Fast Packet-Sequence and Flow-Summary Feature Extensions for Network Traffic Classification

Viraj Bodiwala

1 Introduction

Machine learning models for network traffic classification often rely on fixed-length packet representations, such as the first P packets of a flow encoded using packet sizes and timing information. These representations are attractive because they are simple, lightweight, and compatible with existing libraries such as nPrint and pcapML. However, such representations may discard important flow-level context, including byte volume, protocol composition, and port usage patterns.

In this project, I investigate whether extending a standard packet-sequence representation with lightweight flow-summary features improves classification performance. I compare a baseline packet-sequence representation against an extended representation that incorporates summary statistics, and I evaluate both under standard random splits and temporally stratified splits that better reflect real-world deployment conditions.

This work is motivated by the ML/Net application identification benchmark and leaderboard provided by nPrint:

https://nprint.github.io/benchmarks/application_identification/netml_vpn-nonvpn.html.

2 Data

The dataset consists of a labeled packet capture in `pcapng` format. Each packet includes an annotation in the `opt_comment` field identifying a sample ID and class label. Packets sharing the same sample ID are grouped to form a flow-level sample.

For each sample, I extract packet timestamps, signed packet lengths, transport-layer protocol, and source and destination ports. To control runtime, I cap the number of packets stored per sample while preserving the "first P packets" semantics.

The dataset is downsampled to a fixed target size using stratified sampling by class label, preserving class balance while reducing computational cost.

3 Feature Representations

Each sample is featurized using the first P packets.

3.1 Baseline Packet-Sequence Features

The baseline representation encodes each packet using signed packet length, log-transformed inter-arrival time, and protocol indicators (TCP, UDP, other), yielding a feature vector of dimension $5P$.

3.2 Extended Packet-Sequence + Flow-Summary Features

The extended representation augments the baseline sequence with 16 flow-level summary features, including log-transformed duration, byte statistics, protocol fractions, and source/destination port bucket histograms. The resulting feature dimension is $5P + 16$.

4 Models and Metrics

I evaluate logistic regression with class-balanced loss, random forest, and extra trees classifiers. Performance is measured using balanced accuracy (primary) and macro-averaged F1 score (secondary).

5 Evaluation Methodology

I compare two evaluation settings. In the random stratified split, samples are randomly divided into training and test sets while preserving class balance. In the temporal stratified split, samples are sorted by time within each class and the most recent fraction is assigned to the test set, reducing information leakage and capturing temporal drift.

6 Main Results (P = 20)

Table 1 summarizes the main comparison using $P = 20$ packets per sample.

Under the random split, the baseline representation achieves balanced accuracy between 0.37 and 0.39, with the best baseline performance obtained by the random forest (0.388). The extended representation substantially improves performance, achieving a balanced accuracy of 0.439 with logistic regression and approximately 0.405 with tree-based models.

Under the temporal split, performance decreases across all models. The best baseline temporal performance reaches 0.347 (random forest), while the extended representation achieves a higher balanced accuracy of 0.353 with logistic regression. Although overall accuracy is lower under temporal evaluation, the extended features remain more robust than the baseline.

Table 1: Main results for $P = 20$.

Setting	Model	Balanced Accuracy	Macro F1
Baseline / Random	RF	0.388	0.426
Baseline / Random	LogReg	0.387	0.179
Baseline / Random	ExtraTrees	0.374	0.410
Baseline / Temporal	RF	0.347	0.358
Baseline / Temporal	LogReg	0.304	0.163
Baseline / Temporal	ExtraTrees	0.308	0.307
Extended / Random	LogReg	0.439	0.215
Extended / Random	RF	0.404	0.451
Extended / Random	ExtraTrees	0.407	0.441
Extended / Temporal	LogReg	0.353	0.144
Extended / Temporal	ExtraTrees	0.335	0.332
Extended / Temporal	RF	0.295	0.299

7 Ablation Study: Number of Packets per Sample

To study how performance depends on the number of packets per sample, I perform an ablation over $P \in \{5, 10, 20, 40\}$.

For the baseline representation under random splits, balanced accuracy remains relatively flat across all values of P, ranging from approximately 0.38 to 0.39. Increasing P beyond 5 packets does not yield meaningful gains, indicating early saturation of packet-sequence features.

For the extended representation under random splits, balanced accuracy improves substantially, reaching 0.442 at $P = 5$ and peaking at 0.446 for $P = 10$ using logistic regression. Performance remains strong at $P = 20$ (0.439) but declines slightly at $P = 40$ (0.425), suggesting diminishing returns and potential overfitting at higher packet counts.

Under temporal stratified splits, overall performance is lower across all settings. For the baseline representation, balanced accuracy peaks around 0.35 at $P = 5$ and declines slightly for larger P. The extended representation shows improved robustness, achieving its best temporal performance at small to moderate P, with balanced accuracy around 0.35 at $P = 20$. Larger values of P do not improve temporal generalization.

These trends indicate that most discriminative information is contained in the earliest packets of a flow and that lightweight flow-summary features provide consistent gains across packet counts.

8 Discussion

The results demonstrate that augmenting packet-sequence features with simple flow-level summary statistics yields meaningful and consistent improvements in classification performance. Early packets carry most of the useful signal, and temporal evaluation reveals performance degradation that is hidden under random splits, highlighting the importance of leakage-aware evaluation.

9 Limitations and Future Work

Packet direction is inferred using a simple heuristic based on the first observed source, which may introduce noise. The temporal split is performed per class and does not fully simulate online deployment. Future work could examine feature cost versus accuracy trade-offs, cross-domain transferability, explicit drift modeling, and uncertainty calibration.

10 Reproducibility and Artifacts

All experiments are implemented in a clean Jupyter notebook that runs end-to-end with a single execution. The notebook produces `results/leaderboard.csv` for the main comparison and `results/ablation.csv` for the ablation study, which are used directly in this report.