Towards reusable models in traffic classification

Jan Luxemburk and Karel Hynek

FIT at Czech Technical University in Prague and CESNET

Introduction

The machine learning communities, such as those around computer vision or natural language processing, have developed numerous supportive tools. In contrast, the network traffic classification (TC) field falls behind, and the lack of standard datasets and model architectures holds the entire field back. Our goal is to address this issue. We created CESNET Models, a package comprising pre-trained deep learning models tailored for traffic classification. The included models are trained on public datasets for the task of web service classification. Using the new package, researchers and practitioners can skip the model design from scratch and the collection of large datasets but instead focus on fine-tuning and adapting models to their specific needs, thus accelerating the speed of model development.

Obstacles to reusable models in traffic classification

Reusing of ML models in the TC field has been minimal compared to other fields. One of the reasons is that researchers often do not publish source code for their experiments and model definitions, which is more common in other ML fields. There are additional difficulties that complicate model reuse: differences in computer networks and their users, diverse production environments influencing the requirements in terms of false positive rate and inference speed, or general label differences worldwide.

However, we consider the lack of a standard input feature format to be one of the main obstacles to model reusability in the TC field.

Common model inputs

Flow statistics represent common features describing a network connection, such as the number of transmitted bytes and packets in both directions or the time duration of the connection. More features can be related to TCP flags, histograms, or how the connection ended (e.g., with TCP FIN termination).

Packet sequences describe the first `N` packets of the connection, for each packet including its size, direction, inter-packet time, and sometimes the size of the TCP window or the presence of the PUSH flag.

The payload of the first packet is often used in TC literature. However, its usefulness is questionable as complex ML models learn to extract informative strings, which is a task more suitable for pattern-matching algorithms.

Even though these features are standard across TC literature, their parametrization is not. The following table lists the main differences.

Table 1. Differences in model inputs.

Flow statistics	Feature set Packet size - entire vs. after transport headers Computed from the entire connection vs. the first `N` packets Differences in the flow creation process (e.g., timeouts, hashing, collisions)
Packet sequences	Packet size - entire vs. after transport headers Included vs. excluded TCP SYNs, ACKs Sequence length Extra packet features, such as TCP window size or the presence of the PUSH flag
First packet payload	Size and start offset of the payload

Our contribution

We publish PyTorch implementation and pre-trained weights for models that were used in our previous works. We also started implementing architectures developed by other TC research groups. Models are trained on public datasets of TLS and QUIC traffic—namely CESNET-TLS22, CESNET-QUIC22, and CESNET-TLS-Year22. These datasets are available via another package called DataZoo [1], which streamlines the work with large network traffic datasets.

Broader cooperation of research groups is needed to settle on the model input format and start sharing and reusing TC models more. We believe our open-source tools can help this effort.

Available models

In the current version, the package includes these models:

- **mm-CESNET V1** introduced for the classification of TLS web services [2]. It is provided with weights pre-trained on the CESNET-TLS22 dataset.
- **mm-CESNET V2** updated for a QUIC classification task [3], visualized in Figure 1. It is provided with weights pre-trained on CESNET-QUIC22.
- **1d-CNN** processing packet sequences with ResNet blocks, a small model used in [4].
- **mm-CESNET Enhanced**, an updated version of our multi-modal architecture that includes ResNet blocks, an embedding layer for packet sizes, and has fewer parameters while maintaining the same classification performance.

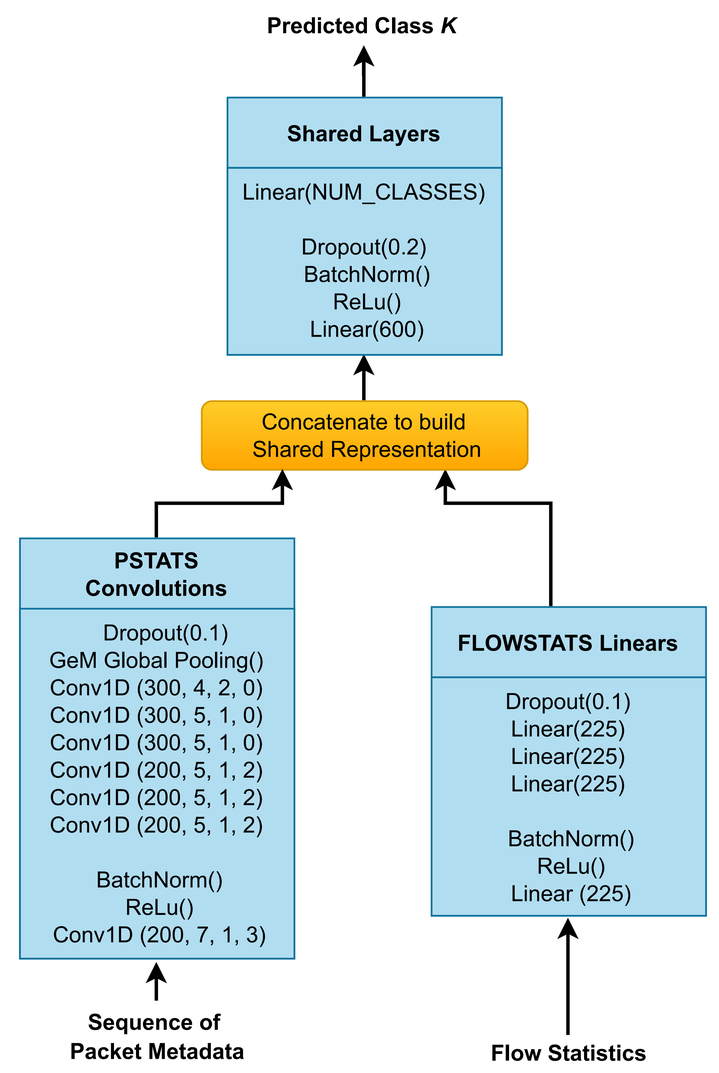

Figure 1. The mm-CESNET V2 model. The parameters represent: `Conv1D(#filters, kernel_size, stride, padding)`, `Linear(#out_features)`, `Dropout(rate)`.

How to use

```
from cesnet_models.models import MM_CESNET_V2_Weights, mm_cesnet_v2

pretrained_weights = MM_CESNET_V2_Weights.CESNET_QUIC22_Week44
model = mm_cesnet_v2(weights=pretrained_weights, model_dir="models/")
transforms = pretrained_weights.transforms
```

Complete examples in Jupyter notebooks are available at the QR code.

References

[1] J. Luxemburk and K. Hynek, "DataZoo: Streamlining traffic classification experiments," in *Proceedings of the 2023 on Explainable and Safety Bounded, Fidelitous, Machine Learning for Networking*, ser. SAFE '23. ACM, 2023, p. 3–7.

[2] J. Luxemburk and T. Čejka, "Fine-grained TLS services classification with reject option," *Computer Networks*, vol. 220, p. 109467, Jan. 2023.

[3] J. Luxemburk, K. Hynek, and T. Čejka, "Encrypted traffic classification: the QUIC case," in *2023 7th Network Traffic Measurement and Analysis Conference (TMA)*, 2023, pp. 1–10.

[4] C. Wang, A. Finamore, P. Michiardi, M. Gallo, and D. Rossi, "Data augmentation for traffic classification," in *Passive and Active Measurement*, 2024, pp. 159–186.