Paper Summary
NetDiffusion: Network Data Augmentation Through Protocol-Constrained Traffic Generation

Sarthak | 2020CS10379

June 2024

Problem Statement

The paper addresses the challenge of generating high-resolution synthetic network traffic traces using diffusion models and to produce synthetic traffic that is statistically similar to real network traffic and conforms to protocol specifications.

Motivation

- Modern networks rely heavily on machine learning for various management tasks, but the availability of labeled network datasets is hindered due to privacy and maintenance concerns
- Public datasets rarely receive updates, making them static and unable to reflect evolving network behaviors
- Current synthetic trace generation methods, typically produce only aggregated flow statistics or a few selected packet attributes
- Hence, exists a need for traffic generation satisying :
  - statistical similarity with real data
  - satisfactory classification accuracy when synthetic statistical attributes are used to augment existing datasets

Key Idea

To use a finely-tuned, controlled variant of a Stable Diffusion model to generate synthetic network traffic that is high fidelity and conforms to protocol specifications. This can be done by converting Network traces to image representation, training a diffusion model upon it and then converting the generated representation back to packet data.

Framework & Methodology

- Network traffic captures (pcaps) are first encoded using the nPrint tool which converts network traffic into standardized bits, where each bit corresponds to a packet header field
- A bit value of 1 indicates the presence of a bit in the packet header, 0 indicates its absence, and -1 represents a missing header bit
- The encoded pcaps are then converted into a matrix format. This matrix is interpreted as an image where each row represents a packet in the network traffic flow. The colors used in the image are green for a set bit (1), red for an unset bit (0), and gray for a vacant bit (-1)
- Diffusion models generate data by modeling the process of noise removal from noisy data. A neural network is trained to predict and remove noise sequentially added to real data, transforming an initial noise vector into a data point drawn from the desired distribution
- A text-to-image diffusion model is fine tuned to generate synthetic image representations based on packet capture-converted images
- Since the image format preserves the essential sequential relationships and complexity of the data, the encoded bits are be directly interpreted back into the packet header fields to reconstruct the original network traffic
- After generating the synthetic network traffic, domain-specific heuristics are applied to modify the traffic details to ensure adherence to protocol rules, such as sequence numbers,checksums etc.

Contributions

- Developed NetDiffusion which converts raw packet captures into images and uses fine-tuned diffusion models to generate synthetic traffic with high resemblance to real traffic and ensures fidelity to protocol specifications and semantic correctness of generated fields
- Conducted a case study evaluation on a curated traffic classification dataset by integrating NetDiffusion-generated network traffic into the real dataset at varying proportions and observed a general increase in accuracy compared to the state-of-the-art generation methods
- The generated network traffic can be converted into raw packet captures suitable for traditional network analysis (non ML) and testing tasks and is compatible with traditional network analysis tools like Wireshark and tcpreplay

Strengths

- Utilizes diffusion models, which are more stable and capture intricate patterns in network traffic better than GANs
- Can generate structured data conforming to specific network properties
- Applicable to a wide range of network analysis and testing tasks, not just ML

Weaknesses & Limitations

- Does not generate application-layer protocol semantics, which are crucial for some network analysis task
- Focuses on generating header traces and does not include payload generation, which can be important for certain security applications, such as deep packet inspection
- Current approach manages protocol rule-compliance post-generation, due to the complexity of inter-dependent constraints during the diffusion process
- Need to refine diffusion models to directly learn and generate time series, providing a more nuanced approach to inter-packet time dependencies
- Currently limited to 1,024 packets per flow sample