Framework for autonomous improvement of network traffic classification

Jaroslav Pesek **Supervisor: Dominik Soukup**

Faculty of Information Technology CTU in Prague, Laboratory of Network Traffic Monitoring

Problem Statement

Current research in the network security domain intensively uses machine learning (ML) and artificial intelligence (AI) to automate processes and **reveal hidden patterns** in data. These technologies, however, require lots of training **high-quality datasets**. Additionally, network infrastructures continuously evolve and thus **network traffic dynamically changes in time** as well. There is an urgent need to adapt machine learning models, update datasets with the latest samples of annotated network traffic and retrain the models regularly to sustain feasible performance. Thus we need:

1. an updated and informative dataset,
2. a method to continuously selecting network samples which hold the best information value in a observed network stream.

Dataset Aging

- Network traffic changes over time.
- Existing datasets can become obsoleted.
- ML models are getting less efficient without retraining using an updated dataset.

$F_1 = 1.0$		Prediction	
		non-DoH	DoH
Reality	non-DoH	93.2%	0%
	DoH	0%	6.8%

(a) Test data from the same time period as the training data.

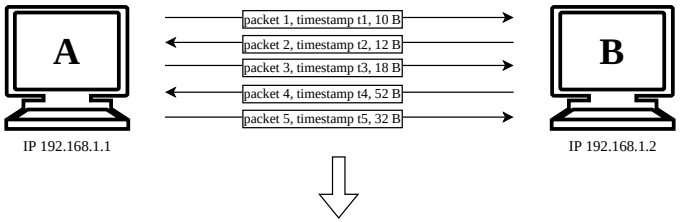
$F_1 = 0.33$		Prediction	
		non-DoH	DoH
Reality	non-DoH	43.7%	0.4%
	DoH	54%	1.9%

(b) Test data captured 6 months after (a)

Figure: DNS over HTTPS classification task. Confusion matrices for two experiments share the same training dataset and model (AdaBoost) but differ in the time of capturing the testing data.

IP Flows

- In network monitoring we employ IP flows instead of packets.
- It is convenient format to describing an enormous volume of network communication.
- IP flow is aggregated from raw packets and grouped by particular connection which is determined by source and destination IP address, ports and time window of *reasonable length*.

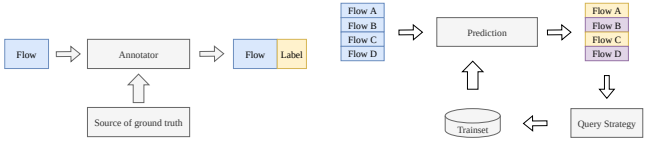


IP src	IP dest	Start	End	Packets src -> dest	Packets dest -> src	Size src -> dest	Size dest -> src
192.168.1.1	192.168.1.2	t1	t5	3	2	60 B	64 B

Figure: Illustration of creating flow from packets.

Active Learning

- The idea is to update the ML model using feedback repeatedly.
- An entity called **Annotator** provides ground truth, i.e. assign the correct label to the IP flow in deterministic way, but annotation is usually more expansive than prediction.
- **Query strategy** select flows to annotate and these flows would update a training dataset.



(a) Annotator assigns a label (class) to input flow and we consider its output as ground truth.

(b) Query strategy selects flows which will be included in train dataset in next iteration.

Figure: Two important parts in active learning.

- Connecting these entities and adding methods to retrieve data and train a new model, we get **active learning loop**.
- We use strictly stream-wise design thus no buffers are used.

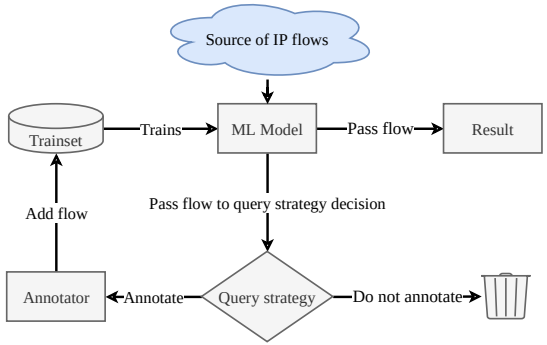


Figure: Scheme of AL Loop. Result could be used for alert, metrics, etc.

Query Strategy

- Query strategy is heuristic approach.
- Usable in stream-wise active learning are
  - **Random** selects the random flow,
  - **Uncertainty of the model** gets samples with highest uncertainty of prediction using probability estimation from the ML model,
  - **Query-By-Committee** uses several models to find the highest disagreement among "committee",
  - **Information Density** selects the sample that is least similar to the samples already selected,
  - **Reinforcement Active Learning** uses feedback: if any strategy selects a sample which were correctly predicted we consider it as superfluous annotation and heuristic gets negative feedback; or positive feedback if vice versa.

Implementation

- Implemented proof-of-concept based on theoretical principles in modular fashion.
- Building blocks of framework are input manager, preprocessor, annotator, query strategy, evaluator for evaluating ML metrics, ML model and postprocessor.

Evaluation

- As a visualization tool Grafana with MariaDB was used.
- The results were assessed offline and online over a longer period of time in CESNET network. Use cases for evaluation were classification of DNS over HTTPS (DoH) and classification of cryptominers.

Table: Comparison of query strategies during the offline experiments with cryptominers detection. All measurements were executed on the same dataset of flows ($n \approx 300k$)

Strategy	Average final F_1	Average query time $[sec/iter.]$
Uncertainty	0.952	0.001
Information Density	0.699	2.592
Query-By-Committee (KL divergence)	0.919	2.820
Random	0.860	0.001

Figure: Comparison of strategies in online setting, visualised accuracy during two days. Random strategy is obviously weak against uncertainty and RAL.

Conclusion and contribution

- Active learning principle is known but its application on network traffic is not properly examined especially in stream-wise setting.
- Developed tool allows for automatic continuous update of both datasets and models.
- Stream-wise query strategies were implemented and experimentally evaluated.
- Some strategies have been shown to be unsuitable for network traffic because they have unreasonably high computational complexity.
- According to the state-of-the-art findings, the random strategy is compelling; but it is easily overcome.
- The developed tool was made open source as ALF (Active Learning Framework), which is one of the outputs of this work in addition to the experimental evaluation of the strategies. ALF is still in active development.

References

[1] Burr Settles.
Active Learning.
Synthesis Lectures on Artificial Intelligence and Machine Learning, 6(1):1–114, June 2012.

[2] Amin Shahraki, Mahmoud Abbasi, Amir Taherkordi, and Anca Delia Jurcut.
Active Learning for Network Traffic Classification: A Technical Study.
IEEE Transactions on Cognitive Communications and Networking, 8(1):422–439, March 2022.
arXiv: 2106.06933.

[3] Václav Bartoš, Tomáš Čejka, and Martin Žádník.
Nemea: Framework for stream-wise analysis of network traffic.
Technical Report 9/2013, CESNET, September 2013.

[4] Indre Zliobaite, Albert Bifet, Geoff Holmes, and Bernhard Pfahringer.
MOA Concept Drift Active Learning Strategies for Streaming Data.
In *Proceedings of the Second Workshop on Applications of Pattern Analysis*, pages 48–55. JMLR Workshop and Conference Proceedings, October 2011.
ISSN: 1938-7228.