Master’s Thesis Presentation

Concept Drift Detection and Adaptation in Federated Learning

Iftekhar Rahman
Matr. Nr. 21863027

Supervisors:

Prof. Dr.-Ing. Stefan Schulte
Institute for Data Engineering
TU Hamburg

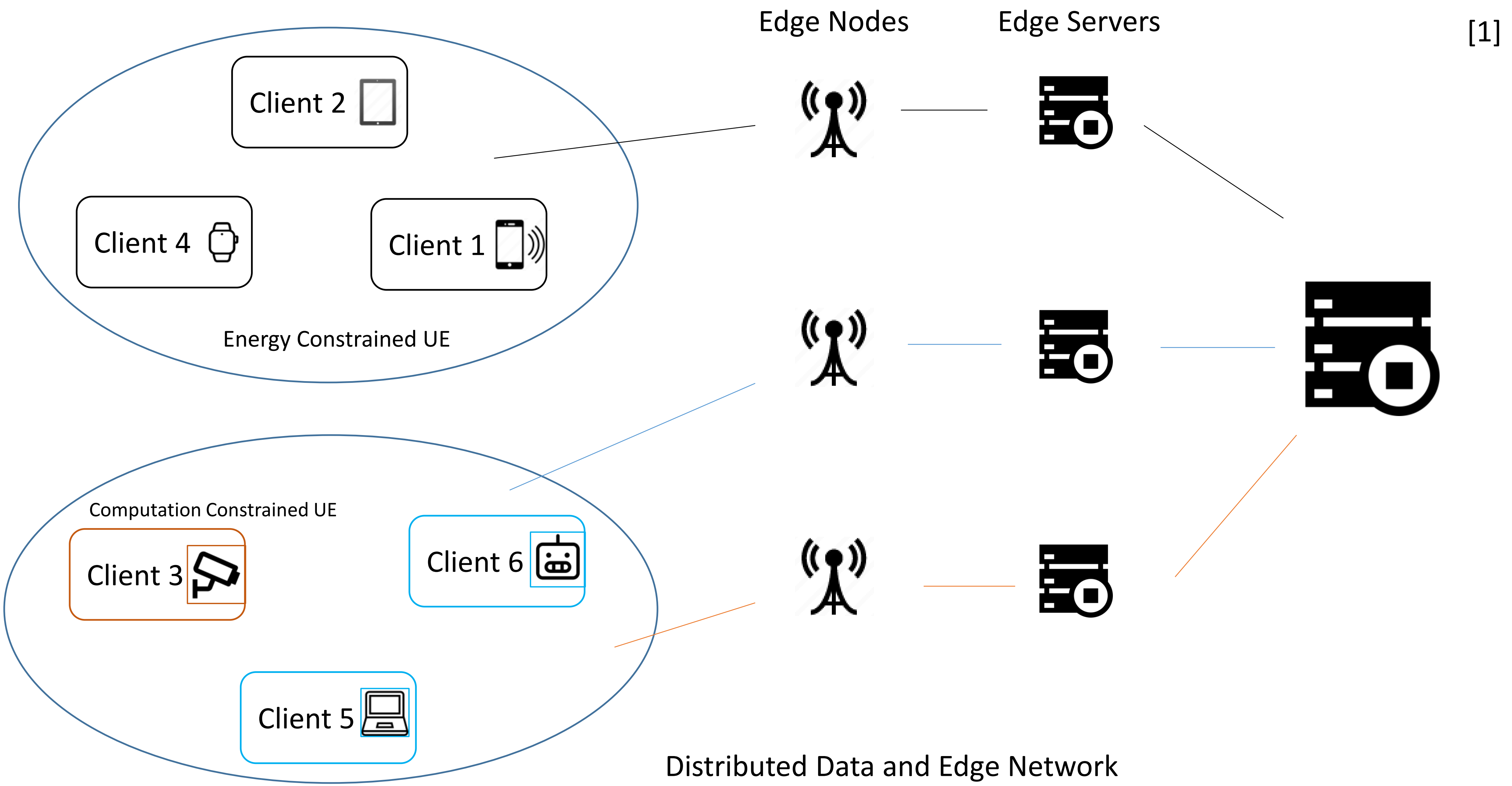
Prof. Dr. Pierre-Alexandre Murena
Human-Centric Machine Learning
TU Hamburg

TUHH

institute for data engineering

Table of Contents

Introduction and Motivation

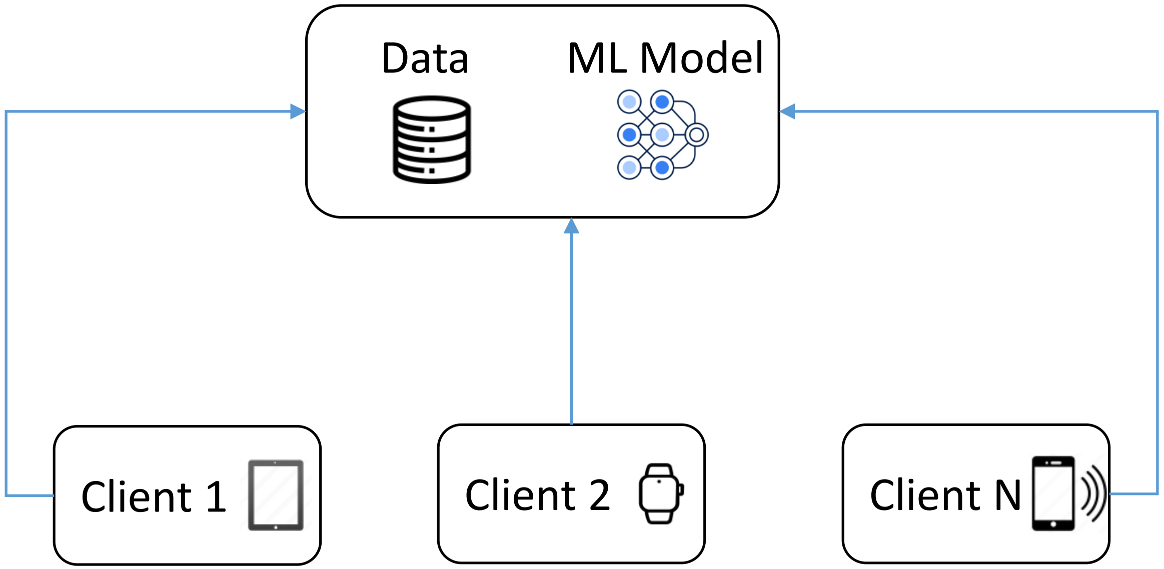


Distributed Data and Edge Network

Introduction and Motivation

Centralized Learning

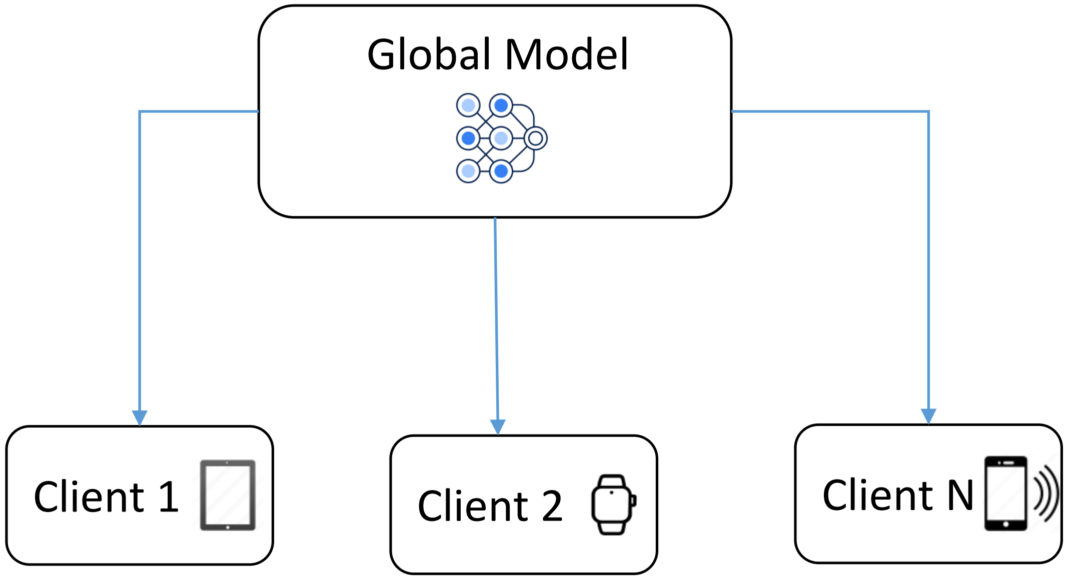
Central Server



- Privacy Issue
- high communication cost
- latency

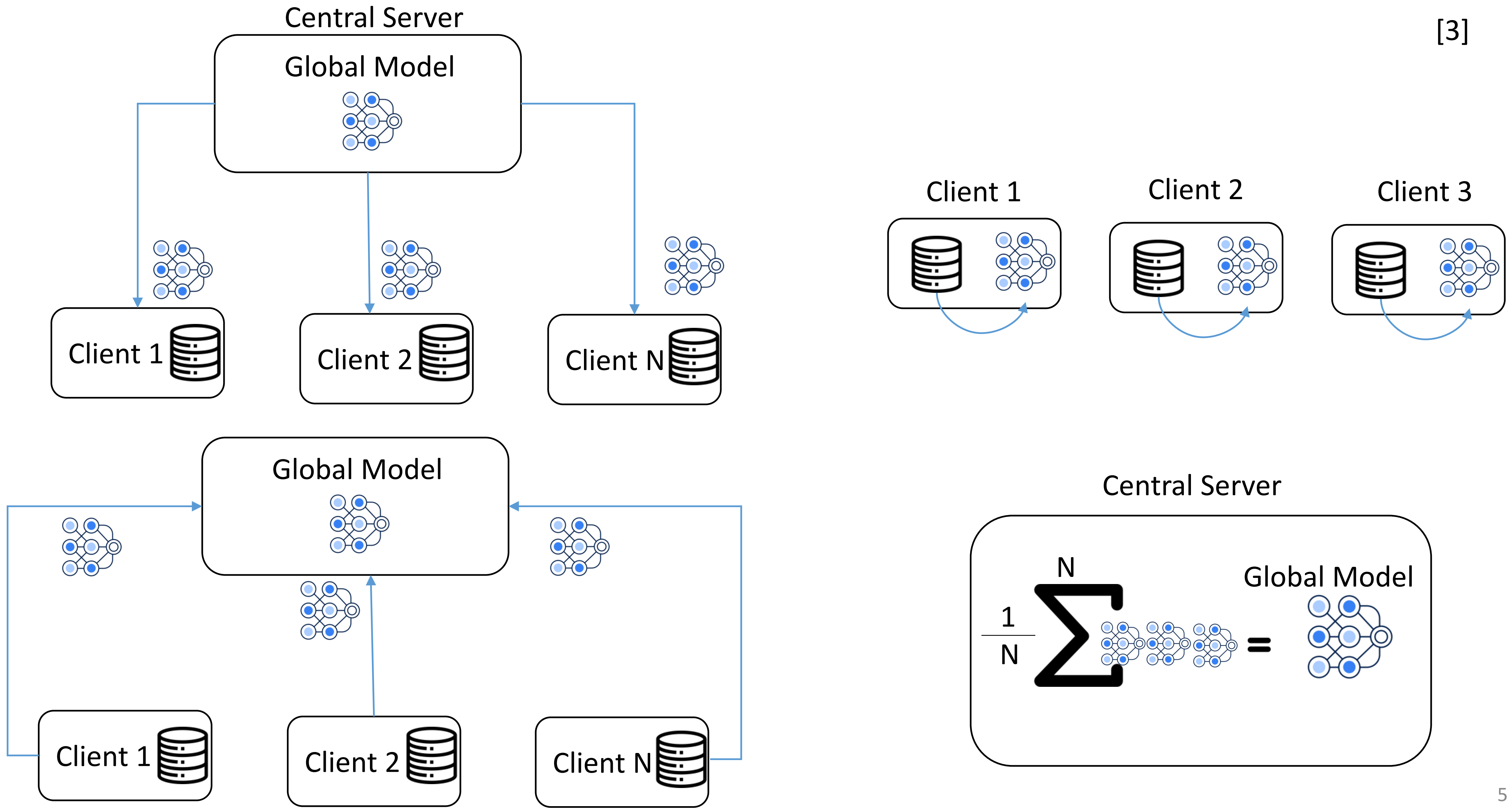
Federated Learning [2]

Central Server



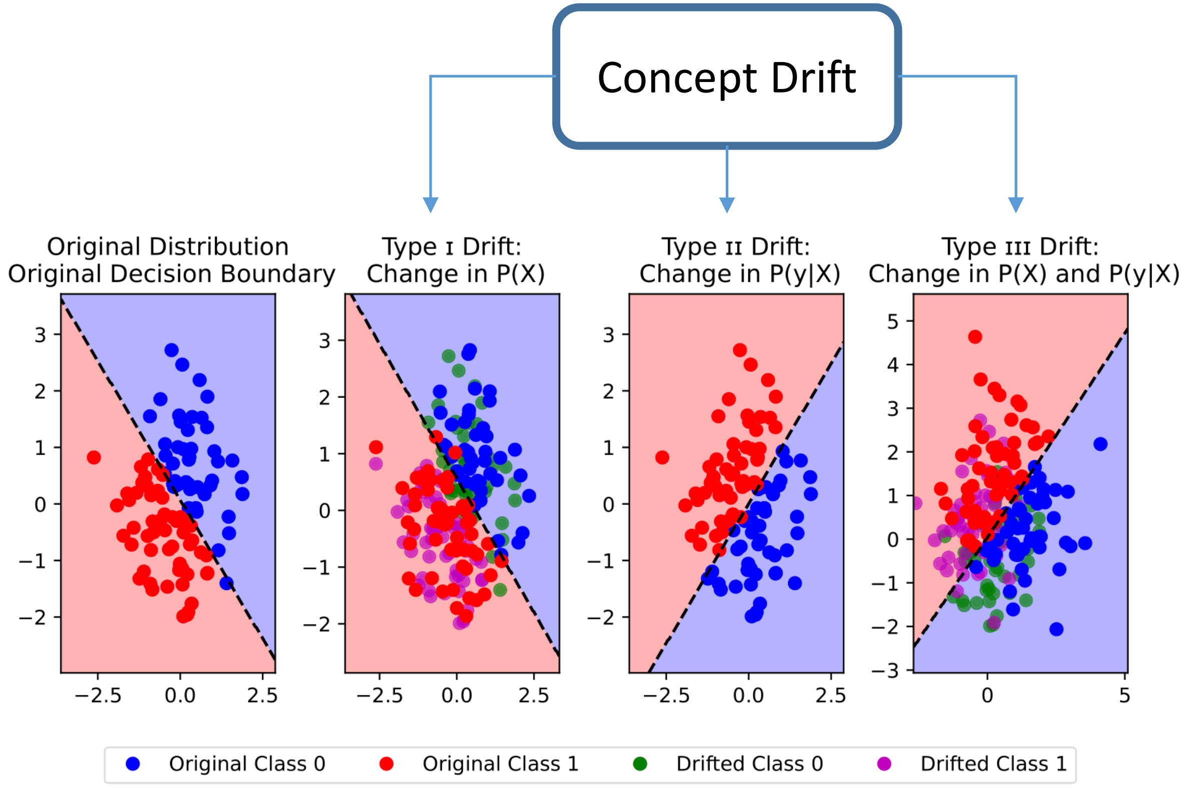
✓ Privacy Issue
✓ high communication cost
✓ latency

Introduction and Motivation (FeDAvg Algorithm)



Introduction and Motivation (Concept Drift)

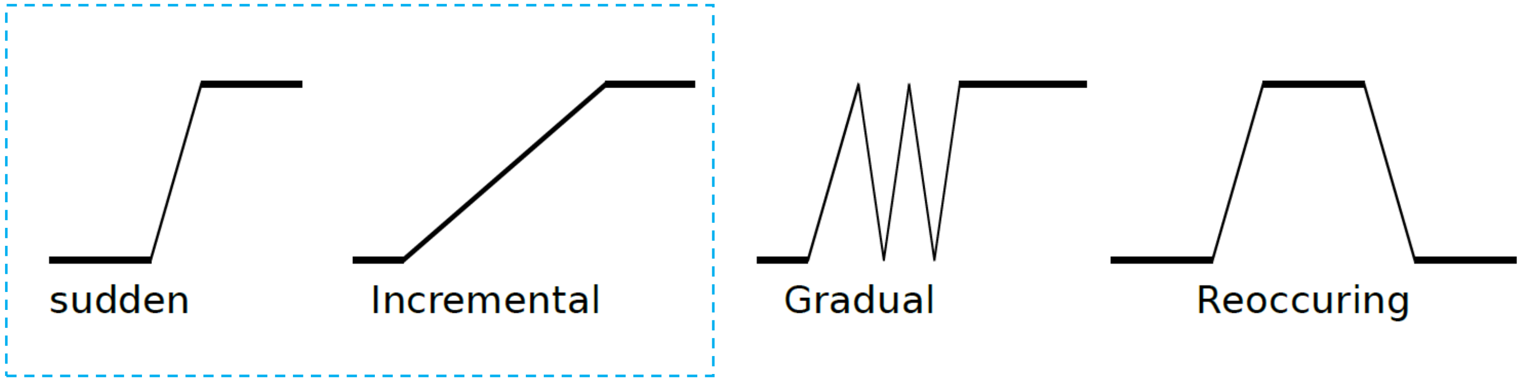
[4]



Problems of concept drift

- ML Model takes a long time to converge
- Degradation of performance (low accuracy and high loss)
- Over time renders it obsolete

- Specially in the case of FL where global model is not able to handle drifted client and non-drifted client at the same time without any adaptation measurement



Related Work (Drift Detection Algorithms)

Drift Detection Algorithms in CL

- Error-rate-based
- Data-Distribution-based
- multiple testing techniques

Error-rate-based

Drift Detection Method (DDM) [9]

- Detects drift by updating min error rate p_{min}, and min standard deviation s_{min}
- Warning threshold : $p_i + s_i \geq p_{min} + 2s_{min}$
- Detection threshold : $p_i + s_i \geq p_{min} + 3s_{min}$
- Can detect abrupt changes

Heoffding's inequality-based Drift Detection Method (HDDM) [10]

- Uses EWMA of two windows to detect concept drift
- Uses Hoeffding's inequality to dynamically determine thresholds
- Performs well both in terms of sudden and gradual drift.

Data-Distribution-based

- Assess the degree of dis-similarity between new and historical data
- Quantifying the distance between distributions [4]

multiple testing techniques

- Multiple testing techniques either in parallel or hierarchical configurations [4]

Related Work (Comparison of FL Concept Drift Adaptation Methods)

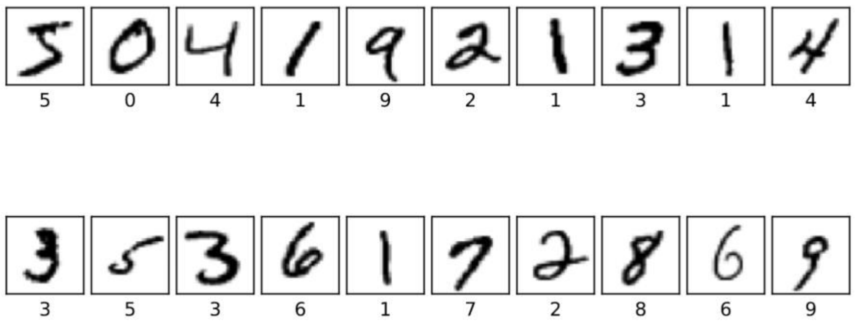
Methods	Sudden	Incremental	Gradual	Recurring	Robust to Outliers	Mixed Drift
FedAvg	No	No	No	No	No	No
Adaptive-FedAvg [5]	Yes	No	No	No	No	No
Saile et al. [17]	Yes	Partially	No	No	No	No
CDA-FedAvg [18]	Yes	Partially	Yes	Yes	No	No
FedNN [19]	Yes	Partially	No	No	No	Partially
FedDrift [20]	Yes	Partially	Partially	Partially	No	Partially
Salazar et al. [21]	Yes	Partially	Partially	Partially	No	Partially
FLAME [22]	Partially	Partially	Partially	Partially	Yes	No
Our Approach	Yes	Yes	Partially	Partially	Yes	Yes

Methods	Implementation	Detection Method Used	Adaptation Method Used
FedAvg	Server-Side	No	No
Adaptive-FedAvg	Server-Side	No	Adaptive Learning Rate
Saile et al.	Client-Side	No	Adaptive Learning Rate
CDA-FedAvg	Client-Side	Distribution Based Method	Memory Retention
FedNN	Client-Side	No	Weight Normalization (WN), Adaptive Group Normalization (AGN)
FedDrift	Server-Side	Global Loss	Multiple Global Model (Clusters)
Salazar et al.	Server-Side	Group Loss (Clusters)	Multiple Global Model (Clusters)
FLAME	Server-Side	Client Validation Loss	Memory Retention
Our Approach	Server-Side	Client Training Loss	Multiple Global Model Approach based on type of Concept Drift

Methodology (Dataset and Model architecture)

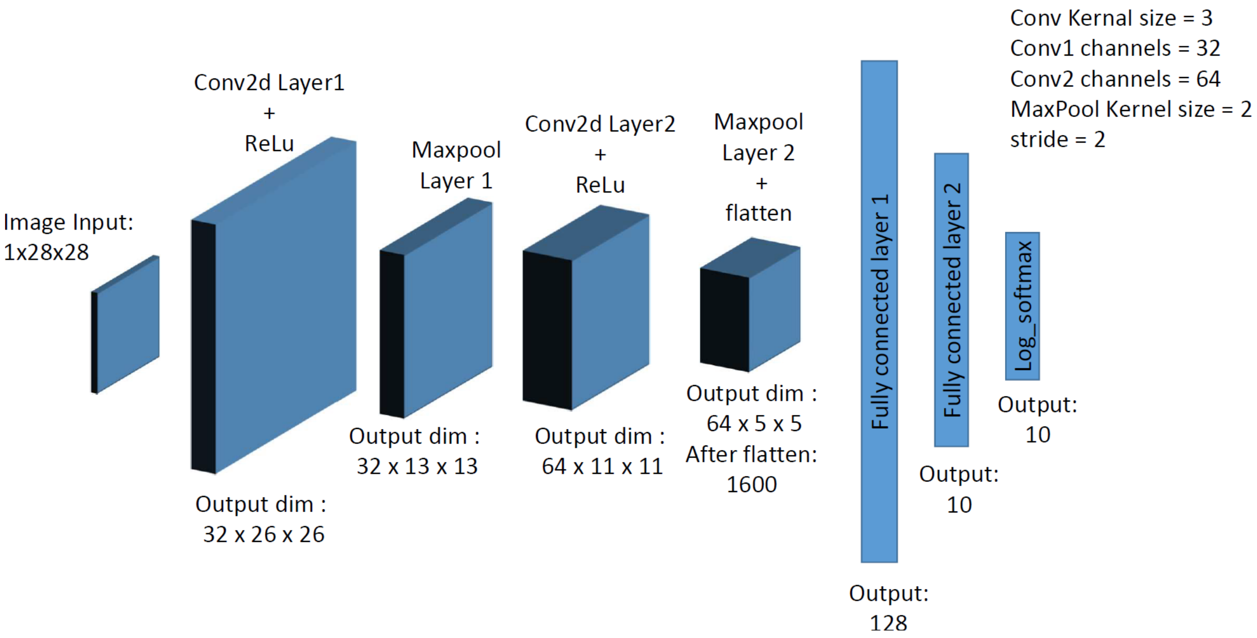
Dataset used for Experiments

[11]



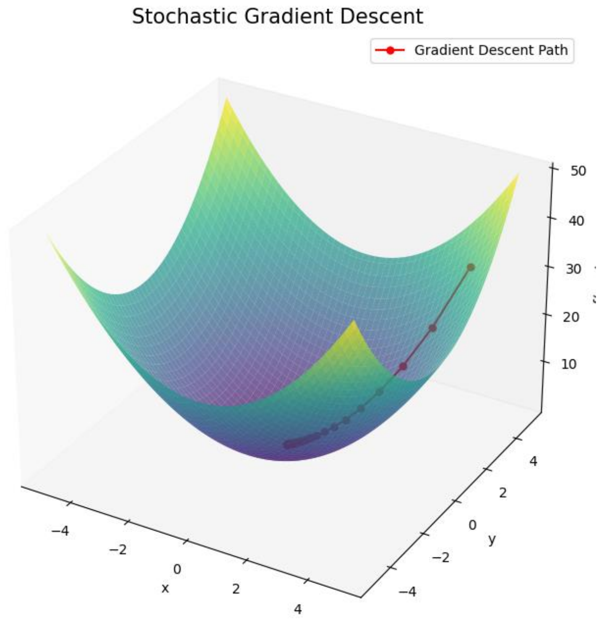
CNN Model used for Experiments

[12]



Adaptive Moment Estimation (Adam)

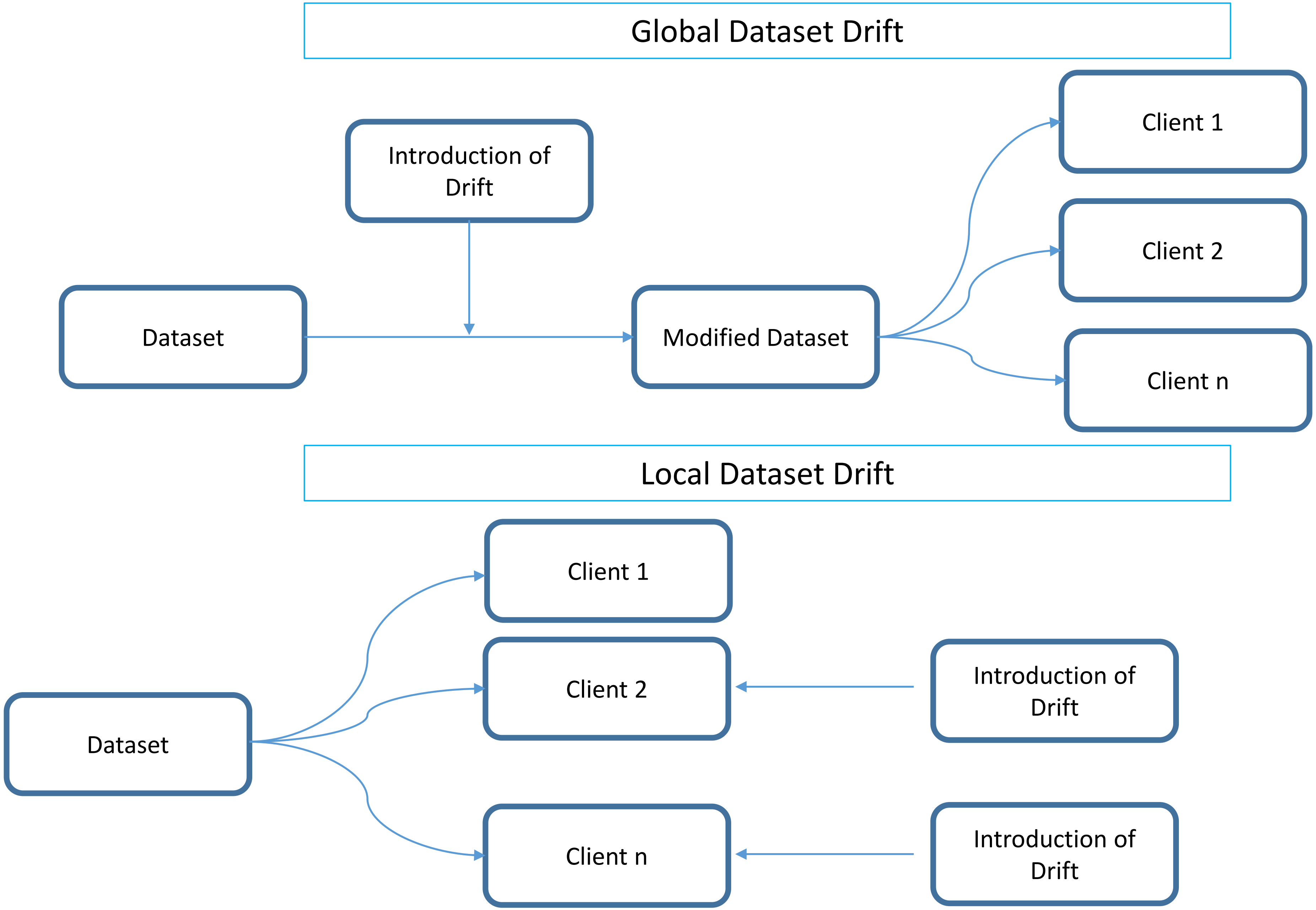
[13]



Adam uses two moment estimates to adaptively update learning rates for each parameter:

- the average direction of the gradients $m_t = \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$
- the magnitude of the gradients, $v_t = \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot {g_t}^2$

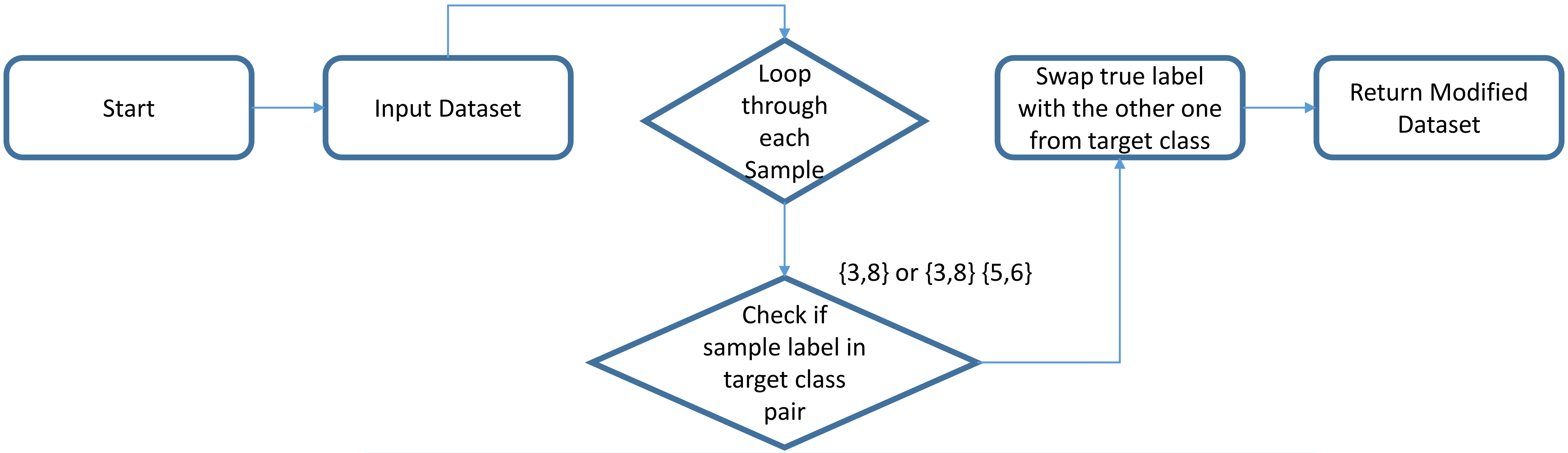
Methodology (Introduction of Drift)



Methodology (Introduction of Drift)

Introduction of Sudden Drift

[5]



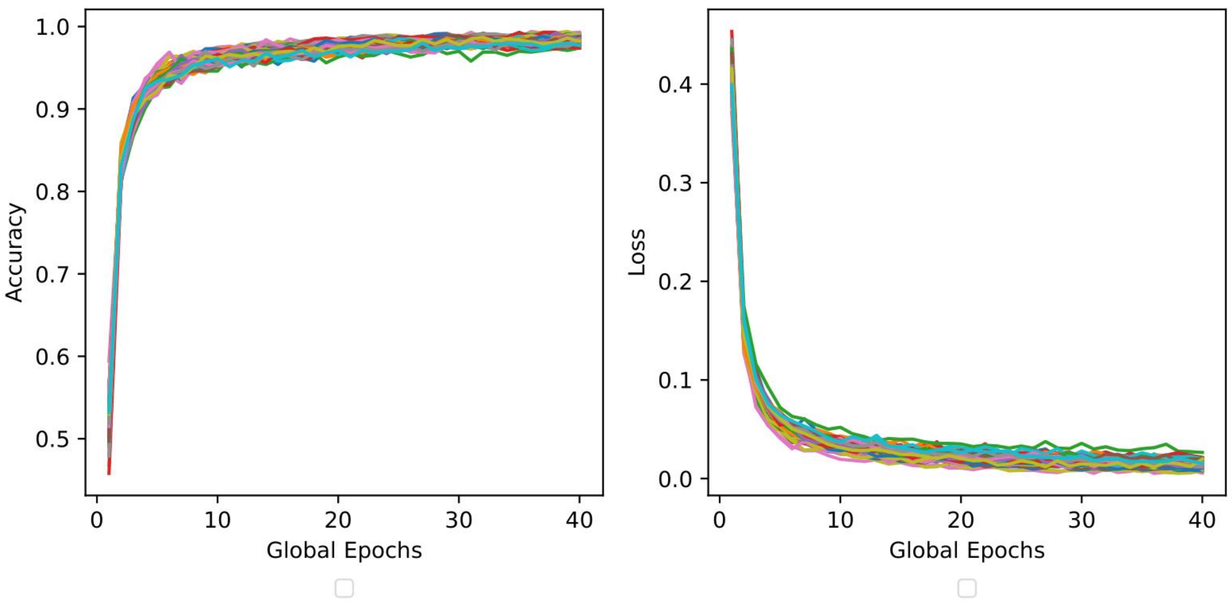
Introduction of Incremental Drift

- $transition\_progress = \frac{current\_epoch - start\_epoch}{end\_epoch - start\_epoch}$
- $\text{rotation\_angle} = \text{transition\_progress} \times \text{max\_rotation}$
- $f_r = \frac{current\_epoch - start\_epoch}{end\_epoch - start\_epoch}$, $n_r = [fr \times N]$

Epoch	Rotation Angle (degrees)	Transition Progress
11	1.5	0.05
15	7.5	0.25
20	15	0.5
25	22.5	0.75
30	30	1

Design and Implementation (Analyzing the Effect of Concept Drift)

[14]



$$Client\ Model\ Accurcay = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions}$$

Client Loss Value:

$$l\,(\hat{y}, y) = \mathrm{L} = [l_1, \ldots\ldots, l_N],$$

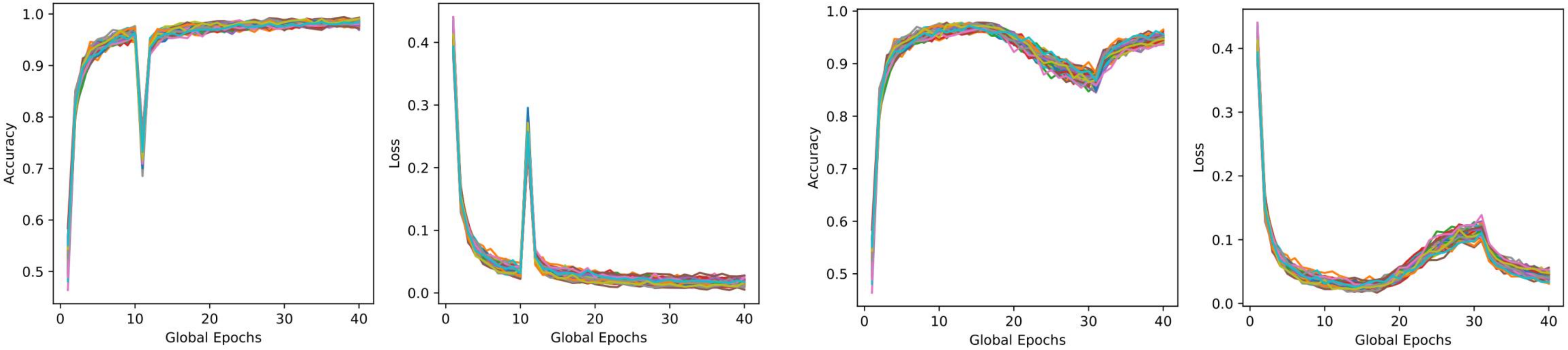
Where , $l_n = -w_{y_n} \log\left(\frac{\exp(\hat{y}_{n,y_n})}{\sum_{c=1}^{M} \exp(\hat{y}_{n,c})}\right) \cdot 1\,\{y_n \neq igore\_index\}$

$$l\,(\hat{y}, y) = \begin{cases} \dfrac{\sum_{n=1}^{N} l_n}{\sum_{n=1}^{N} w_{y_n} \cdot 1\,\{y_n \neq ignore\_index\}}, & if\ reduction = \text{"}mean\text{"} \\ \sum_{n=1}^{N} l_n\ , & if\ reduction = \text{"}sum\text{"} \end{cases}$$

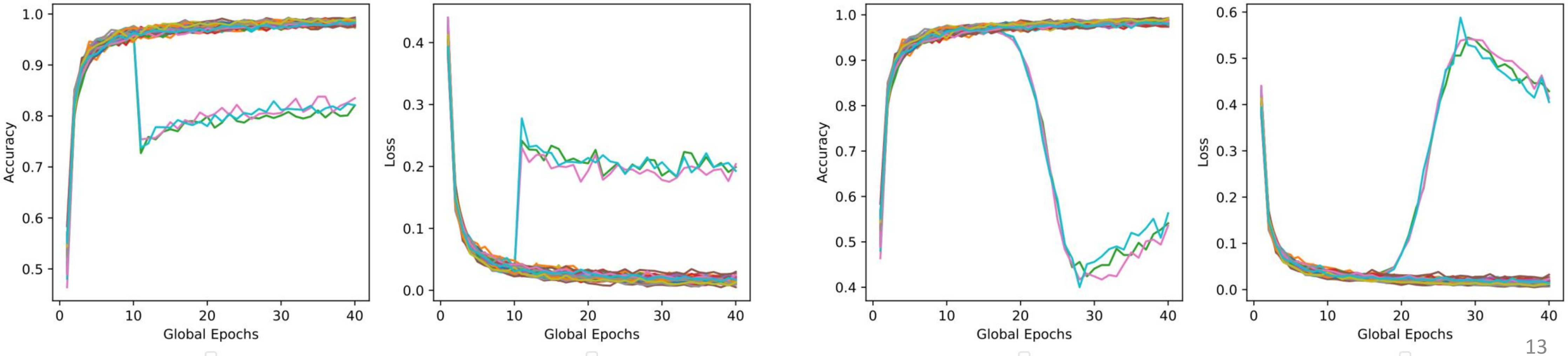
Fig: Federated Learning (FedAvg) client accuracy and loss monitoring for 60 clients and 40 global epochs without any drift

Design and Implementation (Analyzing the Effect of Concept Drift)

Global Dataset Drift

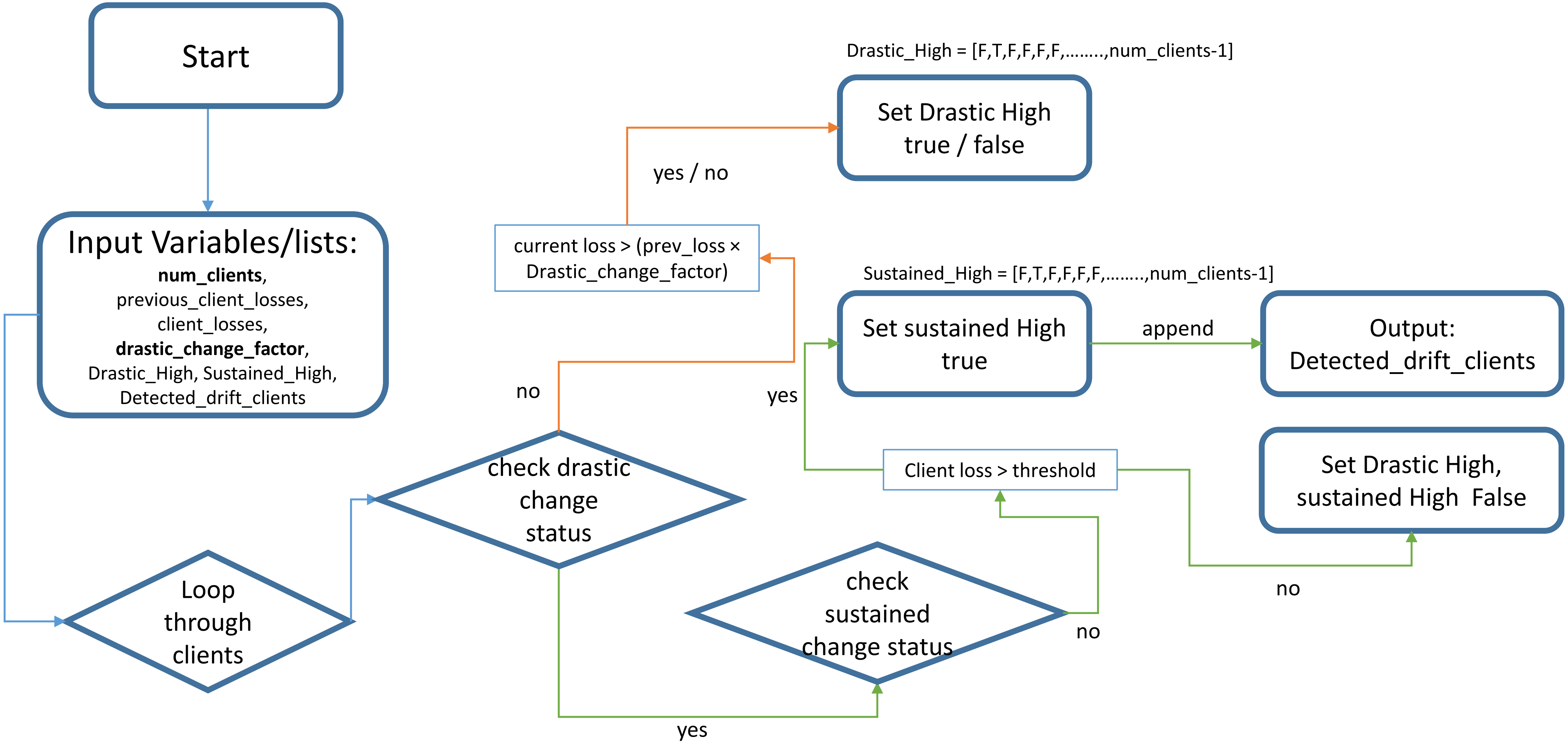


Local Dataset Drift where drift is only introduce to clients 2, 6, 9



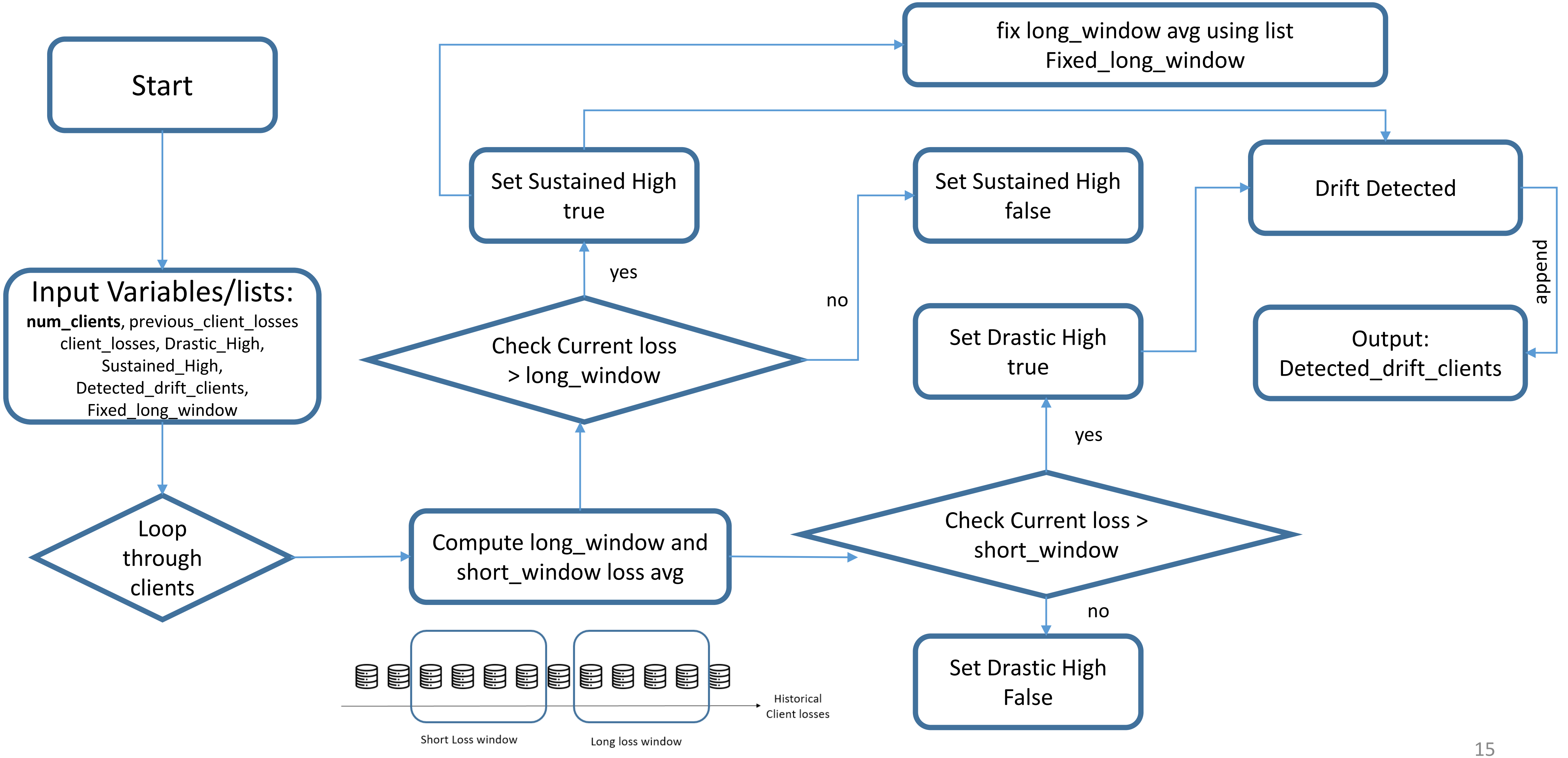
Design and Implementation (Strategies for Concept Drift Detection)

4.1 Server-side Sudden Drift Detection Algorithm



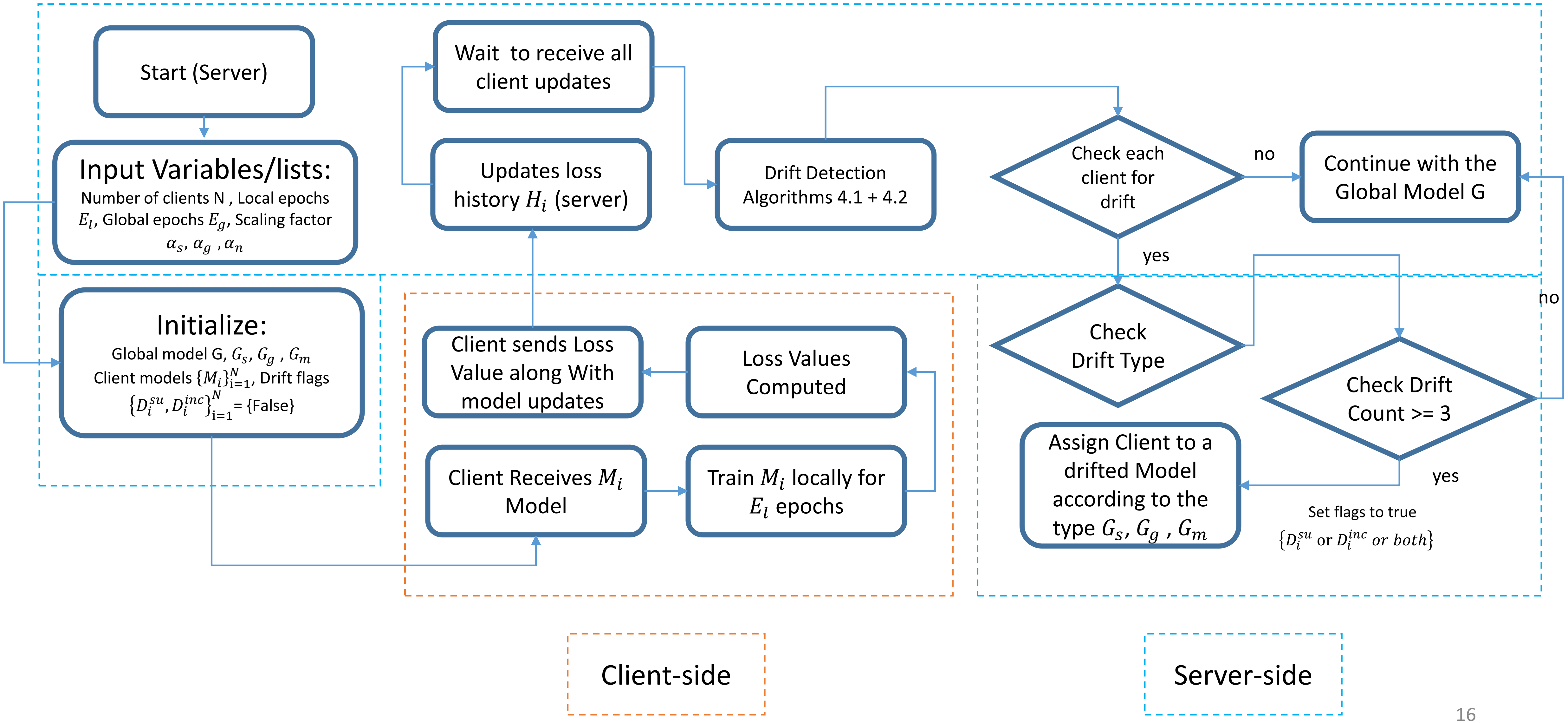
Design and Implementation (Strategies for Concept Drift Detection)

4.2 Server-side Incremental Drift Detection Algorithm



Design and Implementation (Strategies for Concept Drift Adaptation)

4.3 Server-side Concept Drift Detection and Adaptation in FL



Design and Implementation (Strategies for Concept Drift Detection)

Optimizer settings based on drift detection

Client Drift Type	LR	Beta 1	Beta 2
Incremental Drift Only	0.001	0.6	0.7
Both Sudden and Incremental Drift	0.001	0.6	0.7
Sudden Drift Only	0.001	Default	Default
No Drift	0.001	Default	Default

Performance Metrics

- Accuracy = $\frac{Number\ of\ correct\ detection}{Total\ number\ of\ detection\ counts}$ = $\frac{TP+TN}{TP+FP+TN+FN}$
- Precision = $\frac{Number\ of\ correct\ positive\ detection}{Total\ number\ of\ positive\ detection}$ = $\frac{TP}{TP+FP}$
- Recall = $\frac{Number\ of\ correct\ positive\ detection}{Total\ number\ of\ actual\ drifts\ and\ no-drifts}$ = $\frac{TP}{TP+FN}$
- F1 Score = $2 \times \frac{Precision \times Recall}{Precision + Recall}$

Global Optimum Model

$for\ a\ single\ drift:$

$$G_{opt} = \alpha G_d + (1-\alpha)G$$

$for\ sudden,\ icremental\ and\ both\ drift:$

$$G_{opt} = \alpha_s G_s + \alpha_g G_g + \alpha_m G_m + (1-\alpha_s-\alpha_g-\alpha_m)G,$$
$$where\ \alpha_s + \alpha_g + \alpha_m \leq 1$$

Hardware and Software used

Environment :	Kaggle
Implementation of algoritms and FL :	Python, Pytorch [16], Scikit-Multiflow [15]
Number of clients, Local Epochs, Global Epochs, batch size, Training Samples, Client Samples :	60, 1, 40, 32, 60000, 1000

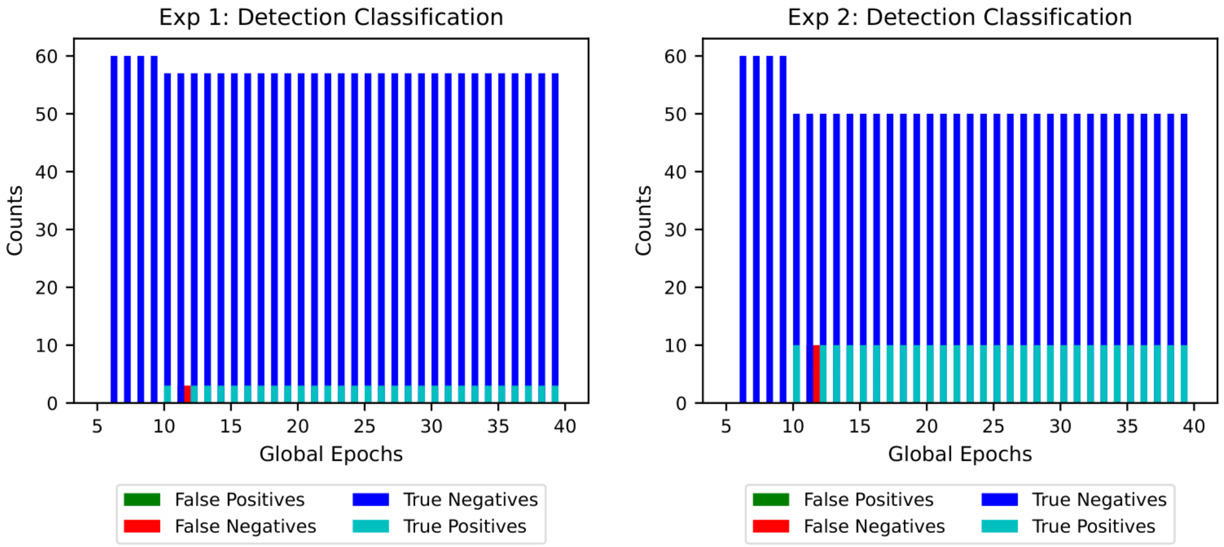
Evaluation (Sudden Drift Detection)

Parameters	Exp 1	Exp 2
Class labels swapped	3 ,8	{3, 8}, {5, 6}
Drifted Clients	3	10

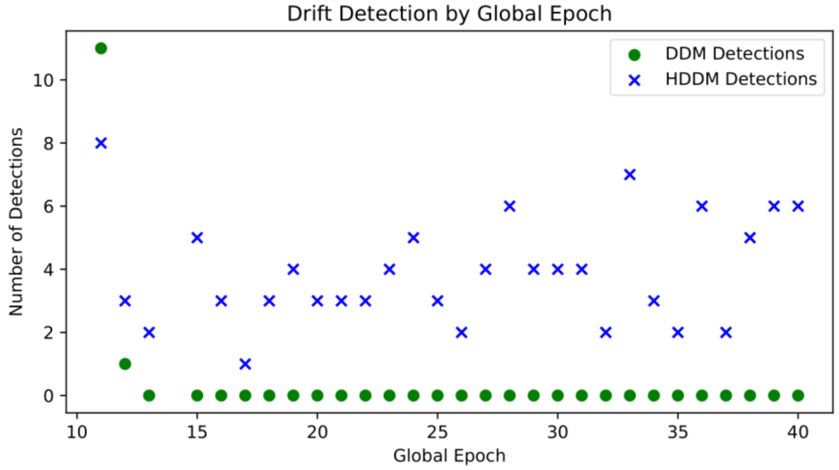
Detection Classification	Algorithm 4.1		DDM+HDDM
	Exp 1	Exp 2	Exp 2
True Positives	87	290	116 (11+113)
True Negatives	2010	1800	1800
False Positives	0	0	0
False Negatives	3	10	184 (289+187)

Detection Metric	Algorithm 4.1		DDM+HDDM
	Exp 1	Exp 2	Exp 2
Accuracy	0.998	0.995	0.9123
Precision	1.0	1.0	1.0
Recall	0.966	0.966	0.386
F1 Score	0.983	0.983	0.557

Algorithm 4.1 Detection Breakdown



DDM + HDDM Detection



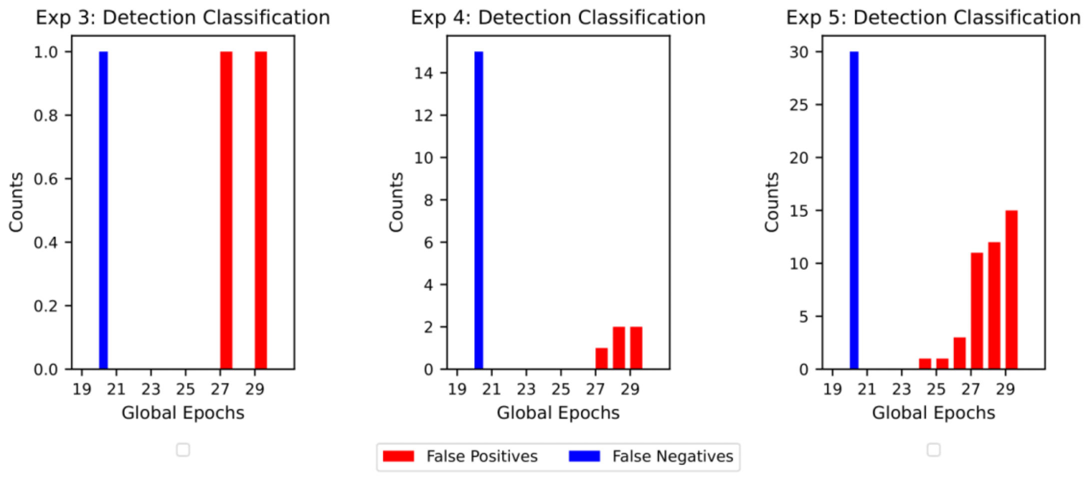
Evaluation (Incremental Drift Detection)

Parameters	Exp 3	Exp 4	Exp 5
Drifted Clients	6	16	30
Percentage of total clients	0.1	0.26	0.5

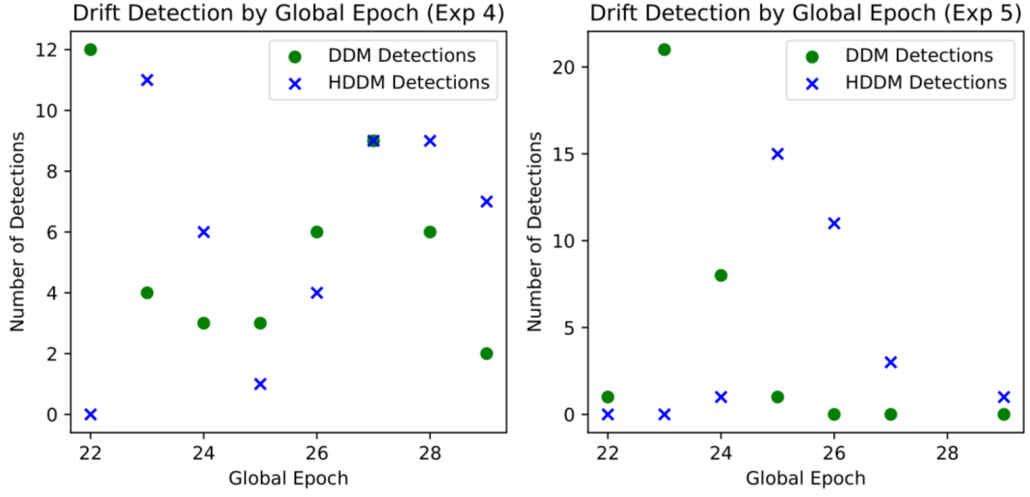
Detection Classification	Algorithm 4.2			DDM+HDDM	
	Exp 3	Exp 4	Exp 5	Exp 4	Exp 5
True Positives	59	145	270	73	62
True Negatives	1378	1275	1097	1280	1140
False Positives	2	5	43	0	0
False Negatives	1	15	30	87	28

Detection Metrics	Algorithm 4.2			DDM+HDDM	
	Exp 3	Exp 4	Exp 5	Exp 4	Exp 5
Accuracy	0.9979	0.9861	0.9493	0.9395	0.8347
Precision	0.9672	0.9666	0.8626	1.0	1.0
Recall	0.98333	0.90625	0.9	0.45625	0.2066
F1 Score	0.9752	0.93548	0.8809	0.6266	0.3425

Algorithm 4.2 False Detections Breakdown



DDM + HDDM Detection



Evaluation (Mixed Drift Detection)

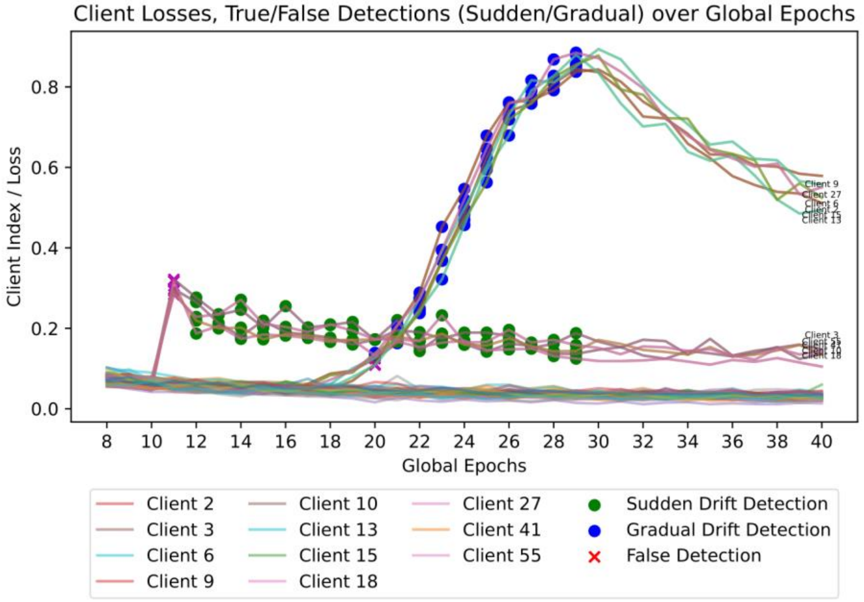
Parameters	Exp 6	Exp 7
incremental_drift_clients	{2, 6, 9, 13, 15, 27}	{2, 6, 9, 13, 15, 27}
class_swap_drift_clients	{3, 10, 18, 41, 55}	{2, 6, 9, 41, 55}

Detection Classification	Algorithm 4.1 + 4.2		DDM+HDDM	
	Exp 6	Exp 7	Exp 6	Exp 7
True Positives	147	119	55 (24+43)	52 (17+47)
True Negatives	1285	1314	1285	1315
False Positives	0	1	0	0
False Negatives	8	6	100 (131+112)	73 (108+78)

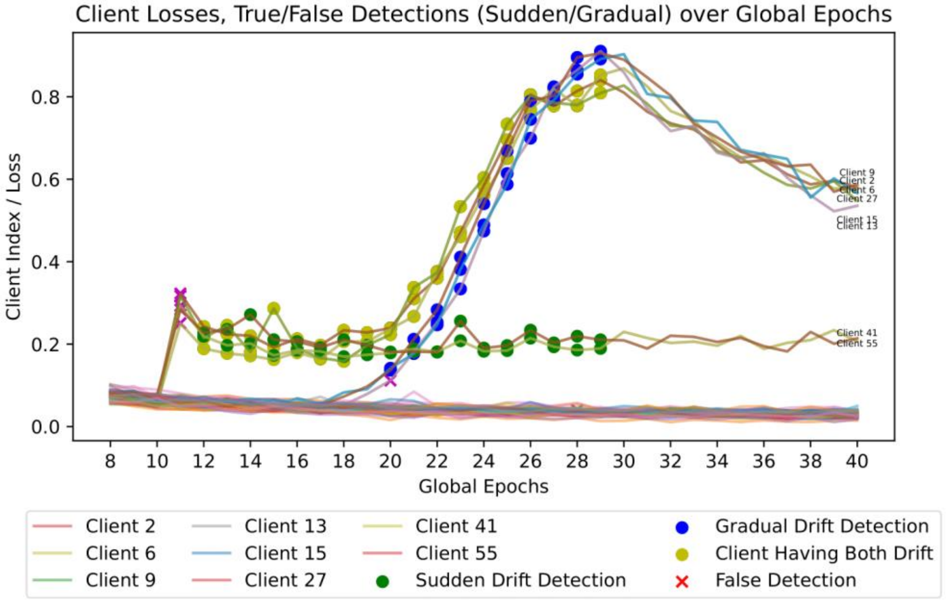
Detection Metrics	Algorithm 4.1 + 4.2		DDM+HDDM	
	Exp 6	Exp 7	Exp 6	Exp 7
Accuracy	0.9944	0.9951	0.9305	62
Precision	1.0	0.9916	1.0	1.0
Recall	0.9438	0.952	0.3548	0.416
F1 Score	0.9735	0.9714	0.5238	0.5875

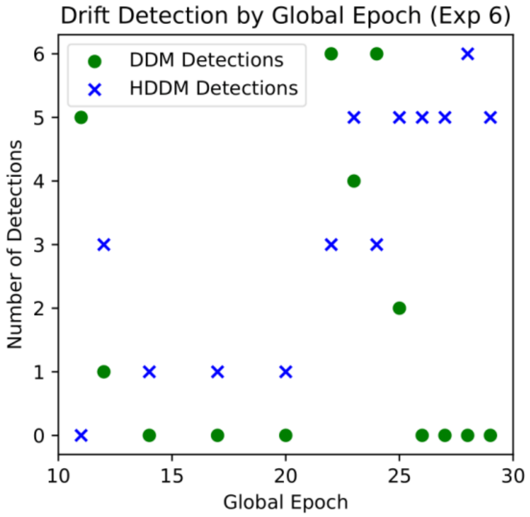
Evaluation (Mixed Drift Detection)

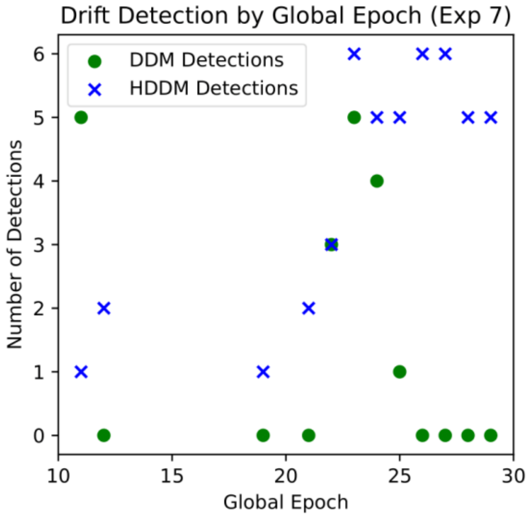
Exp 6: Mixed Drift Detection using Algorithm 4.1+4.2, {2, 6, 9, 13, 15, 27}, {3, 10, 18, 41, 55}



Exp 7: Mixed Drift Detection using Algorithm 4.1+4.2, {2, 6, 9, 13, 15, 27}, {2, 6, 9, 41, 55}

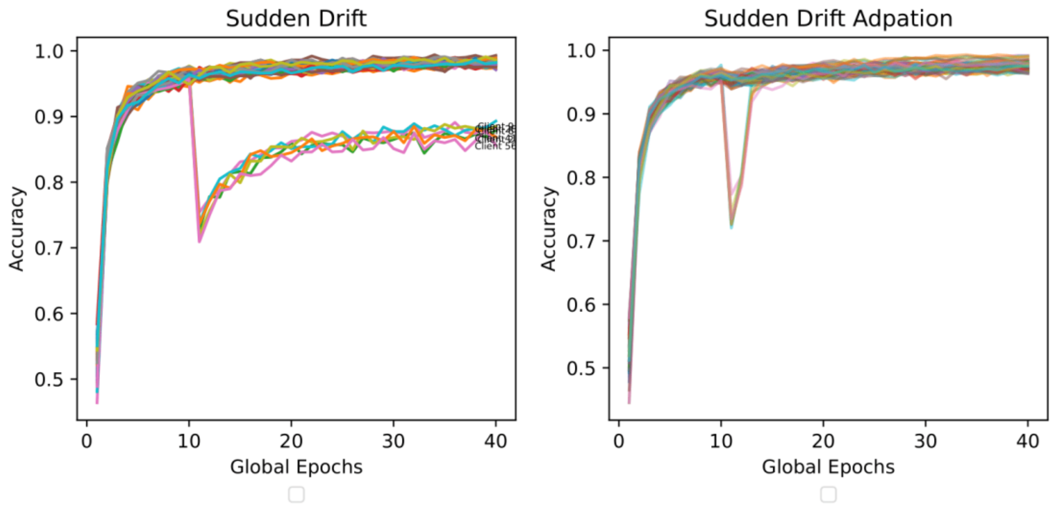




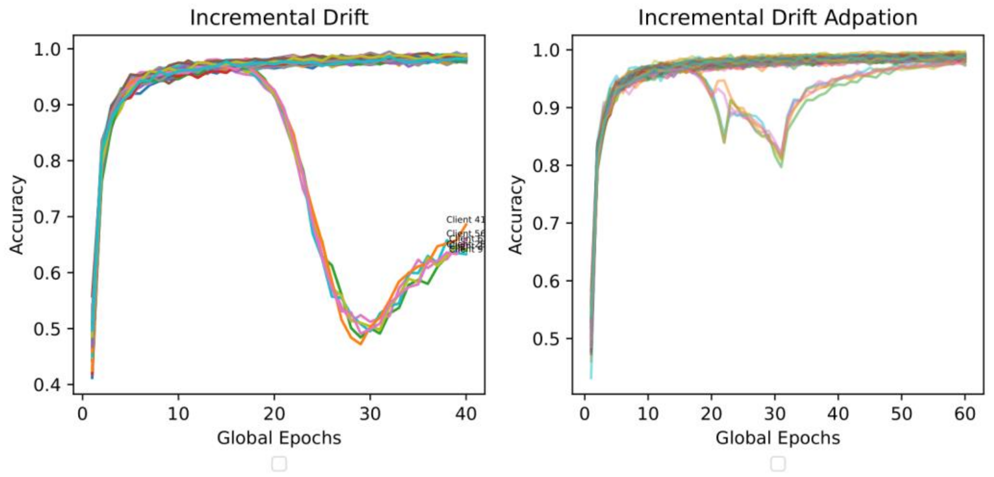


Evaluation (Adaptation)

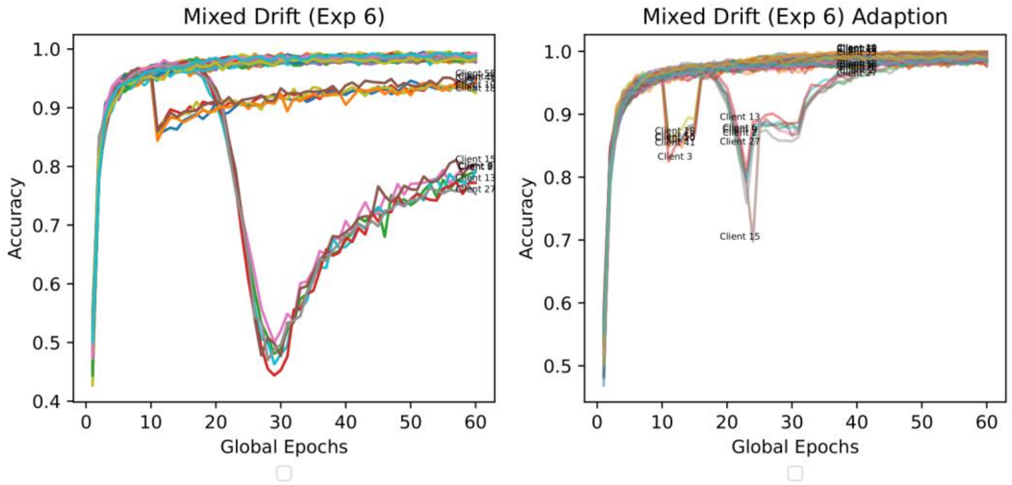
Exp 8: Sudden Drift Adaptation using Algorithm 4.3, {2, 6, 9, 28, 41, 56}



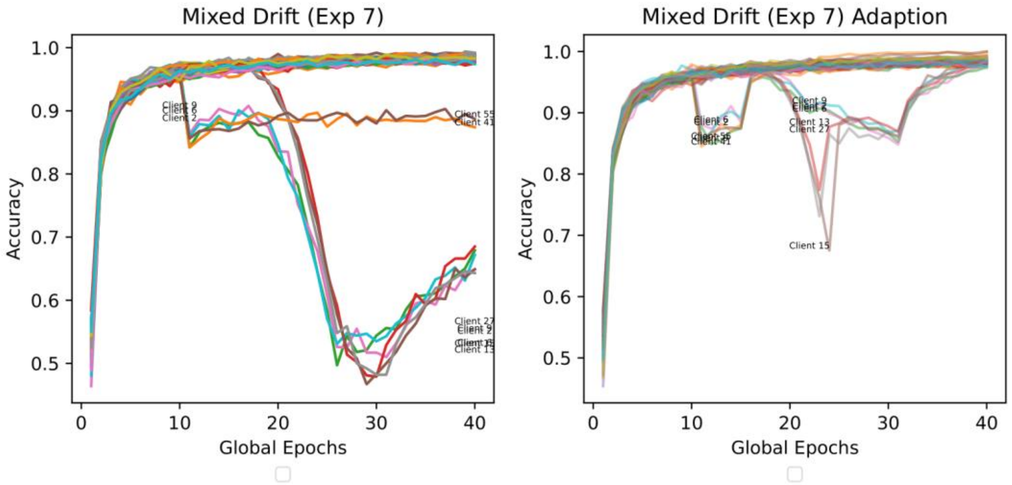
Exp 8 : Incremental Drift Adaptation using Algorithm 4.3, {2, 6, 9, 28, 41, 56}



Exp 6: Mixed Drift Adaptation using Algorithm 4.3, {2, 6, 9, 13, 15, 27}, {3, 10, 18, 41, 55}



Exp 7: Mixed Drift adaption using Adaptation 4.3, {2, 6, 9, 13, 15, 27}, {2, 6, 9, 41, 55}



Evaluation (Global Optimum Model)

$$G_{opt} = \alpha G_d + (1-\alpha)G$$

$$G_{opt} = \alpha_s G_s + \alpha_g G_g + \alpha_m G_m + (1 - \alpha_s - \alpha_g - \alpha_m)G, \qquad where\ \alpha_s + \alpha_g + \alpha_m \leq 1$$

Comparison of FedAvg and Optimized Model Accuracy for Incremental Drift Experiments

Exp No.	Fraction of test data with drift	FedAvg Model Accuracy	Optimized Model Accuracy	Alpha	No. of Clients with drift
Exp 9	0.2	0.9594	0.9602	0.35	3
Exp 10	0.4	0.935	0.9387	0.35	3
Exp 11	0.6	0.917	0.9279	0.35	3
Exp 12	0.8	0.8858	0.9163	0.35	3
Exp 13	1.0	0.8544	0.9066	0.35	3
Exp 14	1.0	0.8451	0.9037	0.35	6
Exp 15	1.0	0.851	0.9056	0.35	16
Exp 16	1.0	0.7644	0.9109	0.35	30
Exp 17	1.0	0.6279	0.9143	0.35	42

Evaluation (Global Optimum Model)

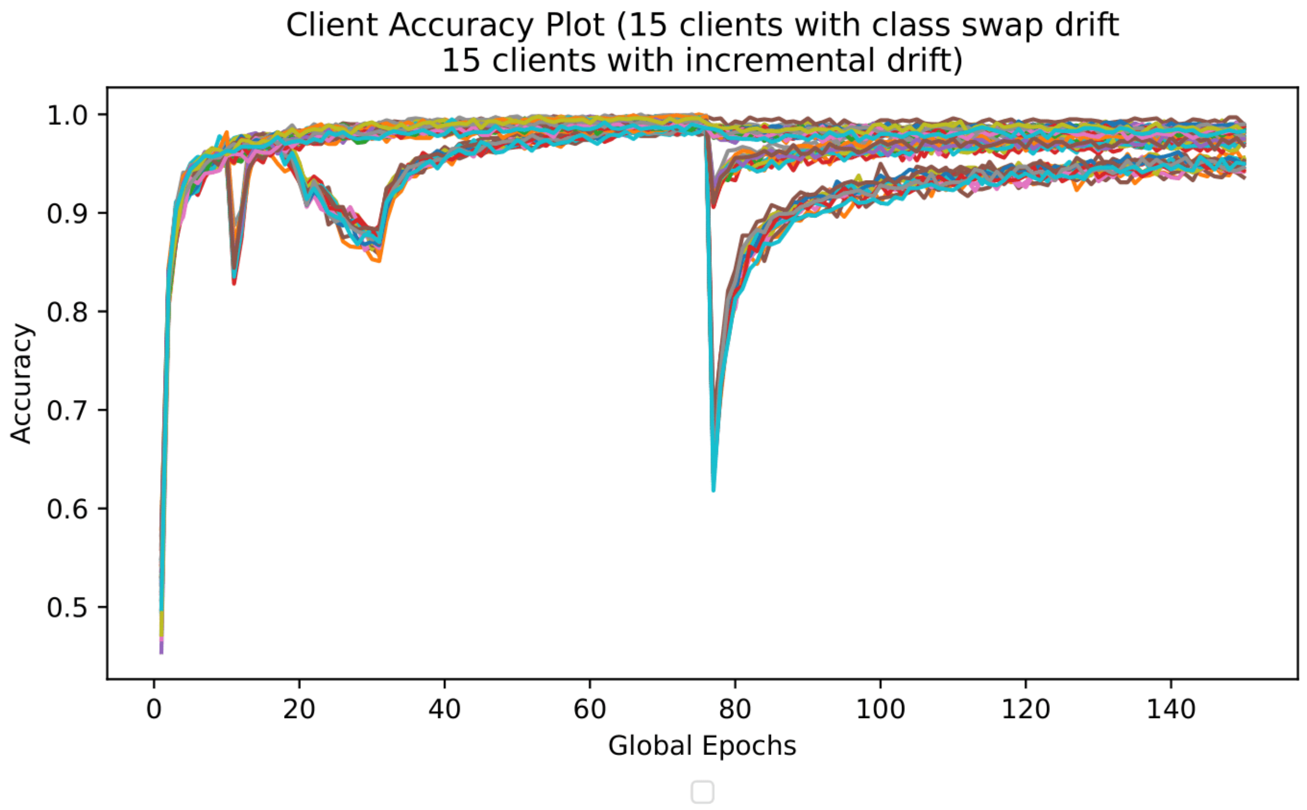
Comparison of FedAvg and Optimized Model Accuracy for Sudden Drift Experiments

Exp No.	Fraction of test data with drift	FedAvg Model Accuracy	Optimized Model Accuracy	Alpha	No. of Clients with drift
Exp 18	0.2	0.9482	0.942	0.35	3
Exp 19	0.4	0.912	0.9064	0.35	3
Exp 20	0.6	0.8703	0.8667	0.35	3
Exp 21	1.0	0.7921	0.9785	1.00	3

Comparison of FedAvg and Optimized Model Accuracy for Mixed Drift Experiments

Exp No.	Fraction of test data with drift	FedAvg Model Accuracy	Optimized Model Accuracy	$(\alpha_s, \alpha_{g,} \alpha_m)$	No. of Clients with drift (sudden, incremental, both)
Exp 22	1	0.7924	0.8924	(0.16, 0.16, 0.16)	(5, 5, 3)
Exp 23	1	0.7913	0.8936	(0.13, 0.13, 0.13)	(10, 10, 6)
Exp 24	1	0.7872	0.8943	(0.13, 0.13, 0.13)	(16, 16, 9)

Evaluation (Replacing Multiple Models with Optimized Model)



Conclusion

- We particularly focused on local dataset drift in this thesis.
- We proposed an active approach to adapt to these drifts by first classifying the type of drift each client faces. Which has shown:
  - excellent precision and accuracy, outperforming DDM and HDDM.
  - Reduced client losses and faster convergence with non-drifted clients.
  - Computationally efficient, placing no additional burden on client devices.
  - The optimized global model consistently outperformed the FedAvg.

Future Work

- Usage of the optimized global model
- Communication Cost
- Using more Dataset

Thank you

References

[1] Wang, Xiaofei, et al. : In-edge ai: Intelligentizing mobile edge computing, caching and communication by federated learning. IEEE Network 33.5, 2019,pages 156-165.

[2] Dishita Naik and Nitin Naik: The Changing Landscape of Machine Learning: A Comparative Analysis of Centralized Machine Learning, Distributed Machine Learning and Federated Machine Learning. In Advances in Computational Intelligence Systems. Springer NatureSwitzerland, 2024, pages 18–28.

[3] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hamp-son, and Blaise Aguera y Arcas: Communication-Efficient Learning of Deep Networks from Decentralized Data. In 20th InternationalConference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research, volume 54, PMLR, 2017, pages1273–1282.

[4] Jie Lu, Anjin Liu, Fan Dong, Feng Gu, João Gama, andGuangquan Zhang: Learning under Concept Drift: A Review.In IEEE Transactions on Knowledge and Data Engineering, volume31 (12), 2019, pages 2346–2363. DOI: 10 . 1109 / TKDE . 2018 .2876857.

[5] Giuseppe Canonaco, Alex Bergamasco, Alessio Mongelluzzo,and Manuel Roveri: Adaptive Federated Learning in Presence of Concept Drift. In 2021 International Joint Conference on Neural Net-works (IJCNN), 2021, pages 1–7. DOI: 10.1109/IJCNN52387.2021.9533710.

[6] Fernando E. Casado, Dylan Lema, Marcos F. Criado, RobertoIglesias, Carlos V. Regueiro, and Senén Barro: Concept drift detec-tion and adaptation for federated and continual learning. In Multime-dia Tools and Applications, volume 81 (3), 2022, pages 3397–3419.ISSN: 1573-7721. DOI: 10.1007/s11042-021-11219-x.

[7] Teresa Salazar, João Gama, Helder Araújo, and Pedro HenriquesAbreu: Unveiling Group-Specific Distributed Concept Drift: A Fair-ness Imperative in Federated Learning. 2024. arXiv: 2402.07586[cs.LG]. URL: https://arxiv.org/abs/2402.07586.

[8] Ellango Jothimurugesan, Kevin Hsieh, Jianyu Wang, Gauri Joshi,and Phillip B Gibbons: Federated learning under distributed con-cept drift. In International Conference on Artificial Intelligence andStatistics. PMLR, 2023, pages 5834–5853.

[9] João Gama, Pedro Medas, Gladys Castillo, and Pedro Rodrigues: Learning with Drift Detection. In Advances in Artificial Intelligence–SBIA 2004. Springer Berlin Heidelberg, 2004, pages 286–295.ISBN: 978-3-540-28645-5.

[10] Isvani Frías-Blanco, José del Campo-Ávila, Gonzalo Ramos-Jiménez, Rafael Morales-Bueno, Agustín Ortiz-Díaz, and Yailé Caballero-Mota: Online and Non-Parametric Drift Detection Meth-ods Based on Hoeffding's Bounds. In IEEE Transactions on Knowledge and Data Engineering, volume 27 (3), 2015, pages 810–823. DOI:10.1109/TKDE.2014.2345382.

References

[11] Li Deng: The MNIST Database of Handwritten Digit Images for Machine Learning Research [Best of the Web]. In IEEE Signal ProcessingMagazine, volume 29 (6), 2012, pages 141–142. DOI: 10.1109/MSP.2012.2211477.

[12] PyTorch Tutorial: Training a Classifier. https : / / pytorch .org/tutorials/beginner/blitz/cifar10_tutorial.html. Accessed: 2024-08-12. 2024.

[13] Ruslan Abdulkadirov, Pavel Lyakhov, and Nikolay Nagornov: Survey of Optimization Algorithms in Modern Neural Networks. InMathematics, volume 11 (11), 2023. ISSN: 2227-7390. DOI: 10 .3390 / math11112466. URL: https : / / www . mdpi . com /2227-7390/11/11/2466.

[14] PyTorch: CrossEntropyLoss. https://pytorch.org/docs/stable / generated / torch . nn . CrossEntropyLoss .html. Accessed: 2024-08-07.

[15] Jacob Montiel, Jesse Read, Albert Bifet, and Talel Abdessalem: Scikit-Multiflow: A Multi-output Streaming Framework. In Journalof Machine Learning Research, volume 19 (72), 2018, pages 1–5.URL: http://jmlr.org/papers/v19/18-251.html.

[16] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al.: Pytorch: An imperative style, high-performance deep learning library. In Advances in neural informationprocessing systems, volume 32, 2019

[17] Finn Saile, Julius Thomas, Dominik Kaaser, and Stefan Schulte:Client-Side Adaptation to Concept Drift in Federated Learning.In 2024 Federated Learning and Trustworthy AI (FLTA) Confer-ence. FLTA Press, 2024, pages 123–134. URL: https://flta-conference.org/paper123.

[18] Fernando E. Casado, Dylan Lema, Marcos F. Criado, RobertoIglesias, Carlos V. Regueiro, and Senén Barro: Concept drift detec-tion and adaptation for federated and continual learning. In Multime-dia Tools and Applications, volume 81 (3), 2022, pages 3397–3419.DOI: 10.1007/s11042-021-11219-x

[19] Myeongkyun Kang, Soopil Kim, Kyong Hwan Jin, Ehsan Adeli,Kilian M. Pohl, and Sang Hyun Park: FedNN: Federated learning on concept drift data using weight and adaptive group normalizations.In Pattern Recognition, volume 149, 2024, pages 110230. DOI:https://doi.org/10.1016/j.patcog.2023.110230.

References

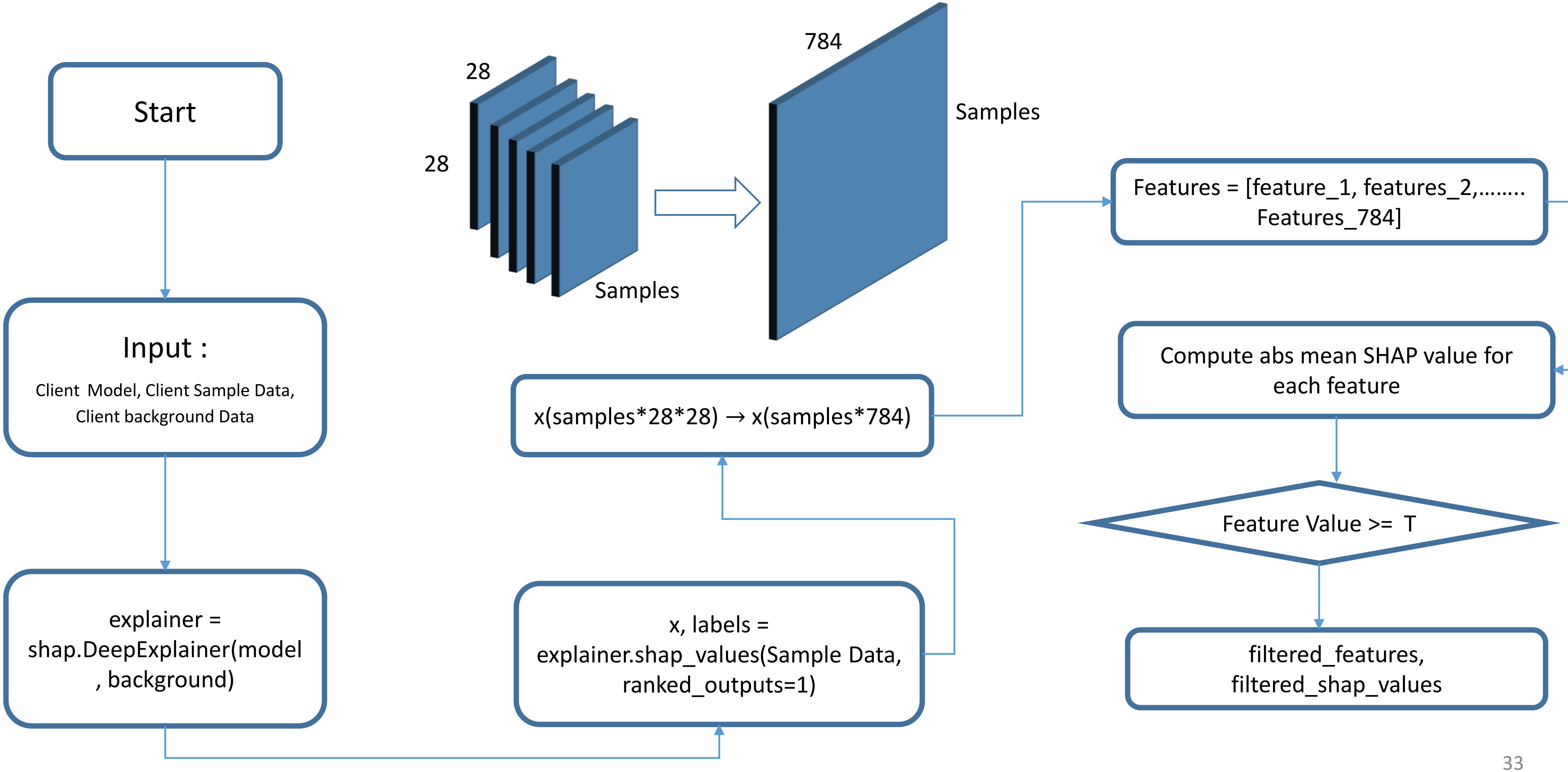
[20] Ellango Jothimurugesan, Kevin Hsieh, Jianyu Wang, Gauri Joshi,and Phillip B Gibbons: Federated learning under distributed con-cept drift. In International Conference on Artificial Intelligence andStatistics. PMLR, 2023, pages 5834–5853.

[21] Teresa Salazar, João Gama, Helder Araújo, and Pedro HenriquesAbreu: Unveiling Group-Specific Distributed Concept Drift: A Fairness Imperative in Federated Learning. 2024. arXiv: 2402.07586[cs.LG].

[22] Ioannis Mavromatis, Stefano De Feo, and Aftab Khan: FLAME:Adaptive and Reactive Concept Drift Mitigation for Federated Learning Deployments. 2024. arXiv: 2410.01386 [cs.LG].

Appendix

Concept Drift Detection Using SHAP Values (SHAP Computation)



Concept Drift Detection Using SHAP Values