Enabling Privacy-preserving Multidimensional Network Telemetry with Autoencoders

Yajie Zhou*, Jason Li†, Gianluca Stringhini‡, Ayse K. Coskun§ and Zaoxing Liu¶
Electrical and Computer Engineering, Boston University
Email: *leszhou, †jli3469, ‡gian, §acoskun, ¶zaoxing@bu.edu

***Abstract*—Network telemetry systems are essential for monitoring network traffic and informing management decisions. However, increasing privacy concerns make user data access and analysis challenging for operators. We introduce PrvTel, a privacy-preserving telemetry system that uses an AutoEncoder model to encode user traffic data and preserve telemetry query ability with differential privacy guarantees. PrvTel features a lightweight model to be stored, facilitates quick training, and executes queries with minimal delay.**

I. Introduction

At the core of network management, telemetry plays a crucial role in understanding what is happening in the network and informing management decisions [1]. However, due to increasing privacy concerns among Internet users, accessing telemetry data and performing query analysis based on user traffic is becoming more challenging. Ideally, telemetry systems should protect user privacy while retaining the ability to estimate various traffic metrics for operation and management.

To this end, we aim to build a telemetry framework that (1) prevents leakage of individual user privacy, (2) preserves the fidelity of query results, and (3) reduces data storage cost for providers. Existing telemetry solutions employ approximate data structures (e.g., sketches [2]) for a provable trade-off between resource efficiency and measurement accuracy. However, it is difficult to leverage sketches to answer all potential telemetry queries [3]. Furthermore, sketch-based systems do not ensure privacy protection for original user data. Other related studies propose using Generative Adversarial Networks (GANs) to generate synthetic network traces for analysis [4]. Although they can offer privacy-fidelity evaluations, they are not well-suited for compressing large volumes of telemetry data and incur large training efforts from varying workloads [4]. Training GAN models is time-consuming and requires intricate hyperparameter tuning to achieve the desired privacy-fidelity trade-off.

In this paper, we introduce PrvTel, a privacy-preserving framework specifically tailored for telemetry data. The framework employs a state-of-the-art AutoEncoder model for original data transformation, protects user privacy against membership inference attacks, and does not compromise high query accuracy. Furthermore, PrvTel features a lightweight model that can (1) save only the encoded features and model, significantly reducing storage space compared to retaining all original data (e.g., 20GB original data after zip compression vs. 200MB encoded data plus model), (2) facilitate quick and straightforward training with our ”fine-tune” module, eliminating the need to train a new complex model for each dataset, and (3) execute queries with minimal delay due to rapid inference during the recovery of transformed data.

In designing PrvTel, we experiment with incorporating various levels of Laplace noise in the original data for training to safeguard user's membership inference privacy. We observe that the AutoEncoder model is capable of capturing the distribution of high-dimensional telemetry data and answering queries within a reasonable error rate (± 0.05).

II. Design

PrvTel consists of three key modules: “Data Processing” to prepare high-dimensional telemetry data for input, “Model Training” to train an AutoEncoder model for each telemetry dataset while incorporating noise to protect user membership inference privacy and maintain query accuracy, and ”Model Fine-Tuning” to rapidly select an optimal set of parameters for newly acquired telemetry datasets.

Data Processing: Real-world telemetry traces are collected from various devices and applications, resulting in no fixed format for the data. We differentiate data into “categorical features” (e.g., load interval) and “continuous features” (e.g., packet rate). Categorical features are transformed into continuous distributions using a transformer, which computes floating-point representatives. Continuous features are transformed into Gaussian variables through Bayesian Gaussian Mixture Modeling. The combined transformed data from both feature types are used as input for the model.

Model Training: The AutoEncoder model is chosen for training due to its natural encoder-decoder split structure. Using input data obtained from data processing, we first introduce Laplace noise to provide controlled randomness to the data, making it more challenging for an attacker to discern individual data points' true values or reverse-engineer the original data. The precise scale of noise depends on the optimal privacy-fidelity tradeoff for the given dataset. With noise-added data, we train the autoencoder model to fit the distribution and allow the model training to introduce additional noise to protect privacy. After training, only latent features and decoder layers are saved on local or cloud computing resources. When processing incoming telemetry queries, the decoder model decompresses the latent features and operates on the noise-added transformed data.

TABLE I: Accuracy with KS-test score values

Features	VAE	GAN	GMM
active-routes-count	0.965	0.877	0.647
paths-count	0.860	0.389	0.746
protocol-route-memory	0.858	0.410	0.728
routes-counts	0.978	0.640	0.677
global__established-neighbors-count-total	0.969	0.932	0.721
perf-stats__global__config-items-processed	0.971	0.358	0.778
perf-stats__vrf__inbound-update-messages	0.944	0.712	0.944
vrf__update-messages-received	0.953	0.712	0.944

Fine-Tuning on New Dataset: Considering the generalizability and deployment cost of our machine learning (ML)-based framework on new telemetry datasets, we propose a fine-tuning module to swiftly adapt the autoencoder model for unseen datasets. Inspired by state-of-the-art domain adaptation and hyperparameter tuning methodologies from the ML community, we initiate adaptation training with a pre-trained model when encountering a new dataset. We then analyze the data distribution to find an optimal set of hyperparameters (e.g., learning rate, batch size, number of epochs, etc.) for continuous model training to achieve the fastest convergence.

III. Preliminary Evaluation

Methodology: We implement a PrvTel flow example in Python and deploy it on the real-world Cisco-ie telemetry dataset [5]. This dataset comprises 15 devices within its network topology. We select one of the available datasets containing normal behavior telemetry data for analysis. The chosen dataset spans an hour-long runtime with no anomalies, featuring application traffic running bidirectionally across the fabric at an aggregate rate of 500Gbps. Out of the 83 total features, we select 8 numerical features associated with network traffic queries to evaluate our framework. The experiments aim to verify whether the PrvTel model's output data (1) maintains high query accuracy, (2) preserves user privacy, and (3) saves data storage space.

Baselines: We compare PrvTel with two baselines. The first is to use GAN and the second is to use Gaussian Mixture Model (GMM) to generate synthetic data. Both are implemented in Python and compared with their most optimal training results.

Querying Accuracy: To assess the querying accuracy of the model output data relative to the original data, we employ the Kolmogorov-Smirnov (KS) test value as a comprehensive metric to measure the similarity of data distributions for each feature. The KS test value represents the maximum absolute difference between the cumulative distribution functions (CDFs) of the two distributions being compared. As shown in Table I, the KS test score and p-value for each feature indicate that the difference between the CDFs of the original and model output distributions is minimal compared with baselines.

Furthermore, we perform two common queries on the data and examine the model output error rate. Figure 1 illustrates that PrvTel's error rate is less than 0.05, which is considered sufficiently accurate for network operators.

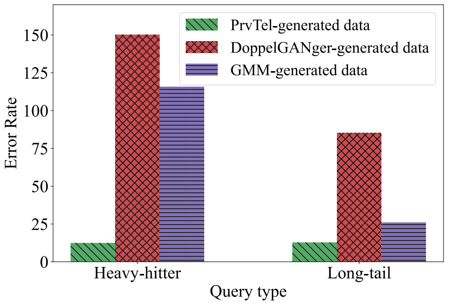

Fig. 1: Accuracy for two Network Traffic Queries: The error rate of query results derived from the PrvTel-generated data, when compared to the original data, is observed to be less than 5%.

Storage Efficiency and training cost: In comparison to the original zip-compressed data, which occupies 1.1GB of storage space, the combination of latent features and the decoder model requires a mere 56MB. This represents a substantial storage reduction of up to 19x, demonstrating the storage-saving efficacy of the PrvTel framework. Compared to the GAN total training time of approximately 14 minutes, PrvTel only takes ≈ 2 minutes, which is more than 7x faster.

IV. Limitations and Future Work

Privacy Preservation Evaluation: We plan to employ Membership Inference Attack (MIA) analysis as a privacy metric to evaluate whether the output data generated by PrvTel effectively protects users' privacy compared to the original data. The primary objective of MIA analysis is to safeguard sensitive information pertaining to individual data points within a dataset used for training a machine learning model. In particular, MIA involves training a classification model to ascertain if a specific data point was part of the training set.

Latent representation use: We can generalize the latent representation from the autoencoder model for other use cases, for example, transfer learning. Once the autoencoder has learned a generic latent representation, it can be fine-tuned on specific telemetry or datasets with smaller amounts of data. This will allow the autoencoder to adapt to different network telemetry cases more efficiently.

References

[1] M. Yu, "Network telemetry: towards a top-down approach," *ACM SIGCOMM Computer Communication Review*, vol. 49, no. 1, pp. 11–17, 2019.

[2] Z. Liu et al., "Nitrosketch: Robust and general sketch-based monitoring in software switches," in *ACM SIGCOMM*, 2019.

[3] Z. Liu, A. Manousis, G. Vorsanger, V. Sekar, and V. Braverman, "One sketch to rule them all: Rethinking network flow monitoring with univmon," in *Proceedings of the 2016 ACM SIGCOMM Conference*, 2016, pp. 101–114.

[4] Y. Yin, Z. Lin, M. Jin, G. Fanti, and V. Sekar, "Practical gan-based synthetic ip header trace generation using netshare," in *Proceedings of the ACM SIGCOMM 2022 Conference*, 2022, pp. 458–472.

[5] "Cisco telemetry data," https://github.com/cisco-ie/telemetry.