Resilient Federated Learning Framework

Supervisor: Prof. Mário Luís Pinto Antunes
Co-supervisor: Prof. Rui Aguiar

08-07-2025

universidade de aveiro
theoria poiesis praxis

30 YEARS instituto de telecomunicações

Leonardo Almeida 102536

Table of Contents

Introduction

Context:

- AI is increasingly used daily
- Sensitive/distributed data is a challenge
- **Solution:** Federated Learning
- Robustness and Resilience are critical
- Resilient Federated Learning Framework

Challenges in FL:

- Node failure and dynamic networks
- Non-iid data, communication overhead and reliable model aggregation
- Existing works lack modularity and resilience

Resilience
The system's ability to maintain its performance and training, even in the presence of node failures, delays, and other adversities

Contributions

- **FlexFL:** Modular and Resilient FL Framework
- Dissertation Document
- Research Grant awarded by University of Aveiro and Instituto de Telecomunicações
- Guidance and Supervision of a PECI group
- 9 Scientific Publications

Status	Number of papers
Published	5
Accepted	3
Under Review	1

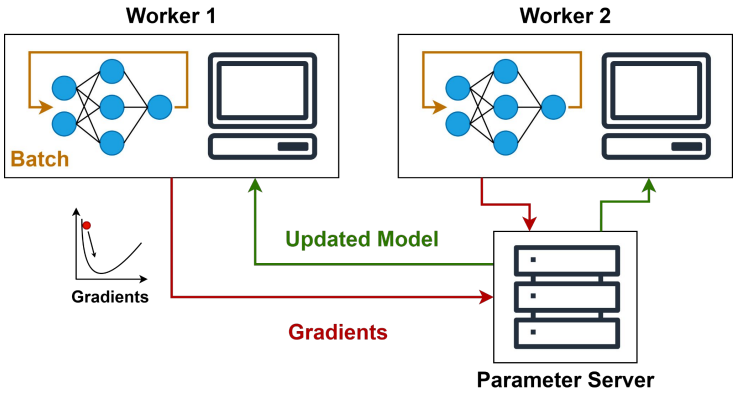
Background and Related work

Federated Learning

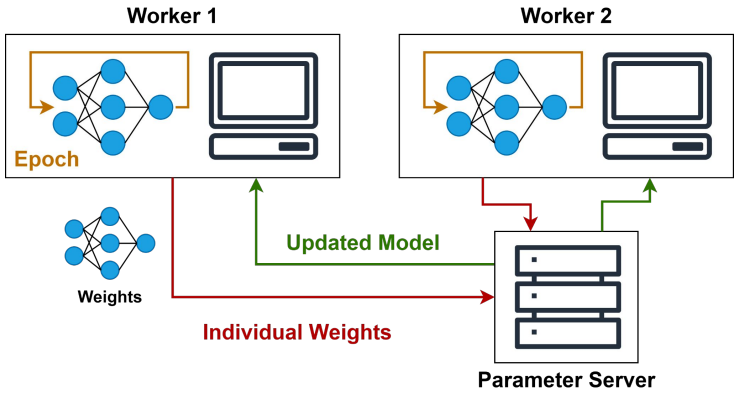
Subset of Distributed Learning

- Data Parallelism: allows data privacy
- **Horizontal FL**, Vertical FL and Federated Transfer Learning

- Centralized vs Decentralized Optimization
- Synchronous vs Asynchronous Scheduling



Centralized Optimization



Decentralized Optimization

Background and Related work

Communication Protocols

Summary and Comparison

Protocol	Scalability	Fault Tolerance	Security	Suitability
MPI	Limited	No built-in support	Relies on SSH	Low, lacks fault tolerance and scalability
MQTT	Moderate	Moderate	TLS, mTLS role-based	High, suitable when using a central broker
Kafka	High	High	TLS, mTLS role-based	Moderate, latency can hinder synchronization
Zenoh	High	High	TLS, mTLS ACL	High, but limited ecosystem maturity

Background and Related work

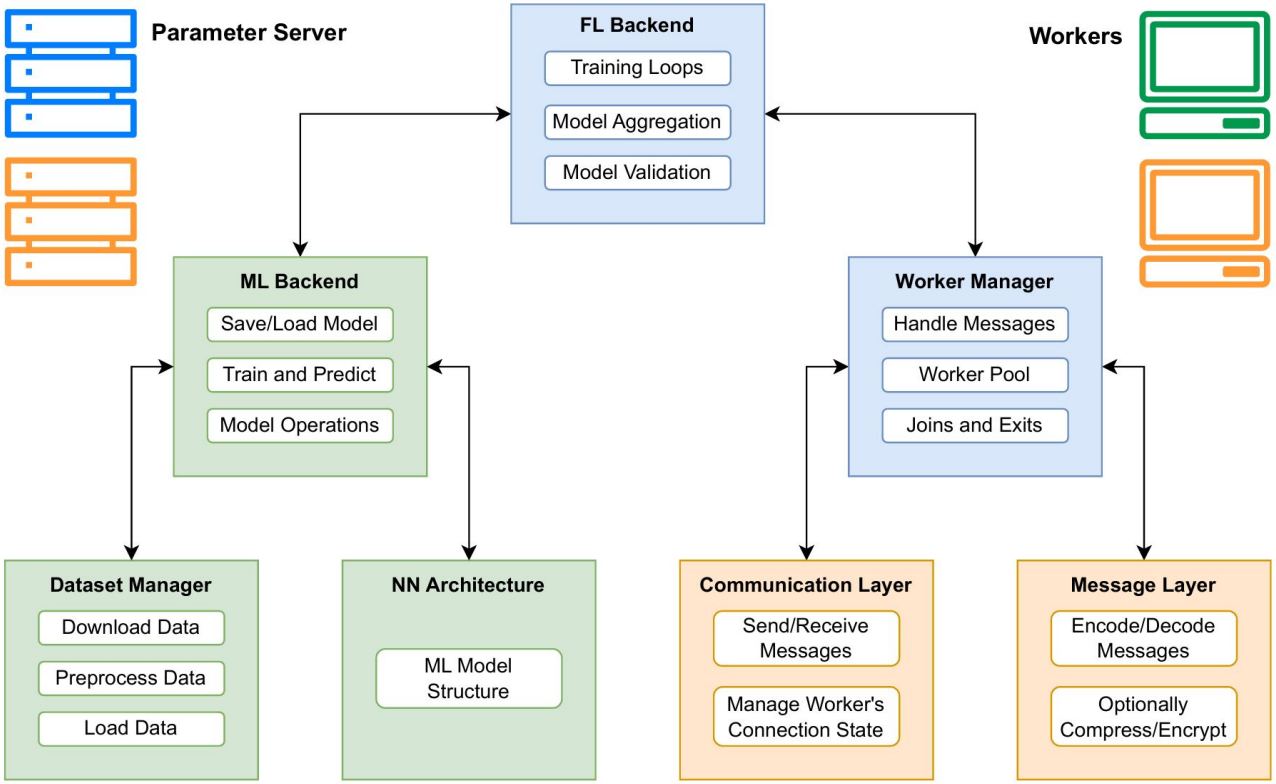
Requirements and Existing Works

Qualitative comparison of the proposed solution with existing solutions

Solution	Resilience	Modularity	Analysis	Code and Documentation	Compliance
Awan *et al.*	X	X	✓	X	25.0%
Jayaram *et al.*	✓	X	✓	X	50.0%
Chen *et al.*	X	X	✓	X	25.0%
Morell and Alba	✓	X	✓	✓	75.0%
Dautov and Husom	X	X	✓	✓	50.0%
HeteroFL	✓	X	✓	✓	75.0%
CoCoFL	✓	X	✓	✓	75.0%
Flower	X	✓	X	✓	50.0%
TTF	✓	X	X	✓	50.0%
Proposed	✓	✓	✓	✓	100.0%

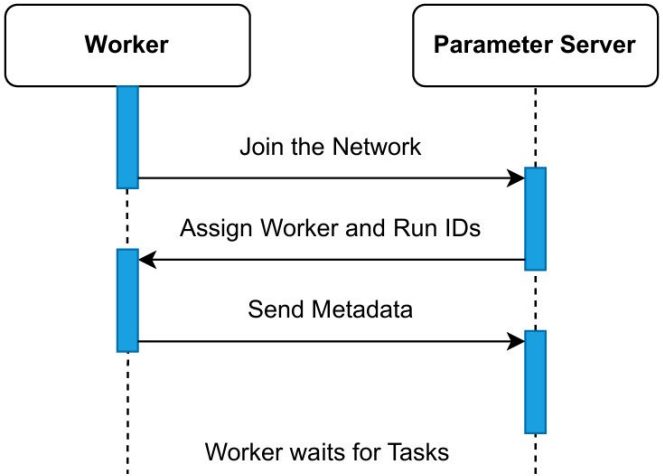
Proposed solution

Architecture



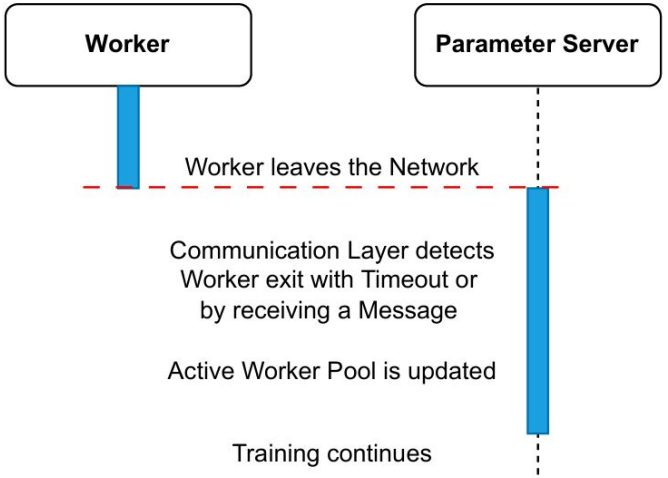
Proposed solution
Communication Strategies

- Adapt training to each worker



Worker Joining Sequence

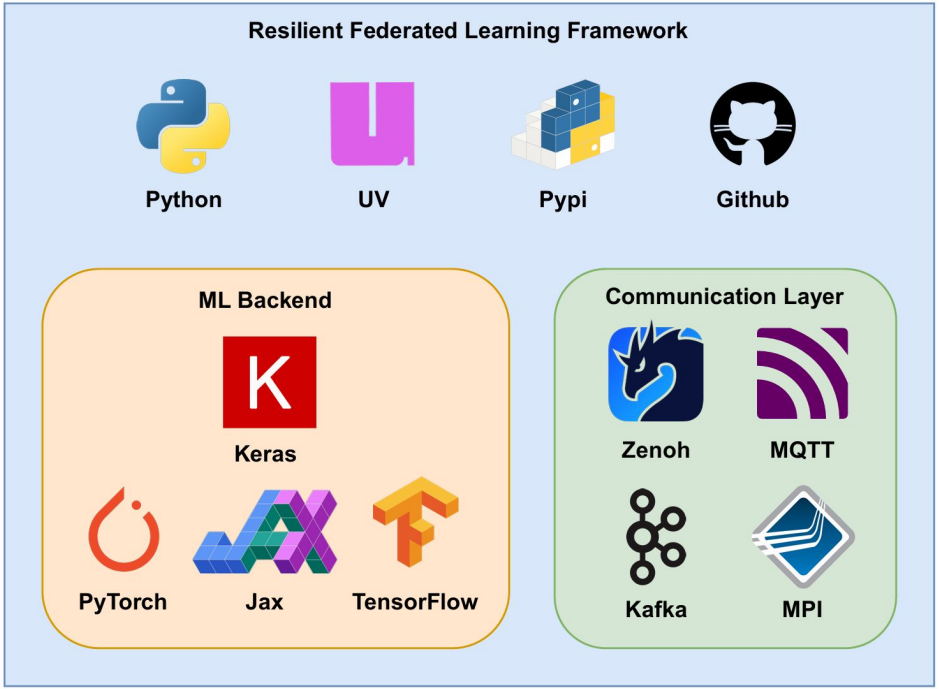
- Dynamic worker participation



Worker Exiting Sequence

Implementation
Software Stack

- **Python:** rich ML ecosystem and ease of development
- **UV:** Project & Dependency Management
- **Git + GitHub:** Version Control
- **PyPI:** Package Distribution
- Multiple ML Backends and Communication Protocols
- Highly modular, flexible and extensible



Implementation

FL Algorithms

All **4 FL Algorithms** were implemented with key **adaptations** to address **resilience**:

- Worker subpool
- Scheduling Policy
- Task completion threshold
- Task rescheduling

Optimizations:

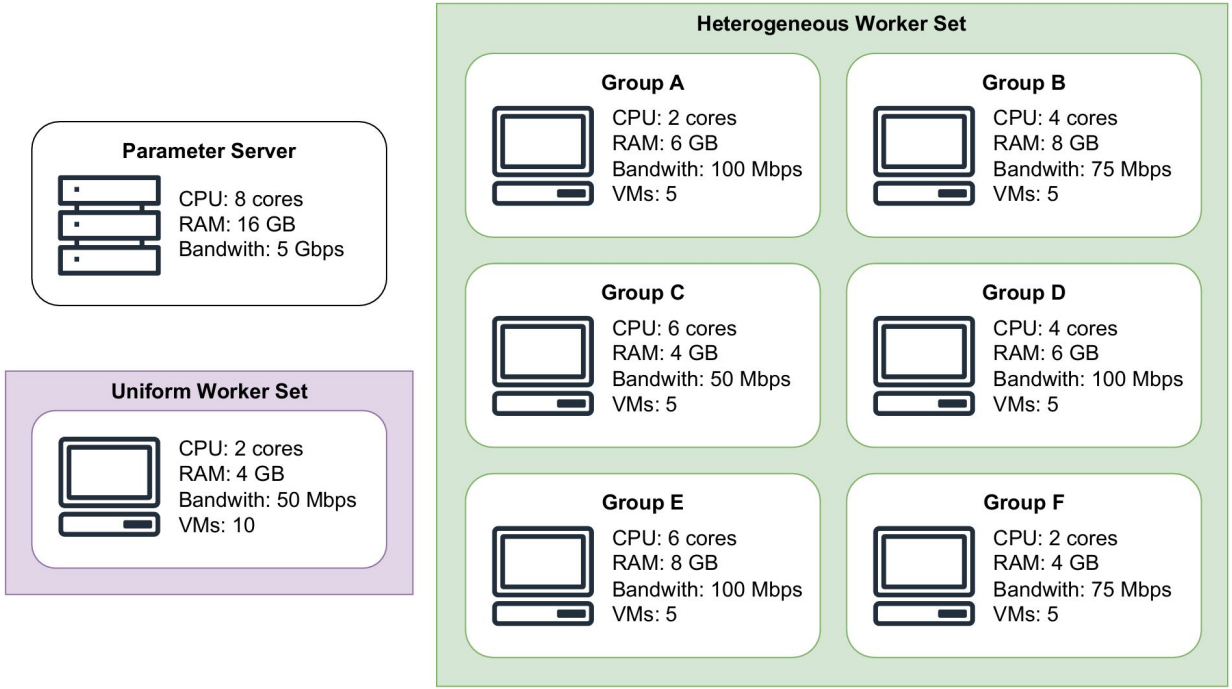
- Model aggregation occurs as updates arrive
  From $N \times M$ bytes to $2 \times M$ bytes
- Training tasks sent before validation phase
  Reduce training time up to $E \times T$ seconds

N: Number of workers **M:** Model size in bytes
E: Number of epochs **T:** Time for validation

Experimental Setup

Environment

- Arm based CPU
- Ubuntu 24.04 LTS
- 2 VMs groups
- Local high speed NTP server

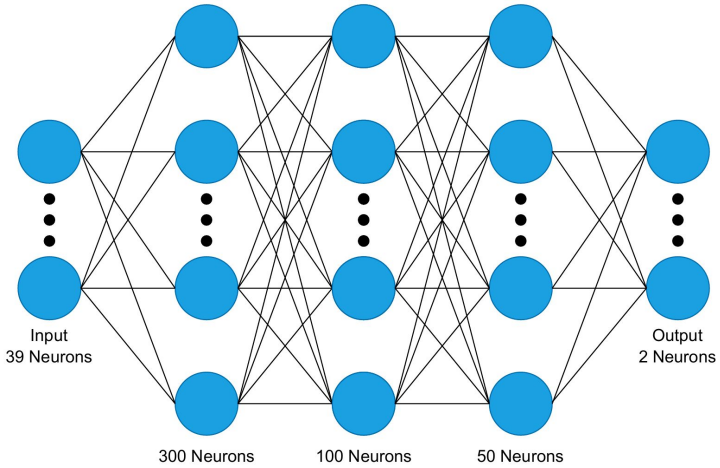


Experimental Setup

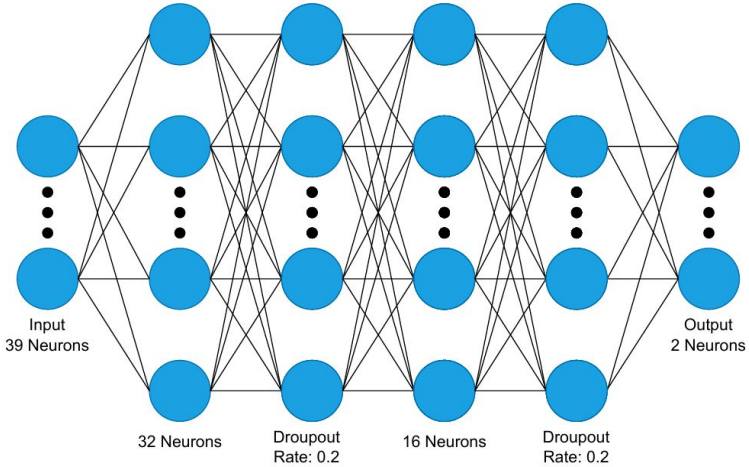
Datasets and Models

- **2 IDS datasets:** UNSW-NB15 and ToN-IoT
- Independent and Identically Distributed
- Each node performs local normalization using its **own statistics**

	UNSW-NB15	ToN-IoT
Samples	2 million	13 million
Data Division Train/Val/Test	70%, 15%, 15%	90%, 5%, 5%
Prediction	Binary Normal/Malicious Classification	



UNSW-NB15 Model



ToN-IoT Model

Experimental Setup

Experimental Scenarios

Scenarios:

- **1:** Impact of Communication Protocols and FL Algorithms
- **2:** Resilience Evaluation under Worker Failures
- **3:** Scalability and Convergence with Large-Scale Failures

Environment:

- 1 and 2 use UNSW-NB15 dataset with the uniform worker set
- 3 uses both datasets and all 40 workers

Hyperparameters used, representing a typical configuration

Hyperparameter	Value
ML Backend	Keras with TensorFlow
Optimizer	Adam
Loss function	Categorical Crossentropy
Batch size	1024
Learning rate	0.0001
Number of Epochs	10
Local Epochs per worker	3
Worker threshold	50%
Workers per epoch	70%

Results - Scenario 1

Impact of Communication Protocols and FL Algorithms

4 FL Algorithms:

- Centralized Sync/Async (CS/CA)
- Decentralized Sync/Async (DS/DA)

4 Communication Protocols:

- MPI for benchmark with Decentralized Sync
- Zenoh, MQTT and Kafka with all 4

Evaluation:

- Average of 3 runs
- After a warm-up run

Results - Scenario 1

Federated Learning Algorithms Comparison

- Centralized methods exchange a higher frequency and volume of messages
- Asynchronous approaches are expected to vary in message count between workers
- Centralized Sync also vary unlike Decentralized Sync

FL Approach	Comm Protocol	Average Number Of Messages	Average Total Payload Size (MB)
Decentralized Synchronous	MPI	**17.4** $\pm$ 0.932	**2.71** $\pm$ 0.164
	Zenoh	17.4 $\pm$ 0.932	2.71 $\pm$ 0.164
	MQTT	17.4 $\pm$ 0.932	2.71 $\pm$ 0.164
	Kafka	17.4 $\pm$ 0.932	2.71 $\pm$ 0.164
Decentralized Asynchronous	Zenoh	17.4 $\pm$ **0.932**	2.71 $\pm$ **0.164**
	MQTT	17.4 $\pm$ **0.932**	2.71 $\pm$ **0.164**
	Kafka	17.4 $\pm$ **1.302**	2.71 $\pm$ **0.229**
Centralized Synchronous	Zenoh	2285.4 $\pm$ **2.044**	403.0 $\pm$ **0.360**
	MQTT	2285.4 $\pm$ **1.935**	403.0 $\pm$ **0.309**
	Kafka	2285.4 $\pm$ **1.069**	403.0 $\pm$ **0.188**
Centralized Asynchronous	Zenoh	**2285.4** $\pm$ 1.302	**403.0** $\pm$ 0.229
	MQTT	2285.4 $\pm$ 1.975	403.0 $\pm$ 0.348
	Kafka	2285.4 $\pm$ 1.302	403.0 $\pm$ 0.229

Results - Scenario 1

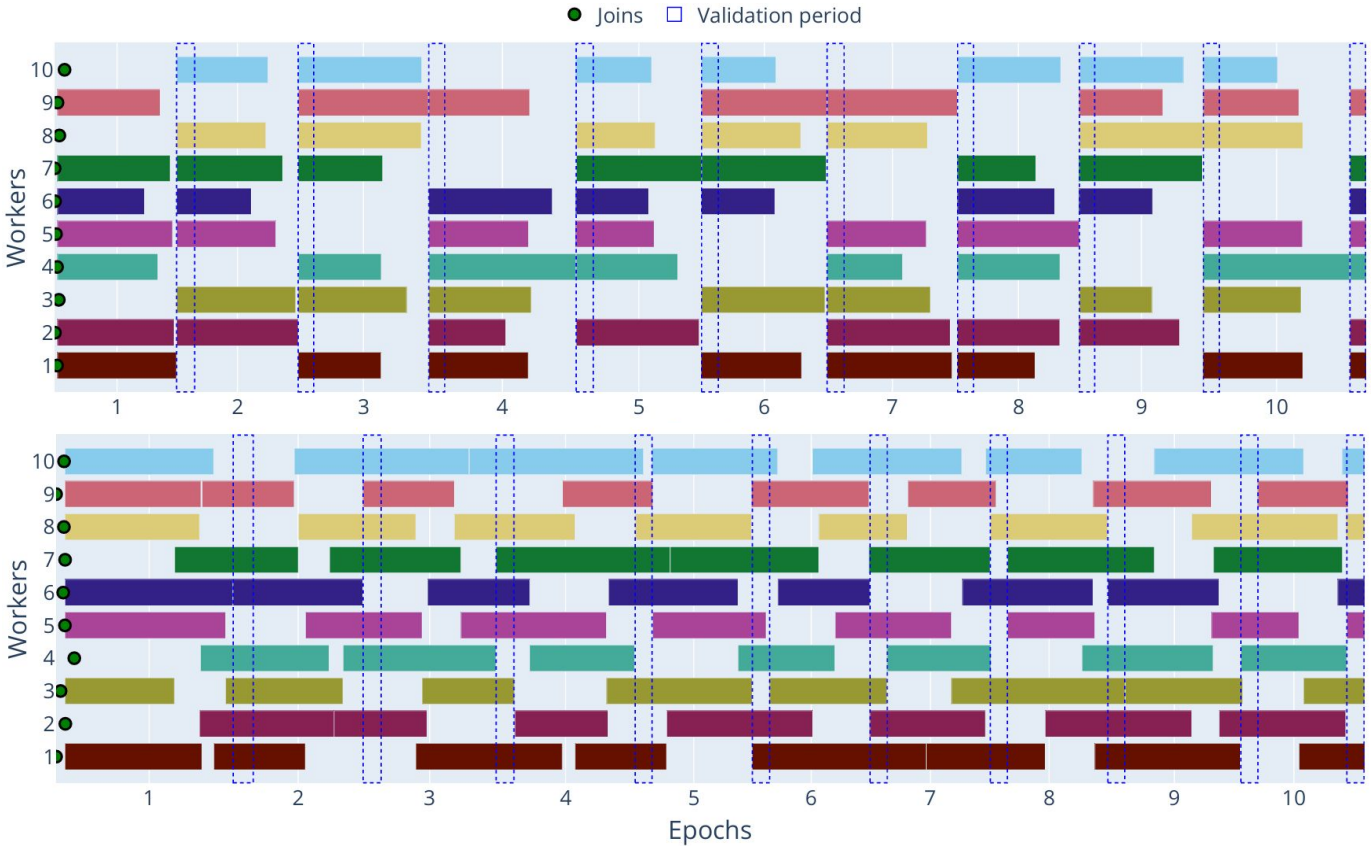
Federated Learning Algorithms Comparison

- Centralized has lower work time and higher run time than Decentralized approaches
- More noticeable in the Centralized Asynchronous setting
- Decentralized Async avoid straggler problem from Decentralized Sync

FL	Comm Protocol	Average Comm Time (s)	Average Work Time (s)	Average Total Run Time (s)
DS	MPI	0.422 ± 0.094	83.92 ± 13.54	139.6 ± 17.88
	Zenoh	0.097 ± 0.021	**82.69** ± 7.823	**136.7** ± 5.091
	MQTT	0.115 ± 0.022	82.51 ± 7.927	138.1 ± 8.274
	Kafka	1.705 ± 0.213	83.43 ± 7.800	**139.9** ± 4.286
DA	Zenoh	0.079 ± 0.014	82.45 ± 8.541	114.2 ± 1.436
	MQTT	0.098 ± 0.016	82.71 ± 7.866	114.8 ± 1.519
	Kafka	1.690 ± 0.182	83.20 ± 7.322	**117.6** ± 0.276
CS	Zenoh	23.35 ± 5.524	**45.35** ± 2.023	**269.2** ± 1.609
	MQTT	25.39 ± 3.492	48.39 ± 3.820	250.9 ± 26.09
	Kafka	360.3 ± 2.206	47.36 ± 2.408	1288.2 ± 6.104
CA	Zenoh	9.737 ± 0.248	47.19 ± 2.287	**728.6** ± 17.57
	MQTT	12.77 ± 0.195	46.58 ± 2.796	744.2 ± 15.76
	Kafka	253.8 ± 3.879	48.37 ± 2.251	1865.0 ± 19.69

Results - Scenario 1

Federated Learning Algorithms Comparison



Decentralized Sync
Zenoh
Avg run time:
136.7 ± 5.091 seconds

Decentralized Async
Zenoh
Avg run Time:
114.2 ± 1.436 seconds

Results - Scenario 1

Communication Protocols Comparison

- Zenoh and MQTT consistently achieved lower communication times than MPI and Kafka
- More evident in Centralized runs
- Similar work time across all settings
- Communication times are slightly higher in Sync approaches

FL	Comm Protocol	Average Comm Time (s)	Average Work Time (s)	Average Total Run Time (s)
DS	MPI	**0.422** ± 0.094	83.92 ± 13.54	139.6 ± 17.88
	Zenoh	**0.097** ± 0.021	82.69 ± 7.823	136.7 ± 5.091
	MQTT	**0.115** ± 0.022	82.51 ± 7.927	138.1 ± 8.274
	Kafka	**1.705** ± 0.213	83.43 ± 7.800	139.9 ± 4.286
DA	Zenoh	0.079 ± 0.014	**82.45** ± 8.541	114.2 ± 1.436
	MQTT	0.098 ± 0.016	**82.71** ± 7.866	114.8 ± 1.519
	Kafka	1.690 ± 0.182	**83.20** ± 7.322	117.6 ± 0.276
CS	Zenoh	**23.35** ± 5.524	45.35 ± 2.023	269.2 ± 1.609
	MQTT	**25.39** ± 3.492	48.39 ± 3.820	250.9 ± 26.09
	Kafka	**360.3** ± 2.206	47.36 ± 2.408	1288.2 ± 6.104
CA	Zenoh	**9.737** ± 0.248	47.19 ± 2.287	728.6 ± 17.57
	MQTT	**12.77** ± 0.195	46.58 ± 2.796	744.2 ± 15.76
	Kafka	**253.8** ± 3.879	48.37 ± 2.251	1865.0 ± 19.69

Results - Scenario 2

Resilience Evaluation under Worker Failures

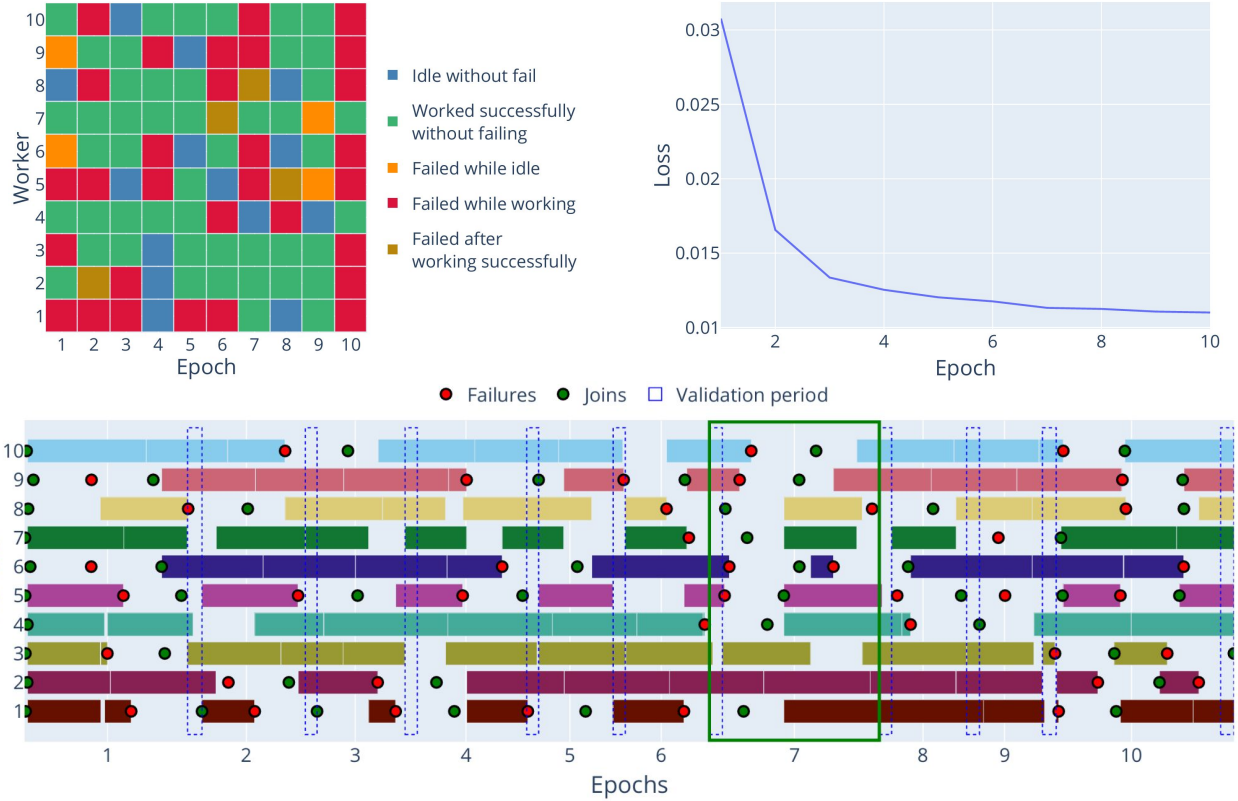
- All runs completed successfully
- Failures increased with failure rate
- Different amount of failure with the same seed
- 1% and 3% failure rate used as a stress test
- Run duration increased with failure rate

Worker Failures in Decentralized Asynchronous FL

Failure Rate	Comm Protocol	Non-Critical Failures	Critical Failures	Total Failures	Total Run Duration (s)
0.5% every Second	Zenoh	3	5	**8**	123.3
	MQTT	2	6	8	**125.0**
	Kafka	1	7	8	121.6
1% every Second	Zenoh	3	**10**	**13**	123.8
	MQTT	2	**12**	14	**129.5**
	Kafka	5	**8**	13	124.1
3% every Second	Zenoh	8	30	**38**	149.4
	MQTT	2	38	40	**155.7**
	Kafka	7	37	44	170.6

Results - Scenario 2

Resilience Evaluation under Worker Failures



Zenoh with 3% failure rate:

- Epoch-by-epoch view of each worker's state
- Chaotic scenario
- Model converged
- Task rescheduling
- Training paused at epoch 7

Results - Scenario 3

Scalability and Convergence with Large-Scale Failures

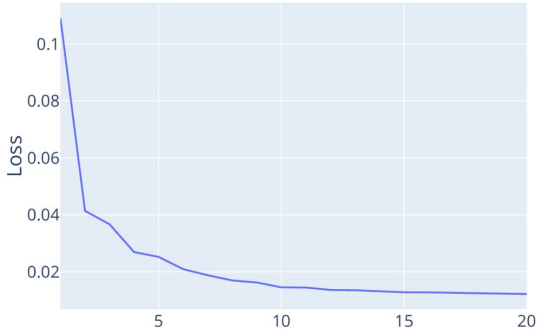
- **40 workers:** less data per worker
- 20 epochs
- Decentralized Async with Zenoh
- 1% Failure rate
- **2 Datasets:** UNSW-NB15 (2 million samples) and ToN-IoT (13 million samples)

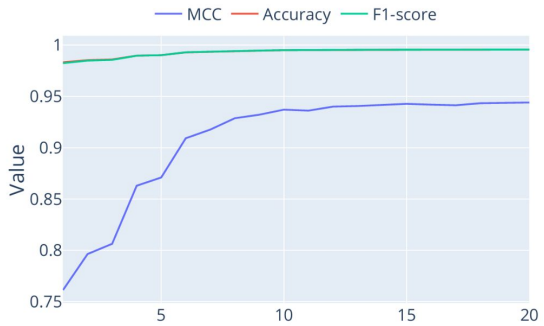
Dataset	Non-Critical Failures	Critical Failures	Total Failures	Total Run Duration (s)
UNSW-NB15	9	8	17	62.6
ToN-IoT	20	27	47	122.5

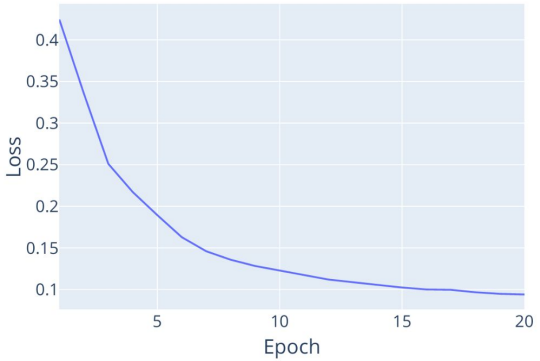
- Faster run duration compared to Scenario 2 (62.6 vs 123.8 secs)
- Higher number of failures
- Run duration difference between datasets

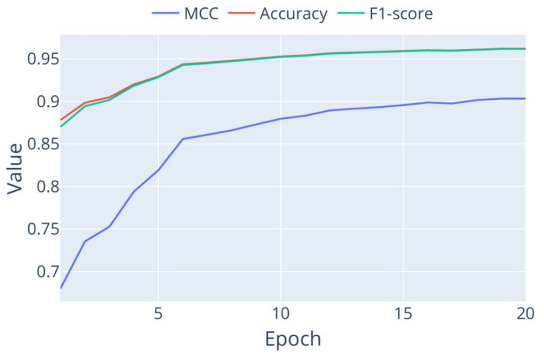
Results - Scenario 3

Scalability and Convergence with Large-Scale Failures









UNSW-NB15:

- Final MCC: 0.944

ToN-IoT:

- Final MCC: 0.903

Results

Demonstration

https://youtu.be/XG3y8HlvgHY

Conclusion

Findings and Accomplishments

- Novel resilient and modular federated learning framework
- Integration with multiple ML backends, communication protocols and FL strategies
- Comparison of different configurations through experimental evaluation
- Resilience given by dynamic worker pool, task rescheduling and completion thresholds
- The framework successfully maintained training continuity and model convergence with large scale worker failures

Conclusion

Limitations and Future Work

Adress Non-IID

FL Algorithms

Local data augmentation/re-sampling

Model aggregation strategies

Scheduling policies

Hierarchical FL

Larger heterogeneous environments

Compression and Encryption

Trade-offs

Performance overhead

Thank you for your attention

Any questions?

universidade de aveiro
theoria poiesis praxis

30 YEARS
instituto de telecomunicações