Traffic Refinery: Cost-Aware Data Representation for Machine Learning on Network Traffic

Francesco Bronzino[*]
fbronzino@univ-smb.fr
LISTIC, Université Savoie Mont Blanc

Paul Schmitt[*]
pschmitt@isi.edu
USC Information Sciences Institute

Sara Ayoubi
sara.ayoubi@nokia-bell-labs.com
Nokia Bell Labs

Hyojoon Kim
hyojoonk@cs.princeton.edu
Princeton University

Renata Teixeira
renata.teixeira@inria.fr
Inria Paris

Nick Feamster
feamster@uchicago.edu
University of Chicago

ABSTRACT

Network management often relies on machine learning to make predictions about performance and security from network traffic. Often, the representation of the traffic is as important as the choice of the model. The features that the model relies on, and the representation of those features, ultimately determine model accuracy, as well as where and whether the model can be deployed in practice. Thus, the design and evaluation of these models ultimately requires understanding not only model accuracy but also the systems costs associated with deploying the model in an operational network. Towards this goal, this paper develops a new framework and system that enables a joint evaluation of both the conventional notions of machine learning performance (*e.g.*, model accuracy) and the systems-level costs of different representations of network traffic. We highlight these two dimensions for two practical network management tasks, video streaming quality inference and malware detection, to demonstrate the importance of exploring different representations to find the appropriate operating point. We demonstrate the benefit of exploring a range of representations of network traffic and present Traffic Refinery, a proof-of-concept implementation that both monitors network traffic at 10 Gbps and transforms traffic in real time to produce a variety of feature representations for machine learning. Traffic Refinery both highlights this design space and makes it possible to explore different representations for learning, balancing systems costs related to feature extraction and model training against model accuracy.

ACM Reference Format:
Francesco Bronzino, Paul Schmitt, Sara Ayoubi, Hyojoon Kim, Renata Teixeira, and Nick Feamster. 2022. Traffic Refinery: Cost-Aware Data Representation for Machine Learning on Network Traffic. In *Abstract Proceedings of the 2022 ACM SIGMETRICS/IFIP PERFORMANCE Joint International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS/PERFORMANCE '22 Abstracts), June 6–10, 2022, Mumbai, India.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3489048.3522637

1 INTRODUCTION.

Network management tasks commonly rely on the ability to classify traffic by type or identify important events of interest from measured network traffic. Over the past 15 years, machine learning models have become increasingly integral to these tasks [2, 3, 5]. Training a machine learning model from network traffic typically involves extracting a set of features that achieve good model performance, a process that requires domain knowledge to know the features that are most relevant to prediction, as well as how to transform those features in ways that result in separation of classes in the underlying dataset. The typical process begins with data (*e.g.*, a raw traffic trace, summary statistics produced by a measurement system); features are then derived from this underlying data. The collection of features and derived statistics is often referred to as the data *representation* that is used as input to the model. Even for cases where the model itself learns the best representation based on its input (*e.g.*, representation learning or deep learning), the designer of the algorithm must still determine the *initial* representation of the data that is provided to model.

Unfortunately, with existing network traffic measurement systems, the first three steps of this process—collection, cleaning, and feature engineering—are often out of the pipeline designer's control. To date, most network management tasks that rely on machine learning from network traffic have assumed the data to be fixed or given, typically because decisions about measuring, sampling, aggregating, and storing network traffic data are made based on the capabilities (and constraints) of current standards and hardware capabilities (*e.g.*, IPFIX/NetFlow). As a result, a model might be trained with a sampled packet trace or aggregate statistics about network traffic—not necessarily because that data representation would result in an efficient model with good overall performance, but rather because the decision about data collection was made well before any modeling or prediction problems were considered.

A central premise of the work in this paper is motivating the need for additional flexibility and awareness in the first three steps of this pipeline for network management tasks that rely on traffic measurements. On the surface, raw packet traces would seem to be an appealing starting point: Any network operator or researcher knows full well that raw packet traces offer maximum flexibility to explore transformations and representations that result in the best model performance. Yet, unfortunately, capturing raw packet traces often proves to be impractical. In large networks, raw packet traces produce massive amounts of data introducing storage and

[*] Both authors contributed equally to this research.

SIGMETRICS/PERFORMANCE '22 Abstracts, June 6–10, 2022, Mumbai, India
© 2022 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-9141-2/22/06.
https://doi.org/10.1145/3489048.3522637

bandwidth requirements that are often prohibitive. Limiting the duration of a pcap collection (*e.g.*, collecting one day's worth of traces) can reduce data storage requirements, but might negatively affect the accuracy of the produced models as the limited capture may not represent network conditions at other times. Conversely, pcaps collected in a controlled laboratory environment might produce models not directly applicable in practice because operational networks include other traffic characteristics that are hard to capture in a lab environment. Due to these reasons, experiments (and much past work) that demonstrate a model's accuracy turn out to be non-viable in practice because the systems costs of deploying and maintaining the model are prohibitive. An operator may ultimately need to explore costs across state, processing, storage, and latency to understand whether a given pipeline can work in its network.

Evaluation of a machine learning model for network management tasks must also consider the operational costs of deploying that model in practice. Such an evaluation requires exploring not only how data representation and models affect model accuracy, but also the systems costs associated with different representations. Sculley *et al.* refer to these considerations as "technical debt" [4] and identified a number of hidden costs that contribute to building the technical debt of ML-systems, such as: unstable sources of data, underutilized data, use of generic packages, among others. This problem is vast and complex, and this paper does not explore all dimensions of this problem. For example, we do not investigate practical considerations such as model training time, model drift, the energy cost of training, model size, and many other practical considerations. In this regard, this paper scratches the surface of systemization costs that applies to machine learning on network traffic, which we believe deserves more consideration before machine learning can be more widely deployed in operational networks.

2 CONTRIBUTIONS.

To lay the groundwork for more research that considers these costs, we develop and publicly release a systematic approach to explore the relationship between different data representations for network traffic and (1) the resulting model performance as well as (2) their associated costs. We present Traffic Refinery, a proof-of-concept reference system implementation designed to explore network data representations and evaluate the systems-related costs of these representations. To facilitate exploration, Traffic Refinery implements a processing pipeline that performs passive traffic monitoring and in-network feature transformations at traffic rates of up to 10 Gbps in software. The pipeline supports capture and real-time transformation into a variety of common feature representations for network traffic; we have designed and exposed an API so that Traffic Refinery can be extended to define new representations, as well. In addition to facilitating the transformations themselves, Traffic Refinery performs profiling to quantify system costs, such as state and compute, for each transformation, to allow researchers and operators to evaluate not only the accuracy of a given model but the associated systems costs of the resulting representation.

We use Traffic Refinery to demonstrate the value of jointly exploring data representations for modeling and their associated costs for two supervised learning problems in networking: video quality inference from encrypted traffic and malware detection. We study two questions:

- *How does the cost of feature representation vary with network speeds?* We use Traffic Refinery to evaluate the cost of performing different transformations on traffic in real-time in deployed networks across three cost metrics that directly affect the ability to collect features from network traffic: in-use memory (*i.e.*, state), per packet processing (*i.e.*, compute), and data volume generated (*i.e.*, storage). We show that for the video quality inference models, state and storage requirements out-pace processing requirements as traffic rates increase. Conversely, processing and storage costs dominate the systems costs for the malware detection. These results suggest that fine-grained cost analysis can lead to different choices for traffic representation depending on different model performance requirements and network environments.
- *Can systems costs be reduced without affecting model accuracy?* We show that different data transformations allow systems designers to make meaningful decisions involving systems costs and model performance. For example, we find that state requirements can be significantly reduced for both problems without affecting model performance, providing important opportunities for in-network reduction and aggregation.

3 CONCLUSIONS.

While it is well-known that *in general* different data representations can both affect model accuracy and introduce variable systems costs, network research has left this area relatively under-explored. Our investigation both constitutes an important re-assessment of previous results and lays the groundwork for new directions in applying machine learning to network traffic modeling and prediction problems. From a scientific perspective, our work explores the robustness of previously published results. From a deployment standpoint, our results also speak to systems-level deployment considerations, and how those considerations might ultimately affect these models in practice, something that has been often overlooked in previous work. Looking ahead, we believe that incorporating these types of deployment costs as a primary model evaluation metric should act as a rubric for evaluating models that rely on machine learning for prediction and inference from network traffic. We release the source code of Traffic Refinery [1] as a reference design so that others can build upon it.

REFERENCES

[1] 2021. Traffic Refinery. https://github.com/traffic-refinery/traffic-refinery.

[2] Raouf Boutaba et al. 2018. A comprehensive survey on machine learning for networking: evolution, applications and research opportunities. *Journal of Internet Services and Applications* 9, 1 (2018), 16.

[3] Thuy TT Nguyen and Grenville Armitage. 2008. A survey of techniques for internet traffic classification using machine learning. *IEEE communications surveys & tutorials* 10, 4 (2008), 56–76.

[4] David Sculley et al. 2015. Hidden technical debt in machine learning systems. In *Advances in neural information processing systems*. 2503–2511.

[5] Jayveer Singh and Manisha J Nene. 2013. A survey on machine learning techniques for intrusion detection systems. *International Journal of Advanced Research in Computer and Communication Engineering* 2, 11 (2013), 4349–4355.