model training
3. Model sharing

Objective

➔ Consolidate normal behavior modeling by sharing knowledge with other participants;

Challenges

➔ Security & Privacy – eg. revealing internals, poisoning, trust [1];
➔ Availability – eg. single point of failure in centralized systems [2];
➔ Resources – eg. high bandwidth consumption when sharing data [3];

Fig 2: Typical CIDS (Collaborative Intrusion Detection System) workflow

Federated Learning as a Collaborative Learning System

1. Model distribution
2. Client-side model training
3. Parameters sharing
4. Global model aggregation

Fig 3: Typical Horizontal Federated Learning workflow for CIDS

Challenges [4]

➔ Heterogeneity – unsuitable global aggregation when participants are too different.
➔ Trust – assessing peer contributions.

**Objective**: Mitigate the impact of *bad* contributions to the local models;

![Graphical representation of the proposed architecture](image)

*Fig 4: Proposed architecture*
Our approach

**Objective:** Mitigate the impact of *bad* contributions to the local models.

➔ How to evaluate models in highly heterogeneous settings?

---

**Fig 4:** Proposed architecture
Objective: Mitigate the impact of bad contributions to the local models;

➔ How to evaluate models in highly heterogeneous settings?
➔ How to set aside dissimilar participants?

Fig 4: Proposed architecture
**Objective:** Mitigate the impact of *bad* contributions to the local models;

- How to evaluate models in highly heterogeneous settings?
- How to set aside dissimilar participants?
- How to identify and discard similar but negative behaviors?

---

Fig 4: Proposed architecture
I. Assessing Contributions with Cross-Evaluation
Methods for filtering contributions

Server-side evaluation [5]

- only applicable in IID settings
- single source of truth

Server-side model comparison [6]

- less related to client data
- more appropriated for high-dimensional features

Client-side evaluation [7]

- high cost in cross-device settings


Advantages

➔ Central server doesn’t need prior knowledge.
➔ Evaluates how each model fits the data (e.g., accuracy).
➔ Exhaustive overview of the entire system at round \( r \).
➔ Keeps the subjectivity of the evaluations.

Drawbacks

➔ Expansive in communication and computation.
➔ Doesn’t scale well.

But...

➔ Cross-silo: Few clients, with reasonable computing capacity.
➔ Slow workflow: long time between rounds.

Fig 5: Cross-evaluation
II. Fighting Heterogeneity with Clustering
Merging heterogeneous contributions

II. Fighting Heterogeneity with Clustering

Global model can either:

- lose accuracy by trying to fit all participants [8];
- dismiss some participants [9].


Merging heterogeneous contributions

II. Fighting Heterogeneity with Clustering

Global model can either:

- lose accuracy by trying to fit all participants [8];
- dismiss some participants [9].


Merging heterogeneous contributions

II. Fighting Heterogeneity with Clustering

Clustering goal:
- regroup similar participants together;
- create an aggregated model per cluster.
Existing clustering approaches for federated learning

Clustering data source

➔ Participants models [10].  
➔ Cross evaluation results.

Clustering for federated learning

➔ Hierarchical clustering [10].


III. Ensuring Quality Contributions with Reputation
Motivation for reputation

Objectives reminder:

➔ weight participants contributions;
➔ detect change that occur over time [12].

Definition [13]

➔ Long-lived entities that inspire an expectation of future interaction;
➔ Capture and distribution of feedback about current interactions (such information must be visible in the future); and
➔ Use of feedback to guide trust decisions.

Evaluation weighting

Similarity

➔ Cluster outliers shouldn’t have too much impact on evaluation.
➔ Ponderate client evaluation using their similarity to other cluster members [14].

Historical considerations

➔ No specific constraints.
➔ Exponential decay: older results fade away.

Dirichlet distribution [15,16]

➔ Multinomial distribution.
➔ Allow discretization of cross evaluation results.

[14] Li Xiong, et al. “PeerTrust: Supporting Reputation-Based Trust for Peer-to-Peer Electronic Communities.” IEEE Transactions on Knowledge and Data Engineering 16, no. 07, 2004


Conclusion

Achievements

➔ Preliminary clustering validation:
  • done at the end of the first round of FL;
  • 8/10 clients are in the correct cluster;
  • empirical demonstration of cross-evaluation.

Future works

➔ Chain the functional blocks:
  • implement clustering and cross-evaluation; into the Flower framework;
  • test the reputation system.

➔ Extensive evaluation:
  • of each atomic block;
  • of the chained system.
Annex 1: Datasets and experimental platform

- "standardized IDS datasets" [17] (UNSW-NB15, BoT-IoT, ToN-IoT, and CSE-CIC-IDS2018)
  - 4 datasets = 4 use cases where clients are distributed in the use cases
  - normalized features among all datasets
- Flower FL Framework (https://flower.dev)

Annex 2: Preliminary clustering results

Fig 7: Rand index for hierarchical clustering with mean cluster interdistance as threshold

\[ RI = \frac{TP + TN}{TP + FP + FN + TN} \]