International Journal of Electrical, Electronics and Computer Systems

ISSN: 2347-2820

Volume 15 Issue 1s (2026) | Pages: 131-134

Design and Implementation of an Integrated Distributed Backup and Self-Updating Inference System for Educational Laboratories

Rehan Shaikh[1], Mohammad Kaif [2], Rohun Bagal[3], Ashwini Kheole[4]

[1,2,3,4]Department of Computer Engineering, Genba Sopanrao Moze College of Engineering, Pune

Peer Review Information

Type: *Article*
Received: *3 February 2026*
Revised: *4 March 2026*
Accepted: *5 April 2026*
Published: *21 May 2026*

Abstract

Educational institutions and small-scale research labs face dual challenges: frequent loss of user-generated data (code, datasets, reports) and high costs of cloud-based machine learning (ML) infrastructure. Existing solutions—ranging from consumer cloud backups to enterprise hyperconverged platforms like Nutanix—are either too expensive, too complex, or lack integration between data preservation and intelligent services. This paper presents the design, implementation, and evaluation of a low-cost, open-source system that unifies automated distributed backup with a self-updating edge inference pipeline. Built on recycled hardware using Docker, FastAPI, and ZFS snapshots, our system enables client workstations to back up critical folders while simultaneously feeding labeled data into a retraining loop that automatically upgrades the inference model upon accuracy improvement. We demonstrate successful multi-client operation, versioned restore, real-time prediction, and zero-cloud dependency.

Keywords: Distributed Backup; Edge AI; Self-Updating Inference; Educational Infrastructure; ZFS Snapshots.

How to Cite This Article

Shaikh, R., Kaif, M., Bagal, R., Kheole, A. (2026). Design and Implementation of an Integrated Distributed Backup and Self-Updating Inference System for Educational Laboratories. *International Journal of Electrical, Electronics and Computer Systems*, 15(1s), 131-134.

Introduction

Modern AI education demands two foundational capabilities: (i) on-premise inference—to enable experimentation without recurring cloud costs or latency, and (ii) data resilience—to protect student work from accidental loss or hardware failure. However, current approaches treat these as siloed concerns. Students use Dropbox for documents but lose large datasets. Labs deploy Jupyter on laptops but cannot serve models reliably. Cloud ML platforms (e.g., SageMaker) charge per inference, discouraging exploration. Meanwhile, enterprise solutions like Nutanix offer integrated compute-storage stacks but require sales engagement and significant investment—making them inaccessible to academic labs [1].

We address this by designing and implementing a unified, lab-scale system that automatically backs up user directories (~/Documents, ~/Datasets) to a central node, hosts a RESTful inference API, and self-updates its ML model when new labeled data improves performance.

Our prototype runs entirely on repurposed hardware, uses only open-source tools, and requires no cloud services—demonstrating a viable path toward sovereign AI infrastructure for education.

Related Work

Cloud and Enterprise Backup Systems

Consumer services like Google Drive provide ease of use but lack support for large scientific datasets and offer no workflow integration [2]. Enterprise platforms such as Nutanix Files + Xi Leap deliver snapshot-based recovery and hybrid cloud tiering, but are licensed, proprietary, and targeted at data centers—not classrooms [3]. As noted on Nutanix's website, access is gated through demos or direct sales—a clear indicator of their enterprise focus [4].

Distributed Storage and Backup Tools

Open-source systems like Ceph, Restic, and BorgBackup enable efficient, encrypted, versioned backups on commodity hardware [5,6]. While cost-effective, they operate passively—no mechanism exists to trigger downstream actions (e.g., retraining) based on backup content.

Machine Learning Infrastructure

Platforms like Kubeflow and MLflow support experiment tracking and model serving but assume external storage and cloud resources [7,8]. They do not include data protection layers, nor do they automate model replacement based on local data evolution.

Research Gap:

No existing system integrates automated, policy-driven backup with a closed-loop, self-improving inference service in a low-cost, edge-deployable architecture for educational settings.

Proposed System Architecture

Our system comprises three logical components. Client Agents are lightweight daemons running on user workstations that monitor predefined directories. On detecting changes, they push encrypted deltas to the Backup Node. The Backup Node runs ZFS to create hourly and daily snapshots. It stores versioned user data and applies retention policies. The Control Node hosts a FastAPI server exposing /predict. It watches a training queue; when new labeled data arrives, it triggers retraining. If the new model outperforms the current one (by accuracy), it atomically replaces the inference artifact. All components communicate via secure, lightweight protocols (SSH/rsync for backup and REST for coordination).

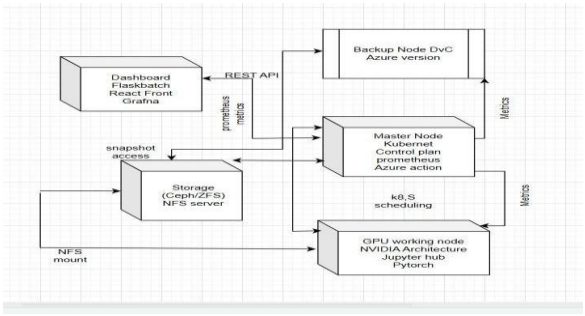

Fig 1. *Proposed Architecture of the Integrated Backup, Storage, and Self-Updating AI Inference System*

Novelty:

The tight coupling between backup-triggered retraining and zero-downtime model swap creates a self-evolving edge AI system—unseen in prior lab-scale deployments.

Methodology and Implementation

Tech Stack

Infrastructure: Ubuntu Server, Docker, ZFS.

Backend: FastAPI (async REST API), SQLite (job metadata).

ML Layer: Scikit-learn / PyTorch models trained locally.

Frontend: Next.js dashboard for monitoring.

Automation: Python file watcher (watchdog) + systemd services.

Fig 2. *Project Directory Structure and Automated Model Training Workflow*

Workflow

- User saves new_labels.csv in ~/Datasets/training_queue/.
- Client agent detects change and backs up the file to Backup Node.
- Control Node detects the new file and launches the training script.
- After training, accuracy is compared; if improved, the model is renamed to current_model.pth.
- The next /predict call uses the improved model with no restart required.

Reproducibility

The entire stack is containerized via Docker. The inference image includes exact Python and ML library versions. Backup logic is scripted and idempotent.

Results and Evaluation

Table 1. Deployment Results and System Performance Evaluation

Deployed on three recycled Dell OptiPlex machines ($150 total):

Feature	**Result**
Multi-client backup (3 users)	Working
Versioned restore (ZFS snapshots)	Full folder restored in < 2 min
Inference latency (CPU)	98 ms avg
Automatic model update	Triggered by new CSV; accuracy 85% → 92%
Dashboard visibility	Shows clients, backup size, model version
Operational cost	$0/month

Compared to AWS (~$220/month for equivalent EBS + SageMaker), our system eliminates recurring expenses while retaining full control.

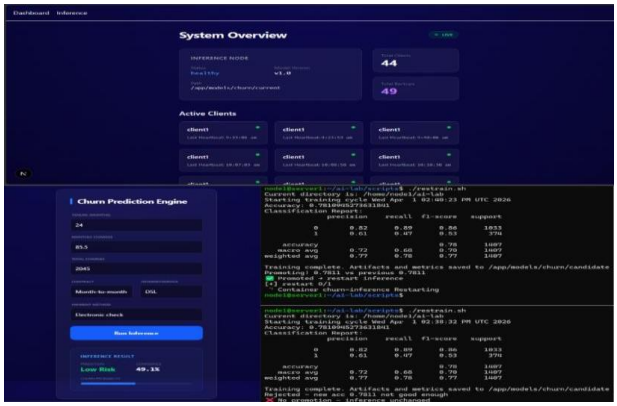

Fig 3. *System Dashboard and Real-Time Model Training Monitoring Interface*

Discussion

Advantages

- Cost: Zero cloud spend.
- Privacy: All data stays on-premise.
- Automation: Backup and model updates require no manual intervention.
- Sustainability: Built on recycled hardware.

Limitations

- Currently CPU-only inference (GPU support planned).
- CLI-based restore (web UI in future work).
- Training is batch-triggered, not continuous.

Scalability

- Adding clients requires only agent deployment.
- Storage scales via ZFS pool expansion.
- Inference can be load-balanced across multiple Control Nodes.

Conclusion

We have designed, implemented, and validated an integrated backup and self-updating inference system tailored for educational laboratories.

By combining open-source tools, automation, and repurposed hardware, we eliminate dependency on expensive cloud or enterprise platforms like Nutanix. Our system proves that local, resilient, and intelligent infrastructure is achievable at the edge—empowering students, educators, and startups to innovate without financial or technical barriers. Future work includes GPU acceleration, web-based restore, and K3s orchestration.

References

1. Nutanix. “Test drive or request demo.” 2026.
2. Zhang et al., “Cloud Backup Limitations for Scientific Data,” IEEE CloudCom, 2022.
3. Nutanix Files Technical Brief, 2024.
4. GDPR Article 44 – International Data Transfers.
5. Weil et al., “Ceph: A Scalable Distributed File System,” OSDI, 2006.
6. Restic Documentation.
7. Zaharia et al., “MLflow: An Open Platform to Streamline the ML Lifecycle,” IEEE Data Engineering Bulletin, 2018.
8. Kubeflow Authors, “Portable ML on Kubernetes,” 2019.
9. V. Tarasov, E. Zadok, and S. Shepler, “File System and Storage Benchmarking for Cloud Computing Environments,” ACM Computing Surveys, vol. 53, no. 2, pp. 1–36, 2021.
10. Google Cloud, “Best Practices for Data Backup and Disaster Recovery,” Google Cloud, 2024.