Incremental Learning in Network Traffic Management: A Fixed-Representation Rehearsal Approach

Francisco Rau[1 3], Petia Georgieva[2], Carlos Herranz[1], Iñaki Val[1], Joaquin Perez[3]

[1] MaxLinear, Inc., Valencia, Spain [2] Instituto de Telecomunicações / IEETA, University of Aveiro, Portugal
[3] Electronic Engineering Dept., Universitat de València, Valencia, Spain • Corresponding author: frau@maxlinear.com

The Problem

- **Traffic Drift** → Classifiers become outdated.
- **Full Retraining** → High computational and storage cost.
- **Naïve Fine-Tuning** → Catastrophic forgetting.

Current Solutions

- **Class-Incremental Learning** → Reduces forgetting but relies on costly replay/distillation mechanisms that conflict with resource-constrained deployments.
- **Fixed-representation combined with rehearsal** → Underexplored approach, restricted to packet-level scenarios.

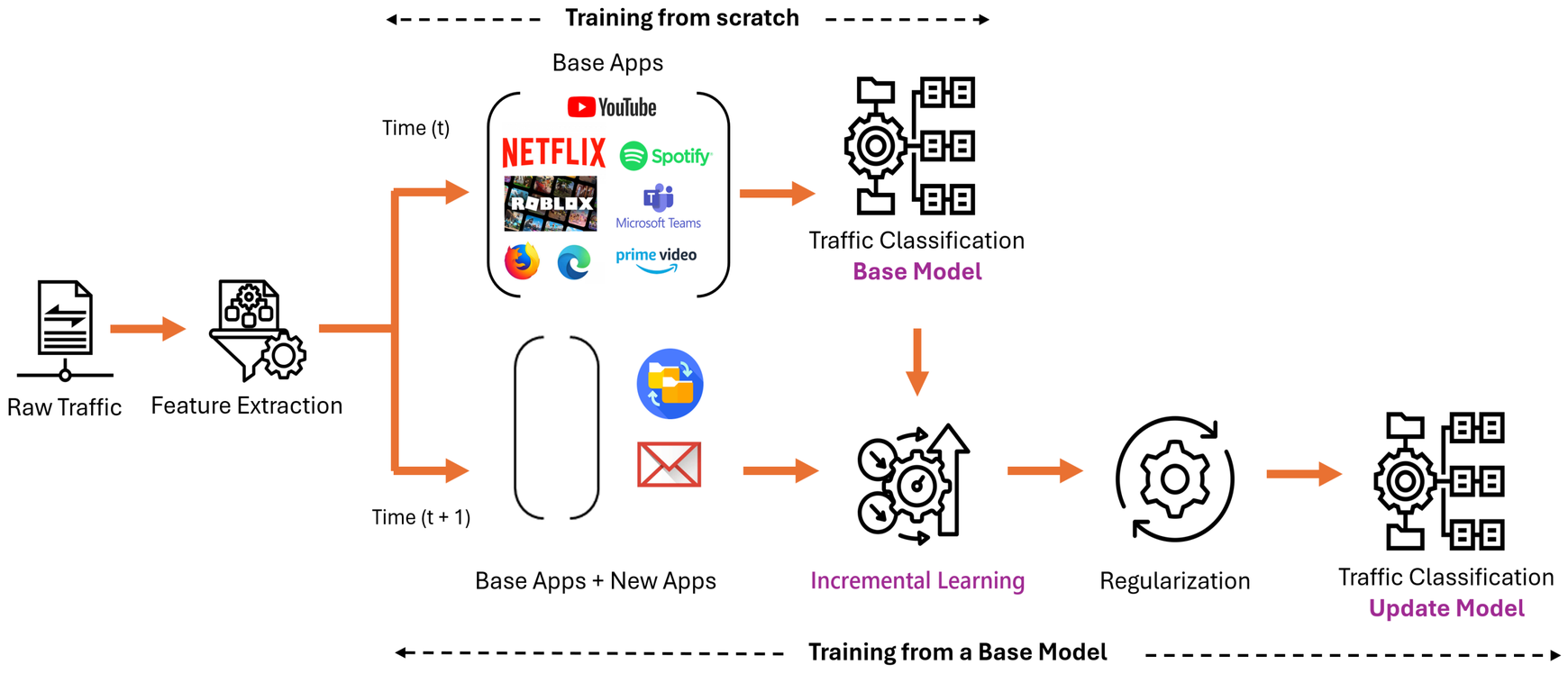


Fig. 1: Stages of the evaluated incremental-learning model — a frozen FCNN backbone with a 10% rehearsal buffer; only the classification head is retrained for the new class(es).

Our Contribution

- First dedicated evaluation *(to our knowledge)* of a **fixed-representation rehearsal** strategy at the **flow level.** Training cost & memory instead of maximum accuracy.
- **Freeze the whole feature-extraction backbone** (hidden layers 1–5); re-initialise and retrain **only the classification head.**
- Rehearsal buffer: **10% of the base training set** (stratified), replayed with all samples of the new class(es).
- Outcome: representational stability + fast updates that avoid catastrophic forgetting without complex pipelines.

Experimental Setup

Dataset: 46,729 labeled flows (VLC + VNAT) — *8 classes, 29 flow-level features (5-s windows), 80/20 train/test split.*

Base model — 6 classes → *Video Streaming, Roblox, Spotify, Teams, YouTube, Web.*
Scenario A (6+1) → *+ File Transfer*
Scenario B (6+2) → *+ File Transfer, + C2 (Command & Control).*

FCNN: 29 → 2048 → 1024 → 768 → 512 → 256 → softmax (ReLU, BatchNorm, Dropout, L2; AdamW, lr 1e-4).

Training time

Wall-clock time to train the model

from scratch → incremental	:	reduction
7 classes → 6 + 1 classes	**1,696 s → 133 s**	**92.16%**
8 classes → 6 + 2 classes	**1,872 s → 192 s**	**89.72%**

Storage

Memory footprint of the training data

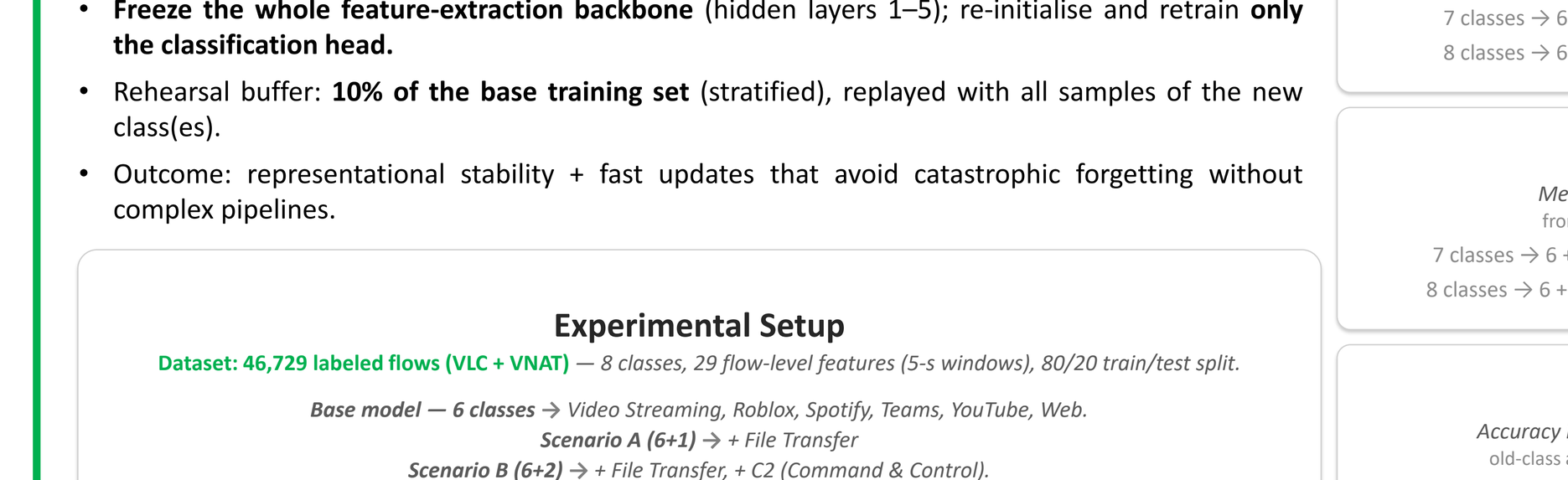
from scratch → incremental	:	reduction
7 classes → 6 + 1 classes	**9.2 MB → 2.1 MB**	**76.88%**
8 classes → 6 + 2 classes	**10.6 MB → 3.4 MB**	**67.50%**

Forgetting

Accuracy lost on old classes when adding new ones

old-class accuracy, from scratch → incremental	:	gap
7 classes → 6 + 1 classes	**86.4% → 85.8%**	**0.56%**
8 classes → 6 + 2 classes	**86.0% → 84.9%**	**1.13%**

Results



(a) Base — 6 classes · 88.0% **(b) +1 class (6+1)** · 87.9% **(c) +2 classes (6+2)** · 87.7%

Fig. 2: Row-normalised confusion matrices (%): base 6-class model, then the incremental model after adding 1 and 2 new classes.

Next Steps

- **Long-Term Evaluation** → Assess performance across longer continuous-learning update sequences.
- **Scalable Scenarios** → Extend beyond 8 classes with imbalanced and evolving traffic arrivals.

This work was supported by the EU Horizon Europe MSCA Doctoral Networks project OWIN6G (GA No. 101119624); Grant PID2023-148671OB-I00 (MICIU/AEI/10.13039/501100011033, ERDF/EU); and the European Union–NextGenerationEU (Bulgaria RRP, BG-RRP-2.004-0005).