OmniFed: A Modular Framework for Configurable Federated Learning from Edge to HPC

Sahil Tyagi, Andrei Cozma, Olivera Kotevska, Feiyi Wang

{tyagis, cozmaal, kotevskao, fwang2}@ornl.gov

Analytics and AI Methods at Scale Group (AAIMS)

Oak Ridge National Laboratory

Abstract

✓ Federated learning (FL) is essential for edge and HPC environments in which data is not centralized, and privacy is critical.

✓ Our framework, **OmniFed**, written in Python and developed over PyTorch, is designed to be modular and decouples configuration, orchestration, communication, and training logic of ML models.

✓ **OmniFed** does the following:
- Enables configuration-driven prototyping and code-level override-only-what-you-need customization
- Comes with over 10 built-in FL algorithms
- Supports different topologies and mixed communication protocols (e.g., MPI, gRPC, AMQP, MQTT) within a single deployment
- Includes privacy mechanisms such as differential privacy (DP), homomorphic encryption (HE), and secure aggregation (SA)
- Provides a compression module to mitigate communication overhead in synchronous FL
- Supports streaming simulation with Apache Kafka for prototyping real-time learning

Introduction

✓ Conventional, centralized AI pipelines are no longer practical as training data becomes increasingly distributed, voluminous, and sensitive.

✓ AI models have already been pretrained on public, internet-scale data; the next frontier is learning from *private, sensitive data* residing at restricted endpoints (e.g., neutron scattering, materials science).

✓ Challenges in FL:
- Data heterogeneity: non-IID, restricted, unbalanced, skewed
- Systems and network heterogeneity: varying compute capability, latency, and bandwidth across clients
- Adversarial nodes: malicious or untrustworthy participants
- Different arrangement or setup of clients in various settings (e.g., hospitals, financial institutions, compute facilities)

✓ Privacy preservation must be an integral component of training.

✓ FL offers a transformational opportunity for the next generation of secure and distributed AI where models are trained or fine-tuned across institutional, geographic, and disciplinary boundaries.

✓ Thus, FL is crucial for scientific collaborations, national security applications, and industrial partnerships.

✓ An ideal FL framework should thus enable rapid prototyping of new algorithms without boilerplate, implement custom training topologies, deploy different training and cluster configurations, and run different parallelization schemes and communication models.

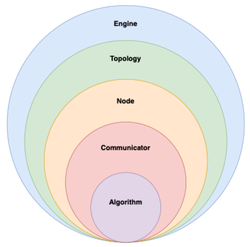
Core Vision and Components

✓ **OmniFed**'s design:
- Modularity and extensibility
- Clear and layered abstractions
- Configuration-driven, schema-based workflow
- Agnosticism to topology and communication protocols
- Developer-friendly design with consistent and intuitive usage patterns

✓ Design philosophy:
- Clear separation of concerns across components
- Modules that expose clear, minimal interfaces
- Override-only-what-you-need paradigm (e.g., add/remove privacy, compression, etc. with minimal code change)
- Single-file algorithm plug-ins

✓ Software stack:
- Implementation language: ***Python***
- ML backend: ***PyTorch***
- Configuration management: ***Hydra***
- Distributed execution: ***Ray***

✓ **OmniFed**'s core components (*on the right*):
i. Engine
ii. Topology
iii. Node
iv. Communicator
v. Algorithm



❖ ***Engine***:
- Provides orchestration endpoint for FL experiments
- Launches and coordinates distributed jobs
- Manages node life cycle and resource allocation

❖ ***Topology***:
- Defines structure, coordination, and communication patterns between distributed *Nodes*
- Supports templates for centralized, decentralized, and hierarchical setups
- Enables custom topology definition via graph-based representation

❖ ***Node***:
- Represents any participant in the federation that manages model state, data, and device resources
- Interfaces with the *Communicator* for data exchange
- Executes the local *Algorithm* implementations

❖ ***Communicator***:
- Provides abstraction that exposes a unified API for different underlying protocols (e.g., MPI for HPC settings, gRPC for geo-distributed devices, messaging queues for IoT, middleware)

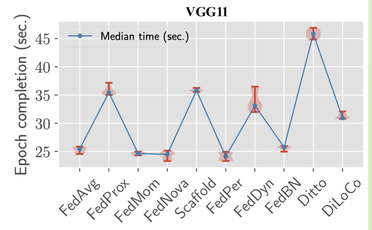
❖ ***Algorithm***:
- Implements the inter-/intranode training logic, such as federated averaging (FedAvg), federated proximal averaging (FedProx), and DiLoCo (distributed low-communication)

Preliminary Results

✓ **OmniFed** enables training with different FL algorithms with a single line change to the Hydra configuration YAML file (*below right*):

✓ Currently supported algorithms:
- FedAvg
- FedProx
- FedMom
- FedNova
- Scaffold
- Moon
- FedPer
- FedDyn
- FedBN
- Ditto
- DiLoCo

```
1  defaults:
2    - override topology: centralized
3    - override algorithm: fedavg
4    - override model: resnet18
5    - override datamodule: cifar10
6
7  topology:
8    _target_: src.omnifed.topology.CentralizedTopology
9    num_clients: 8
10   inner_comm:
11     _target_: src.omnifed.communicator.GrpcCommunicator
12     master_port: 50051
13     master_addr: 127.0.0.1
14
15 algorithm:
16   _target_: src.omnifed.algorithm.FedAvg
17   lr: 0.01
18
19 global_rounds: 2
```



✓ With a simple change (at line 16), users can compare the parallel performance (e.g., epoch completion) vs. statistical quality (e.g., test loss, accuracy) b/w FL algorithms (*shown above and below*).

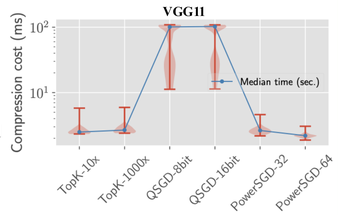
Algorithm	ResNet18	VGG11	AlexNet	MobileNetV3
FedAvg	99.32%	**86.6%**	87.9%	81.35%
FedProx	99.26%	86.31%	87.98%	**82.96%**
FedMom	99.14%	66.39%	63.85%	48.98%
FedNova	91.18%	14.1%	58.1%	22.27%
Moon	**99.46%**	81.67%	87.28%	81.4%
FedPer	90.9%	26.93%	82.94%	14.59%
FedDyn	99.31%	86.18%	**88.78%**	79.15%
FedBN	99.33%	86%	88.7%	78.65%
Ditto	73.64%	5.5%	40%	9.84%
DiLoCo	84.88%	5.1%	45.17%	15.47%

Table 1: Convergence quality of various FL algorithms.

✓ To mitigate communication cost during synchronization, **OmniFed** has a compression module integrated with the communicator.

```
1 inner_comm:
2   _target_: src.omnifed.communicator.TorchDistCommunicator
3   port: 28670
4   compression:
5     _target_: src.omnifed.communicator.compression.TopK
6     k: 100x
```

✓ The following gradient compression techniques are currently implemented: *sparsification* (TopK), *quantization* (QSGD), and *low-rank approximation* (PowerSGD).



✓ Like algorithms, compression can be enabled or disabled with a few line changes from the YAML configuration.

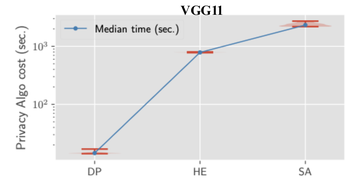
Compression specified via YAML configuration (top), and overhead of different methods over varying degrees of compression (bottom)

Ongoing and Future Work

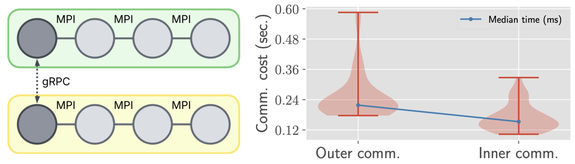
✓ Security and privacy are critical aspects for trustworthy FL.

```
1 inner_comm:
2   _target_: src.omnifed.communicator.TorchDistCommunicator
3   port: 28670
4   privacy:
5     _target_: src.omnifed.privacy.DifferentialPrivacy
6     \epsilon: 10.0
7     \delta: 0.00001
```

✓ Features such as DP, HE, and SA enable users to easily compare the compute cost of different privacy mechanisms (*right*).



✓ With its modular design, **OmniFed** supports custom topologies with mixed communication protocols, as shown below:



✓ ***Example of cross-facility FL***: (*Above left*) A hierarchical topology is shown with densely connected nodes *within* groups and sparse connections *across* groups. The outer communicator corresponds to gRPC use for exchanging data across groups, and MPI is used as the inner communicator for aggregation within a group.

✓ (*Above right*) MPI-based collectives like Allreduce are faster than centralized communication over low-latency, high-bandwidth links.

✓ With **OmniFed**'s modular design, developers can apply compression ***only*** over the outer communicator and aggregate full model updates among clients within a group.

Want to get involved?
Contact Sahil Tyagi (tyagis@ornl.gov).

GitHub repository ➡

Acknowledgments

✓ This work has been authored by UT-Battelle LLC under contract DE-AC05-00OR22725 with the US Department of Energy (DOE).

✓ The US government retains and the publisher, by accepting the article for publication, acknowledges that the US government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this manuscript, or allow others to do so, for US government purposes.

✓ DOE will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (https://www.energy.gov/doe-public-access-plan).