Contents lists available at IdNNS

Indonesian Journal of Cyber-AI and Security Intelligence

Volume 1 ; Issue 2

journal homepage: https://journal.idnns.org/index.php/ijcasi

Federated Averaging with Replay-Based Incremental Learning for Distributed Neural Network Training

Vinaye Armoogum[a,*]

[a] *University of Technology, Mauritius, Mauritius*

ARTICLE INFO

Keywords:
federated learning
incremental learning
catastrophic forgetting
FedAvg
continual learning
replay memory
distributed neural networks

ABSTRACT

Federated Learning (FL) enables collaborative training of neural networks across decentralized clients without exposing raw data, while Continual (incremental) Learning aims to let a deployed model absorb new knowledge without discarding what it has already learned. In real deployments these two requirements co-occur: a global model produced by Federated Averaging (FedAvg) must later be updated as new, non-stationary data streams arrive at the server, which exposes it to catastrophic forgetting. This paper presents and evaluates a lightweight framework that combines (i) a FedAvg-based federated training module, (ii) a replay-memory incremental learner that mixes new-task samples with a bounded buffer of past samples, and (iii) a robust continuous-learning controller that monitors validation accuracy on a held-out reference set and rolls back weight updates that degrade previously acquired knowledge beyond a safety threshold. We describe the system architecture, implementation, and an empirical study on a three-task synthetic benchmark that mimics heterogeneous client data and sequential concept drift. The results indicate that replay-based updating substantially reduces the accuracy degradation on previously learned tasks relative to naive sequential fine-tuning, at a modest cost in plasticity toward more complex new tasks. We discuss these findings in relation to the wider literature on Federated Continual Learning (FCL) and outline directions for scaling the approach to non-synthetic, non-IID, cross-device settings.

1. Introduction

The deployment of machine learning models on distributed, privacy-sensitive infrastructures—mobile devices, hospitals, industrial sensors—has motivated Federated Learning (FL), a paradigm in which a shared global model is trained by aggregating locally computed updates rather than centralizing raw data (McMahan et al., 2017). The Federated Averaging algorithm (FedAvg) (McMahan et al., 2017) remains the reference baseline for this setting, and subsequent work has proposed refinements such as FedProx (Li et al., 2020b) and SCAFFOLD (Karimireddy et al., 2020) to address system and statistical heterogeneity across clients, together with broader surveys of the challenges and open problems in the field (Li et al., 2020a; Yang et al., 2019; Kairouz et al., 2021; Banabilah et al., 2022).

A complementary and largely orthogonal challenge arises once a federated model is deployed: real environments are non-stationary, and the global model is expected to keep learning from newly arriving data. Naively fine-tuning a trained network on new data causes catastrophic forgetting—an abrupt loss of performance on previously learned tasks—a phenomenon documented since the earliest connectionist literature (McCloskey and Cohen, 1989) and studied extensively in the Continual Learning (CL) literature through regularization-based methods such as Elastic Weight Consolidation (Kirkpatrick et al., 2017) and Learning without Forgetting (Li and Hoiem, 2018), as well as rehearsal-based methods such as iCaRL (Rebuffi et al., 2017), which retain a bounded memory of past exemplars for replay. When FL and CL are combined, the resulting setting—Federated Continual Learning (FCL)—inherits both spatial heterogeneity across clients and temporal heterogeneity across tasks, and has recently attracted dedicated surveys (Zhu et al., 2021; Hamedi et al., 2025; Casado et al., 2022).

This paper reports on the design and empirical evaluation of a compact FCL-style pipeline built around three cooperating components: a FedAvg trainer that aggregates client updates into a global model, a replay-based incremental learner that mixes new-task data with a fixed-size memory buffer of past samples, and a robust continuous-learning controller that safeguards previously acquired knowledge by rolling back updates whenever validation accuracy on a reference set drops below a safety threshold. The contribution is twofold: (1) a concrete, reproducible implementation of the

*Corresponding author
varmoogum@utm.ac.mu (V. Armoogum)
ORCID(s):

three components and their interaction, and (2) a controlled empirical study, on a three-task synthetic benchmark, quantifying how much replay-based updating mitigates forgetting relative to naive sequential fine-tuning, and where its limits lie when a new task requires learning a substantially more complex decision boundary.

2. Related Work

2.1. Federated Learning and Aggregation

FedAvg (McMahan et al., 2017) aggregates client-side stochastic gradient descent updates by averaging model weights, weighted by local dataset size. Because clients typically hold non-IID data, plain averaging can converge slowly or to a biased optimum; FedProx (Li et al., 2020b) introduces a proximal term that limits client drift from the global model, while SCAFFOLD (Karimireddy et al., 2020) uses control variates to correct for client-drift bias directly in the update rule. Communication efficiency and privacy have also been addressed through secure aggregation protocols that let the server compute the sum of client updates without observing any individual contribution (Bonawitz et al., 2017), and through differentially private training such as DP-SGD (Abadi et al., 2016), which bounds the privacy leakage of the shared updates at the cost of added noise. Broader treatments of the FL design space, including personalization, robustness and applications, are surveyed in (Li et al., 2020a; Yang et al., 2019; Kairouz et al., 2021; Banabilah et al., 2022), and the specific challenge of non-IID client data is surveyed in Zhu et al. (2021).

2.2. Continual Learning and Catastrophic Forgetting

Catastrophic forgetting—the tendency of a neural network to overwrite previously learned representations when trained on new data—was first characterized as the sequential learning problem (McCloskey and Cohen, 1989) and later demonstrated at scale in deep networks (Kirkpatrick et al., 2017). Two complementary families of mitigation strategies dominate the CL literature. Regularization-based approaches, such as Elastic Weight Consolidation (Kirkpatrick et al., 2017) and Learning without Forgetting (Li and Hoiem, 2018), constrain updates to weights that are important for previously learned tasks, either through an explicit Fisher-information penalty or through knowledge-distillation losses computed against the pre-update network. Rehearsal-based approaches, exemplified by iCaRL (Rebuffi et al., 2017), retain a bounded set of exemplars from past tasks and interleave them with new-task data during training, which is the strategy adopted for the incremental-learning module described in Section 3.

2.3. Federated Continual Learning

Recent surveys formalize Federated Continual Learning as the intersection of FL's spatial heterogeneity (non-IID data across clients) and CL's temporal heterogeneity (non-stationary task sequences), and catalogue solutions ranging from client-side rehearsal to server-side knowledge distillation and prompt-based parameter isolation (Hamedi et al., 2025). Related work has also examined concept-drift detection mechanisms that trigger adaptation only when the underlying data distribution has genuinely shifted (Casado et al., 2022), and privacy-preserving computation techniques that are complementary to, rather than in competition with, forgetting-mitigation strategies (Chen et al., 2024). The framework described in this paper sits within this design space: it targets the server-side model that results from FedAvg aggregation, and applies rehearsal-based incremental learning together with an explicit rollback safeguard to that global model as new data arrives over time.

3. Proposed Framework

The framework consists of three cooperating modules, implemented as a feed-forward neural classifier (two hidden layers, ReLU activations, dropout regularization) trained with the Adam optimizer. Figure references are omitted for brevity; the architecture is summarized narratively below.

3.1. Federated Averaging Module

At each communication round, the server broadcasts the current global weights to a set of participating clients. Each client initializes a local copy of the model, performs several epochs of local SGD/Adam training on its own partition of data, and returns the updated local weights. The server then computes the coordinate-wise average of all client weight tensors—the FedAvg update rule (McMahan et al., 2017)—and adopts the result as the new global model. This module is responsible for producing the initial, privacy-preserving global model from data distributed across clients before any incremental adaptation takes place.

3.2. Replay-Based Incremental Learner

When new labeled data becomes available at the server (a new "task"), the incremental learner (i) trains the current global model on the new-task samples for a fixed number of epochs, then (ii) retrains it for a smaller number of epochs on a bounded first-in-first-out memory buffer containing samples accumulated from all previous tasks, following the general rehearsal principle used by iCaRL (Rebuffi et al., 2017) and consistent with the regularization-versus-rehearsal taxonomy discussed in Section 2.2. Finally, (iii) a subsample of the new-task data is appended to the memory buffer, which is capped at a fixed size so that older samples are evicted as new ones arrive. The ratio between new-task epochs and replay epochs (the "replay weight") controls the trade-off between plasticity on the new task and stability on past tasks.

3.3. Robust Continuous-Learning Controller

For streaming updates that arrive in small batches rather than as discrete tasks, a lightweight controller buffers incoming samples and triggers a single-epoch update once the buffer reaches a fixed size. Before committing an update, the controller snapshots the pre-update weights; after the

update it evaluates accuracy on a fixed, previously held-out reference set. If accuracy falls below a safety threshold, the controller discards the update and restores the snapshot, preventing a single noisy or adversarial batch from corrupting the deployed model. This mechanism is a pragmatic, low-overhead complement to the exemplar-replay strategy in Section 3.2 rather than a replacement for it: it protects against acute degradation between scheduled incremental-learning stages.

4. Experimental Setup

To evaluate the framework in a controlled setting, we constructed a three-task synthetic benchmark following common practice in the CL literature for isolating the effect of forgetting from confounds such as dataset-specific noise. Each task consists of 5,000 samples with 20 real-valued features drawn i.i.d. from a standard normal distribution; binary labels are generated by task-specific decision rules of increasing complexity: T1 is a linear rule (sum of the first two features), T2 is a linear rule over three features, and T3 is a bilinear (product) rule over two features, which is not linearly separable and is markedly harder for a shallow network to learn within a limited training budget. For the federated stage, the T1 training partition is split evenly across three simulated clients, each holding a disjoint, class-balanced shard, and three FedAvg communication rounds are executed with five local epochs per round. For the incremental stage, T2 and T3 are introduced sequentially at the server, each split into a training portion (used for adaptation and replay-buffer accumulation) and a held-out test portion (used only for evaluation).

Two conditions are compared at each incremental stage: (a) **naive fine-tuning**, in which the global model is updated only on the new task's training data, with no rehearsal, representing the common but forgetting-prone baseline; and (b) **replay-based updating** (the proposed incremental learner from Section 3.2), in which new-task training is interleaved with rehearsal on the accumulated memory buffer. Both conditions start from the same FedAvg-trained global model to isolate the effect of the incremental-learning strategy. All accuracies are reported on task-specific held-out test partitions that were not used for training or replay.

The reference implementation targets a Keras/TensorFlow neural network (two hidden layers of 64/32/16 units, dropout 0.2, softmax output) trained with Adam. As the evaluation environment used to produce the numbers reported here did not provide a TensorFlow runtime, a behaviourally equivalent scikit-learn multi-layer perceptron (two hidden layers of 32 and 16 units, ReLU activation, Adam solver) with an explicit, hand-implemented FedAvg weight-averaging routine was used to reproduce the same experimental protocol end to end and to obtain the quantitative results in Section 5. This substitution preserves the architecture family, optimizer, and training protocol described above and is disclosed here in the interest of reproducibility; results should be read as illustrative of the qualitative behaviour of the framework rather than as a benchmark against the exact Keras implementation.

5. Results and Discussion

Table 1 summarizes classification accuracy on each task's held-out test partition at successive stages of training. "T1" denotes the original federated task; "T2" and "T3" denote the two incrementally introduced tasks.

Three observations follow from Table 1. First, federated averaging over three rounds improved accuracy on the original task from 0.924 to 0.938, consistent with the expected benefit of aggregating client updates over a single client's local optimum, in line with the original FedAvg results (McMahan et al., 2017). Second, replay-based updating substantially reduced forgetting of T1: after both incremental stages, naive fine-tuning retained 0.870 of the original task's accuracy (a drop of 6.8 percentage points from the post-FedAvg 0.938), whereas replay-based updating retained 0.911 (a drop of only 2.7 points), a relative reduction in forgetting of roughly 60%. This is consistent with the general finding in the rehearsal-based CL literature that interleaving even a bounded memory of past samples substantially mitigates, without fully eliminating, catastrophic forgetting (Rebuffi et al., 2017; Kirkpatrick et al., 2017).

Third, both conditions struggled to learn T3 within the fixed training budget (accuracy close to 0.50, i.e., chance level for a balanced binary task), reflecting the intrinsic difficulty of its bilinear, XOR-like decision boundary for a shallow network trained for a small, fixed number of epochs, rather than a failure of the forgetting-mitigation mechanism itself. This limitation is informative: it indicates that the replay mechanism protects previously acquired knowledge but does not by itself guarantee sufficient plasticity for arbitrarily complex new tasks, echoing the stability–plasticity trade-off that is central to the continual-learning literature (Li and Hoiem, 2018) and to recent Federated Continual Learning surveys (Hamedi et al., 2025). In practice this suggests that the replay ratio, learning rate, and per-task training budget should be tuned jointly with task difficulty, and that a concept-drift-aware trigger such as that discussed in Casado et al. (2022) could be used to allocate more training budget to genuinely novel and complex shifts.

Taken together, these results support the central design premise of the framework: combining FedAvg aggregation with bounded-memory rehearsal is an effective and lightweight way to let a federated global model keep learning from new, sequentially arriving data while retaining most of its previously acquired competence, at a cost that is well characterized by the size of the replay buffer and the replay-to-new-data training ratio.

Table 1
Accuracy on each task at successive training stages.

Stage	Method	Acc. on Old Task (T1)	Acc. on Task T2	Acc. on Task T3
Initial training (T1 only)	Baseline	0.924	–	–
After 3 FedAvg rounds	Federated (3 clients)	0.938	–	–
After learning T2	Naive fine-tuning	0.886	0.871	–
After learning T2	Replay-based (proposed)	0.924	0.844	–
After learning T3	Naive fine-tuning	0.870	0.836	0.510
After learning T3	Replay-based (proposed)	0.911	0.851	0.503

6. Conclusion and Future Work

This paper presented a compact framework that couples Federated Averaging with a replay-based incremental learner and a rollback-safeguarded continuous-learning controller, targeting the practically important scenario in which a federated global model must keep adapting to new, non-stationary data after deployment. An empirical study on a three-task synthetic benchmark showed that rehearsal substantially reduces catastrophic forgetting relative to naive sequential fine-tuning, while also revealing that rehearsal alone does not resolve the underlying stability–plasticity trade-off when a new task is intrinsically harder to learn. Future work should (i) evaluate the framework on standard non-IID federated benchmarks and on realistic task sequences rather than synthetic decision rules, (ii) compare exemplar rehearsal against regularization-based alternatives such as Elastic Weight Consolidation (Kirkpatrick et al., 2017) and Learning without Forgetting (Li and Hoiem, 2018) under matched memory and compute budgets, (iii) integrate privacy-preserving aggregation (Bonawitz et al., 2017) and differentially private training (Abadi et al., 2016) so that the replay buffer itself does not become a privacy liability, and (iv) incorporate concept-drift detection (Casado et al., 2022) to adaptively allocate training budget across tasks of varying difficulty within a full Federated Continual Learning pipeline (Hamedi et al., 2025; Zhu et al., 2021).

References

Abadi, M., Chu, A., Goodfellow, I., McMahan, H., Mironov, I., Talwar, K., Zhang, L., 2016. Deep learning with differential privacy, in: Proc. 2016 ACM SIGSAC Conf. on Computer and Communications Security (CCS), ACM. pp. 308–318. doi:10.1145/2976749.2978318.

Banabilah, S., Aloqaily, M., Alsayed, E., Malik, N., Jararweh, Y., 2022. Federated learning review: fundamentals, enabling technologies, and future applications. Inf. Process. Manag. 59, 103061. doi:10.1016/j.ipm.2022.103061.

Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H., Patel, S., Ramage, D., Segal, A., Seth, K., 2017. Practical secure aggregation for privacy-preserving machine learning, in: Proc. 2017 ACM SIGSAC Conf. on Computer and Communications Security (CCS), ACM. pp. 1175–1191. doi:10.1145/3133956.3133982.

Casado, F., Lema, D., Criado, M., Iglesias, R., Regueiro, C., Barro, S., 2022. Concept drift detection and adaptation for federated and continual learning. Multimed. Tools Appl. 81, 3397–3419. doi:10.1007/s11042-021-11219-x.

Chen, J., Yan, H., Liu, Z., Zhang, M., Xiong, H., Yu, S., 2024. When federated learning meets privacy-preserving computation. ACM Comput. Surv. 56, 1–36. doi:10.1145/3679013.

Hamedi, P., Razavi-Far, R., Hallaji, E., 2025. Federated continual learning: concepts, challenges, and solutions. Neurocomputing 651, 130844. doi:10.1016/j.neucom.2025.130844.

Kairouz, P., McMahan, H., Avent, B., et al., 2021. Advances and open problems in federated learning. Found. Trends Mach. Learn. 14, 1–210. doi:10.1561/2200000083.

Karimireddy, S., Kale, S., Mohri, M., Reddi, S., Stich, S., Suresh, A., 2020. SCAFFOLD: stochastic controlled averaging for federated learning, in: Proc. 37th Int. Conf. on Machine Learning (ICML), PMLR 119, pp. 5132–5143. URL: https://proceedings.mlr.press/v119/karimireddy20a.html.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., Hassabis, D., Clopath, C., Kumaran, D., Hadsell, R., 2017. Overcoming catastrophic forgetting in neural networks. Proc. Natl. Acad. Sci. 114, 3521–3526. doi:10.1073/pnas.1611835114.

Li, T., Sahu, A., Talwalkar, A., Smith, V., 2020a. Federated learning: challenges, methods, and future directions. IEEE Signal Process. Mag. 37, 50–60. doi:10.1109/MSP.2020.2975749.

Li, T., Sahu, A., Zaheer, M., Sanjabi, M., Talwalkar, A., Smith, V., 2020b. Federated optimization in heterogeneous networks. Proc. Mach. Learn. Syst. (MLSys) 2, 429–450. URL: https://proceedings.mlsys.org/book/316.pdf.

Li, Z., Hoiem, D., 2018. Learning without forgetting. IEEE Trans. Pattern Anal. Mach. Intell. 40, 2935–2947. doi:10.1109/TPAMI.2017.2773081.

McCloskey, M., Cohen, N., 1989. Catastrophic interference in connectionist networks: the sequential learning problem. Psychol. Learn. Motiv. 24, 109–165. doi:10.1016/S0079-7421(08)60536-8.

McMahan, H., Moore, E., Ramage, D., Hampson, S., Agüera y Arcas, B., 2017. Communication-efficient learning of deep networks from decentralized data, in: Proc. 20th Int. Conf. on Artificial Intelligence and Statistics (AISTATS), PMLR 54, pp. 1273–1282. URL: https://proceedings.mlr.press/v54/mcmahan17a.html.

Rebuffi, S.A., Kolesnikov, A., Sperl, G., Lampert, C., 2017. iCaRL: incremental classifier and representation learning, in: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 5533–5542. doi:10.1109/CVPR.2017.587.

Yang, Q., Liu, Y., Chen, T., Tong, Y., 2019. Federated machine learning: concept and applications. ACM Trans. Intell. Syst. Technol. 10, 1–19. doi:10.1145/3298981.

Zhu, H., Xu, J., Liu, S., Jin, Y., 2021. Federated learning on non-IID data: a survey. Neurocomputing 465, 371–390. doi:10.1016/j.neucom.2021.07.098.