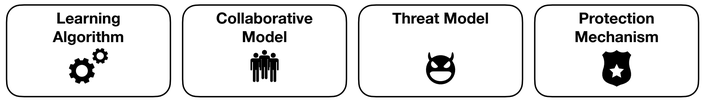
SoK: Privacy-Preserving Collaborative Tree-based Model Learning

Sylvain Chatel Apostolos Pyrgelis Juan Ramon Troncoso-Pastoriza Jean-Pierre Hubaux

École Polytechnique Fédérale de Lausanne

Overview

Tree-based models are among the most efficient machine learning techniques nowadays due to their accuracy, interpretability, and simplicity. Recent needs for more data and privacy protection call for collaborative privacy-preserving solutions. We systematize the literature on four axes:



We identify the strengths and limitations of these works and provide for the first time a framework analyzing the information leakage occurring in distributed tree-based model learning.

Learning Algorithm

Tree-based models are supervised learning techniques used for regression or classification tasks:

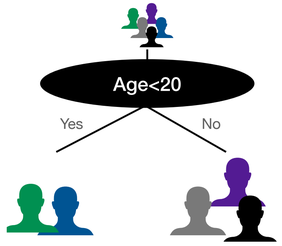


Figure 1. Example of node in a tree.

- Greedy algorithms rely on deterministic gains: information gain, Gini index, max, . . .
- Random algorithms select the split at random or select at random a subset of the features.
- Ensemble algorithms evaluate multiple trees and average their outputs.
- Boosting can combine weak learners into a stronger model (e.g., AdaBoost, GBDT, XGBoost).

Collaborative Model

Data can be shared in various ways impacting how the learning needs to be distributed

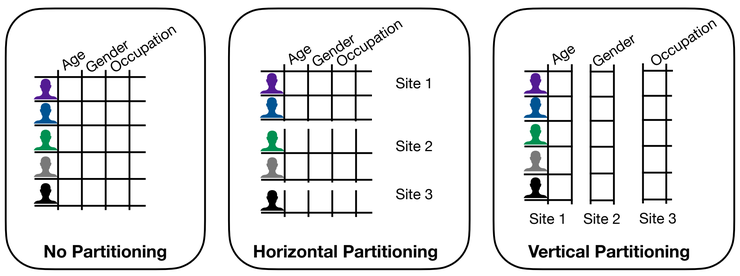


Figure 2. Data partitioning.

Depending on the entities involved, several collaborative models exist:

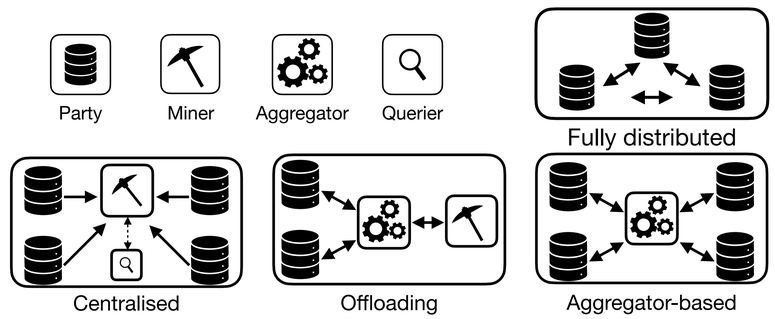


Figure 3. Collaborative models.

Depending on the collaborative model and the learning algorithm, different types of information may be exchanged between the parties calling for protection.

Threat Model

Most works consider the semi-honest setting. The malicious scenario is envisioned in seldom cases and rely on quality checks or cryptographic approaches that induce constraints on the communication and computation overhead.

Privacy Enhancing Technologies

Perturbation techniques

Create a surrogate dataset by perturbation or from the same distribution.
→ No formal guarantees.
→ Potentially vulnerable to attacks.

Secure Multiparty Computation

Enable almost exact learning for numerous collaborative models.
→ Potential overhead.
→ Lack tolerance to unavailability.
→ Trust assumptions.

Secure Hardware

Parties use secure enclaves for storage and learning.
→ Different trust assumptions.
→ Need countermeasures to side channel attacks.
→ Trending in cloud/industry.

Differential Privacy

Need to define which entity adds the noise, at which stage, and how much.
→ Need to preserve the budget (i.e., relax the learning).
→ Utility and privacy trade-off.

Homomorphic Encryption

Enable computation on ciphertexts directly.
→ Limited functionalities.
→ Comparisons not trivial.
→ Computation overhead.

Hybrid Solutions

Obtain benefits of multiple PETs:
→ Reduce privacy budget consumption.
→ Reduce communication overhead.
→ Enable comparisons.

Leakage Analysis

From the literature, we systematize the type of information shared during the learning process.

- **Raw data:**
  Can be inherent to the collaborative model or follow from improper protection.
- **Intermediate values:**
  Local or global, they can be used to infer information about the training data: e.g., counts, features, gradients, . . .
- **Final model:**
  Could be used for inference attacks and might be proprietary.

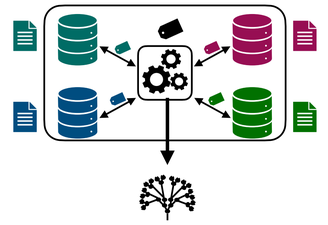


Figure 4. Leakage in collaborative tree-based model learning.

Open Challenges

Our deep analysis of privacy-preserving collaborative solutions for decision-tree induction enables us to identify some open problems and challenges in this field.

1. **In depth leakage analysis:**
   Investigate privacy leakage from intermediate values and quantify the risk associated with it.
2. **Malicious threat models:**
   Port existing solutions to settings with malicious entities without hampering utility.
3. **End-to-end protection for tree induction:**
   Protect raw data, intermediate values, and final model
4. **Resilience and fault tolerance:**
   Cope with unavailable and/or entities leaving the collaboration during the learning.

Conclusion

The collaborative model and the learning model pilot what information needs to be exchanged to build trees.

Most solutions modify the tree learning algorithm or add new entities to cope with these challenges.

Hybrid solutions combining DP and cryptography are the most promising as they can offer the benefits of both.

Our leakage identification framework can help designers categorize and reason about what information is exchanged during the tree-based model learning.

Our paper is available at: