A Survey on Federated Learning Systems: Vision, Hype and Reality for Data Privacy and Protection

<https://arxiv.org/abs/1907.09693>

# Chapter 1

A survey paper exploring Federated Learning systems.

They wish to enable collaborative training of machine learning models but under privacy restrictions it makes it really hard. The main points of federated learning systems are to achieve effectiveness, efficiency, and privacy. This paper will analyze the system components which are six different aspects, data distribution, machine learning model, privacy mechanism, communication architecture, scale of federation and motivation of federation.

Data islands - Isolated data hotspots, like hospitals, although they can’t be shared through their sensitive nature.

Biggest problem is developing a good predictive accuracy while obeying policies and regulations that will protect their privacy. Researchers have been looking at deep neural networks, gradient boosted decision trees, logistic regression, support vector machines, and even linear regression models to be used while protecting other people’s data.

# Chapter 2

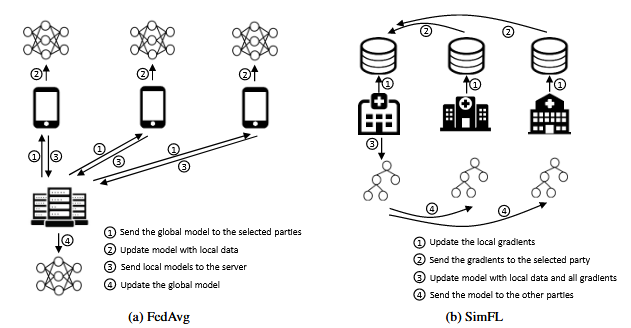
Federated Learning systems - multiple parties collaboratively train machine learning models without giving away their raw data. When its practical, then an evaluation metric, then the performance of the model learned by federated learning, should be better than the local model with the same architecture.

Federated database systems.These has autonomy which means without the database they could still run. Heterogeneity, the system can be different inside the FDBS. Distribution may be different for each Database system, and will do what it needs to increase the reliability.

Three components of Federated Learning systems, parties, manager, and communication computation framework to train the model.

Parties - We need to think about the hardware capacity that these parties will be using. The second is scale and stability so that you can handle many issues that could happen like connection loss, and cross-silo setting. The last is looking at the data distribution of their data for these parties.

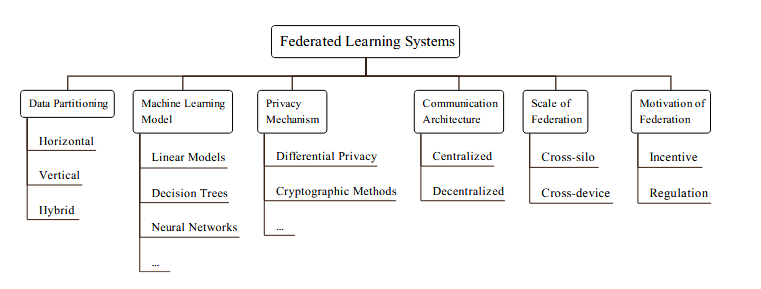
Manager - Powerful central server, that will lead in the communication between parties, and trains the global machine learning model. Needs to be stable. There can be a main manager where all parties communicate with one server and that server has the labels, or it can be a distributed server, where all parties contribute equally to the training of the global machine learning models training. (decentralized) (Seen as below)



Federated averaging - The server averages the global model with local data, and then the global model is sent back to the parties.

SimFL - First servers update their gradients, then send it to a selected party, they update the model with all the gradients collected, update the model, and send it back to all parties.

# Chapter 3



Data Partitioning

1. Horizontal - same feature space but no insertion on the sample space. Since they are the same feature space then you can use local data to update local models and just send it to the global model. (Example - “Hey Siri”, same application different voices)
2. Vertical - Similar sample space but different feature space. They need entity alignment techniques, and overlapped data are used to train the model, using encryption methods, to keep the privacy.

Machine Learning Models

Neural Networks, Decision Trees, Gradient boosting trees, ensemble methods, ….

Privacy Mechanism

Model Parameters can still leak private information, and people would attack models to infer the raw data. THere are many privacy mechanisms like differential privacy, and k-anonymity.

Cryptographic Methods - Encrypt their messages before sending, operate on the encrypted message, and decrypt the output to get the final result. (High computation)

Differential privacy - injection random noise, to protect against inference attacks, although there are less accurate models.

Many more privacy methods

Parties/attacks can also sabotage the model that is being created.

Architecture - Central or Decentralized (Blockchain), referenced above. (Main server or No)

Scale of Federation

1. Cross-silo FLS - the parties are data center (Huge amount of data,computation power and small amount of parties)
2. Cross-device - Large amount of parties, but small amount of data and computation power.

Motivation of Federation - Incentive to actually go through with federation, why let them use your sacred data. For what purpose are you trying to get out of this.

# 4. Existing Studies

‘**SGD-based Methods:**

* + **FedAvg:** Typical and practical framework, reduces communication rounds.
  + **FedSVRG:** Based on SVRG and DANE, can achieve better accuracy in some cases.
  + **FedProx:** Addresses non-IID data challenges by limiting local model updates.
  + **SCAFFOLD:** Applies variance reduction for non-IID data.
  + **FedNova:** Normalizes local models before averaging.
  + **Personalized FL algorithms:** Per-FedAvg, pFedMe, etc., aim for personalized models.
  + **Robust FL algorithms:** Agnostic FL, FedRobust, for robustness against distribution shifts.
  + **FedDF:** Aggregates models with different architectures using knowledge distillation.
  + **Vertical FL:** FedBCD, FDML.
* **Neural Networks:**
  + **PFNM:** Probabilistic federated neural matching, outperforms FedAvg.
  + **FedMA:** Federated matched averaging, layer-wise matching scheme.
  + **Split learning:** SplitNN, divides neural network for vertical FL.
* **Trees:**
  + **GBDT (Gradient Boosting Decision Tree):**
    - Horizontal FL: Zhao et al., Li et al., Liu et al.
    - Vertical FL: Liu et al. (Federated Forest), Cheng et al. (SecureBoost)
* **Linear/Logistic Regression:**
  + **Horizontal FL:** Nikolaenko et al. (privacy-preserving ridge regression), Chen et al.
  + **Vertical FL:** Sanil et al. (secure regression), Hardy et al. (two-party logistic regression)

**Key Considerations:**

* **Data distribution:** Horizontal vs. vertical FL.
* **Non-IID data:** Requires specialized algorithms (FedProx, SCAFFOLD, FedNova, personalized FL).
* **Model heterogeneity:** FedDF for aggregating models with different architectures.
* **Privacy-preserving techniques:** Differential privacy, encryption, secure multi-party computation.

**Further Considerations**

* Combining federated learning with other machine learning techniques (e.g., multi-task learning, reinforcement learning)
* Optimization techniques for federated learning
* Handling stragglers and noise in federated learning

**4.2.2 Communication Efficiency:**

* Optimizing communication costs is crucial due to its significant impact on training efficiency.
* Several techniques focus on reducing communication size:
  + **Structured/sketched updates:** compress updates before sending.
  + **Federated distillation:** communication depends on output dimension, not model size.
  + **Data augmentation:** make data IID to reduce communication needs.
  + **Sparse ternary compression:** robust to non-IID data and large parties.
* Beyond size, communication architecture is also important:
  + **Cross-silo FL topology design:** optimizes throughput for faster training.

**4.2.3 Privacy, Robustness and Attacks:**

* Exchanged model parameters can still leak sensitive information.
* Techniques for privacy guarantees:
  + **Differential privacy:** adds noise to protect individual data points or entire datasets.
  + **Secure multi-party computation:** securely aggregates parameters without revealing them.
  + **Combining differential privacy and secure multi-party computation:** provides stronger guarantees.

**Attacks on FL:**

* **Backdoor attacks:** aim to inject malicious updates and compromise the global model.
  + Methods include model poisoning, distributed backdoor attacks, and edge-case backdoors.
  + Defenses exist, but attackers are constantly evolving their techniques.
* **Byzantine attacks:** involve manipulating authenticated devices to disrupt the network.
  + Existing robust aggregation rules from distributed learning can be adapted to FL.
  + Byzantine-robust federated learning approaches are also being developed.
* **Inference attacks:** try to reconstruct training data from model parameters or gradients.
  + While possible, effective defenses are still under development.

**Additional Considerations:**

* Fairness in FL algorithms: ensuring all parties benefit equally from the model.
* Incentive mechanisms: motivating devices to participate and contribute honestly.

They talk about some applications like FedGKT to reduce the computational overhead to train a small part of a whole ResNet. They created a Federated recommender system, even created a variant of LSTM called coupled input and forget gate.

**FedML:**

* **Strengths:**
  + Comprehensive: Supports various ML models, FL algorithms, and computing paradigms.
  + Baseline implementations for FedAvg, FedNAS, Vertical FL, and split learning.
* **Weaknesses:**
  + Some experimental results are preliminary.

**FedEval:**

* **Strengths:**
  + Uses "ACTPR" model for evaluation (accuracy, communication, time, privacy, robustness).
  + Docker containers for isolated evaluation environment.
  + Supports FedSGD and FedAvg with MLP and LeNet models.
* **Weaknesses:**
  + Limited to horizontal algorithms and a few models.

**OARF:**

* **Strengths:**
  + Measures different FL components (algorithms, encryption, privacy, communication).
  + Realistic data partitioning using public datasets.
  + Tests both horizontal and vertical algorithms.
* **Weaknesses:**
  + No reported experimental results currently.

## **Additional Benchmark: Edge AIBench**

* **Focus:** Testbed for federated learning applications.
* **Scenarios:** ICU patients, surveillance, smart home, autonomous vehicles.
* **Current Status:** Open-source implementation, no reported results.

Overall, FATE, PaddleFL, and FedML try to provide algorithm-level APIs for users to use directly, while TFF and PySyft try to provide more detailed building blocks so that the developers can easily implement their FL process. Table 2 shows the comparison between the open-source systems. At the algorithm level, FATE is the most comprehensive system that supports many machine learning models under both horizontal and vertical settings. TFF and PySyft only implement FedAvg, which is a basic framework in FL as shown in Section 4.2. PaddleFL supports several horizontal FL algorithms currently on NNs and logistic regression. FedML integrates several state-of-the-art FL algorithms such as FedOpt and FedNova. Compared with FATE, TFF, and FedML, PySyft and PaddleFL provide more privacy mechanisms. PySyft covers all the listed features that TFF supports, while TFF is based on TensorFlow and PySyft works better on PyTorch. Based on the popularity on GitHub, PySyft is currently the most impactful federated learning system in the machine learning community

When designing a Federated Learning System it needs to be effective, user privacy, efficiency, autonomy.

# Chapter 6

This section explores three potential applications of federated learning (FL) across different industries:

**Mobile Services:**

* **Benefits:** Protects user privacy while enabling accurate predictions (e.g., next word suggestion).
* **Challenges:**
  + Limited resources on individual devices (computation, bandwidth).
  + Robustness against user churn.
  + Balancing privacy with inference attacks.
  + User incentives for data contribution.

**Healthcare:**

* **Benefits:** Enables collaborative research and diagnosis while ensuring patient privacy.
* **Challenges:**
  + Training on hybrid partitioned data (horizontal and vertical).
  + Designing decentralized FL systems robust against malicious actors.
  + Implementing privacy-preserving techniques (e.g., secure multi-party computation).

**Finance:**

* **Benefits:** Facilitates cooperation between financial institutions while protecting customer privacy.
* **Challenges:**
  + Training models on vertically partitioned data with user IDs.
  + Implementing privacy-preserving record linkage.
  + Designing cross-silo and decentralized FL systems.
  + Motivating participant institutions through shared interests.

## **Overall Insights**

Federated learning offers a promising approach for collaborative learning while preserving data privacy. However, each application domain presents unique challenges requiring tailored solutions. Addressing these challenges, particularly regarding resource limitations, robustness, privacy protection, and incentive mechanisms, is crucial for realizing the full potential of FL across various industries.

# Chapter 7

**7.1 Heterogeneity:**

* **Dynamic Scheduling:** Adapting to changes in the number of participating devices (new entries, departures).
* **Diverse Privacy Restrictions:** Accommodating different privacy requirements of participants.
* **Intelligent Benefits:** Designing fair incentive mechanisms based on data contributions.

**7.2 System Development:**

* **System Architecture:** Moving beyond limited frameworks like FedAvg to support diverse learning algorithms and aggregation methods.
* **Model Market:** Integrating FL with model sharing and trading platforms for increased flexibility and adoption.
* **Benchmark:** Establishing a robust and comprehensive benchmark for evaluating FL systems across various metrics.
* **Data Life Cycles:** Designing data handling frameworks within FL that consider security and privacy throughout the entire data lifecycle.

**7.3 FL in Domains:**

* **Internet-of-Things:** Balancing privacy, security, and resource constraints in edge computing environments.
* **Regulations:** Addressing how FL complies with existing regulations like GDPR and ensuring explainability of models.

## **Overall Takeaways:**

* Heterogeneity poses significant challenges in terms of managing dynamic participation, diverse privacy needs, and fair incentive mechanisms.
* System development requires flexible architectures, model marketplaces, robust benchmarks, and data life cycle management.
* Applying FL in specific domains like IoT necessitates tailoring solutions to resource constraints and regulations.

The highlighted challenges offer exciting opportunities for research and development to enhance the practicality, efficiency, and security of federated learning across diverse applications.