Federated Learning under Distributed Concept Drift

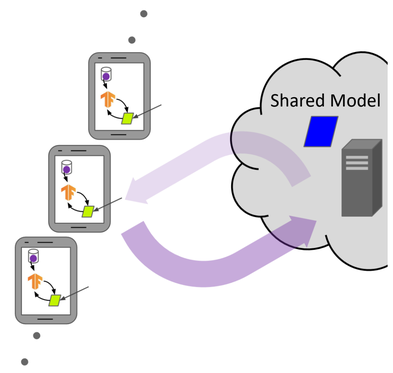
Ellango Jothimurugesan[1], Kevin Hsieh[2], Jianyu Wang[1], Gauri Joshi[1], Phillip B. Gibbons[1]

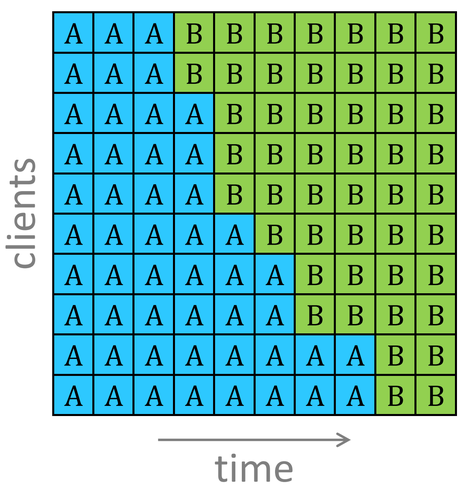
[1] Carnegie Mellon University, [2] Microsoft Research

Distributed Concept Drift

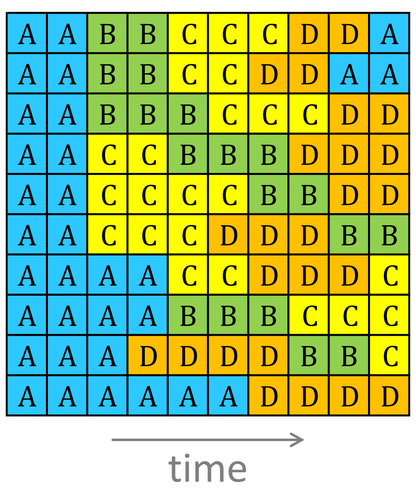
- Data are decentralized and continuously arriving over time
- At each client c and each time step t, data are drawn from a distribution $P_c^{(t)}(x, y)$
- *Concept drift* occurs if $P_c^{(t)} \neq P_c^{(t-1)}$
- Distributed concept drift poses previously unaddressed challenges:

Federated Learning (FL): collaborative training across clients without sharing raw training data

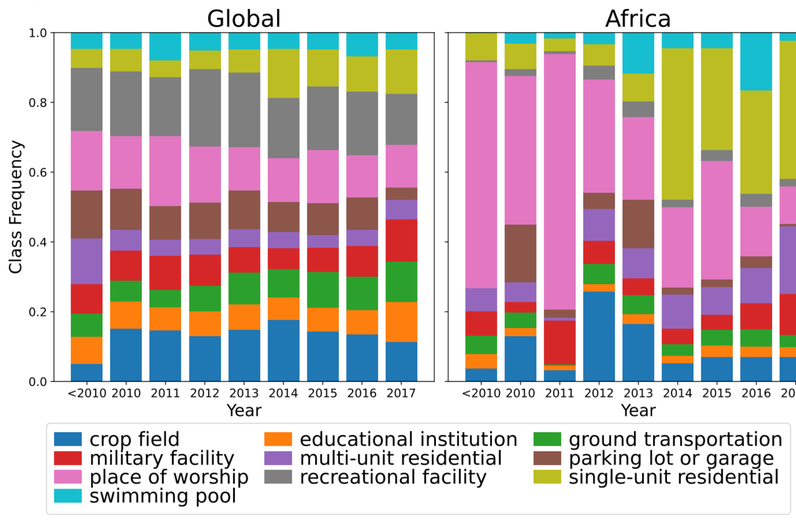




Drifts can occur staggered in time across clients



Multiple concepts can arise at the same time



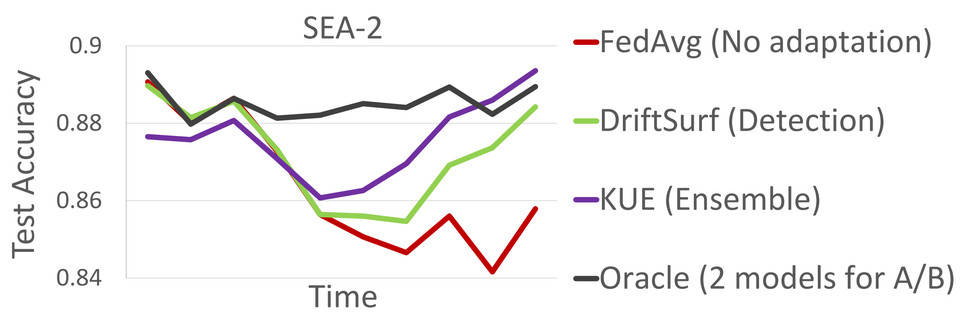
Drifts can evolve at varying rates—the global drift is small relative to the local drift for Africa (FMoW dataset)

- Objective: High accuracy on test data at each client, at each time step

Centralized Algorithms are Suboptimal

Experimental observations from the 2-concept staggered drift:

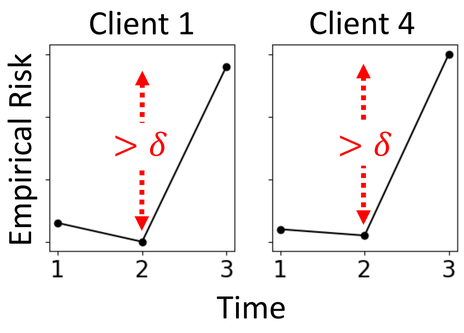
- Locally, drift occurs abruptly; globally, drift is slow and hard to detect
- No single global model works well for all clients when multiple concepts exist
- Ensemble algorithm (KUE) also suboptimal—new models are trained over a mixture of both concepts



FL Solution: FedDrift

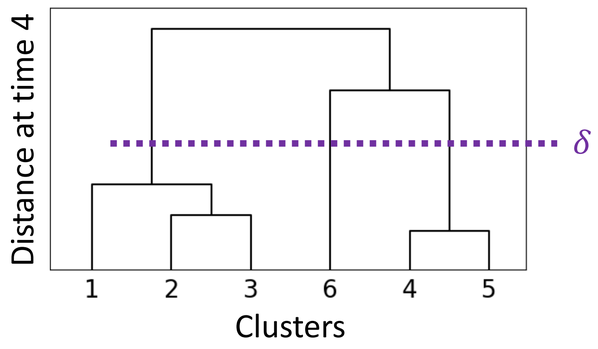
- FedDrift employs multiple models, each trained by a cluster of clients
- Challenge: determining the right number of clusters. FedDrift runs 2 subroutines each time step:

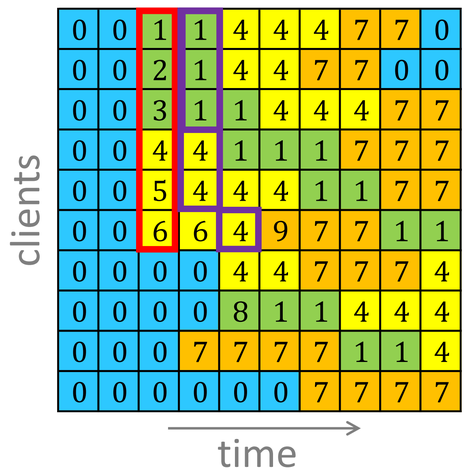
1. Eager Splitting
Isolate clients into individual clusters via local drift detection of size δ



2. Lazy Merging
Merge clusters via hierarchical clustering up to distance δ

$d(i, j)$ = risk of model trained by cluster i over data of cluster j, relative to data of cluster i





Clustering learned on MNIST-4
Color: Ground-truth
Number: Cluster ID

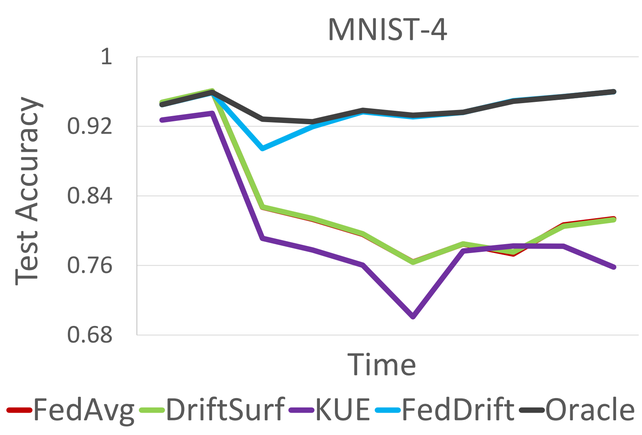
- Unified hyperparameter δ can be interpreted as tolerance for performance loss
- Iterative merging accounts for new model warm-up where cluster distances vary over time

Experimental Results

FedDrift achieves significantly higher and more stable accuracy than existing baselines, and similar accuracy to an Oracle algorithm

Average test accuracy across clients and time (5 trials)

	SINE-2	CIRCLE-2	SEA-2	MNIST-2	SEA-4	MNIST-4	FMoW
Oblivious	52.11 ± 1.79	88.38 ± 0.17	86.46 ± 0.22	87.37 ± 0.16	85.40 ± 0.09	82.95 ± 0.03	58.57 ± 0.07
DriftSurf	84.18 ± 1.40	92.34 ± 0.38	87.20 ± 0.27	93.26 ± 0.52	85.55 ± 0.13	82.97 ± 0.09	58.45 ± 0.19
KUE	86.86 ± 0.17	93.71 ± 0.14	87.25 ± 0.94	90.44 ± 0.44	85.09 ± 0.86	79.89 ± 0.26	33.11 ± 6.09
AUE	86.00 ± 0.95	92.84 ± 0.19	87.48 ± 0.07	92.22 ± 0.05	85.47 ± 0.12	82.07 ± 0.47	54.23 ± 0.14
Window	86.28 ± 0.64	93.72 ± 0.14	87.94 ± 0.10	92.34 ± 0.07	85.72 ± 0.13	81.43 ± 0.44	58.88 ± 0.15
Adaptive-FedAvg	74.10 ± 10.03	86.26 ± 0.00	86.77 ± 0.53	92.18 ± 0.05	85.25 ± 0.27	81.64 ± 0.04	52.82 ± 0.21
IFCA+Window	**98.49 ± 0.13**	94.31 ± 1.62	**88.04 ± 0.17**	91.76 ± 0.50	86.17 ± 1.00	81.27 ± 0.43	49.40 ± 0.76
CFL+Window	96.92 ± 1.84	96.04 ± 1.56	87.81 ± 0.32	90.66 ± 0.35	86.06 ± 0.11	80.51 ± 0.72	58.82 ± 0.11
FedDrift-Eager	97.53 ± 0.13	**97.82 ± 0.17**	87.51 ± 0.88	**95.52 ± 0.11**	87.61 ± 1.26	90.69 ± 1.20	61.77 ± 0.51
FedDrift	97.43 ± 0.06	**97.82 ± 0.19**	87.29 ± 0.75	95.48 ± 0.08	**88.13 ± 0.76**	**93.80 ± 0.08**	**64.84 ± 0.33**
Oracle	98.45 ± 0.03	97.84 ± 0.22	87.76 ± 0.98	95.54 ± 0.11	88.79 ± 0.41	94.30 ± 0.08	-



Algorithms evaluated

- Oblivious: FedAvg (McMahan et al. ‘17) with no adaptation
- DriftSurf (Tahmasbi et al. ‘21): drift detection
- KUE (Cano and Krawcyzk ‘20) and AUE (Brzezinski and Stefanowski ‘13): ensembles
- Window: forget past one time step
- Adaptive-FedAvg (Canonaco et al. ‘21): single-model FL with adaptive learning rate
- IFCA (Ghosh et al. ‘21) and CFL (Sattler et al. ‘20): static clustered FL algorithms extended with the window method
- FedDrift-Eager: variant of FedDrift that eagerly merges all drifted clients each time step—more efficient than hierarchical clustering, but inaccurate when multiple concepts arise
- Oracle: idealized algorithm that uses ground-truth clustering