Paper Summary
Practical GAN-based Synthetic IP Header Trace Generation using NetShare

Sarthak | 2020CS10379

June 2024

Problem Statement

The paper addresses the challenge of generating synthetic packet and flow header traces for network tasks (such as telemetry, anomaly detection, and provisioning) which are often unavailable due to privacy concerns.

Motivation

- Data holders who hold traces are usually unwilling to share raw traces
- Techniques based on tabular data GANs, fail to capture essential correlations across header fields with large ranges of values
- Differentially-private learning approaches often result in poor fidelity for networking datasets

Key Idea

To leverage Generative Adversarial Networks (GANs) to create synthetic packet and flow header traces that satisfies fidelity metrics specified by domain experts and downstream applications, focusing on IPv4 header 5-tuple fields.

Framework & Methodology

- Reformulate header trace generation as a time series generation problem of generating flow records for the entire trace rather than a per-epoch tabular approach
- Data from different measurement epochs are merged into one giant trace, divided by flows and header fields are encoded based on domain knowledge

- Flow traces are then evenly sliced into fixed-time chunks with explicit flow tags and each chunk is trained using DoppelGANger (Time series GAN)
- If Differential Privacy (DP) is not required, the model from the first chunk is used as a pre-trained model. For DP, a model pre-trained on public data is fine-tuned using DP-SGD
- Generated fields are mapped back to their natural representations (e.g., using nearest-neighbor search for IP2Vec embeddings)
- The data is converted to PCAP or NetFlow datasets by merging packets or NetFlow records according to timestamps
- Implemented using TensorFlow 1.15, with DP-SGD implemented using tensorflow-privacy 0.5.0

Contributions

- Developed NetShare, an end-to-end system for generating synthetic network traces using GANs
- Incorporated differentially-private model training and combined public data with private data to enhance privacy-fidelity tradeoffs
- Open sourced the codebase for public access

Strengths

- Achieves significant improvements in accuracy over existing methods with 46% more accuracy than baseline approaches across all distributional metrics and traces
- Efficient training process that handles large datasets effectively and enhances privacy while maintaining high fidelity of generated traces

Weaknesses & Limitations

- Does not capture stateful session semantics (e.g., TCP sessions), application layer protocol semantics (e.g., HTTP headers), packet payloads, or fine-grained temdoes not currently generate payloads, which are much higher-dimensional than the headers generated in this workporal properties (e.g., distribution of inter-arrival times of packets)
- Does not generate payloads, which are much higher-dimensional than the headers generated in this work
- The model may memorize some fields without memorizing others, and it is unclear how to measure packet closeness since fields have different units

- NetShare should be used with care to ensure that the privacy requirements of the data holder are accounted for as gener- ative models can memorize and leak individual records