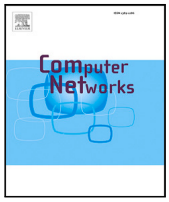
Computer Networks 251 (2024) 110616

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet



NetDiffus: Network traffic generation by diffusion models through time-series imaging

Nirhoshan Sivaroopan [a,*], Dumindu Bandara [a], Chamara Madarasingha [b], Guillaume Jourjon [c], Anura P. Jayasumana [d], Kanchana Thilakarathna [e]

[a] Department of Electronic and Telecommunication, University of Moratuwa, Sri Lanka
[b] University of New South Wales, Sydney, Australia
[c] CSIRO, Australia
[d] Colorado State University, CO, USA
[e] University of Sydney, Australia

ARTICLE INFO

Keywords:
Synthetic data generation
ML classification
Network traffic generation
Diffusion models

ABSTRACT

While Machine-Learning based network data analytics are now commonplace for many networking solutions, nonetheless, limited access to appropriate networking data has been an enduring challenge for many networking problems. Causes for lack of such data include complexity of data gathering, commercial sensitivity, as well as privacy and regulatory constraints. To overcome these challenges, we present a Diffusion-Model (DM) based end-to-end framework, NetDiffus, for synthetic network traffic generation which is one of the emerging topics in networking and computing system. NetDiffus first converts one-dimensional time-series network traffic into two-dimensional images, and then synthesizes representative images for the original data. We demonstrate that NetDiffus outperforms the state-of-the-art traffic generation methods based on Generative Adversarial Networks (GANs) by providing 66.4% increase in the fidelity of the generated data and an 18.1% increase in downstream machine learning tasks. We evaluate NetDiffus on seven diverse traffic traces and show that utilizing synthetic data significantly improves several downstream ML tasks including traffic fingerprinting, anomaly detection and traffic classification. The code has been included at https://github.com/Nirhoshan/NetDiffus.

1. Introduction

Many network planning, monitoring, and related computing tasks depend on network data analytics in and off the network [1]. These tasks are often driven by Machine Learning (ML) models which require a large amount of real network measurements to train the models. However, having access to appropriate network data traces is increasingly becoming challenging [2–5]. First, due to the complexity of modern networks and the sheer volume of data being transferred, deploying data collection tools requires significant expertise and cost. Second, privacy and regulatory constraints have made many types of network data inaccessible or restricted in use for other purposes such as network management [4,5]. Third, due to the commercial sensitivity of the data, many organizations do not share data with others, even among different departments of the same organization, let alone with the research community [2].

To overcome these issues, *synthetic data generation* has become a promising alternative which has been an emerging topic in both networking and computing domains. While there are many techniques and tools for data packets generation such as NS-3 [6], and iPerf [7] to satisfy a given model or a distribution, they fail to faithfully mimic the intricacies of real traces, particularly the real user behavior. However, recent ML-based approaches are capable of learning from traces to overcome this limitation [2–4,8,9]. Among them, generative model based solutions such as DoppleGanger [2], NetShare [3], SyNIG [10] and CTGAN [8] have shown superior performance in terms of representing practical network constraints and issues. However, GANs that have been the basis for many state-of-the-art (SOTA) network traffic generation tools suffer from mode-collapse, vanishing gradients, and instability unless the hyper-parameters are properly selected [11].

Poor data fidelity in network traffic synthesis has motivated us to address the following three research questions:

- RQ1 — How to further increase the fidelity of the network traffic data generation?

* Corresponding author.
E-mail address: nirhosh98@gmail.com (N. Sivaroopan).

https://doi.org/10.1016/j.comnet.2024.110616
Received 26 February 2024; Received in revised form 27 May 2024; Accepted 23 June 2024
Available online 27 June 2024

- RQ2 — How to optimize the overall data generation process for better efficiency?
- RQ3 — How useful this synthetic data generation method would be?

To address the RQ1, we explore how we can leverage recent advances in Diffusion Models (DM), which is an emerging generative AI (Artificial Intelligence) architecture, to generate synthetic network traffic. DM has shown outstanding performance compared to generative models such as GANs in particular with controlled image generation with models such as Dall-E. The ability to control the generated output makes DMs ideally suited for synthetic data generation for training ML models as it allows the generation of balanced datasets. This will lead to generating more robust and accurate trained models. Nevertheless, DMs have not yet been investigated for network traffic generation. *To the best of our knowledge, this is the first attempt at utilizing Diffusion Models for network data generation uncovering its benefits for network traffic analysis and related computing tasks*.

Despite DMs undergoing numerous advancements in data synthesis, the most notable progress has been achieved in the realm of image processing [11]. Throughout the years, DMs have consistently delivered high-resolution images in the generation of synthetic data. Therefore, to further address RQ1, we propose NetDiffus, a framework for network traffic generation using DM leveraging time-series imaging. This framework enables high-fidelity synthetic data, addressing the challenge of achieving satisfactory classification accuracy in network traffic analysis and the current issue with existing generative models in their limited ability to accurately extract features, which leads to a lack of resemblance with real data.

To further optimize the process to address RQ2, we perform multiple steps of pre-processing to enrich the features in the raw data. First, we convert 1D network traces to a specific image format called Grammian Angular Summation Field (GASF) [10,12] to capture important features from 1D network traces. GASF images can encode features such as packet sizes, inter-packet times and most importantly the correlation among the 1D time-series samples onto an image in 2D space making it a rich source of information for ML models. Second, to reduce the computational demand and improve the feature learning process in DMs, we apply several simple image processing techniques such as contrast adjustment and image resizing on GASF images. Third, in order to increase the features being captured by the generative models, we add gamma correction to the transformed images. Finally, this enhanced data is used to train DMs and synthetic data from the trained models are used to improve various downstream ML tasks. This method of data enrichment enables the DM to improve its performance in generating data. Note that, unlike recent works [2,3], NetDiffus does not generate meta-data at this stage.

To address RQ3, we leverage a wide range of network traffic: video, web and IoT, and show that a standard DM can generate data with higher fidelity surpassing the baseline compared. For example, NetDiffus achieves 28.0% and 85.6% fidelity improvement compared to SOTA GAN based models, DoppelGanger [2] and NetShare [3] respectively. Moreover, we utilize these synthetic data to train ML models related to different network-related tasks, such as traffic fingerprinting, anomaly detection, and classifications in data-limited scenarios. Even without combining with original data, NetDiffus can achieve almost the same accuracy of original data or improved accuracy of 1%–57% in those tasks. Comparing with the baselines above, NetDiffus synthetic data can improve classification performance by 4.7–32.3% in the corresponding ML tasks.

Based on the above contributions, we summarize that, the novelty in our approach stems from leveraging DM, the SOTA generative model which has undergone multiple changes to produce high quality images, and imaging time-series data with a high representative imaging mechanism and implementing multiple preprocessing steps to refine the raw data for generation. The goal of this work is to generate network traffic features, either raw (e.g., packet size) or pre-processed (e.g., bytes downloaded by a group of packets [13]), in 2D GASF format and use it directly in improving downstream ML tasks [12,14,15].

The rest of the article is as follows. In Section 2, we present the *Related work* summarizing the recent literature. Next, we provide basic knowledge and motivation to understand NetDiffus process in Section 3, *Background*. Section 4, *Design of NetDiffus*, provides detailed steps of the data generation mechanism followed by *Evaluation setup* in Section 5. Section 6 presents a range of *Results* and analysis we conducted to validate the performance of data generation leveraging various downstream ML tasks. *Limitations and future directions* of NetDiffus are discussed in Section 7.2, and finally, we provide *Conclusion* of NetDiffus in Section 8.

2. Related work

2.1. Synthetic network traffic generation

A Plethora of work has been done in network traffic data generation domain. [2–4,16,17]. Markov models and recurrent neural networks (RNNs) are commonly utilized in prior network traffic generation models [2,9]. Despite the generalization they provide, their fidelity in domain-specific generative tasks remains still limited. Particularly, RNNs typically require longer training times and are susceptible to the gradient vanishing problem. In the recent past, Generative Adversarial Network (GAN) based models have revolutionized network traffic synthesis, offering powerful capabilities. Generator and discriminator architecture used in GANs can effectively extract the characteristics of network traces with further modification to preserve temporal attributes [2–4,8,10,16]. Despite their promise, GANs suffer from major issues such as mode collapsing, and unstable and inflexible training which results in synthetic data with lower fidelity to the original data [2,11]. For example, the authors in [10], propose additional post-processing steps to overcome the lower fidelity in synthetic data from GAN models.

2.1.1. Shift from GANs to DMs

Recently, DM models have been proposed to address many issues bound with GAN architectures. For example, the DMs comparably higher stability and scalability while being robust to the mode collapsing. Authors of [11] have conducted and in-depth comparison and proved that DMs perform better than GANs in image domain. In [18], authors show that diffusion models have beaten GANs in realistic and stable image and video generation. Though, DMs are widely used in image, audio, text-to-image and image-to-text generation [11,17,19], to the best of our knowledge, no prior work has been conducted for synthetic network traffic generation. We discuss about DMs in detail in Section 3.1.

2.2. Imaging time series data

Converting 1D data into 2D images is widely studied in many works [10,12,20–23]. One motivation for such conversion is the improved performance in downstream analysis tasks, especially in ML-based classification [12] that leverages Convolutional Neural Networks (CNNs) [12,20]. Furthermore, such image representations are rich with information for ML tasks [23]. Using Gramian matrix and Markov Transition Fields (MTF), the authors in [12] have converted 1D data into specific images formats called Gramian Angular Summation Field (GASF) and Gramian Angular Difference Field (GADF). These data formats are derivatives of Gramian Angular Field (GAF), generated by converting the time series into a polar coordinate system and mapping the correlation between the 1D samples. These GASF, GADF and MTF images are typically used for the shorter time-series data. To convert longer time-series into images, Fukino et al. [21] have suggested to use Recurrence plot (RP) of Recurrence plots (RPofRPs) where RPs

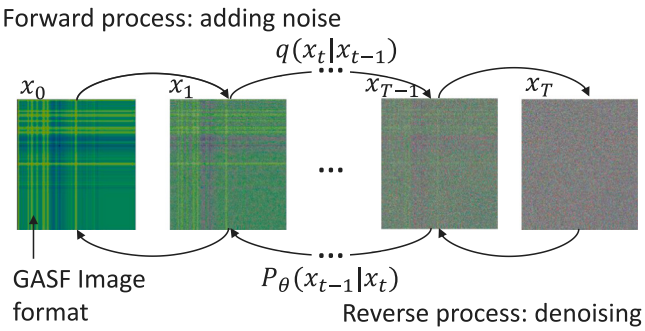


Fig. 1. Simple illustration of DM training process taking GASF image as the input.

are used hierarchically to represent local and global regularities. Apart from that different other image formats and conversions, such as direct conversion of 1D trace features to matrix [24], spectral images [22,23]. For example, the authors in [24] convert multiple features extracted from 1D traces to 2D matrix to be presented as heatmaps. Spectral images generated using 1D traces are used in [22,23] for biomedical signals (e.g., ECG) that are less complex and has periodic patterns that can be easily learnt by ML models compared to network traffic patterns.

2.3. ML in traffic classification tasks

A plethora of work has been done in encrypted traffic analysis leveraging ML approaches [25–27].

Anomaly detection in network traffic is an important activity in the network domain which helps the maintenance of the network systems by actively or passively monitoring network traffic [27,28].

For example, authors in [27] introduce C-LSTM (i.e., CNN model used with Long Short Term Memory module) architecture to extract robust features from raw packet data. Security and privacy is another important aspect of networks which affects both infrastructure, services and the users. In terms of privacy, Li et al. in [26] fingerprints a closed set of video streams leveraging different types of NN models (i.e., CNN, RNN and MLP (Multi Layer Perceptron)). The authors in [29,30] develop a hierarchical classification model based on XGBoost classifiers which can classify traffic into different network providers and then into individual videos from each platform. For all these applications, realtime performance is important [25,29]. For example, in [29], Kattadige et al. fingerprint video traffic by observing the packets at the beginning of the flow. The Authors in [29] propose 360° and normal video classification engine which extracts 360° video flows in near realtime.

In contrast to the previous work, we demonstrate how DMs can be used for network traffic generation by converting time series distribution to 2D GASF images. These images accurately capture feature distributions including correlations between 1D sample points further supporting downstream ML tasks.

3. Background and motivation

3.1. DM: Background and promise

Fundamentally, DM falls into likelihood-based methods, outperforming many generative AI models such as GANs, VAEs (Variational Autoencoders) in multiple aspects. DM models contains two main processes: forward and backward pass. During the forward pass, we gradually add noise, through multiple steps, until the given input (e.g., image or audio sample) becomes complete random noise (i.e., top row: Fig. 1). Assume that the input image, $\mathbf{x_0}$ (i.e., GASF image in our case) is sampled from a distribution, $q(\mathbf{x})$. Then, Gaussian noise is added to the samples in T steps creating a noisy sample sequence of $\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_T}$, where $\mathbf{x_T}$ is a complete random noise. Here, the amount of noise added is controlled by a variance schedule, $\{\beta_t \in (0,1)\}_{t=1}^{T}$. Since the noise level added in each step is sufficiently low, this process can be set to a conditional Gaussian, which is expressed in Eqs. (1) and (2) and, further combining with Markov assumption.

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t I) \tag{1}$$

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}) \tag{2}$$

During the reverse diffusion process, process starts from the $\mathbf{x_T}$ which is sampled from $\mathcal{N}(0, I)$. Given that β_t is a small value, the distribution, $q(x_{t-1}|x_t)$ is also considered as a Gaussian. To estimate this conditional diffusion, it is trained a model, p_θ, which is typically a DNN such as U-Net. This reverse diffusion model is denoted in Eqs. (3) and (4).

$$p_\theta(\mathbf{x}_{0:T}) = p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) \tag{3}$$

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(x_t, t), \sum_\theta(\mathbf{x}_t, t)) \tag{4}$$

This trained DNN acts as a generative model which produces images from pure noise distributions. Many recent works have used DMs in tasks such as image, audio, text-to-image and image-to-text generation, nevertheless, to the best of our knowledge, have not been used in network traffic generation with various network traffic types [11,17,19,31].

Compared to the related work in Section 2, DM can generated synthetic data in higher fidelity. Particularly, in the experiments in [10], the authors observed that traditional GAN approaches are struggling to keep a higher a fidelity in synthetic data in special forms such as GASF and GADF. However, we observe that DM models are strongly capable of learning subtle feature differences in GASF formats maintaining higher fidelity. Moreover, compared to which have more distribution coverage, scalability and stability in training and is a solution for issues such as mode collapse, instability and less flexibility in GAN models [11].

3.2. GASF images and conversion from 1D to 2D

In this section, we delve into the theoretical intricacies of the conversion process from 1D traces to 2D GASF images, providing an illustrated example in Fig. 2. We will discuss the significant potential that GASF images hold for the data generation capabilities of NetDiffus in greater detail in Section 4.

Let us begin by denoting the 1D signal we wish to convert into 2D GASF images as X with values ranging from 0 to 1. So, X can be represented as a sequence of values: $X = x_1, x_2, x_3, \ldots, x_n$, where n is the total number of data samples. Our conversion process involves transforming these values into polar coordinates using Eqs. (5) and (6).

$$\theta_i = \arccos(x_i), \quad x_i \in X \tag{5}$$

$$rad_i = \frac{t_i}{C} \tag{6}$$

where, t_i is the timestamp of ith sample and C is a constant factor to regularize the radius. These polar coordinates serve as the foundation for creating the Grammian matrix, which represents the inner product between the cosine angles of the samples from the time series. This matrix forms the basis of the Grammian Angular Field (GAF). Furthermore, the polar coordinate system allows us to leverage the angular perspective by considering the trigonometric sum between each point, helping identify the temporal relations within different time intervals.

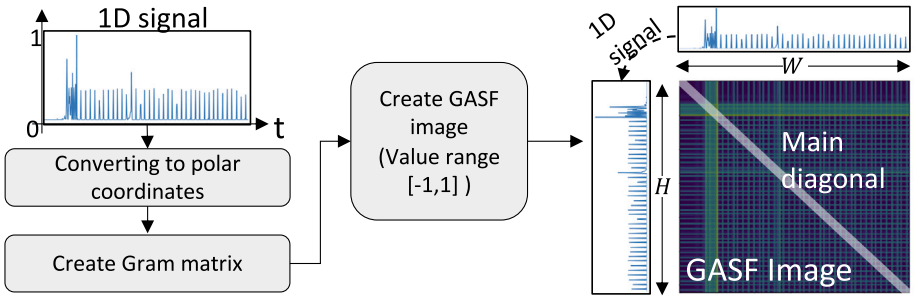
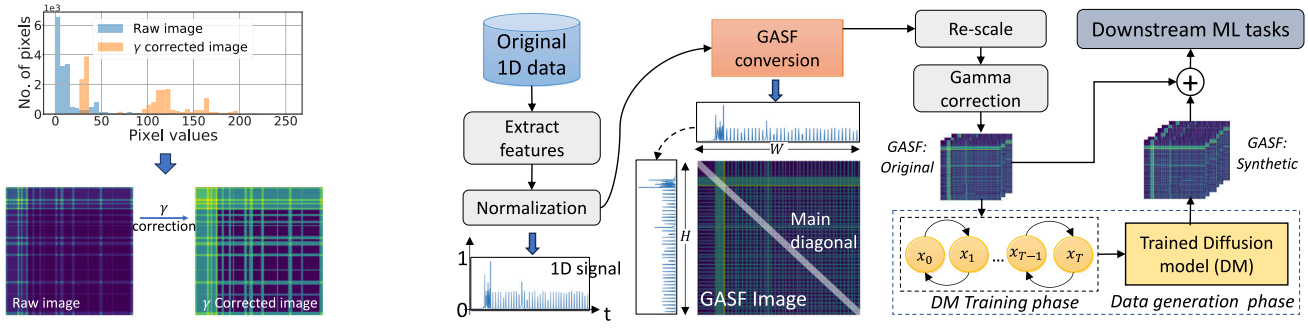


Fig. 2. Converting 1D trace to 2D GASF image.



(a) Histogram distribution between raw and γ corrected image

(b) Overall design of NetDiffus including sample 1D signal and a GASF image and the key design steps in described in Section 4.

Fig. 3. Impact of gamma correction taking sample GASF image and overall design of NetDiffus.

Eq. (7) illustrates how the GASF is constructed. Here, I denotes the unit row vector $[1, 1, \ldots, 1]$, X' is the transpose of X, and $i, j = 1, 2, 3...n$.

$$GASF = [\cos(\theta_i + \theta_j)] = X'.X - \sqrt{I - X^2}'.\sqrt{I - X^2} \quad (7)$$

Additionally, by setting the sample range of GASF images within [0,1], we establish a bijective property that allows the time series to be reconstructed from the image. Fig. 2 provides a visual representation of a sample GASF image, where the width (W) and height (H) of the image align with the trace's length. The main diagonal of the image corresponds to the time-series signal and encodes feature amplitudes, inter-packet gaps, and other characteristics. If needed, the corresponding 1D signal can be reconstructed using Eq. (8).

$$Trace = \frac{\sqrt{Y + 1}}{2} \quad (8)$$

Here Y is the vector consisting the elements in the diagonal of the GASF image [12,14]. GASF images possess several notable properties. As evident in Fig. 2, these images are symmetric around the main diagonal. Based on Eq. (7), the main diagonal directly corresponds to the initial 1D trace. Relations between different 1D samples are distributed within prominent vertical and horizontal strips, accentuated by peaks and troughs. Notably, constant values in a certain segment of the 1D signal are represented as constant pixel value patches on the GASF image. Importantly, in the context of NetDiffus data generation, our aim is to preserve these properties to ensure higher fidelity in the synthetic data.

4. Design of NetDiffus

In this section, we present the design strategies along with their basis used to develop NetDiffus. We start with individual design strategies and conclude with presenting the overall NetDiffus design.

Table 1
L3 classification for D1 data using original 1D traces and corresponding GASF images.

Platform	80% data		40% data		20% data	
D1-	1D	2D	1D	2D	1D	2D
YT	90.5	**92.5**	85.5	**87.0**	71.0	**75.0**
Stan	91.0	**99.0**	96.0	**98.0**	90.0	**93.0**
Netflix	83.0	**100.0**	84.0	**97.0**	78.0	**88.0**

4.1. Capturing important feature attributes

DNNs are designed to learn hidden features of input data. However, identifying subtle features such as the relation between the samples, frequency-related patterns from 1D signals or time series, etc., requires complex models and rigorous training processes. Manually extracting such features prior to model training is non-trivial as it can enable the model to learn those features efficiently and improve the data fidelity in generative AI tasks. Moreover, such manually featured samples have the potential to increase the performance (i.e., accuracy) in ML based classification or prediction tasks. Conversion of 1D traces to 2D images has been a proven method in ML domain in which we can extract meaningful features (e.g., relation between 1D samples, frequency distribution etc.) for ML tasks [12,20–23]. To analytically measure this impact, we conducted a preliminary experiment comparing the accuracy in a selected ML classification task for several datasets taking 1D network traffic signals and its 2D image representation (i.e., as in GASF format described in Section 3.2) as the input. Table 1 reports the results.

In this experiment, we leverage, one of the key datasets in our evaluation (see Section 5.1 for further details). The dataset contains video streaming related network traffic from three different content providers and each platform contains 20 unique videos. The intended classification task is to fingerprint the traces from these 20 videos. During the training phase, 80% of the traces from each video, repeatedly

collected during data acquisition, were used to train the ML models and the remaining 20% traces were used for the test. As mentioned above, in 2D scenario, we converted all the 1D traces to a 2D format, in this case, GASF images described in Section 3.2. 1D and 2D classification models have the same architecture except the 1D and 2D convolutional layers in the respective models and the results are given for different proportions of training data. For example, in *80% of data* we use all the available data to train the models and in *40% of data*, only half of the available training data was used.

For all proportions of training data, 2D GASF exceeds its corresponding 1D time series accuracy by 1.5%–13% showing the improved performance by GASF conversion. The main reason has been the rich source of feature space provided by GASF conversion, particularly by mapping the relation between 1D sample points onto one 2D plane as mentioned in Section 3.2. Hence, we leverage this 1D to 2D conversion in NetDiffus data generation, on the one hand, in favor of increasing the fidelity in synthetic data generation as the underlying architectures of these generative AIs are again DNNs as in traditional ML classifiers such as those used in the above experiment. On the other hand, synthetic data in 2D format can provide better performance in various downstream ML tasks compared to 1D counterparts as we show in Section 6 later.

4.2. Highlighting hidden features

As we operate in 2D domain, enhancing the contrasts of GASF images can further highlight subtle feature variations that can be effectively learnt by DMs and improve the fidelity. We leverage standard gamma correction on raw GASF images according to the equation, $I_c = A * I_r^{\gamma}$, where I_r, I_c, A and γ are gamma corrected image, raw image, a constant and gamma variable respectively. We empirically set $\gamma = 0.25$ and $A = 1$. Fig. 3(a) shows the histogram distribution of sample raw and gamma-corrected images. We notice that this process separates the pixel values into distinct ranges increasing the image contrasts and emphasizing the feature variations.

4.3. Supporting fast and stable training

DMs typically require high computational power and time. Hence, keeping the GASF image size similar to the trace length can lead to a longer training time and insufficient resources. As a solution, we resize the images to a fixed smaller resolution and feed low resolution images for DM training. We use `OpenCV.resize()` method with `INTER_AREA` interpolation method which resamples image pixels based on area relations and is the preferred method for image decimation [32]. We empirically decide the image size without affecting the downstream ML performance as image resizing can potentially drop high-frequency information from the images. Also, we max normalize GASF pixel range to [0,1] dividing each pixel by the global maximum value, 255. This makes DM training process faster and more stable. Note that we leverage vectorized operations in `Python-numpy` to speed up these pixel level operations.

4.4. Overall design of NetDiffus

Fig. 3(b) presents the overall process of NetDiffus. We start with time-series feature extraction from related datasets and max normalization. Then, the 1D signals are converted to GASF images which are further enhanced by gamma conversion and resizing the images. Finally, we use these original GASF images to train DMs. Unless otherwise noted, from each dataset, we use the first 80% of the data for synthetic data generation and keep the remaining as the test dataset for the downstream ML tasks. We set diffusion steps to 1000 and a standard U-Net model with 5 layers for the denoising process. The synthesized GASF images are used to improve various downstream ML tasks combined with original GASF data. We will release all the model details with the artifacts.

Interestingly, we observe that a basic DM architecture suffices to generate GASF images with high fidelity. At this stage of NetDiffus, we do not construct the corresponding 1D traces from the GASF images for downstream analysis, on the one hand, for a variety of ML-based analysis, a 2D image is a suitable format [12,14,15,33]. On the other hand, we observed improved ML classification performance with 2D GASF data compared to its 1D counterpart (see Section 4.1). We keep further evaluations with reconstructed 1D data in our future work.

Table 2
Summary of datasets.

Dataset	No. of classes	Traces per class	Downstream ML task
D1-YouTube [30]	20	100	video fingerprinting
D1-Stan	20	100	
D1-Netflix	20	100	
D2-DF [34]	20	900	web fingerprinting
D3-Google [35]	10	1000	IoT smart home activity recognition
D3-Alexa	10	1000	

5. Evaluation setup

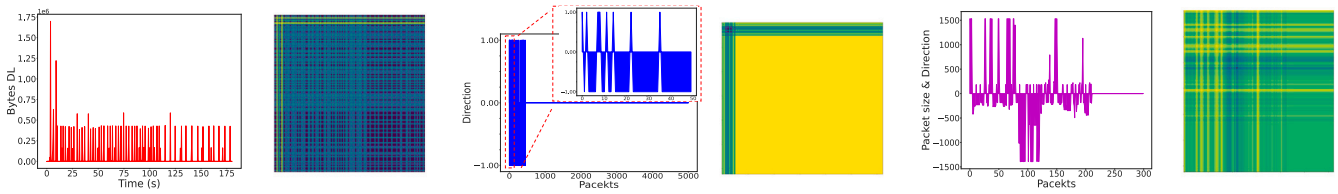
5.1. Dataset

In NetDiffus, we leverage three main datasets, reported in Table 2, representing three different network traffic.

D1: Streaming videos: We selected videos from **YouTube (YT)**, **Stan** and **Netflix**, 20 from each with 3 min duration, and streamed each video multiple times to generate 100 traces. Here, a trace represents a time series distribution of bytes downloaded (dl) values. While streaming, we passively captured network packets which are binned into non-overlapping 0.25 s bins to extract *Total bytes dl* feature in each bin, which can substantially represent the properties in video streaming sessions. Videos are streamed under not controlled bandwidth conditions in an enterprise network. Binning can highlight different video-specific features (e.g., quality switching) in network traces and increase the performance in downstream ML tasks [13,25]. The intended ML task is to fingerprint a given trace into one of the 20 videos from a given platform. Fig. 4(a)-(left) shows a sample trace from **D1-YT** dataset showing the periodic data transmission, which is common in most of the video streaming task, mainly due to the adaptive bit rate controlling algorithm in proprietary applications [25].

D2: Accessing web pages: We selected publicly available web surfing dataset from [34] which has 20 websites. The extracted features include packet direction (i.e., (+1) for uplink and (−1) for downlink packets) and inter-packet gaps. Each trace has 5000 fixed number of samples. Traces with over 5000 samples are truncated and otherwise 0 padded up to 5000 samples. Each traffic trace is classified into one of the 20 classes (i.e., website) as a website fingerprinting task. Fig. 4(b)-(left) illustrates such packet direction distribution as time-series for a randomly selected trace from **D2-DF**-Class 1.

D3: Traffic generated by IoT smart-home devices We collected this dataset by passively monitoring the network traffic from two smart home assistance devices **Google Home** [36] and **Amazon alexa** [37] smart home devices. A user gives 10 different commands for each device and the device communicates with its cloud server to execute the related activity. By repeating each command, we collected 1000 traces and each trace is kept at 300 packet length following the same trace truncation and 0 padding approach in **D2**. Captured features include packet size, direction and inter-packet gaps. The intended ML task is to classify each trace into one of the activities. Each trace has packet direction and the size as shown in Fig. 4(c)-(left) for **D3-Google**-Class 1.



(a) **D1-YT**: Class 1 (b) **D2-DF**: Class 1 (c) **D3-Google**: Class 1

Fig. 4. Sample traces from **D1-YT**, **D2-DF** and **D3-Google**, subfigure (left): time series (1D signal) feature distribution, (right): corresponding GASF transformation of the time series feature.

The datasets we selected consist of a wide range of features, for example from raw packet sizes to aggregated *Total bytes dl* values by bins that can be used for a wide range of ML tasks. This further verifies the robustness of NetDiffus for different network-related feature generation along with the efficacy of GASF-based synthetic data for downstream tasks. Particularly, these variations in 1D domain are properly represented by the 2D GASF images as shown in Fig. 4 while increasing the robustness in NetDiffus data generation process.

Based on how we utilize the data for ML training, there are three main scenarios; (*i*) **original**: use only original data, (*ii*) **synth**: use only synthetic data, (*iii*) **ori+synth**: combine original data with synthetic. Unless otherwise noted, we separate data (i.e., network traffic traces) from each class from each dataset into 80%/20% train and test splits which are used to train and test both DMs in data generation and ML models in downstream tasks.

5.2. Benchmark models

DoppelGanger (DG) [2]**:** A GAN based approach which generates both metadata and traffic features of the traces while finding their correlations. We train NetDiffus using one of their datasets, Wikipedia Web Traffic (WW), first, to compare DG with NetDiffus while preserving its original attributes, and second, to show the robustness of NetDiffus to different datasets. The related ML task is to classify a given WW trace into four classes based on the type of access to Wiki pages (e.g.,mobile, desktop etc.).

NetShare [3]**:** SOTA GAN-based method for packet/flow header generation taking them as time-series data compared to tabular format. Though the base model considered is DG, the authors claim that with the proposed packet/flow data epochs merging mechanism, the scalability of the generation has been increased with improved fidelity in synthetic data. We train Netshare using **D3-Google** and **D3-Alexa** datasets as these datasets are compatible with packet-level data generation in Netshare.

Note: DG and NetShare have outperformed many other ML based and statistical approaches [38–41], and therefore, we exclude other GAN methods and ML based approaches from the comparison. Since these models synthesize 1D data, we convert 1D synthetic traces from those models to GASF images before the comparison. Further, the code setup can be found at https://github.com/Nirhoshan/NetDiffus.

6. Results

Now, we present the results and analysis leveraging synthetic data from NetDiffus for downstream ML tasks. We start from measuring the fidelity of the synthetic data followed by a comparison with benchmarks. Then, we present how the synthetic data can improve the performance in three downstream use cases that are based on ML. Finally, we present how the 2D GASF generation can outperform its 1D counterpart showing improved ML classification accuracy in two selected datasets.

6.1. Analysis of data fidelity

We compare the histograms between **original** and **synth** GASF images from **D1**, **D2** and **D3** in Fig. 5. Overall, original and synthetic data have a similar distribution preserving a wide range of pixel values. We observe a noticeable difference in the distribution of **D3-Alexa** in Fig. 5(d). This is due to the higher similarity between the classes that has hindered DMs learning unique differences between **D3-Alexa** classes. These high overlaps between histograms verify that attributes such as packet size, and temporal correlations that were mapped onto the original GASF images are still maintained in NetDiffus synthetic data.

As another metric, we use the metric FID (Frechet Inception Distance) [42] to evaluate fidelity in synthetic traces.[1] A lower FID score means that the original and synthetic images have a close distribution and vice-versa.

In these experiments, we pair-wise compare n randomly selected synthetic traces from each class with corresponding original traces used to train the model. Fig. 6(a) shows that, overall, **D1** data has a lower FID score compared to others which is less than 9 on average. **D3–Alexa**, shows the highest FID score, 25.8 ($\pm$11.9), because of the overlap in the features between classes for some traces in the original dataset.

Figs. 6(b) and 6(c) compares the FID scores between NetDiffus and the baselines: DG and Netshare respectively. Overall, NetDiffus outperforms all the baselines by an average FID value difference of 72.8 which is equivalent to 66.4% gain. NetDiffus surpasses DG by 4.1 (i.e., a gain of 28%) showing its applicability to new datasets, and the model significantly outperforms Netshare by around 105 of average FID score difference (i.e., a gain of 85.6%). The main reason for lower fidelity in Netshare is that the model is not able to learn packet size distributions and correlations over a wide range (e.g., 0–1500 bytes) despite the model robustness claimed [3]. Though we do not evaluate NetDiffus fidelity in 1D domain after reversing GASF conversion, the higher fidelity in 2D GASF format indicates the high fidelity in 1D domain as well.

6.2. Comparison with benchmarks

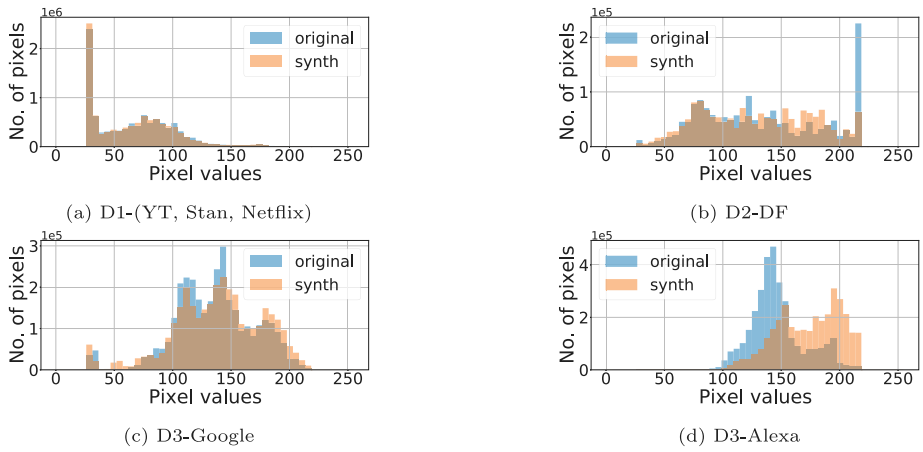
6.2.1. Comparison between different 2D image formats

Table 3 reports the comparison between GASF format used in NetDiffus and different other image format used to 1D to 2D conversion [12]. These image formats include GADF[2] and MTF,[3] which transfer the unique patterns and the relation between the samples in 1D traces as we observed in GASF format. We compare the ML accuracy in

[1] FID is a metric used to measure fidelity in images. FID values < 20 are commonly observed for high fidelity data [42,43].

[2] This is similar to GASF. But rather than taking the summation between the cosine angles, we take the difference between them here.

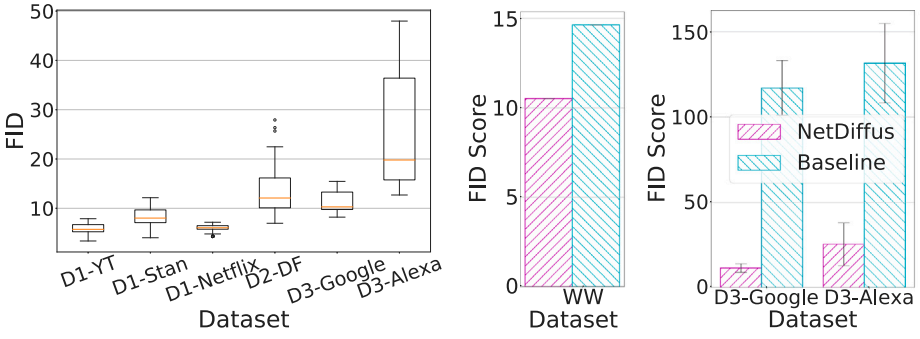
[3] The time-series will be binned in the amplitude axis and the image will be created by applying markov transition theory to the bins and extending it to every elements.



(a) D1-(YT, Stan, Netflix) (b) D2-DF

(c) D3-Google (d) D3-Alexa

Fig. 5. Histogram distribution between original and NetDiffus synthetic data.



(a) FID score by datasets (b) DG (c) Netshare

Fig. 6. Fidelity by individual datasets and comparison with baselines.

Table 3
Comparison with different image formats.

Image (type)	Dataset	Classifier Per. Accuracy(%)	DM Per. FID score	Invertible
GASF	D1-YT	92.5	12.5	Y
	D1-Stan	99.0	16.3	Y
GADF	D1-YT	91.0	14.2	N
	D1-Stan	96.0	18.6	N
MTF	D1-YT	57.0	95.6	N
	D1-Stan	18.0	121.2	N

related downstream ML tasks, and fidelity in synthetic data in terms of FID-score and invertibility (i.e., 2D to 1D conversion). Results show that both GASF and GADF perform well on the generation and classification tasks on D1-YT and D1-Stan. Nevertheless, GASF outperforms GADF slightly by 1.5%–3% higher accuracy in ML and 1.7–2.3 lower FID-score in fidelity. MTF, on the other hand, shows the lower performance other two image types. Since, GASF conversion allows us to freely move between image and trace (refer Eq. (8)), we chose GASF as the imaging method for NetDiffus.

6.2.2. ML accuracy comparison with baselines

Fig. 7(a) compares the NetDiffus with DG model using five classification ML algorithms (Convolutional Neural Network (CNN), XGBoost, Multi-Layer Perceptron (MLP), Naive Bayes (NB), Random Forest (RF)). We classify the type of access to Wiki pages, e.g. mobile, desktop, etc. in WW trace for **original** and **synth** scenarios taking an equal number of synthetic samples from NetDiffus and DG. In all ML models, NetDiffus exceeds the DG accuracy, which is 4.67% on average. Similarly, in Fig. 7(b), for both **D3-Google** and **D3-Alexa** datasets, NetDiffus outperforms Netshare by an average difference of 32.3% and 17.3% respectively. These results indicates that NetDiffus can outperform many SOTA data generation models.

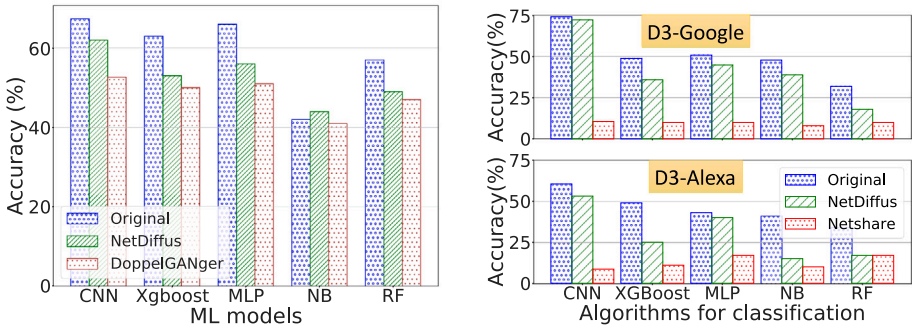
6.3. Performance in different ML algorithms

By considering multiple downstream ML algorithms in Fig. 7, we further show that NetDiffus synthetic data can be used to evaluate different downstream ML algorithms as well. This is important when utilizing synthetic data to tune models in load balancing, cluster scheduling etc. [2]. A key property of synthetic data to achieve this goal is they should have accuracy trends similar to that of original data in different algorithms. Except for the **D3-Alexa**-MLP evaluation in Fig. 7(b)-bottom, NetDiffus synthetic data follows similar accuracy patterns to its original data in all other cases. For example, in Fig. 7(a), CNN, XGBoost and MLP show higher classification accuracy with both original and NetDiffus data and both datasets show lower accuracy with NB.

6.4. Improved ML performance in use-cases

6.4.1. Surveillance through traffic fingerprinting

ML has been widely applied for the surveillance of network traffic, nonetheless, it can show limited performance due to the shortage of training data. In this use-case, we analyze how traffic fingerprinting tasks can be optimized by NetDiffus synthetic data. We leverage a hierarchical ML classifier with three layers (L), which is illustrated in Fig. 8. At L1, we classify the input samples by the *Traffic type* (e.g., video, web or IoT) followed by *Platform type* (e.g., Video into



(a) DG: WW dataset (b) Netshare: D3-G & D3-A

Fig. 7. Comparison with baselines G:Google, A:Alexa.

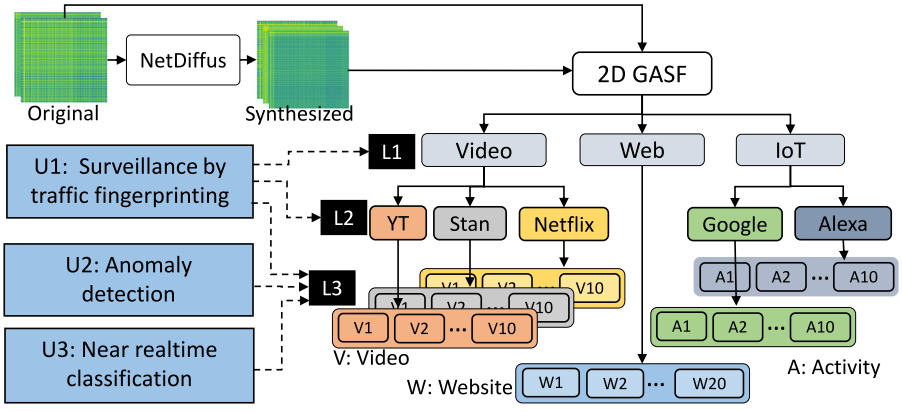


Fig. 8. Hierarchical classification model used in use case analysis.

Table 4
Accuracy of hierarchical classification model.

Layer (type)	Data used	Original	Synth	Ori+Synth
L1 (*Traffic*)	**D1+D2+D3**	99.0	100.0	100.0
L2 (*Platform*)	**D1**	100.0	100.0	100.0
	D3	97.0	97.5	98.5
L3 (*Class*)	**D1-YT**	84.5	91.0	92.5
	D1-Stan	92.5	98.5	99.5
	D1-Netflix	97.5	100.0	100.0
	D2-DF	92.0	83.8	93.64
	D3-Google	74.4	72.5	77.0
	D3-Alexa	60.5	53.1	62.0

YT, Stan, Netflix and IoT into Google and Alexa) classification at L2. Finally, L3 corresponds to classifying samples from each platform into individual classes (e.g., YT into V1(Video 1), V2, ..., V10 and Google IoT data into A1 (Activity 1), A2, ..., A10).

Results and observations: Table 4 reports classification accuracy. We observe both L1 and L2 provide over 95% accuracy in both **original** and **synth** scenarios. L3 is a challenging task compared to L1 and L2 due to the higher number of classes and the similarities in traces we notice between the classes. In **D1**, video fingerprinting task we see 3.5–6.5% accuracy improvement in **synth** scenario compared to **original** data mainly due to the high fidelity in synthetic data

Though we see 5.83% average accuracy drop in **D2** and **D3** in **synth** scenarios compared to **original**, referring to recent literature [2,3] and considering the difficulty in tasks, we believe such accuracy levels are still acceptable. However, by combining original with synthetic data, we achieve improved accuracy compared to **original** scenario by 1%–8%.

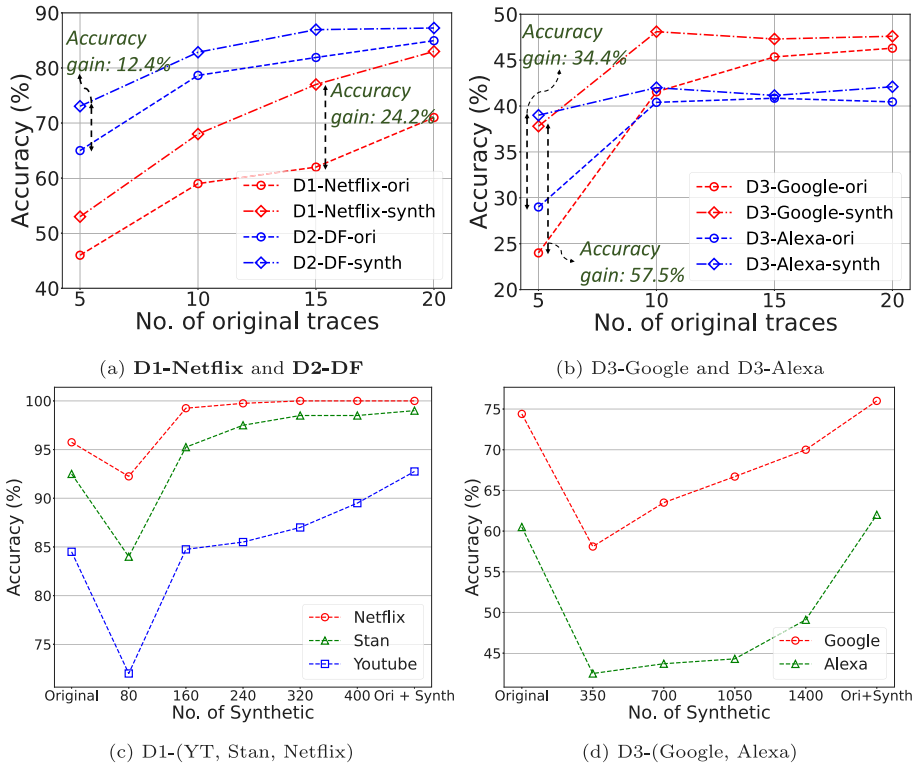
A limited number of original traces is a challenging scenario which hinders the above ML performance. To see the support by NetDiffus to improve the downstream ML accuracy, we change the number of original traces on NetDiffus data generation and add the resulting synthetic traces to train the ML models. Fig. 9 reveals that when the number of original traces is limited, NetDiffus synthetic traces can exceed the accuracy of original data. In Fig. 9(a), **D2-DF synth** achieves 12.4% accuracy gain and, in Fig. 9(b), **D3-Google synth** obtain 57.5% accuracy gain compared to corresponding **original** scenarios. This is in contrast to the lower performances in **synth** scenario in Table 4 and highlights the advantages of NetDiffus in data-limited use cases.

Figs. 9(c) and 9(d) illustrates the accuracy variation when gradually adding the synthetic data. We start from the **original** scenario and the numerical values in the x-axis show the number of synthetic images added in **synth** scenario (i.e., having only synthetic images). Last index in the x-axis indicates the **ori+synth** scenario. Fig. 9(c) shows that even with 160 synthetic data all three datasets in **D1** can surpass or achieve the same accuracy of **original** data. Though we observe a sudden drop in **synth** scenario for **D3** data, adding synthetic data shows a gradual increase in accuracy for **D3** data in Fig. 9(d), and eventually exceeds the **original** accuracy in **ori+synth** scenario.

6.4.2. Anomaly detection

Anomaly detection often struggles with collecting sufficient malicious data to train models. We extend the **D1** video fingerprinting at L3 while creating a class imbalance when training the ML models to mimic real world anomaly detection. We assume that randomly selected two classes with a limited number of traces are anomalies and another set of five classes with all available training traces are legitimate.

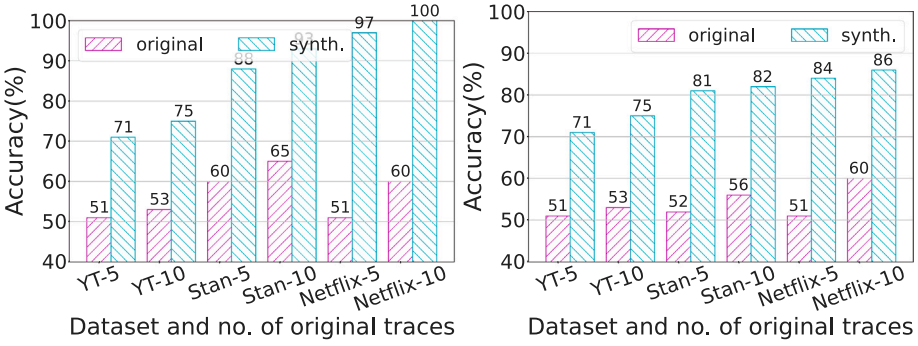
We analyze two sub-cases. (i) *sub-case 1*: Ground truth data is available for both legitimate and anomaly classes. In this case, we simply calculate the accuracy of anomaly trace classification. To mimic the shorter duration and further generalize anomalous behavior, we limit the trace length from 180 s (Full trace) to first 45 s. (ii) *sub-case*



(a) **D1-Netflix** and **D2-DF** (b) D3-Google and D3-Alexa

(c) D1-(YT, Stan, Netflix) (d) D3-(Google, Alexa)

Fig. 9. Accuracy improvement with a limited number of original data for **original** and **synth** scenarios and the impact of number of synthetic data on L3 classification accuracy.



(a) Full-trace (b) 45s long trace

Fig. 10. Anomaly detection, *sub-case 1*: 5 & 10 traces from anomaly classes with ground truth data for model training. Results are given for two different trace lengths: Full and 45 s long.

2: Ground truth labels are available only for legitimate classes. During the test phase, we measure the uncertainty of classification based on the entropy of classification results following a deep ensembling approach [44]. For the legitimate and anomaly traces, a lower and a higher uncertainty are expected respectively.

Results and observations: In *sub-case 1* (Fig. 10), for *Full* and 45 s lengths, adding synthetic traces provide 54.6(±18.3)% and 48.5(±9.0)% of average gain respectively, compared to having only original data. We see that gain achieved is higher when the number of original data is limited. For example, **Netflix-5** achieves 90.2% average accuracy gain compared to **Netflix-10** which has only 66.7%. Fig. 11 presents the uncertainty comparison for legitimate and anomaly detection. In a gist, synthetic traces reduce the uncertainty of the predictions for legitimate samples (e.g., 0.75 on average at 1600 **synth** traces), whereas, the uncertainty for anomaly samples remains high. This higher uncertainty score is an indication to decide whether a given trace is an anomaly [44]. This is further verified in the extended graph showing the score for different datasets at 800 **synth** data step. We notice a 0.74 and 1.01 average uncertainty difference between legitimate and anomaly data for **Stan** and **Netflix** respectively further evidencing the support of NetDiffus for anomaly detection.

6.4.3. Near real-time classification

We analyze the NetDiffus support for near real-time classification representing a scenario in which only a part of the network trace is extracted without waiting for the entire trace. We assume that we can identify the beginning of the network trace. The corresponding GASF images are generated by cropping the initial GASF images from the bottom and right directions representing the 1D trace with limited data. We leverage L3 classification in Section 6.4.1. We train multiple classifiers for different trace lengths which are measured in trace length for **D1** and percentage number of packets for **D2** and **D3**.

Results and observations: Fig. 12(a) shows that with having only 45 s worth of data, **D1** data can achieve over 92% accuracy in **ori**+**synth** scenario which is 5.7% accuracy gain compared to **original**. Though **synth** accuracy is less than the **original** scenario, on the one

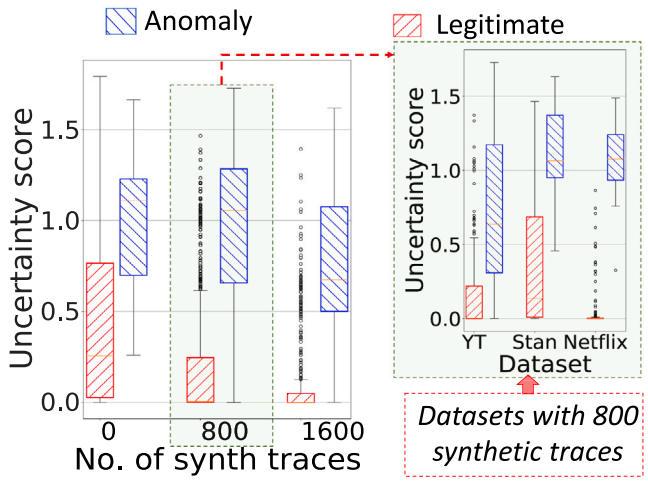


Fig. 11. Anomaly detection, *sub-case 2*: Uncertainty score variation when adding synthetic data.

Table 5
Comparison between 1D vs 2D DMs.

Dataset	Original (%)	Synth (%)		Ori+Synth (%)	
		1D	2D	1D	2D
D1-Netflix	97.5	62.2	100.0	94.8	100.0
D3-Google	74.0	42.3	72.5	73.2	76.0

hand, **ori+synth** accuracy always outperforms **original** scenario in **D2** and **D3** data in Fig. 12(b) and Fig. 12(c) respectively. On the other hand, **synth** accuracy follows the same increasing trend in **original** and **ori+synth**, and eventually, reducing the accuracy gap with **original**, for example in **D2**. Empirically, we observe GASF conversion takes time in the millisecond range (i.e., around 10 ms) without affecting overall inference process.

6.5. Comparison with 1D DM

We compare L3 classification for **D1-Netflix** and **D3-Google** between synthetic data from 1D and 2D DMs from NetDiffus. We follow, a similar architecture to NetDiffus DM when developing corresponding 1D DM and use the 1D traces to train them. Once the 1D synthetic traces are generated, we convert them to 2D GASF images for downstream ML classification. Table 5 reports that in all scenarios, synthetic data from 2D DMs in NetDiffus overpass the data from 1D DMs and achieves almost the same **original** accuracy. For instance, in **synth** scenario, NetDiffus data achieve 37.8% and 30.2% accuracy improvement over 1D DMs data for **D1-Netflix** and **D3-Google** respectively. We further measure a lower FID score for 2D DMs data which is 77% and 36% less than synthetic data from 1D DM in **D1-Netflix** and **D3-Google** respectively. These results reveal that 2D DMs in NetDiffus can perform better than their 1D counterpart.

7. Discussion

7.1. Applicability of NetDiffus

In the motive to enhance network traffic data generation for improved fidelity and classification accuracy in network traffic analysis, we have achieved superior results in both data fidelity and ML classification accuracy relative to current SOTA network traffic generative models. We demonstrated the effectiveness of the data generated by NetDiffus, through three use cases that can be applied for different practical scenarios:

- Surveillance through fingerprinting - Applicable for deep network traffic analysis
- Anomaly detection - Applicable for use cases such as identification of unauthorized access patterns, and potential network security breaches etc.
- Near real-time classification - Applicable for early detection network attacks by collecting a minimal amount of real data.

In all three scenarios, NetDiffus outperformed existing SOTA models significantly.

By using the high-resolution generative model DM, NetDiffus is able to produce data of exceptional fidelity.

Even though our study involves network traffic data, this methodology is applicable for time series. Network administrators and mobile network administrators can collect their sample data and use them to synthesize highly reliable data for content filtering or parental controlling with anomaly detection. Further, mobile network operators might want to perform the network analysis with minimum data collected to provide quick services to their users. In the near real-time usecase we highlight how NetDiffus can be used for such purposes with minimum data collected.

7.2. Limitations and future directions

We identify the reconstruction of time series from GASF and privacy awareness in synthetic data generation as two limitations in our current work and we aim to address those in the following future directions.

7.2.1. Reconstruction of time series from GASF

Our research has demonstrated that when it comes to classification and synthetic data generation, working in the image domain yields superior results compared to the 1D domain (refer Tables 1 and 5). As a result, we have chosen not to delve deeply into the comprehensive analysis of 1D traces by applying the inverse GASF transformation on synthesized images. It is important to note that the reconstructed traces, as they stand now, may not closely resemble the initial pre-processed traces due to resizing or rescaling the GASF images. However, it is possible to generate high resolution images without rescaling using the current implementation of NetDiffus, if there is a requirement of reconstructing 1D traces. In such a scenario, the reconstructed trace would indeed be a perfect replica of the initial pre-processed trace.

7.2.2. Privacy aware NetDiffus

The resizing of GASF images, and the subsequent loss of fidelity in the reconstructed traces, has the potential to mitigate privacy leakage form data. This privacy improvement arises from the fact that the reconstructed signal becomes an aggregated representation of the initial 1D trace, making it more challenging for sensitive information to be directly discerned from the reconstructed data. Essentially, the aggregation process blurs certain details, which can be viewed as a privacy-boosting feature.

At this stage, our study does not provide an in-depth analysis of the trade-off between privacy and utility. We defer the exploration of this trade-off to future research endeavors. To incorporate privacy-aware generation, we have intentions to integrate differential privacy measures into the denoising process of diffusion models (DMs) to create datasets that are privacy-conscious.

Differential privacy is a critical concept in data privacy and security. It involves adding controlled noise or perturbations to data or query results to protect individuals' sensitive information while still providing useful insights from the data. In the context of NetDiffus, this means that during the process of synthesizing data, we will introduce differential privacy techniques to ensure that the resulting datasets do not inadvertently disclose private or sensitive information.

However, it is essential to consider that this aggregation process also comes with its own set of trade-offs. While it enhances privacy, it has

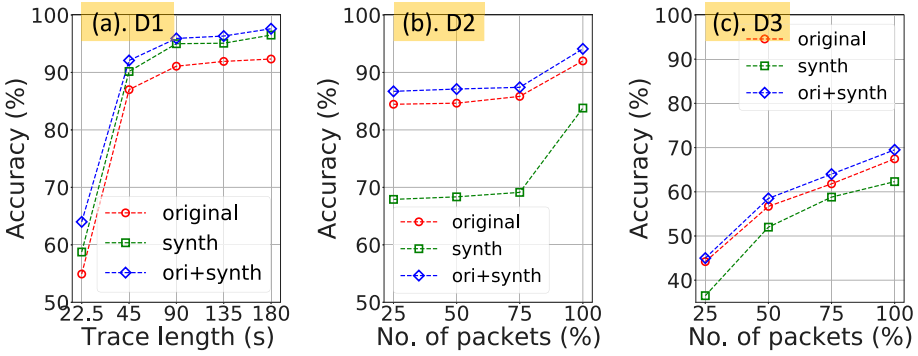


Fig. 12. Performance of L3 classification for different trace lengths/No. of packets.

the downside of potentially removing subtle feature differences or fine-grained details present in the original data. These nuanced distinctions can be valuable for downstream analysis tasks or applications where precise information is necessary.

This approach aligns with the broader trend of incorporating privacy safeguards into data analysis and ML models, especially when dealing with sensitive or personal information. By making privacy a fundamental consideration in the NetDiffus framework, we aim to provide a robust solution that not only yields valuable insights from network data but also respects individuals' privacy rights and data protection regulations.

7.2.3. Extended data scope of NetDiffus

NetDiffus was initially designed with a focus on network traces that contain detailed packet-level features. However, there is a clear need to extend the capabilities of NetDiffus to handle flow-level data, which is commonly accessible to network providers. Flow-level data is valuable for a more comprehensive analysis of network behavior. Furthermore, despite our investigation of three prominent types of internet traffic – video, IoT, and web, which collectively constitute the majority of online activity – it is crucial to subject NetDiffus to evaluation using expanded datasets that encompass diverse applications, such as file transfers. This broader evaluation will help determine the model's adaptability and usefulness across a wider range of network scenarios.

While our current implementation has been tailored for single-variate time series, it is worth highlighting that NetDiffus can be readily extended to accommodate multi-variate time series data. Additionally, we can incorporate metadata by mapping it to integer values and combining these channels into a single image representation. Currently, NetDiffus employs a fixed lower image size, a choice made to expedite model training and enhance scalability. To introduce more flexibility in handling various 1D trace lengths, we intend to explore stable/latent diffusion models that utilize lower-dimensional embeddings of input data [45,46]. This approach has the potential to improve both the scalability and the fidelity of the data representation.

In scenarios where NetDiffus is applied to longer traces, particularly those with tens of thousands of data points, it may be necessary to employ additional pre-processing steps. These steps could involve segmenting the trace into smaller frames, applying GASF imaging to each frame, and then stacking these images as channels. Alternatively, we can consider implementing the techniques recommended by the authors of RpofRps [21] for handling long traces with Recurrence Plots (RP). These strategies help manage computational complexity while still retaining essential details from the entire trace. Given the volatile nature of network traffic, it is imperative to continuously train NetDiffus to adapt to significant changes in traffic patterns over time. Consistent training ensures that the model remains capable of capturing and understanding evolving network behaviors.

7.2.4. Acceleration of data generation phase in NetDiffus

It is essential to note that NetDiffus operates in an offline mode, which means that the time required for data generation does not significantly impact the overall process. To further enhance efficiency, we can explore advanced techniques like parallel processing with GPUs or TPUs [47], which can substantially accelerate the data generation phase. This optimization ensures that the model can efficiently handle large-scale data processing tasks.

8. Conclusion

We introduced NetDiffus, a novel network traffic generation tool based on diffusion models (DMs). NetDiffus first converts time-series network traffic data into a special image format called Grammian Angular Summation Field (GASF) and then, uses that 2D data for data synthesis using DMs. The resulting 2D GASF images represent traffic data that has the same properties of the original data but with unseen variations that are useful for various downstream tasks such as ML model training. Moreover, this GASF conversion enables us to represent network traffic in a visually interpretable manner, making it easier to analyze and understand.

Leveraging three different traffic types: video, web and IoT, we demonstrate that NetDiffus can provide synthetic data with high fidelity verifying the quality of data generation. This has been further proven by the increased performance in various downstream ML tasks including network traffic fingerprinting, anomaly detection and near-realtime classification in the network, by using the synthetic data for ML model training. We further showed that NetDiffus outperforms many state-of-the-art data synthesis approaches proposed for network traffic data, highlighting the usefulness of GASF image conversion followed by the application of DMs.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] G. Dimopoulos, I. Leontiadis, P. Barlet-Ros, K. Papagiannaki, Measuring video QoE from encrypted traffic, in: Proceedings of the 2016 Internet Measurement Conference, 2016, pp. 513–526.

[2] Z. Lin, A. Jain, C. Wang, G. Fanti, V. Sekar, Using GANs for sharing networked time series data: Challenges, initial promise, and open questions, in: Proceedings of the ACM Internet Measurement Conference, 2020, pp. 464–483.

[3] Y. Yin, Z. Lin, M. Jin, G. Fanti, V. Sekar, Practical GAN-based synthetic IP header trace generation using netshare, in: Proceedings of the ACM SIGCOMM 2022 Conference, 2022, pp. 458–472.
[4] C. Kattadige, S.R. Muramudalige, K.N. Choi, G. Jourjon, H. Wang, A. Jayasumana, K. Thilakarathna, VideoTrain: A generative adversarial framework for synthetic video traffic generation, in: 2021 IEEE 22nd International Symposium on a World of Wireless, Mobile and Multimedia Networks, WoWMoM, IEEE, 2021, pp. 209–218.
[5] M.R. Shahid, G. Blanc, H. Jmila, Z. Zhang, H. Debar, Generative deep learning for Internet of Things network traffic generation, in: 2020 IEEE 25th Pacific Rim International Symposium on Dependable Computing, PRDC, IEEE, 2020, pp. 70–79.
[6] NS3, NS-3 network simulator, 2023, https://www.nsnam.org/. (Online; Accessed 19 June 2023).
[7] Iperf.fr, iPerf - The ultimate speed test tool for TCP, UDP and SCTP, 2023, https://iperf.fr/. (Online; Accessed 19 June 2023).
[8] L. Xu, M. Skoularidou, A. Cuesta-Infante, K. Veeramachaneni, Modeling tabular data using conditional GAN, Adv. Neural Inf. Process. Syst. 32 (2019).
[9] H. Redžović, A. Smiljanić, M. Bjelica, IP traffic generator based on hidden Markov models, parameters 1 (2) 1.
[10] N. Sivaroopan, C. Madarasingha, S. Muramudalige, G. Jourjon, A. Jayasumana, K. Thilakarathna, SyNIG: Synthetic network traffic generation through time series imaging, in: 2023 IEEE 48th Conference on Local Computer Networks, LCN, IEEE, 2023, pp. 1–9.
[11] P. Dhariwal, A. Nichol, Diffusion models beat GANs on image synthesis, Adv. Neural Inf. Process. Syst. 34 (2021) 8780–8794.
[12] Z. Wang, T. Oates, Imaging time-series to improve classification and imputation, 2015.
[13] F. Bronzino, P. Schmitt, S. Ayoubi, G. Martins, R. Teixeira, N. Feamster, Inferring streaming video quality from encrypted traffic: Practical models and deployment experience, Proc. ACM Measur. Anal. Comput. Syst. 3 (3) (2019) 1–25.
[14] A. Shankar, H.K. Khaing, S. Dandapat, S. Barma, Epileptic seizure classification based on gramian angular field transformation and deep learning, in: 2020 IEEE Applied Signal Processing Conference, ASPCON, IEEE, 2020, pp. 147–151.
[15] Ò. Garibo-i Orts, N. Firbas, L. Sebastiá, J.A. Conejero, Gramian angular fields for leveraging pretrained computer vision models with anomalous diffusion trajectories, Phys. Rev. E 107 (3) (2023) 034138.
[16] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial networks, Commun. ACM 63 (11) (2020) 139–144.
[17] F.-A. Croitoru, V. Hondru, R.T. Ionescu, M. Shah, Diffusion models in vision: A survey, IEEE Trans. Pattern Anal. Mach. Intell. (2023).
[18] M. Stypułkowski, K. Vougioukas, S. He, M. Zięba, S. Petridis, M. Pantic, Diffused heads: Diffusion models beat GANs on talking-face generation, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2024, pp. 5091–5100.
[19] M. Zhang, Z. Cai, L. Pan, F. Hong, X. Guo, L. Yang, Z. Liu, Motiondiffuse: Text-driven human motion generation with diffusion model, 2022, arXiv preprint arXiv:2208.15001.
[20] A.B. Said, A. Erradi, Deep-Gap: A deep learning framework for forecasting crowdsourcing supply-demand gap based on imaging time series and residual learning, in: 2019 IEEE International Conference on Cloud Computing Technology and Science, CloudCom, IEEE, 2019, pp. 279–286.
[21] M. Fukino, Y. Hirata, K. Aihara, Coarse-graining time series data: Recurrence plot of recurrence plots and its application for music, Chaos 26 (2) (2016) 023116.
[22] K.E. Smith, A.O. Smith, Conditional GAN for timeseries generation, 2020, arXiv preprint arXiv:2006.16477.
[23] K.E. Smith, A.O. Smith, A spectral enabled GAN for time series data generation, 2021, arXiv preprint arXiv:2103.01904.
[24] A.N. Sayed, Y. Himeur, F. Bensaali, From time-series to 2D images for building occupancy prediction using deep transfer learning, Eng. Appl. Artif. Intell. 119 (2023) 105786.
[25] C. Kattadige, A. Raman, K. Thilakarathna, A. Lutu, D. Perino, 360NorVic: 360-degree video classification from mobile encrypted video traffic, in: Proceedings of the 31st ACM Workshop on Network and Operating Systems Support for Digital Audio and Video, 2021, pp. 58–65.
[26] Y. Li, Y. Huang, R. Xu, S. Seneviratne, K. Thilakarathna, A. Cheng, D. Webb, G. Jourjon, Deep content: Unveiling video streaming content from encrypted WiFi traffic, in: 2018 IEEE 17th International Symposium on Network Computing and Applications, NCA, IEEE, 2018, pp. 1–8.
[27] T.-Y. Kim, S.-B. Cho, Web traffic anomaly detection using C-LSTM neural networks, Expert Syst. Appl. 106 (2018) 66–76.
[28] G. Marín, P. Casas, G. Capdehourat, Rawpower: Deep learning based anomaly detection from raw network traffic measurements, in: Proceedings of the ACM SIGCOMM 2018 Conference on Posters and Demos, 2018, pp. 75–77.
[29] C. Kattadige, K.N. Choi, A. Wijesinghe, A. Nama, K. Thilakarathna, S. Seneviratne, G. Jourjon, SETA++: Real-time scalable encrypted traffic analytics in multi-Gbps networks, IEEE Trans. Netw. Serv. Manag. 18 (3) (2021) 3244–3259.
[30] K. Choi, A. Wijesinghe, C. Kattadige, K. Thilakarathna, S. Seneviratne, G. Jourjon, SETA: Scalable encrypted traffic analytics inmulti-Gbps networks, in: 2020 IEEE 17th International Conference on Local Computer Networks, LCN, IEEE, 2020.
[31] R. Corvi, D. Cozzolino, G. Zingarini, G. Poggi, K. Nagano, L. Verdoliva, On the detection of synthetic images generated by diffusion models, in: ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, IEEE, 2023, pp. 1–5.
[32] OpenCV, Geometric image transformations, 2023, https://docs.opencv.org/3.4/da/d54/group_imgproc_transform.html. (Online; Accessed 19 June 2023).
[33] X. Wan, J. Luo, Y. Liu, J. Xu, X. Mei, Y. Chen, A novel arrhythmia classification method based on GASF and optimized inception-resnet-v2, Available at SSRN 4176688.
[34] P. Sirinam, M. Imani, M. Juarez, M. Wright, Deep fingerprinting: Undermining website fingerprinting defenses with deep learning, in: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, 2018, pp. 1928–1943.
[35] K.N. Choi, T. Dahanayaka, D. Kennedy, K. Thilakarathna, S. Seneviratne, S.S. Kanhere, P. Mohapatra, Passive activity classification of smart homes through wireless packet sniffing, in: 2020 19th ACM/IEEE International Conference on Information Processing in Sensor Networks, IPSN, IEEE, 2020, pp. 347–348.
[36] Google, Compare smart displays, URL https://store.google.com/au/magazine/compare_nest_hub?hl=en-GB.
[37] Alexa, Amazon Alexa, URL https://alexa.amazon.com/.
[38] C. Esteban, S.L. Hyland, G. Rätsch, Real-valued (medical) time series generation with recurrent conditional GANs, 2017, arXiv preprint arXiv:1706.02633.
[39] J. Yoon, D. Jarrett, M. Van der Schaar, Time-series generative adversarial networks, in: Advances in Neural Information Processing Systems, vol. 32, 2019.
[40] A. Cheng, PAC-GAN: Packet generation of network traffic using generative adversarial networks, in: 2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference, IEMCON, IEEE, 2019, pp. 0728–0734.
[41] S. Xu, M. Marwah, N. Ramakrishnan, Stan: Synthetic network traffic generation using autoregressive neural models, 2020, ArXiv Preprint arXiv:2009.12740.
[42] M.J. Chong, D. Forsyth, Effectively unbiased fid and inception score and where to find them, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 6070–6079.
[43] S. Ravuri, O. Vinyals, Classification accuracy score for conditional generative models, in: Advances in Neural Information Processing Systems, vol. 32, 2019.
[44] M. Sensoy, L. Kaplan, M. Kandemir, Evidential deep learning to quantify classification uncertainty, in: Advances in Neural Information Processing Systems, vol. 31, 2018.
[45] A. Blattmann, R. Rombach, H. Ling, T. Dockhorn, S.W. Kim, S. Fidler, K. Kreis, Align your latents: High-resolution video synthesis with latent diffusion models, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 22563–22575.
[46] P. Schramowski, M. Brack, B. Deiseroth, K. Kersting, Safe latent diffusion: Mitigating inappropriate degeneration in diffusion models, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 22522–22531.
[47] C. Madarasingha, K. Thilakarathna, A. Zomaya, OpCASH: Optimized utilization of MEC cache for 360-degree video streaming with dynamic tiling, in: 2022 IEEE International Conference on Pervasive Computing and Communications, PerCom, IEEE, 2022, pp. 34–43.

Nirhoshan Sivaroopan has completed his BSc (Hons) in Electronics Engineering from Electronic and Telecommunication Engineering at the University of Moratuwa. He completed an internship under the supervision of Dr. Kanchana Thilakarathna, focusing on applications of diffusion models for privacy aware deep learning models at the University of Sydney, where he served as a visiting research affiliate.

Dumindu Bandara is a final year undergraduate student pursuing a degree in Electronic and Telecommunication Engineering at the University of Moratuwa. He completed an internship under the supervision of Dr. Kanchana Thilakarathna, focusing on applications of diffusion models for privacy aware deep learning models at the University of Sydney, where he served as a visiting research affiliate. Dumindu's research interests encompass generative deep learning models, privacy concerns in deep learning models, hardware acceleration of deep learning models, and signal processing in graph domains.

Chamara Madarasingha is a Research Associate at School of Electrical Engineering and Telecommunications at University of New South Wales, Australia. He completed his PhD in Computer Science at the School of Computer Science at the University of Sydney and his BSc (Hons) from the University of Moratuwa, Sri Lanka. His current research interests include 360-Degree/VR video streaming optimization, Edge computing and networks, Privacy and Security.

Guillaume Jourjon is a principal scientist at CSIRO. He received his Ph.D. from the University of New South Wales and the Toulouse University of Science in 2008. Before his Ph.D., he received an engineering degree from the ISAE. He also received a DEUG in Physics and Chemistry (Major) and Mathematics (Minor) from the University of Toulouse III. His research areas of interest are related to Distributed Computing, Software Defined Network, in-network Computing, and the Security and Privacy of Networked Systems.

Anura P. Jayasumana is a Professor in Electrical & Computer Engineering at Colorado State University. He received Ph.D. and M.S. in Electrical Engineering from Michigan State University and B.Sc. in Electronic and Telecommunications Engineering with First Class Honors from University of Moratuwa, Sri Lanka. His current research interests include Internet of Things, detection of weak distributed patterns, mining of network-based data for radicalization detection, and synthetic data generation. He is currently an ACM Distinguished Speaker.

Kanchana Thilakarathna is a Senior Lecturer at the School of Computer Science, The University of Sydney. He was a visiting academic at the Electrical and Computer Engineering at the University of Maryland. He received his PhD from University of New South Wales in 2015 with the Malcolm Chaikin Prize, awarded to the Best Engineering PhD Thesis, and then worked as a Research Scientist at the Information Security and Privacy research group at Data61-CSIRO, Australia. His research interests include cybersecurity, network security, data privacy, mobile and distributed computing. He is a recipient of many research grants from industry and government organizations including Google Faculty Award and Facebook/Meta Research Awards.