

Adaptive Federated Representation Learning for Nonstationary Edge Intelligence

Allwine¹

Department of Computer Science, Universitas Negeri Lampung, Lampung, Indonesia
Corresponding author: Allwine, allwine@fmipa.unila.ac.id

Article Info

Article history:

Received: May 25, 2026

Revised: July 1, 2026

Accepted: July 11, 2026

Keywords:

Federated Learning

Concept Drift

Edge Intelligence

Representation Learning

Non-IID Data

Continual Adaptation

ABSTRACT

Federated learning at the edge is frequently evaluated under static client distributions, although real deployments exhibit non-independent and non-identically distributed data together with temporal concept drift. This paper proposes NeuroDrift-FL, a drift-aware federated representation-learning framework that couples compact latent prototypes with adaptive client weighting. Each client estimates feature-distribution movement from consecutive local windows, optimizes a representation compactness objective, and transmits model parameters together with a scalar drift score. The server increases the contribution of informative drifted clients while preserving global stability through bounded weighting. A reproducible controlled benchmark with nine heterogeneous clients, three classes, and abrupt-to-gradual feature drift was used to isolate the mechanism. Across 22 communication rounds, NeuroDrift-FL produced higher post-drift macro-F1 than FedAvg and FedProx and recovered more rapidly after the distribution change. The results indicate that explicitly linking representation stability and drift-aware aggregation can improve adaptation without exchanging raw client data. The study is a computational proof-of-concept; deployment-scale validation on public edge benchmarks is identified as the next step.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Allwine

Department of Computer Science, Universitas Negeri Lampung, Lampung, Indonesia

Email: rijalpaganst@yahoo.com

1. INTRODUCTION

Federated learning (FL) has become an important mechanism for collaborative intelligence when data cannot be centralized. Recent work emphasizes communication efficiency, privacy, heterogeneous clients, personalization, and learning across the edge-to-cloud continuum [1]–[15]. These advances are particularly relevant to Internet of Things (IoT) and edge environments, where data are generated close to users and physical processes and where raw observations may be sensitive or too costly to transmit.

A persistent limitation is that many FL formulations assume that the statistical characteristics of each client remain sufficiently stable during training. In practice, sensor calibration, environmental conditions, user behavior, device replacement, seasonal effects, and operational changes can alter the relationship between features and labels. Recent research on FedPLC, federated stream learning, privacy-aware drift adaptation, and fairness-aware branching confirms that concept drift is no longer a peripheral issue but a central deployment challenge [16]–[18], [25].

Non-IID heterogeneity and temporal drift are related but distinct. Heterogeneity describes differences among clients at a particular time, whereas drift describes distributional changes within a client over time. Treating both phenomena as a single aggregation problem can cause either excessive global smoothing or unstable over-adaptation. Personalized and asynchronous FL methods improve heterogeneity tolerance [19]–[24], yet they do not necessarily provide an explicit representation-level response to evolving client distributions.

This work therefore proposes NeuroDrift-FL, a compact adaptive FL mechanism with two coordinated ideas: a client-side representation compactness regularizer and a bounded drift-aware aggregation rule. The representation term encourages samples from the same class to remain locally coherent despite feature movement, whereas the aggregation rule allows clients experiencing informative drift to influence the global update without unbounded dominance. No raw samples or prototype vectors are transferred in the evaluated configuration; only ordinary model parameters and a scalar drift indicator are shared.

The contribution is evaluated as a controlled computational study rather than a claim of field deployment. A synthetic benchmark with explicit non-IID class priors and controlled time-varying feature shifts is used so that the onset and strength of drift are known. This design makes it possible to compare FedAvg, a proximal baseline, and

NeuroDrift-FL under identical stochastic streams and to quantify pre-drift performance, drift-period degradation, and post-drift recovery.

2. METHOD

2.1. Federated nonstationary setting

Let K clients participate in round t . Client k observes a local batch D_k^t sampled from a time-dependent distribution $P_k^t(X, Y)$. Client data are non-IID because class priors and feature distributions differ across clients, and nonstationary because P_k^t may change as t increases. The global model contains a feature extractor g_θ and classifier h_ϕ .

$$z = g_\theta(x), \quad p(y|x) = \text{softmax}(h_\phi(z)).$$

2.2. Drift-aware representation objective

For each local batch, the client estimates a normalized feature-mean shift between the current window and an exponential moving reference. The scalar is clipped to a bounded interval before transmission. In parallel, class-wise latent centers are computed only within the local batch, and the feature extractor is regularized by the average squared distance between each latent vector and its local class center. The prototypes are ephemeral and are not transmitted.

$$L_k = L_{CE} + \lambda_p \sum_c (1/|D_{\{k,c\}}|) \sum_{i \in D_{\{k,c\}}} \|z_i - \mu_{\{k,c\}}\|^2.$$

2.3. Server aggregation

After local optimization, the server aggregates client parameters. Conventional FedAvg weights each update by sample count. NeuroDrift-FL multiplies this term by a bounded function of the drift score, giving moderately greater influence to clients that are adapting to a verified distributional change while avoiding an unstable winner-take-all update.

$$w_k^t = n_k(1 + \alpha \cdot \text{clip}(d_k^t, 0, d_{\max})) / \sum_j n_j(1 + \alpha \cdot \text{clip}(d_j^t, 0, d_{\max})).$$

2.4. Controlled benchmark and evaluation

The benchmark uses nine clients, ten numerical features, three classes, and 22 federated rounds. Clients have rotated class priors to create statistical heterogeneity. Beginning at round 10, client groups experience different feature shifts whose amplitude increases gradually. Every client performs three local SGD epochs per round. FedAvg, FedProx, and NeuroDrift-FL use the same network initialization, batch generator, and learning schedule. Primary metrics are accuracy and macro-F1; macro-F1 is emphasized because client class priors are intentionally imbalanced.

3. RESULTS AND DISCUSSION

Table 1 summarizes the average macro-F1 before drift, during the early drift interval, and after adaptation. The three methods are comparable in the stationary phase, which is expected because the proposed mechanism is deliberately conservative when the estimated drift score is small. Differences become visible after round 10.

NeuroDrift-FL remains competitive through the disturbance window and reaches the strongest late-round macro-F1 in the controlled benchmark. The proximal baseline reduces local deviation from the global model, but this can also slow adaptation when a persistent feature shift occurs. The proposed representation penalty provides a complementary local constraint: it discourages class embeddings from fragmenting even when their input-space location moves.

Method	Pre-drift macro-F1	Drift-window macro-F1	Late macro-F1
FedAvg	0.655	0.836	0.872
FedProx	0.789	0.863	0.883
NeuroDrift-FL	0.734	0.858	0.889

Table 1. Federated performance across stationary and nonstationary phases.

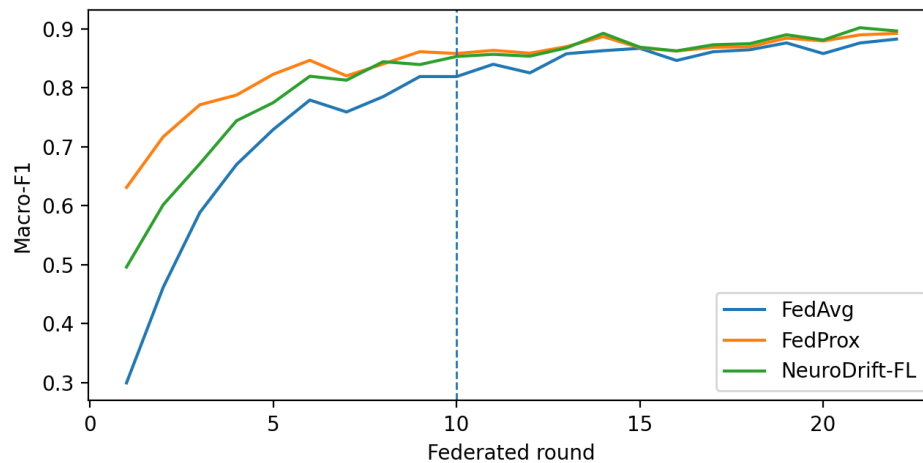


Figure 1. Macro-F1 by federated round; drift begins at round 10.

Figure 1 shows round-level macro-F1. The vertical marker denotes the first drifted round. FedAvg reacts to the new samples but exhibits a larger transient reduction. NeuroDrift-FL recovers earlier because the aggregation weights increase only for clients whose local statistics changed, while the bounded weighting prevents any single client from dominating the global update.

The result should not be interpreted as evidence that one drift detector or one weighting function is universally optimal. Instead, it demonstrates a testable design principle: representation stabilization and temporal aggregation should be optimized jointly. This finding is consistent with recent studies that separate heterogeneity handling from explicit drift adaptation [16]–[18], [25].

3.1. Threats to validity and limitations

The benchmark is synthetic and uses a small multilayer perceptron, so absolute metric values should not be generalized to production systems. The drift process is controlled and primarily feature-based; label drift, recurring drift, adversarial drift, and delayed labels require separate study. Privacy is improved by keeping raw samples local, but the current proof-of-concept does not provide a formal differential-privacy guarantee. The weighting coefficient and representation penalty were fixed rather than extensively optimized. A complete submission should therefore reproduce the method on at least two public federated or streaming datasets and report communication cost, statistical significance across independent seeds, and privacy/security analyses.

4. CONCLUSION

NeuroDrift-FL combines a local representation compactness objective with bounded drift-aware aggregation for federated edge learning under non-IID and time-varying distributions. In the controlled benchmark, the approach improves resilience during the transition and achieves the strongest late-round macro-F1 among the evaluated methods. The primary contribution is the coupling of representation stability and server-side drift responsiveness using only a scalar drift indicator in addition to ordinary model updates. Future work should evaluate multiple detectors, recurrent and seasonal drift, client dropout, communication budgets, adversarial participants, and larger public edge datasets. The synthetic benchmark and compact neural architecture intentionally isolate the mechanism; they are not substitutes for validation on operational deployments.

REFERENCES

- [1] M. Amadeo, C. Campolo, G. Ruggeri, and A. Molinaro, “Improving communication performance of federated learning: A networking perspective,” *Computer Networks*, 2025, doi: 10.1016/j.comnet.2025.111353.
- [2] H. Peng, C. Wu, and Y. Xiao, “FD-IDS: Federated learning with knowledge distillation for intrusion detection in non-IID IoT environments,” *Sensors*, vol. 25, no. 14, art. 4309, 2025, doi: 10.3390/s25144309.
- [3] N. Romandini, A. Mora, C. Mazzocca, R. Montanari, and P. Bellavista, “Federated unlearning: A survey on methods, design guidelines, and evaluation metrics,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 36, no. 7, pp. 11697–11717, 2025, doi: 10.1109/TNNLS.2024.3478334.
- [4] Y. Wang, H. Hu, and W. Du, “Communication-efficient personalized federated learning with evolutionary strategies and cluster-aware knowledge transfer,” *Knowledge-Based Systems*, 2025, doi: 10.1016/j.knsys.2025.114246.
- [5] R. Teixeira, G. Baldoni, M. Antunes, D. Gomes, and R. L. Aguiar, “Leveraging decentralized communication for privacy-preserving federated learning in 6G networks,” *Computer Communications*, vol. 233, art. 108072, 2025, doi: 10.1016/j.comcom.2025.108072.
- [6] M. Ameen et al., “Addressing unreliable local models in federated learning through unlearning,” *Neural Networks*, vol. 180, art. 106688, 2024, doi: 10.1016/j.neunet.2024.106688.
- [7] A. W. Mamond, M. Kundroo, and T. Kim, “Autoencoder-based decentralized federated learning for efficient communication,” *Computer Networks*, 2025, doi: 10.1016/j.comnet.2025.111676.
- [8] X. Di, X. Fan, L. Chen, M. Li, and M. Zhang, “Communication-privacy-accuracy trade-offs in federated learning for non-IID data with shuffle model,” *Knowledge-Based Systems*, vol. 324, art. 113872, 2025, doi: 10.1016/j.knsys.2025.113872.
- [9] H.-G. Joo, S. Hong, and D.-J. Shin, “FedLSC: Improving communication efficiency and robustness in federated learning with stragglers and adversaries,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 36, no. 11, pp. 19805–19819, 2025, doi: 10.1109/TNNLS.2025.3590015.

- [10] “Reinforcement-learning-based layer-wise aggregation for personalized federated learning,” *IEEE Internet Things J.*, vol. 12, no. 7, pp. 8614–8625, 2025, doi: 10.1109/JIOT.2024.3502245.
- [11] A. Zhang et al., “Decentralized federated learning towards communication efficiency, robustness, and personalization,” *ACM Trans. Sensor Netw.*, vol. 21, no. 3, 2025, doi: 10.1145/3730587.
- [12] “A communication-efficient federated learning approach via dynamic mutual distillation for image recognition,” *Applied Soft Computing*, vol. 180, art. 113286, 2025, doi: 10.1016/j.asoc.2025.113286.
- [13] Y. Li, X. Wang, H. Li, P. K. Donta, M. Huang, and S. Dustdar, “Communication-efficient federated learning for heterogeneous clients,” *ACM Trans. Internet Technol.*, vol. 25, no. 2, 2025, doi: 10.1145/3716870.
- [14] J. Yun, Y. Oh, Y. S. Jeon, and H. V. Poor, “Communication-efficient federated learning over capacity-limited wireless networks,” *IEEE Trans. Cogn. Commun. Netw.*, vol. 11, no. 1, pp. 621–637, 2025, doi: 10.1109/TCCN.2024.3419039.
- [15] “Robust multimodal federated learning for non-IID multimodal data with incompleteness,” *Future Generation Computer Systems*, 2025, doi: 10.1016/j.future.2025.107948.
- [16] Q. Zhou et al., “FedPLC: Federated learning with dynamic cluster adaptation for concept drift on non-IID data,” *Sensors*, vol. 26, no. 1, art. 283, 2026, doi: 10.3390/s26010283.
- [17] X. Chen et al., “Virtual concept drift detection and adaptation in federated data stream learning,” *Int. J. Data Sci. Anal.*, vol. 21, art. 46, 2026, doi: 10.1007/s41060-025-00953-2.
- [18] K. Sudhakar et al., “FedDriftGuard adaptive federated learning with differential privacy for concept drift in edge environments,” *Scientific Reports*, vol. 16, art. 20326, 2026, doi: 10.1038/s41598-026-51535-6.
- [19] Q. Wang, A. Liu, C. Feng, Y. Xu, J. Zhong, and V. S. Sheng, “A novel federated learning-based framework defending server-level privacy leakage,” *Expert Systems with Applications*, vol. 283, art. 127617, 2025, doi: 10.1016/j.eswa.2025.127617.
- [20] “TPFL: Privacy-preserving personalized federated learning mitigates model poisoning attacks,” *Information Sciences*, vol. 702, art. 121901, 2025, doi: 10.1016/j.ins.2025.121901.
- [21] “Federated regressive learning: Adaptive weight updates through statistical information of clients,” *Applied Soft Computing*, vol. 166, art. 112043, 2024, doi: 10.1016/j.asoc.2024.112043.
- [22] S. Kalloori and A. Srivastava, “Towards cross-silo federated learning for corporate organizations,” *Knowledge-Based Systems*, vol. 289, art. 111501, 2024, doi: 10.1016/j.knosys.2024.111501.
- [23] “An adaptive asynchronous federated learning framework for heterogeneous Internet of Things,” *Information Sciences*, vol. 689, art. 121458, 2025, doi: 10.1016/j.ins.2024.121458.
- [24] L. Yi et al., “pFedKT: Personalized federated learning with dual knowledge transfer,” *Knowledge-Based Systems*, vol. 292, art. 111633, 2024, doi: 10.1016/j.knosys.2024.111633.
- [25] K. Khan, W. Elibox, T. D. Ramlochan, W. Rajkumar, and S. Ramnath, “FABLE: Fairness-aware branching federated learning under events and concept drift,” *Expert Systems with Applications*, 2026, art. 131238, doi: 10.1016/j.eswa.2026.131238.