(Reproducing:) Practical GAN-based synthetic IP header trace generation using **NetShare (SIGCOMM'22)**

Akanksha Cheeti, Annus Zulfiqar, Ashwin Nambiar,
Hasan Amin, Murayyiam Parvez, Syed Abubaker

Why should we generate traces?

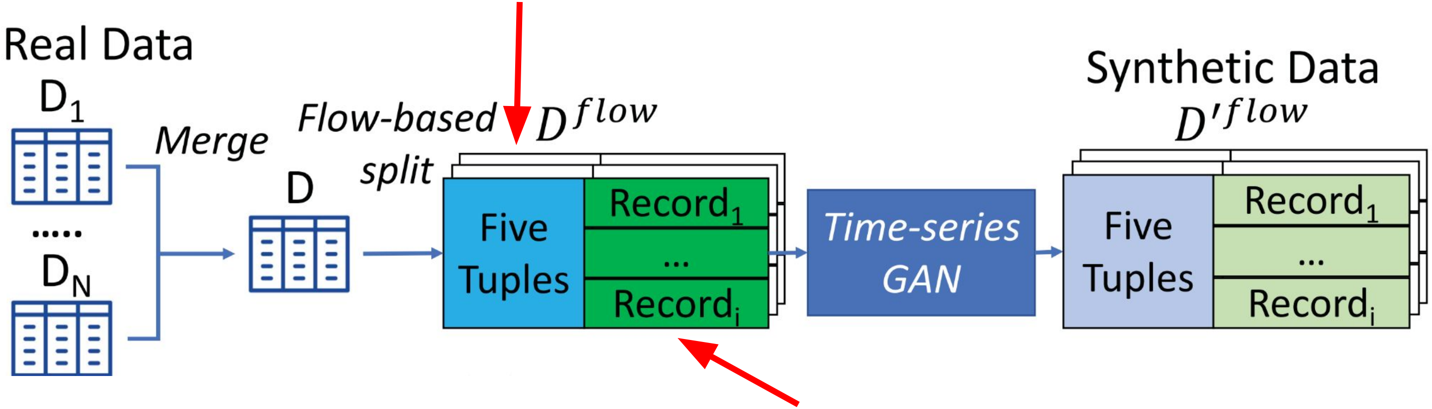
- Scarcity of ***shareable*** networking data
- Many downstream applications rely on authentic traces
  - Data-driven network monitoring algorithms
  - Anomaly detection and fingerprinting
  - Testing and benchmarking new hardware and software etc.
- An obvious solution: **Collect** shareable data
  - Simulations, large testbeds, etc.
  - But costly, time-consuming, difficult…
- An alternate: ***Generate*** shareable data!
  - How? Generative models

Strawman Approach: Tabular GANs

Can’t generate high-fidelity traces and don’t perform well on downstream applications

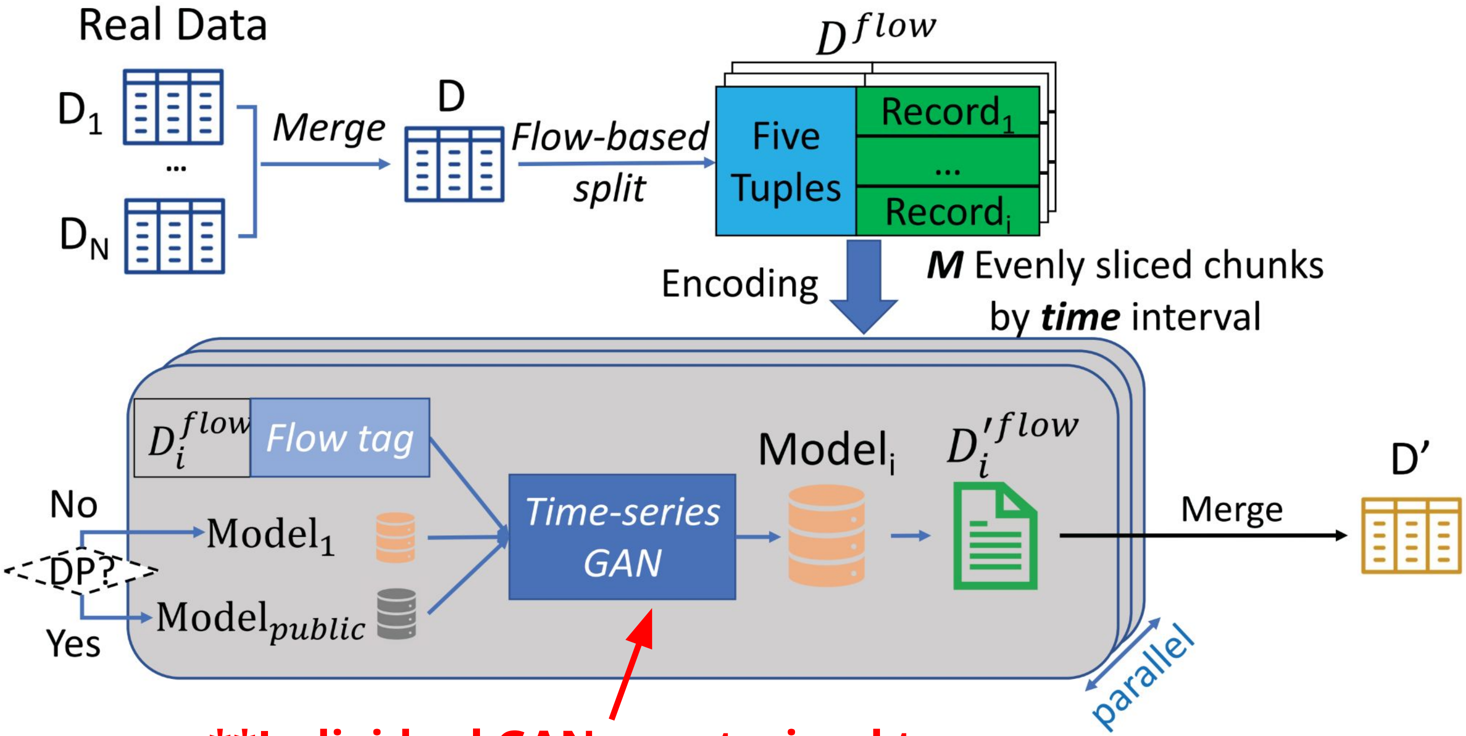
Enter NetShare: The DoppleGANger Approach

****Flow-based Split: GANs will be trained to generate flows instead of arbitrary sequences of packets**



****Time-based Split: Further split each flow into time duration-based chunks (say 10 parts of the flow)**

Enter NetShare: The DoppleGANger Approach



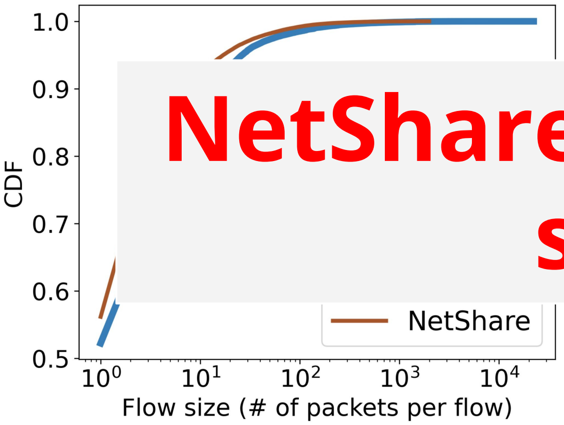
****Individual GANs are trained to generate individual chunks of flows**

Evaluations

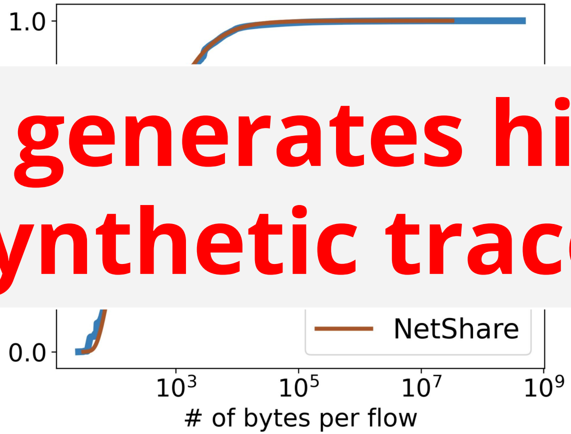
Fidelity of Generated Traces

****High correlation across several packet properties**

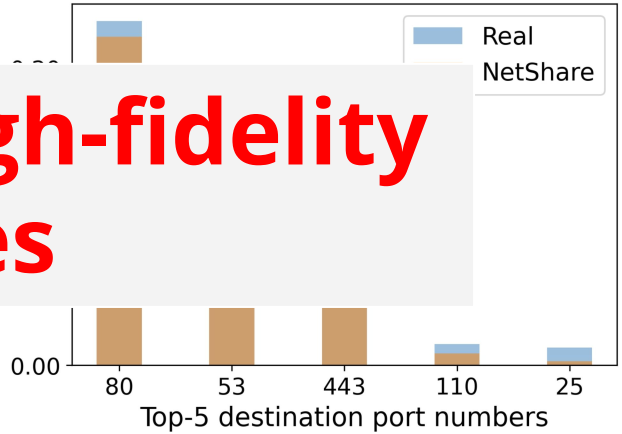
Flow Size



Flow Volume



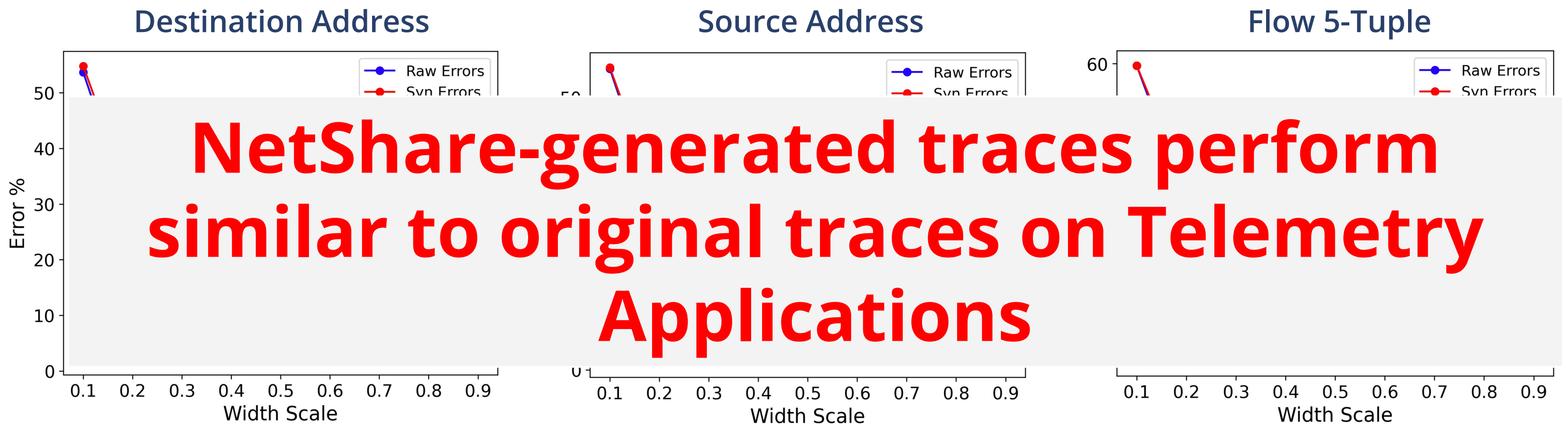
Packet Fields



NetShare generates high-fidelity synthetic traces

Downstream Applications

Count-Min Sketch Top-10% Heavy-Hitters Detection Error Rate



NetShare-generated traces perform similar to original traces on Telemetry Applications

Downstream Applications

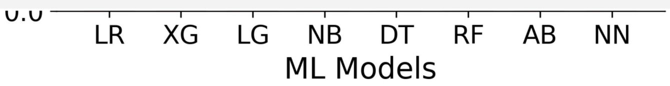
****High correlation of performance rank order**

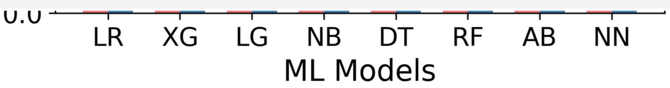
Botnet Detection

Anomaly Detection using Packet Fields

NetShare-generated traces perform with high-accuracy on Anomaly Detection Tasks

NetShare preserves the relative rank-order of algorithm performance





Conclusions

Time Series GANs effective at flow-level for trace generation

Use Time Series GANs to produce specific chunks of flows

High-fidelity synthetic traces that perform like raw traces on many downstream applications

Privacy can be preserved using Differentially Private – SGD

Thank You!

Problem

- Scarcity of *shareable* networking data
- *Why we need such networking data?* To develop (data-driven) network monitoring algorithms, for anomaly detection and fingerprinting, for testing and benchmarking new hardware and software etc.
- An obvious solution: *Collect* shareable data. But costly, time-consuming, difficult…
- An alternate: *Generate* shareable data! How? Generative models

Prequel (or what the paper didn't really do)

Prequel (or what the paper didn't really do)

Generative Models

- Generative models approximate underlying data distribution given access to finite set of samples from it.
- *Deep* generative models use deep learning for the purpose.
- *GANs* – minimax game between generator and discriminator.

$$\min_G \max_D \mathbb{E}_{\mathbf{x}\sim P_{data}(\mathbf{x})} \log[\mathrm{D}(\mathbf{x})] + \mathbb{E}_{\mathbf{z}\sim P_{\mathbf{z}}(\mathbf{z})}[\log(1 - \mathrm{D}(\mathrm{G}(\mathbf{z}))]$$

- *VAEs* – latent variable model involving encoder and decoder.

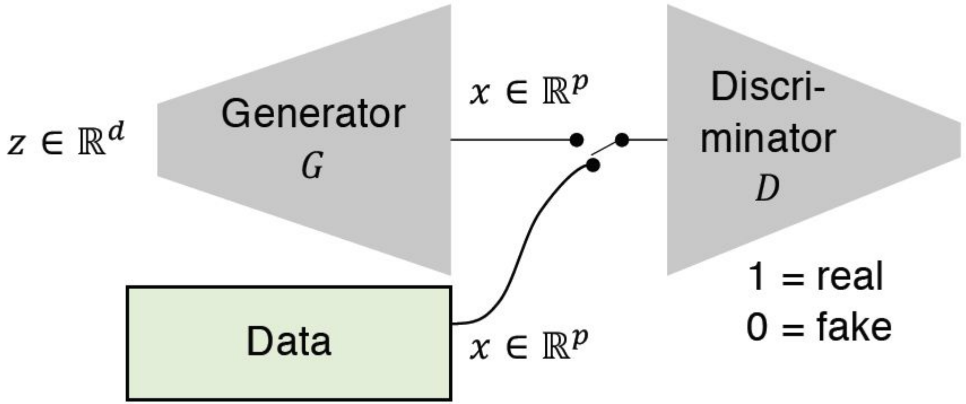
$$\min_\theta \sum_{\mathrm{x}\in\mathcal{D}} \max_\lambda \mathbb{E}_{q_\lambda(\mathrm{z}|\mathrm{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - D_{KL}(q_\lambda(\mathbf{z}|\mathbf{x})||(p_\theta(\mathbf{z}))$$

Prequel (or what the paper didn't really do)

GAN

- Minimax game between generator and discriminator.

$$\min_G \max_D \mathbb{E}_{\mathbf{x}\sim P_{data}(\mathbf{x})}\log[\mathrm{D}(\mathbf{x})] + \mathbb{E}_{\mathbf{z}\sim P_{\mathbf{z}}(\mathbf{z})}[\log(1-\mathrm{D}(\mathrm{G}(\mathbf{z}))]$$

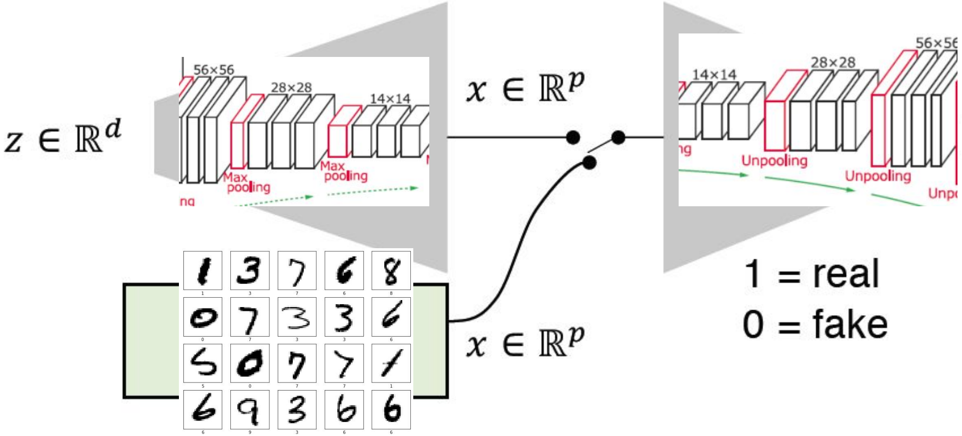


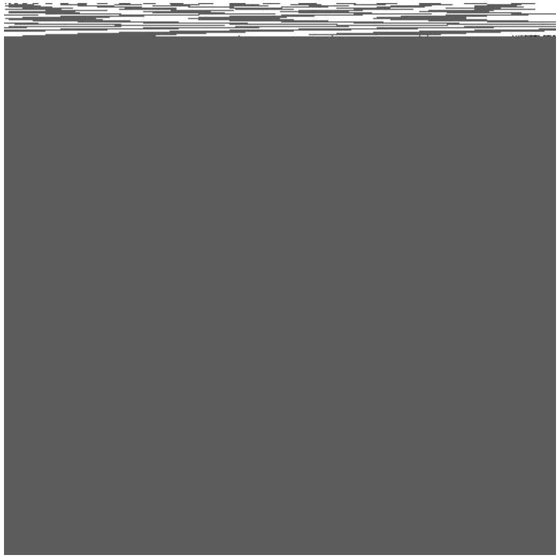
Prequel (or what the paper didn’t really do)

GAN

- Minimax game between generator and discriminator.

$$\min_{G}\max_{D}\ \mathbb{E}_{\mathbf{x}\sim P_{data}(\mathbf{x})}\log[\mathrm{D}(\mathbf{x})] + \mathbb{E}_{\mathbf{z}\sim P_{\mathbf{z}}(\mathbf{z})}[\log(1-\mathrm{D}(\mathrm{G}(\mathbf{z})))]$$



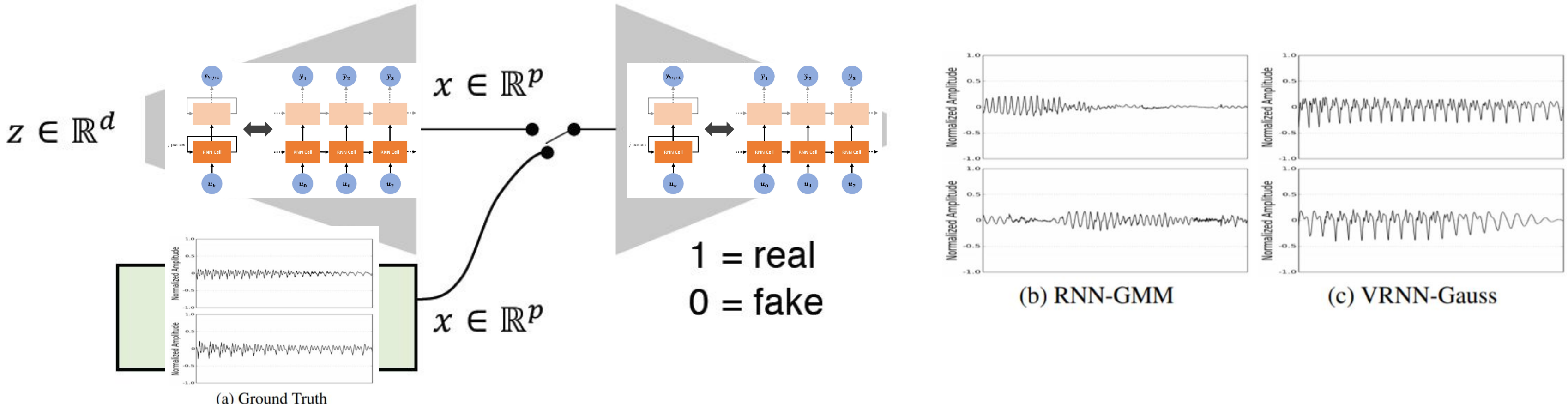


Prequel (or what the paper didn't really do)

GAN – for Time Series Data

- Minimax game between generator and discriminator.

$$\min_G \max_D \mathbb{E}_{\mathbf{x}\sim P_{data}(\mathbf{x})} \log[D(\mathbf{x})] + \mathbb{E}_{\mathbf{z}\sim P_{\mathbf{z}}(\mathbf{z})}[\log(1 - D(G(\mathbf{z}))]$$

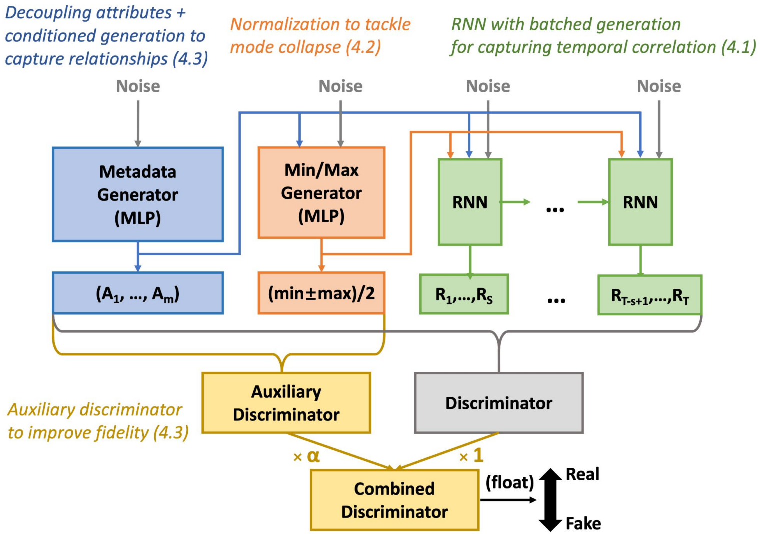


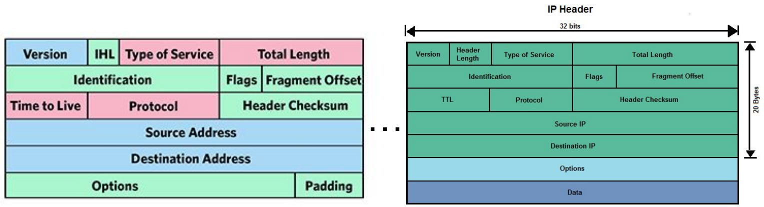
Prequel (or what the paper didn't really do)

GAN – for Time Series Data in Networking

- *DoppelGANger* [IMC '20]

- Domain-specific modifications to capture long-term temporal correlations between measurements and metadata





Prequel (or what the paper didn’t really do)

GAN – for Time Series Data in Networking

- *DoppelGANger* [IMC ‘20]
- Domain-specific modifications to capture long-term temporal correlations between measurements and metadata
- Generates realistic data but privacy concerns remain

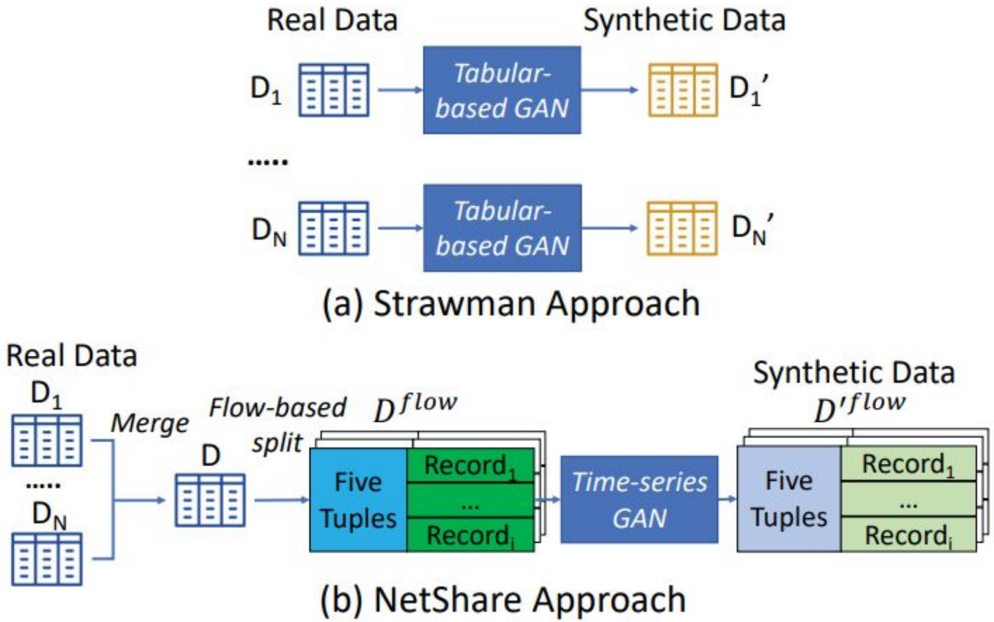
...

Research Gaps

- Not good (enough) **fidelity**
- Not good (enough) **scalability**
- Poor **privacy**
- Solution: NetShare (or DoppelGANger 2.0)

NetShare

1. Header trace generation as flow-based time series generation

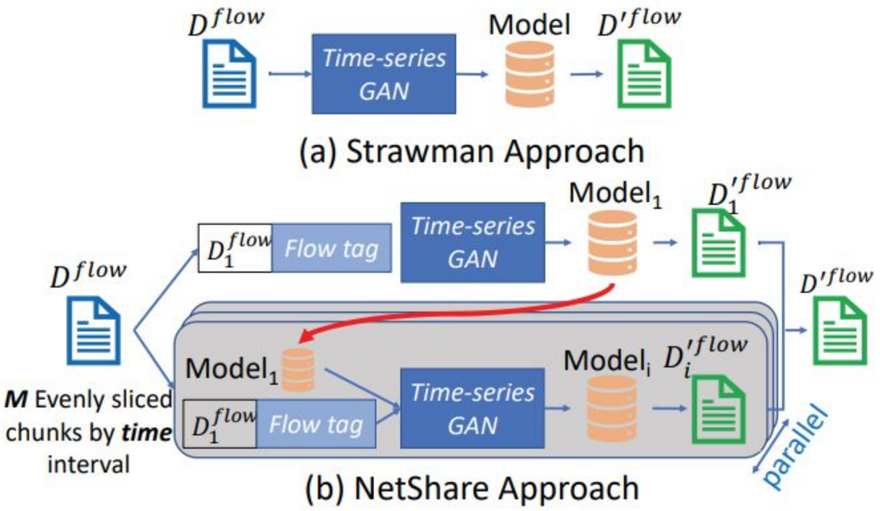


NetShare

1. Header trace generation as flow-based time series generation
2. Data preprocessing using bitwise encoding and IP2Vec

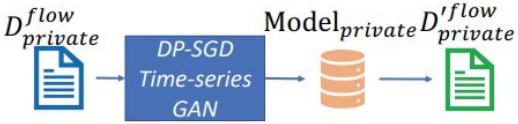
NetShare

1. Header trace generation as flow-based time series generation
2. Data preprocessing using bitwise encoding and IP2Vec
3. Smart parallel training using time-spaced chunks

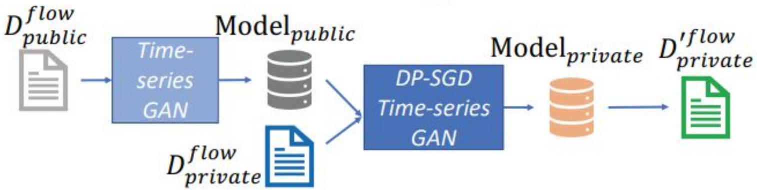


NetShare

1. Header trace generation as flow-based time series generation
2. Data preprocessing using bitwise encoding and IP2Vec
3. Smart parallel training using time-spaced chunks
4. Optimizing model pre-trained on public dataset using DP-SGD on private dataset



(a) Strawman Approach



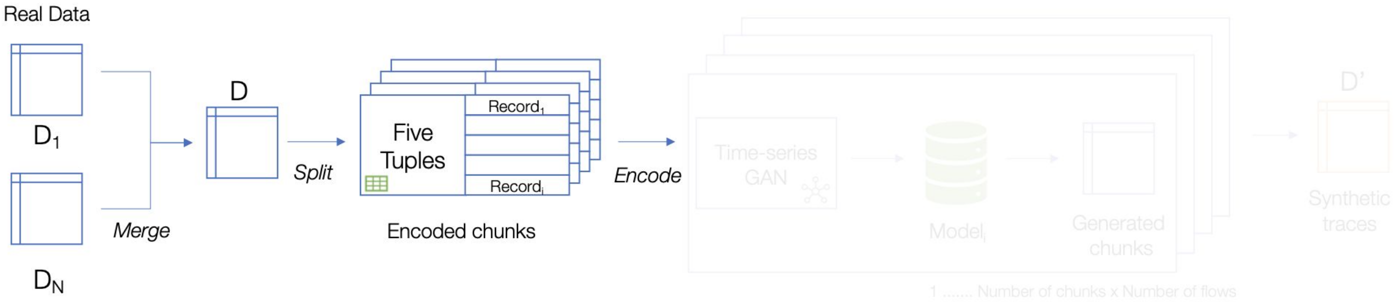
(b) NetShare Approach

NetShare

1. [Header trace generation as flow-based time series generation with IP2Vec data encoding] Distinct data preprocessing for improving **fidelity**
2. [Smart parallel training using time-spaced chunks] Distinct parallelization for improving **scalability**
3. [Optimizing model pre-trained on public dataset using DP-SGD on private dataset] Distinct training routine for improving (differential) **privacy**

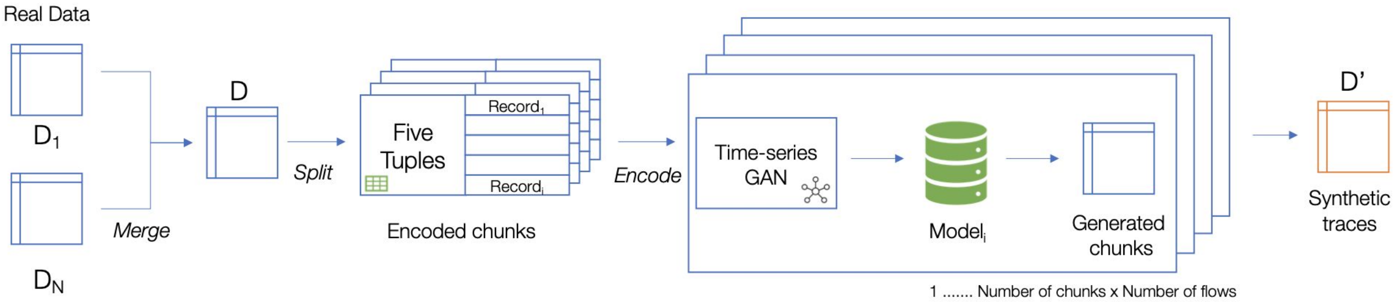
NetShare

1. Distinct data preprocessing for improving **fidelity**



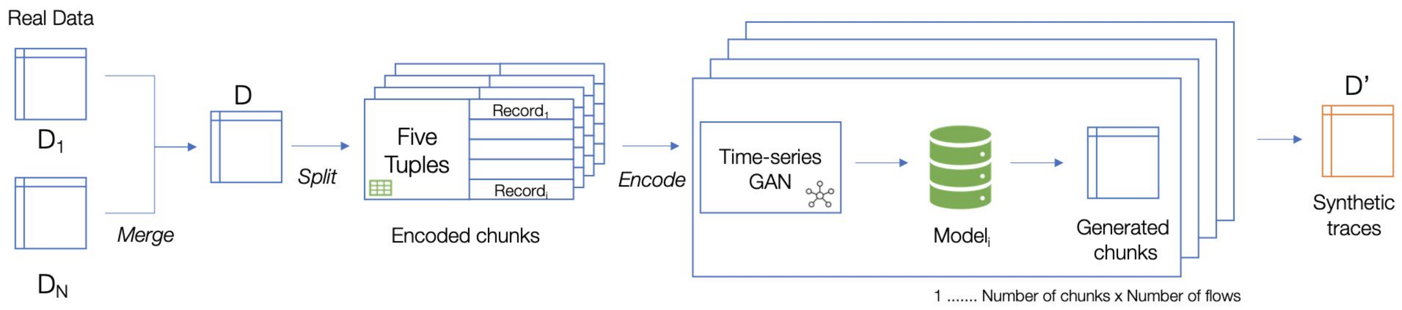
NetShare

1. Distinct data preprocessing for improving **fidelity**
2. Distinct parallelization for improving **scalability**



NetShare

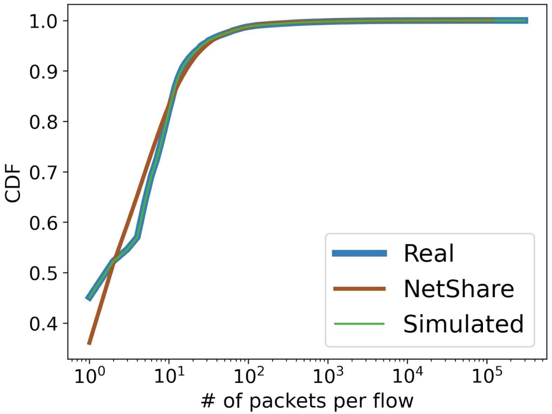
1. Distinct data preprocessing for improving **fidelity**
2. Distinct parallelization for improving **scalability**
3. Distinct training routine for improving (differential) **privacy**

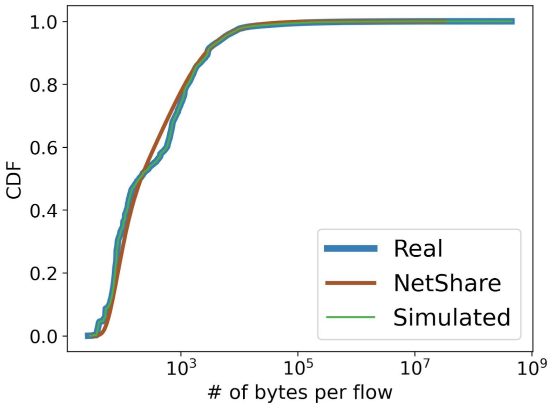


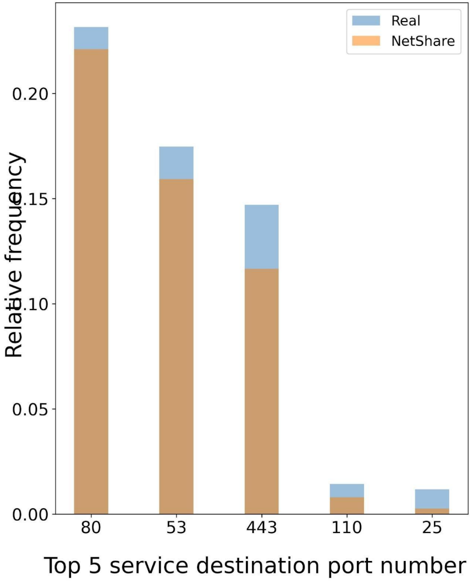
But Does It Work?

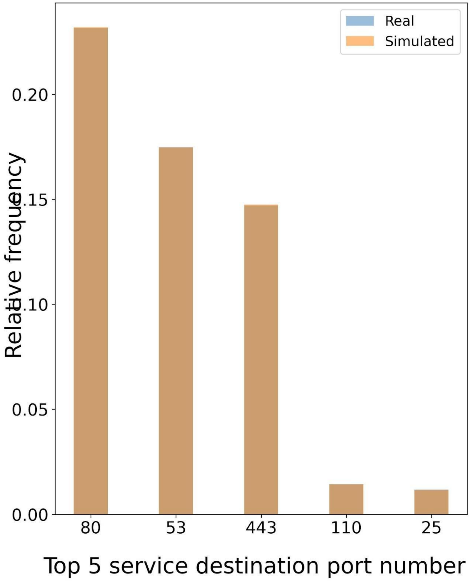
- Yes
- Reproduced and extended multiple experiments that demonstrate:
    - High **fidelity** between real and synthetic data, as measured by difference in distribution as captured by Jensen-Shannon Divergence and Earth Mover's Distance
    - Decent **privacy**, as measured under the differential privacy framework)
    - (Typically) minor drop in **performance on downstream tasks**, as measured by change in error on header-based anomaly detection and sketch-based network telemetry tasks

High Fidelity



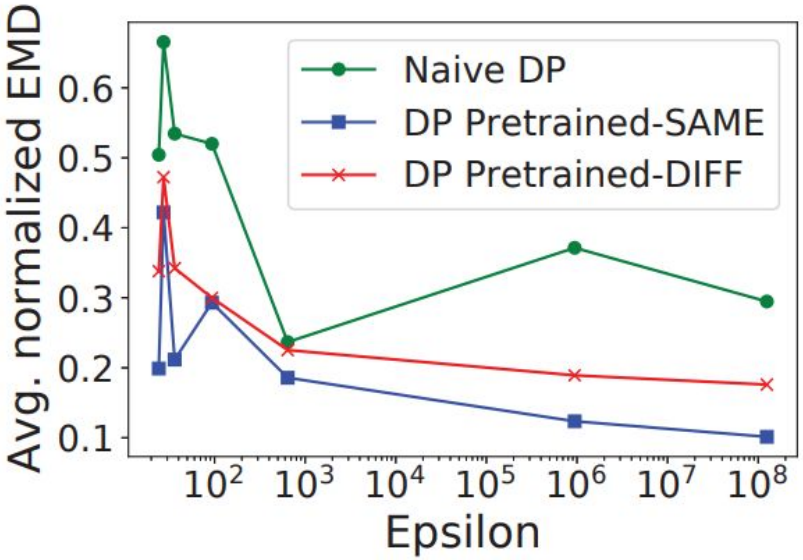




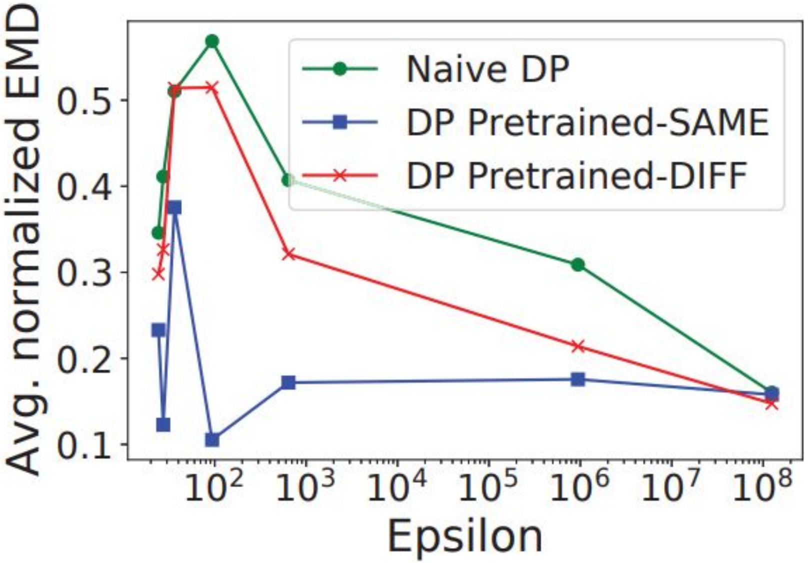


Decent Differential Privacy

- $M(z; D) \leq e^{\epsilon} M(z; D') + \delta$
- Smaller values of ϵ and δ give more privacy

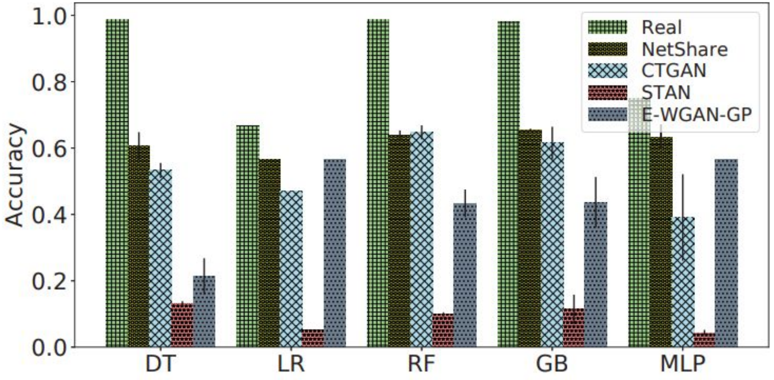


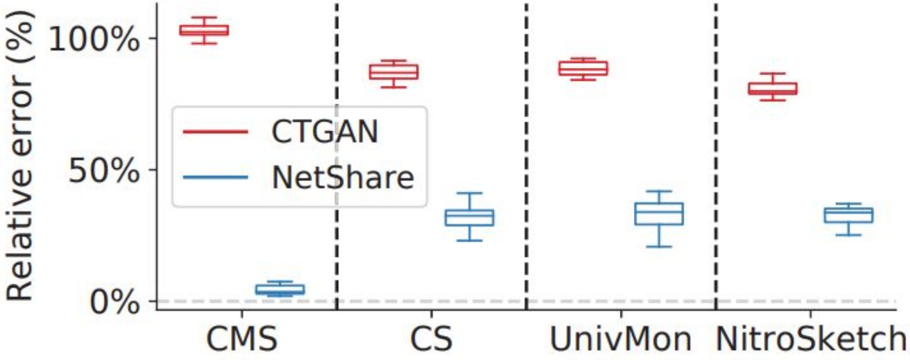
NetFlow (UGR16)



PCAP (CAIDA)

Good Performance on Downstream Tasks





(a) CAIDA (HH: Destination IP)

Takeaways

- NetShare: A machine learning solution for synthetic header generation with high fidelity while preserving reasonable privacy
- Modifies a recent domain-specific GAN-variant, with particular focus on more suitable data engineering
- Very methodical updates? No.

Takeaways

- NetShare: A machine learning solution for synthetic header generation with high fidelity while preserving reasonable privacy
- Modifies a recent domain-specific GAN-variant, with particular focus on more suitable data engineering
- Very methodical updates? No. But do they work? Yes.

Takeaways

- NetShare: A machine learning solution for synthetic header generation with high fidelity while preserving reasonable privacy
- Modifies a recent domain-specific GAN-variant, with particular focus on more suitable data engineering
- Very methodical updates? No. But do they work? Yes. And can they be improved? Likely yes, with plenty to try out (GAN with transformer-based generator and discriminator, diffusion-based generative model, even better parallelization etc.)