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

Sarthak | 2020CS10379

June 2024

Problem Statement

The paper addresses the challenge of managing network traffic using machine learning models, with a focus on the cost and performance implications of different data representations.

Motivation

- Network management relies heavily on machine learning to analyze traffic for performance and security
- The features that the model relies on, and the representation of those features, ultimately determine model accuracy
- Existing approaches often neglect the systems cost implications of data representation
- Existing network traffic measurement capabilities capture either flow-level statistics or perform fixed transformations on packet captures

Key Idea

The key idea is to create a framework that jointly evaluates machine learning performance and the system costs associated with different representations of network traffic enabling a balanced approach to optimize both.

Framework & Methodology

- Uses state-of-the-art packet capture libraries like PF_RING to access packets with minimal overhead
- Caches the map of remote IP addresses to services accessed by users. The categorizations are based on DNS queries and IP prefixes.
- Employs a pool of worker processes to handle traffic in parallel, leveraging multicore CPU architectures. Flow clustering ensures that packets from the same flow are processed by the same worker, reducing cross-core communication.
- The system aggregates and exports flow features at regular intervals, specified in the configuration file
- Collected data is temporarily stored and periodically uploaded to a remote location for further analysis and model input

Contributions

- Developed a new framework 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
- Implemented Traffic Refinery, which monitors network traffic at 10 Gbps and transforms them in real time to produce a variety of feature representations
- Showed the need for additional flexibility and awareness in the Data Collection, Cleaning and feature engineering for network management tasks that rely on traffic measurements
- Designed and exposed an API so that Traffic Refinery can be extended to define new representations, released evaluations and Traffic Refinery as open source software

Strengths

- Supports capture and real-time transformation into a variety of common feature representations for network traffic
- Provides a holistic approach to evaluate both performance and costs

Weaknesses & Limitations

- Focuses mainly on video quality inference and malware detection, which might not generalize to all network management tasks
- Does not investigate practical considerations such as model training time, model drift, the energy cost of training, model size etc.
- Implementing the framework in live networks might introduce operational overhead due to computations for various feature representations