Computers & Security 145 (2024) 103993

Contents lists available at ScienceDirect

Computers & Security

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

Benchmarking of synthetic network data: Reviewing challenges and approaches

Maximilian Wolf [a,*], Julian Tritscher [b], Dieter Landes [a], Andreas Hotho [b], Daniel Schlör [b]

[a] Center for Responsible Artificial Intelligence, University of Applied Sciences and Arts Coburg, Friedrich-Streib-Str. 2, Coburg, Germany
[b] Center for Artificial Intelligence and Data Science, University of Würzburg, Campus Hubland Nord, Emil-Fischer-Straße 50, Würzburg, Germany

ARTICLE INFO

Keywords:
NetFlow
Synthetic data
Generator
GPT
GAN
Benchmark
Evaluation

ABSTRACT

The development of Network Intrusion Detection Systems (NIDS) requires labeled network traffic, especially to train and evaluate machine learning approaches. Besides the recording of traffic, the generation of traffic via generative models is a promising approach to obtain vast amounts of labeled data. There exist various machine learning approaches for data generation, but the assessment of the data quality is complex and not standardized. The lack of common quality criteria complicates the comparison of synthetic data generation approaches and synthetic data.

Our work addresses this gap in multiple steps. Firstly, we review and categorize existing approaches for evaluating synthetic data in the network traffic domain and other data domains as well. Secondly, based on our review, we compile a setup of metrics that are suitable for the NetFlow domain, which we aggregate into two metrics Data Dissimilarity Score and Domain Dissimilarity Score. Thirdly, we evaluate the proposed metrics on real world data sets, to demonstrate their ability to distinguish between samples from different data sets. As a final step, we conduct a case study to demonstrate the application of the metrics for the evaluation of synthetic data. We calculate the metrics on samples from real NetFlow data sets to define an upper and lower bound for inter- and intra-data set similarity scores. Afterward, we generate synthetic data via Generative Adversarial Network (GAN) and Generative Pre-trained Transformer 2 (GPT-2) and apply the metrics to these synthetic data and incorporate these lower bound baseline results to obtain an objective benchmark. The application of the benchmarking process is demonstrated on three NetFlow benchmark data sets, NF-CSE-CIC-IDS2018, NF-ToN-IoT and NF-UNSW-NB15. Our demonstration indicates that this benchmark framework captures the differences in similarity between real world data and synthetic data of varying quality well, and can therefore be used to assess the quality of generated synthetic data.

1. Introduction

Problem. The NetFlow format (Claise, 2004) is widely used in the cybersecurity domain in the field of NIDS (Ring et al., 2019). In order to build, test, and evaluate these systems, labeled benchmark data sets are required. Due to privacy concerns and the complexity of labeling real world data, such data are available only to a limited extent (Ring et al., 2018). Moreover, to apply self-learning systems, training data tailored to the monitored network architecture are required. The lack of labeled data can be reduced by applying synthetic data generators that can be used to enrich real world data, or create data with similar properties, without the disclosure of sensitive network information. There are several existing approaches to generate synthetic NetFlow data (Ring et al., 2018; Manocchio et al., 2021; Yin et al., 2022), but there is no consensus in objectively evaluating the synthetic data generated, making it difficult to compare generation approaches (Goncalves et al., 2020). This lack of evaluation guidelines for synthetic data has also been noted by Borji (2021), Dankar et al. (2022) and Koochali et al. (2022). In essence, one wants to determine whether the generated data are similar to real world data and embody the properties of real world data. This is particularly important when comparing various approaches for synthetic data generation based on the data outputs. The missing guidelines and differences in the evaluation methodologies within the literature make objective evaluations of data and comparisons between generator approaches challenging.

Objective. This work takes a step further to close this gap by proposing a multi-metric evaluation framework for synthetic NetFlow data. Firstly, an overview of synthetic data evaluation metrics has to be

[*] Corresponding author.
E-mail addresses: maximilian.wolf@hs-coburg.de (M. Wolf), tritscher@informatik.uni-wuerzburg.de (J. Tritscher), dieter.landes@hs-coburg.de (D. Landes), hotho@informatik.uni-wuerzburg.de (A. Hotho), schloer@informatik.uni-wuerzburg.de (D. Schlör).

https://doi.org/10.1016/j.cose.2024.103993
Received 25 January 2024; Received in revised form 10 May 2024; Accepted 4 July 2024
Available online 9 July 2024

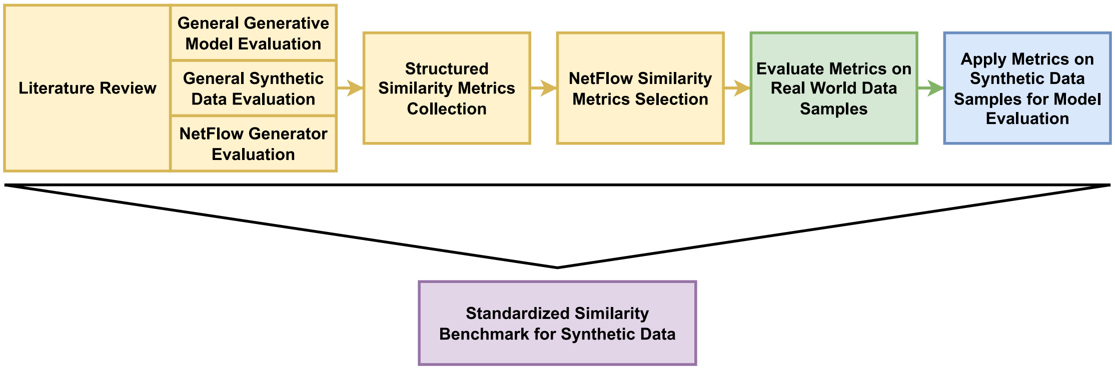

Literature Review
General Generative Model Evaluation
General Synthetic Data Evaluation
NetFlow Generator Evaluation
Structured Similarity Metrics Collection
NetFlow Similarity Metrics Selection
Evaluate Metrics on Real World Data Samples
Apply Metrics on Synthetic Data Samples for Model Evaluation
Standardized Similarity Benchmark for Synthetic Data

Fig. 1. A structured and step-wise overview of the paper's approach.

compiled. Second, metrics have to be selected that are suitable for the NetFlow domain to create a standardized benchmark. Thirdly, the suitability has to be tested on real NetFlow data to evaluate how well they determine similarity and dissimilarity of NetFlow data. Finally, the metrics need to be applied to synthetic data to validate the practicability of the framework.

Approach. The approach of this work is structured in different steps, which are depicted in Fig. 1 and feature three main contributions. Firstly, our literature review on synthetic data evaluation and the evaluation of generative models reveals a lack of standardization in assessing synthetic data.

Secondly, based on this review, we compile a structured collection of metrics to create our benchmark framework. Our metric selection for the benchmark incorporates data-driven measures regarding, for example, data distribution, correlations and population characteristics, and domain-driven metrics regarding, for example, syntax checks and the application for NIDS.

Thirdly, our validation on real data samples shows that the selected metrics effectively distinguish samples from different data sets, while samples from the same set yield high similarity scores. We create baseline similarity scores for specific data sets that are helpful for synthetic data evaluation.

Finally, we generate synthetic data based on established generative models, namely GAN (Goodfellow et al., 2014; Gulrajani et al., 2017) and Generative Pre-trained Transformer 2 (GPT-2) (Radford et al., 2019). The models are trained on NetFlow benchmark data sets, which are designed for the creation and testing of NIDS. We generate data samples during the training process of the generative model, which allows the evaluation of the overall training process in terms of synthetic data quality. Finally, the similarity metrics are applied to synthetic data to assess the data quality objectively.

Overall, the benchmark presented in this work provides an objective, standardized, and model-agnostic assessment process for evaluating and comparing synthetic data generators. In summary, we contribute a benchmark for synthetic NetFlow data with meaningful baseline references. This allows the evaluation of the training process of generative models by utilizing data similarity to real data to objectively compare different generative models.

Contributions. The contributions of this work can be summarized as follows:

1. We conducted a comprehensive literature review on synthetic NetFlow quality assessment and discuss the suitability of established metrics on NetFlow data.
2. Using the reviewed metrics, we construct a benchmarking system.
3. We demonstrate that the selected metrics effectively distinguish samples from different real NetFlow data sets.
4. We conduct a case study of our framework, with synthetic data from two generative models to demonstrate the application of the framework on synthetically generated NetFlow data.
5. We publish our code as well as our benchmark data so that it can be used by other researchers to evaluate their generated data.

Structure. The remainder of the paper is structured as follows. We review current benchmark approaches for synthetic data from diverse data domains and explicitly from the NetFlow domain in Section 2. Then, we define the fundamentals for our benchmark in Section 3. Next, we collect general approaches to measure quality and data similarity from literature and discuss them in the context of NetFlow data in Section 4. Afterward, we define the benchmark setup for the evaluation of the synthetic data in Section 5. Finally, we test our selected metrics on public NetFlow benchmark data sets in Section 6 and demonstrate our benchmark setup for synthetic data, which are created by generative models in Section 7. Our conclusions and future research directions are given in Section 8.

2. Related work

The generation and evaluation of synthetic data are ongoing areas of research. Generally, synthetic data should accurately capture the properties and distribution of the original data, while avoiding the creation of exact duplicates of the training data and still being realistic overall. This section discusses previous work which targets the evaluation of multiple synthetic data sets or generator models. This section discusses multiple aspects of synthetic data evaluation. At first, we briefly discuss model-specific evaluation settings for GANs. Afterward, related works, which comprehensively evaluate synthetic data in other domains, e.g. medicine and finance, are reviewed. Finally, we analyze approaches of the NetFlow domain, where synthetic data are evaluated based on different evaluation metrics.

2.1. Evaluation of models (GANs)

Borji (2021) presents an updated overview of various approaches for the evaluation of GAN-based data generators. As image generation constitutes a significant segment within synthetic data generation, the majority of methodologies focus on assessing image data based on fidelity and diversity. The author concludes that, for example, the Inception Score (IS) (Salimans et al., 2016), Fréchet Inception Distance (FID) (Heusel et al., 2017), Precision and Recall (P&R) (Sajjadi et al., 2018), and Perceptual Path Length (PPL) (Karras et al., 2018) are popular metrics, but objective and comprehensive evaluation needs further research in the future. IS and FID require a pre-trained classifier (InceptionNet) trained on ImageNet, which is available in the image domain, but not in the domain of NetFlows. The metrics of Precision and Recall are leveraged to evaluate the GAN's generator. These metrics use the pre-trained InceptionNet to obtain classification results. The similarity of generated data and real data is evaluated via precision, while the coverage of the data distribution is measured by recall to detect mode collapse, for example. PPL evaluates the entanglement of the generator's latent space, where a less curved latent space shall generate perceptually smoother transitions, in contrast to a highly curved one. This measure is strongly related to the models' architecture (GANs) and cannot be applied to synthetic data from other models in general. This work highlights the necessity for a comprehensive evaluation model, yet its approach, being specific to GANs and focused on the image domain, falls short of the required universality and transferability to other models and domains. Our benchmark focuses on creating an evaluation method that goes beyond the image domain, offering a versatile, model-agnostic approach suitable for the unique aspects of NetFlow data.

2.2. Synthetic data evaluation benchmarks in other domains

Choi et al. (2017) generates synthetic patient records in the medical domain. They apply an adapted GAN model called medGAN to generate data with high-dimensional discrete variables. The synthetic data are evaluated through a qualitative evaluation by a medical doctor and a privacy risk evaluation based on presence and attribute disclosure. Patient data are subject to data protection, which requires that data generators, trained on real data, need to be checked for data disclosure. The authors examine two variants of disclosure, presence disclosure, and attribute disclosure based on the calculation of hamming distances between patient data records. They compare their medGAN to different GAN variants and other generative models like Variational Autoencoders and demonstrate that the medGAN is able to generate high quality data, while showing a limited risk of privacy data disclosure. In contrast to their setup, our benchmark does not include the subjective evaluation via human experts. The application of the Hamming distance is defined for numerical values only and cannot be directly applied to NetFlows, which include numerical and categorical attributes.

Goncalves et al. (2020) focus the generation of synthetic medical data targeting cancer classification. They compare different models for synthetic data generation, which are probabilistic models, classification-based imputation models, and GANs. They compare the generated data based on different metrics like Cluster-Analysis, Support Coverage, Pearson Correlation, Privacy Disclosure, Train on Synthetic, Test on Real (TSTR) and Train on Real Test on Synthetic (TRTS) classification results. They conclude that none of the tested generation approaches outperforms the others in all metrics. In our benchmark we incorporate some of their evaluation metrics like Pearson Correlation, TSTR and TRTS, the selection process will be discussed in greater detail in the next sections.

Ehrhart et al. (2022) apply conditional GANs for the generation of physiological sensor data for stress detection. They focus on the generation of time series sensor data which are evaluated via visual inspection of the time series data as individual data points and sequences, as well as expert evaluation based on five experts from the field of human sensing from various domains of physiology. Additionally, they apply a TSTR (called TGTR in their work) and a Classifier Two-Sample Test. They compare different GAN-Architectures and conclude that the unstable training behavior of GANs hinder a search of optimal hyperparameters. The mode collapse, of the GAN model complicates the synthetic data generation further. They suggest the application of alternate GAN architectures like Wasserstein Generative Adversarial Network (WGAN) for future research. In accordance with their work, we apply a WGAN for synthetic data generation, as well as the evaluation by TSTR, but extend the evaluation with additional metrics.

Koochali et al. (2022) suggest standardized assessment methods for the evaluation of synthetic time series data. They adapt widely used image domain measurements like Inceptions Score (IS) (Salimans et al., 2016) and Fréchet Inception Distance (FID) (Heusel et al., 2017) to the time series domain, resulting in the measures Inception Time Score (ITS) and Fréchet Inception Time Distance (FITD). These metrics are based on IS and FID which require a pre-trained established classifier. The evaluation of generated data is supplemented by a classifier-based evaluation in the settings of TSTR and TRTS As a showcase for the evaluation protocol, a conditional GAN is trained and tested on 80 different time-series data sets from the UCR-Dataset repository (Chen et al., 2015). The TSTR and TRTS settings will be applied in our experiments to evaluate the application of synthetic data in the context of anomaly detection-based NIDS.

Dankar et al. (2022) give an overview of different metrics for the evaluation of synthetic data. They categorize the metrics in four main classes, namely attribute fidelity, bivariate fidelity, population fidelity, and application fidelity. Finally, the synthetic data of four different data generators and 19 data sets are evaluated using the quality metrics Hellinger Distance for attribute fidelity, Pairwise Correlation Distance for bivariate fidelity, Propensity Score, and Prediction Accuracy for population fidelity to assess the overall utility of synthetic data generators. The comparison of the four selected metrics suggests that there is no strong correlation between the individual metrics. Therefore, all metrics should be considered for multivariate quality analysis of synthetic data. In our work, the hierarchy of data-driven similarity metrics (Fig. 2) is influenced by their metric categorization.

Schlör (2022) focuses on anomaly detection in transactional data. Variational Autoencoder and GAN are used to generate synthetic transaction data, which are evaluated in multiple ways. Firstly, the probability distributions of the synthetic data points are estimated via Kernel Density Estimation, where a Gaussian Kernel is applied. Next, the probability distributions are compared via Jensen–Shannon divergence and the Earth-Movers Distance. Secondly, the feature correlations of the real and the synthetic data are compared via the Mean Absolute Error. The correlational analysis includes Pearson's correlation coefficient, Uncertainty Coefficient and Correlation Ratio. Thirdly, the data distributions are plotted in violin plots and histograms for visual comparison. While other studies primarily focus on Pearson's correlation coefficient, Schlör (2022) broadens the scope by also analyzing correlations in both numerical and categorical data. Our benchmark setup incorporates this comprehensive feature correlational analysis for synthetic data, since NetFlow data consist of a mixture of numerical and categorical features and their analysis is an important aspect of data quality.

2.3. Evaluation of NetFlow generator models

Ring et al. (2018) evaluate the effect of data preprocessing on the generation of NetFlow traffic via WGAN. They test a normalized version of input data, a binary transformation of data, and an embedding version. Their evaluation setup includes a visual comparison of distributions, syntax checks, and the calculation of Euclidean distances based on statistical values. Their experiments indicate that the binary transformation and the embedding version work well for synthetic data generation.

This work applied data-driven aspects, e.g. the comparison of distributions and domain-driven aspects, e.g. syntax checks for the evaluation. While syntax checks can be applied as objective measurement, visual comparison of distributions is a subjective measure since it depends on the viewers' opinion.

Guo et al. (2021) apply GANs to generate synthetic network data for oversampling. Their goal is to use synthetic data for oversampling minority classes, thereby enhancing the performance of classification models. Their experimental setup compares their approach against various other oversampling methods such as random oversampling. They evaluated their approach through classification performance (F1-Score and AUC-PR) of the models, for which the training data were oversampled by the tested approaches. Their evaluation methodology targets the application of synthetic data exclusively. Their experiments show that their approach surpasses the other oversampling techniques.

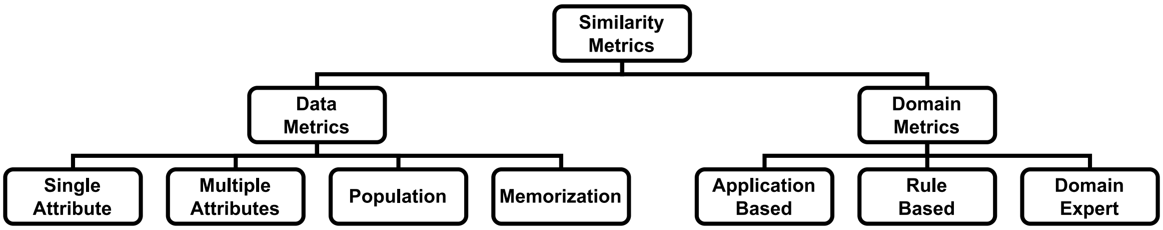


Fig. 2. A structured overview of different similarity metrics.

Liu et al. (2021) applies a WGAN as an oversampling approach as well. Their approach is tested against other oversampling methods like Random oversampling and different variants of SMOTE (Chawla et al., 2002). The oversampling methods are tested on the data sets NSL-KDD (Tavallaee et al., 2009), UNSW-NB15 (Moustafa and Slay, 2015), and CICIDS-2017 (Sharafaldin et al., 2018b). In their experiments they apply different classifiers, e.g. Naive Bayes, Decision Tree, Random Forest, Gradient Boosting Decision Tree, and Support Vector Machine. Classification performance is evaluated via standard classification metrics namely accuracy, precision, recall, and F1-Score. They report that their oversampling method can effectively improve detection performance. The application of synthetic data for oversampling targets the generation of proxy data that can be applied instead of real data. Our work will first focus on this notion, since the generation of proxy data is a more general application setting.

Charlier et al. (2019) apply GANs for the generation of synthetic network attacks, which is closely related to oversampling, since attack data are commonly represented in minority classes. They test their model on two benchmark data sets, namely NSL-KDD (Tavallaee et al., 2009) and CICIDS2017 (Sharafaldin et al., 2018b). The generated data are evaluated via visual inspection of histograms for each attribute of the generated data. They report an adequate convergence of real and synthetic data in their results. The evaluation methodology of this work will not be used, as visual inspection is not objective.

Yin et al. (2022) compare different GAN-based design choices to build their NetShare model for the generation of synthetic network data based on packet or NetFlow traces. They compare several GAN models from previous works to their approach on multiple data sets. The results are evaluated based on distributional measures of different attributes using empirical cumulative distribution function (CDF) plots, distance measures like Jensen–Shannon divergence or Earth movers distance, and via the classification quality based on classifiers in the TSTR setting. Our work will not only include distributional measures and the testing of data application through TSTR, but also extend this concept to a broader range of metrics compared to their work.

Nekvi et al. (2023) focus on the generation of synthetic IoT network traffic with the original GAN. They use a self-recorded data set consisting of IoT-traffic and DDoS attacks. In their experiments, they evaluate the effect of the batch size on the training behavior and generate data after each training epoch of the GAN model. They evaluate data quality exclusively through TSTR and report that it is a good metric to identify quality data during the training and generation process. Due to the inconsistent output quality of the GAN during the training process, it is necessary to identify high-quality NetFlow batches in retrospect. The TSTR setup is based on four classification models used for binary classification (normal or attack) that are evaluated via accuracy. In contrast to this work, our work evaluates the synthetic data with TSTR as well but extends the evaluation to many more metrics.

Kholgh and Kostakos (2023) apply the OpenAI's GPT-3 API, where they construct a pipeline for synthetic traffic generation. The generation model is based on GPT-3 variants that have been fine-tuned for packet data generation. The pipeline contains the following steps: The user requests specific traffic scenario like normal traffic, or certain attack scenarios like Ping-of-Death. A Flow generator collects NetFlow traffic, given the specified type of traffic, and parses it into text. Finally, the NetFlow text is given to the Transformer model to create packet-based data in text format based on the NetFlow text. The packets are parsed in the packet format afterward. They test their approach on the public available benchmark data sets DARPA, KDD99 and TON-IoT. They evaluate their model based on loss and accuracy and the generated packet data via the so-called success rate, which measures packets that can be sent to the Internet and trigger a form of reply like a Ping, DNS query or answer from an HTTP server, etc. Our approach focuses on the generation of NetFlow data instead of packet-based data, as well as the application and comparison of open-source generative models.

2.4. Conclusion on related work

In general, GAN models are a popular choice for synthetic data generation for NetFlow and other domains and will be applied in our benchmark case study. Moreover, this overview highlights the diversity and inconsistency in evaluation methodologies for synthetic data, particularly when it comes to network or NetFlow data, which emphasizes the need for an objective measurement to systematically evaluate the quality of synthetic data sets and generators. Additionally, there are various relevant theoretical dimensions for the evaluation of synthetic data and generative model quality in other domains that need to be explored and validated in the NetFlow domain. While some statistical measurements like distance measures and correlations within data can be generally applied to different types of data, domain-specific measures or tasks are required as well to evaluate generated data comprehensively. The demand for a comprehensive evaluation setup for synthetic data is also stated in various works of other data domains (Borji, 2021; Dankar et al., 2022; Koochali et al., 2022).

A brief survey on evaluation metrics is given in Section 4, where several metrics discussed in this section are presented in detail.

3. Foundations

This section outlines the key elements of the data and machine learning techniques employed in our benchmark. In the following, the NetFlow file format and synthetic data are defined. Next, we describe two generative models, the Wasserstein GAN and GPT-2, which are employed to synthesize data.

3.1. NetFlow

The NetFlow file format (Claise, 2004) describes the data exchanged in a session between the source and destination IP in an aggregated format. Therefore, the meta-information of the connection is aggregated in time periods. NetFlow itself contains meta-information such as the duration of transmissions, the transport protocol, the ports of source and destination, the bytes sent and the amount of packets sent (Claise, 2004). Unlike packet-based traffic captures, NetFlow does not contain any payload and requires less storage capacity.

There exist different tools to capture and convert network traffic into specified NetFlow formats, therefore, the NetFlow formats of published NetFlow data sets can differ. Since our work will focus on measures for data similarity, we have to evaluate our measures on data, which are converted by the same NetFlow converter into the same format, to avoid the risk of converter artifacts affecting our similarity measures. The data used in the benchmark are NetFlows from the data sets NF-CSE-CIC-IDS2018, NF-ToN-IoT and NF-UNSW-NB15 based on Sarhan et al. (2021), who converted the original data sets with the same NetFlow-Converter to the same NetFlow-Format for comparability.

3.2. Synthetic data

Dankar et al. (2022) define synthetic data based on summarized definitions of previous works (Hu, 2018; Ruiz et al., 2018; Park et al., 2018), as artificial data that mimic the statistical properties of real data, without containing identifiable information about real data. In addition to creating a shareable and privacy-safe data set for further experimentation, synthetic data can be applied to augment existing data, completely replace real data, or act as a reasonable proxy for real data (Goncalves et al., 2020). Our benchmark focuses on the ability to mimic the statistical properties of real data and applying synthetic data as a replacement for, or a reasonable proxy to, real data.

3.3. Wasserstein generative adversarial network

The GAN architecture, introduced by Goodfellow et al. (2014) is a generative model which consists of two components, Generator (G) and Discriminator (D). The model is trained through an adversarial game, where G tries to create fake data that mimic real data, while D tries to distinguish real from fake data. G receives noise as input and attempts to generate data that resemble real data from noise. D is provided with real or fake data and must differentiate between both. Based on D's loss, both networks are updated.

Based on the original GAN, multiple model variants attempt to increase its performance. The WGAN (Arjovsky et al., 2017) is similar to its original, but the original loss function is updated to the Wasserstein Distance. In addition to improved training stability, this loss function allows the acwgan to model discrete distributions over latent spaces. While Arjovsky's implementation uses weight clipping to enforce differentiability of the Wasserstein loss, Gulrajani et al. (2017) apply a gradient based penalty to enforce it via a soft constraint.

The GAN models are originally applied in the domain of image synthesis but were adapted to other domains, e.g. NetFlows (Ring et al., 2018) as well. In our benchmark, the model is applied to NetFlow data which are encoded into continuous numerical representations, since neuronal networks are not able to process categorical attributes by default. The generated data are then decoded into the original NetFlow format, before they are evaluated.

3.4. Generative pretrained transformer 2

GPT-2 (Radford et al., 2019) is a large-scale language model based on multiple transformer decoder blocks. The model has been successfully applied to various tasks in the domain of natural language processing such as reading comprehension, question answering, and translation (Radford et al., 2019).

The training of the model consists of two phases, which are pre-training and fine-tuning. The self-supervised pre-training is applied via next token predictions, where the model predicts the next token given a sequence of previous tokens. This allows GPT-2 to model the overall data distribution. Afterward, the model can be fine-tuned with labeled data for specific tasks. Our work will focus exclusively on the generative properties of the pre-training phase to generate NetFlow data.

4. Evaluation metrics review

This section targets the literature review for synthetic NetFlow quality assessment. The evaluation of synthetic data in general is an ongoing research topic. According to Dankar et al. (2022) there is currently no general guideline on evaluation. Based on our research, there are multiple metrics to measure the similarity of real and synthetic data. The following survey covers various aspects of quality metrics from previous works that have been applied to determine the similarity between synthetic and real data in the domain of NetFlow data, as well as other data domains like images or medical data. Table 1 lists various approaches for evaluating synthetic data, highlighting the lack of common evaluation guidelines, as evidenced by the minimal overlap of the metrics applied in different studies. Furthermore, Table 1 highlights several metrics that were already applied in other domains but not yet in the domain of NetFlows.

The various measures identified in our survey, can be hierarchically categorized, as depicted in Fig. 2. There exist two main categories of similarity metrics. The first category is entirely data-driven and independent of the data domain, while the second category exploits domain knowledge to evaluate synthetic data. The diagram in Fig. 2 gives a structured overview, while the details for each category are discussed in the following.

4.1. Data metrics

In order to compare real data with synthetic data, data-based measures can be applied to compare various characteristics of data sets. These measures can be applied to a broad range of data sets, since the metrics are domain independent.

4.1.1. Measures for single attributes

The attribute fidelity of data sets can be compared by using different types of distance measures, that compare the distance of real and synthetic data. The scope of the single attribute metric is to compare the properties e.g. the distribution of single attributes in the multivariate data distribution to determine their similarity. Common choices for these distance measurements are based on probability distributions, such as Kullback–Leibler divergence (KLD) (Kullback and Leibler, 1951) or Jensen–Shannon divergence (JSD).

Probability distributions from data sets. Metrics that compare the similarity of distributions require a distribution P_D generated from sampled data points D in the application of measures for distributional divergence. In our setting, a Gaussian Kernel-Density-Estimation (Scott, 2015) $KDE(D_x)$ is applied for numerical features D_x of D. Categorical Attributes D_x are converted into an empirical distribution via relative frequency counts called $P_c(D)$ according to the work of (Yin et al., 2022).

$$P_D = \begin{cases} KDE(D_x) & \text{if } D_x \text{ numerical} \\ P_c(D_x) & \text{if } D_x \text{ categorical} \end{cases} \tag{1}$$

Kullback–Leibler divergence. Given the observed values of synthetically generated data as a probability distribution P_S and some known realistic data as probability distribution P_R, the KLD allows the comparison of both probability distributions through measuring the relative entropy, and can be used as a distance through

$$d_{\mathrm{KL}_X}(P_R \parallel P_S) = \sum_{x \in X} P_R(x) \log \frac{P_R(x)}{P_S(x)} \tag{2}$$

for the sample space X. The resulting KLD is a widely established distance function that is commonly used as a measure in generative settings (Dankar et al., 2022).

Table 1
Literature review of the evaluation metrics used for synthetic data from the NetFlow(highlighted via ^{NF}) or other domains, references in the paper column refer to their application in earlier works.

Category	Metric	Paper
Single attributes	Jensen–Shannon divergence	Schlör (2022) and Yin et al. (2022)^{NF}
	Hellinger distance	Dankar et al. (2022)
	Wasserstein/Earth-Movers distance	Yin et al. (2022)^{NF}
	Support-Coverage	Goncalves et al. (2020)
Multiple attributes	Pearson correlation	Goncalves et al. (2020), Dankar et al. (2022) and Schlör (2022)
	Correlation ratio	Schlör (2022)
	Uncertainty coefficient (Theils U)	Schlör (2022)
	Pairwise Correlation Distance Pearson	Goncalves et al. (2020) and Dankar et al. (2022)
	Mean absolute error	Schlör (2022)
Population	Discriminator	Goodfellow et al. (2014)
	Classifier-Two-Sample Test	Lopez-Paz and Oquab (2016) and Ehrhart et al. (2022)
	Propensity score	Dankar et al. (2022)
	Euclidean distance	Ring et al. (2018)^{NF}
	Cluster analysis	Woo et al. (2009) and Goncalves et al. (2020)
Memorization	Memorization-Informed-Frechet Inception Distance	Bai et al. (2021)
	Presence-Disclosure	Choi et al. (2017) and Goncalves et al. (2020)
	Attribute-Disclosure	Choi et al. (2017) and Goncalves et al. (2020)
Rule based	Syntax checks	Ring et al. (2018)^{NF}
Domain expert	Visual comparison of distributional plots	Ring et al. (2018)^{NF}, Charlier et al. (2019)^{NF} and Schlör (2022)
	Visual inspection of data	Choi et al. (2017) and Ehrhart et al. (2022)
Application based	Train on Real, Test on Synthetic (TRTS)	Goncalves et al. (2020) and Dankar et al. (2022)
	Train on Synthetic, Test on Real (TSTR)	Goncalves et al. (2020) and Dankar et al. (2022) Yin et al. (2022)^{NF} and Nekvi et al. (2023)^{NF}
	Oversampling	Guo et al. (2021)^{NF} and Liu et al. (2021)^{NF}

Jensen–Shannon divergence. JSD is a bounded symmetric variant of KLD, whose square root may be used as a distance metric, in contrast to KLD satisfying the triangle inequality. It is defined by

$$\underset{X}{d_{\mathrm{JS}}}\left(P_R \parallel P_S\right) = \sqrt{\frac{1}{2}\left(\mathrm{KL}_X(P_R \parallel P_M) + \mathrm{KL}_X(P_S \parallel P_M)\right)} \tag{3}$$

where

$$P_M = \frac{1}{2}(P_R + P_S) \tag{4}$$

JSD constitutes another popular choice for evaluating data generation (Yin et al., 2022). Next, we calculate the average over all features i of D, where each feature is transformed into a probability distribution P_D^i

$$\overline{d}_{JS}(R,S) == \frac{1}{n}\sum_{i}^{n}\left|\underset{X}{d_{\mathrm{JS}}}\left(P_R^i \parallel P_S^i\right)\right| \tag{5}$$

Schlör (2022) generates synthetic transactional data and applies JSD to compare the distributions of real and synthetic data for the comparison of different generators. Yin et al. (2022) generates synthetic packet-based and NetFlow data via their NetShare model to compare it with various GAN-based approaches. One of their applied metrics is the JSD.

Earth mover's/Wasserstein-1 distance. Earth Mover's Distance (EMD) distance (sometimes also referred to as Wasserstein-1 distance) represents the necessary probability weight to move when transforming probability distribution Q to probability distribution P, and is formally given as

$$d_{\mathrm{EM}}(R,S) = \inf_{\gamma \in \Pi(R,S)} \underset{(x,y)\sim\gamma}{\mathbb{E}} \left[\|x-y\|\right], \tag{6}$$

with $\Pi(P,Q)$ denoting the set of all possible joint distributions $\gamma(x,y)$ with marginals P and Q. Here, the infimum finds the way of distributing the probability weight between both probability distributions with the least cost.

Yin et al. (2022) use the EMD as another evaluation metric for the performance comparison of several generator models for NetFlows. Schlör (2022) does not use the EMD. The author argues that this metric is an optimization metric applied in the WGAN and therefore, this model is explicitly optimized toward this metric, which prevents a fair comparison of WGAN to other models with another optimization criterion.

Support coverage. Support Coverage S_c (Goncalves et al., 2020) measures the average ratio of the cardinalities based on the support of the variable. Support represents the proportion of a particular value of a variable.

$$S_c(R,S) = \frac{1}{V}\sum^{V}\frac{|S^V|}{|R^V|} \tag{7}$$

R^V and S^V are the support of the Vth variable for the real and synthetic data and V is the set of random variables representing the variables to be generated.

Goncalves et al. (2020) generate medical data sets and use the Support Coverage to measure the difference of the support value of a variable in the real data compared to the synthetic data. They use the metric in the comparison of data generation approaches to determine if all the categories in the real data appear in the synthetic data, as well.

Hellinger distance. Hellinger distance, used by Dankar et al. (2022) for the evaluation of synthetic data, is another distance metric based on probability distributions. Given two probability distributions P_R and P_S, the Hellinger distance d_{H} is calculated via

$$d_{\mathrm{H}}(P_R, P_S) = \frac{1}{\sqrt{2}}\left\|\sqrt{P_R} - \sqrt{P_S}\right\|_2. \tag{8}$$

Dankar et al. (2022) favor this measure for the comparison of real and synthetic data distributions in the area of medical data, over the KL-Divergence since its value ranges from 0 to 1 and is therefore easy to interpret. A comparison with other metrics such as the JSD, which also has a value range of 0 to 1, is not included in their argumentation.

4.1.2. Measures for multiple attributes

Aside from matching the distribution of single attributes, synthetically generated data also need to match the relationship between multiple attributes. Here, correlations between multiple attributes are calculated separately on real and synthetic data, with correlations of synthetic data expected to closely match the observed correlations within real data. We argue that the correlations reflect the relationship between two attributes and can be applied to assess integrity of the underlying data.

This evaluation poses the additional obstacle of mixed data types. While many established correlation measures may be used to investigate the correlation of two numerical or two categorical attributes, a deeper investigation of correlations also requires measures that enable investigation of inter-correlation between numerical and categorical attributes. Our correlational measures are closely following Schlör (2022), who successfully evaluated synthetic tabular data in the domain of fraud detection systematically via various correlational measures.

Pearson's correlation coefficient. For correlations between numerical attributes, Pearson's correlation coefficient (cor_P) is an established measure that has been successfully used to evaluate synthetic data (Goncalves et al., 2020; Dankar et al., 2022; Schlör, 2022). The cor_P is calculated through the standardization of the covariance between both attributes by

$$cor_P(X,Y) = \frac{\mathbb{E}\left[(X - \mathbb{E}(X))(Y - \mathbb{E}(Y))\right]}{\sigma_X \sigma_Y} \tag{9}$$

with σ_X, σ_Y denoting the standard deviations of the observed values of two attributes, denoted as random variables X and Y. Dankar et al. (2022) and Goncalves et al. (2020) use the cor_P matrices of synthetic and real data in the medical domain and compare the two matrices based on the Pairwise Correlation Distance to measure the difference between them. Schlör (2022) applies the cor_P as well as Uncertainty Coefficient and Correlation Ratio in the area of transactional data to calculate correlation metrics of real and synthetic data, but compares the differences using the mean absolute error metric. The attributes of a NetFlow can be categorized into numerical and categorical attributes, therefore the Pearson's correlation coefficient can be used to determine correlations between numerical attributes.

Uncertainty coefficient. To calculate the correlations between categorical attributes, the uncertainty coefficient U (also referred to as Theil's U) (Theil, 1970) can be applied. With samples given from two discrete random variables X and Y, their joint distribution $P_{X,Y}(x,y)$ and conditional distribution $P_{X|Y}(x \mid y)$, the uncertainty coefficient can be calculated. The uncertainty coefficient is directly based on the entropy H of a single distribution and the conditional entropy $H(X \mid Y)$, defined as

$$H(X) = -\sum_{x \epsilon X} P(x) \log P(x) \tag{10}$$

$$H(X \mid Y) = -\sum_{x \epsilon X, y \epsilon Y} P(x,y) \log \frac{P(x,y)}{P(x)} \tag{11}$$

$$cor_U(X,Y) = 1 - \frac{H(X \mid Y)}{H(X)}. \tag{12}$$

Similar to the Pearson's correlation coefficient, the Uncertainty Coefficient can be applied to NetFlows for measuring the correlation between categorical attributes.

Correlation ratio. The correlation (cor_η) (Fisher, 1992) can be used to measure the correlation between categorical and numerical attributes, as a measure based on the variance of the numerical attribute. The cor_η divides the data of the numerical attribute Y into subsets Y_x per distinct attribute values of each categorical attribute $x \in X$, and compares the variances across these sets to the variance across all data,

$$cor_\eta(X,Y) = \frac{\sum_x |Y_x|(\bar{Y}_x - \bar{Y})^2}{\sum_y (y - \bar{Y})^2}, \tag{13}$$

where $|Y|$ denotes the number of data entries within Y, $\bar{Y} = \mathbb{E}[Y]$ denotes the mean over Y and $\bar{Y}_x = \mathbb{E}[\{Y_i : X_i = x\}]$ the mean over each $Y_i \in Y$ for which the feature X_i has the value x.

Just like the correlation metrics mentioned above, the Correlation Ratio can be applied to NetFlows for measuring the correlation between numerical and categorical attributes.

Pairwise correlation distance. The Pairwise Correlation Distance (PCD) measure how much the pairwise feature correlation for two data sets is given by calculating the difference in terms of Frobenius norm (Goncalves et al., 2020) by

$$\mathrm{PCD}(R,S) = \|cor(R) - cor(S)\|_F, \tag{14}$$

for real R and synthetic S data correlation matrices $cor(\cdot)$ that captures all pairwise feature correlations within a data set. The smaller the PCD value, the closer are the two data sets in terms of the underlying correlation measure.

Mean absolute error. Another way of aggregating the correlation matrix $cor(\cdot)$ into a single value is the Mean Absolute Error (MAE) of the two correlation matrices (Schlör, 2022), which is beneficial since the resulting values are in the interval of the underlying correlation measures and are therefore restricted to the interval of 0 to 1. It is defined as

$$\mathrm{MAE}(cor(R), cor(S)) = \frac{1}{{n_f}^2} \sum_i^{n_f} \sum_j^{n_f} \left| cor(R_i, R_j) - cor(S_i, S_j) \right| \tag{15}$$

where R_i, R_j and S_i, S_j are the features of the real R and synthetic S data at index i and j and n_f is the number of overall features.

4.1.3. Population

Discriminator. The Discriminator of GAN models, which were successfully applied to NetFlows (Ring et al., 2018; Yin et al., 2022), learns to separate real and synthetic data and is an essential part of the GAN architecture. Inspired by this discriminator concept, an unsupervised anomaly detection model can be applied to separate real and synthetic data. Synthetic data are labeled as class $synthetic$, while real data are labeled as class $real$. The model is trained on synthetic data and tested on real data. If the synthetic data are able to capture the properties of the real data, the model should classify the real data as normal in the test setting, resulting in many false positives, because the model cannot separate the two data distributions. If the synthetic data do not cover the distribution of the original data, the model will separate data outside the original data distribution as anomaly, resulting in few false positives. Overall, the Discriminator model performance is evaluated via the well-known false positive rate (FPR), where a high FPR indicates synthetic data close to the original data.

Classifier two-sample test. The Classifier Two-Sample Test (Lopez-Paz and Oquab, 2016) applies a binary classifier model. The real and synthetic data get labels according to their class (real or synthetic). The data of both classes are then used to create training and test sets for the model. Under the assumption that the train data and synthetic data have similar properties, the trained binary classifier shall not be able to separate real and synthetic data in the test set. This results in a random guess probability in a two-class classification setting. The expected accuracy score of the classifier is around 0.5. The Classifier Two-Sample Test is sensitive to the choice of the underlying binary classifier model. In essence, a binary classifier is a Discriminator model that pursues to separate the data sets.

Propensity score. Propensity Score (Woo et al., 2009), originally introduced to measure the utility of modified or anonymized data, can be applied to evaluate synthetic data from generative models shown by Dankar et al. (2022). Woo et al. (2009) evaluate different models to calculate propensity scores and conclude that the model-choice is crucial for the application of the propensity score, as a utility measure. The Propensity score is calculated in multiple steps. First, the original and masked data are combined, but the labels for the original and masked data indicate their origin. Second, the probability that a data point belongs to the masked data class is calculated, the so-called propensity score. Third, the predicted distributions of the propensity scores in the original and masked data are compared. The higher the similarity between the distributions, the greater the utility of the data should be, according to Woo et al. (2009).

Euclidean distance. The Euclidean distance measures distance between numeric vectors (Han et al., 2011). It is defined as

$$d(r,s) = \sqrt{(r_1 - s_1)^2 + \cdots + (t_p - s_p)^2} \tag{16}$$

where p represents the number of dimensions in the vectors r and s. In general, one can apply a distance function on vectors to measure their similarity under the assumption that vectors with a low distance represent vectors with similar properties. The NetFlow data are mixed type data by default, but the Euclidean distance is defined numerical vectors exclusively and requires future research to adapt this metric to mixed-type NetFlow data.

Ring et al. (2018) applied this metric to calculate the Euclidean distance between the probability distributions of real and synthetic NetFlows, thus determining the effects of data preprocessing on the data generation performance of GANs.

Cluster-analysis measure. Another measure for data population similarity is the Cluster-Analysis Measure. The Cluster-Analysis Measure is evaluated in the setting of original and masked data similarity (Woo et al., 2009) and the setting of real and synthetic data similarity (Goncalves et al., 2020). Let N_O and N_M be sizes of the two groups for a random partition of a data set. The percent of observations c belonging to group O can be calculated by

$$c = N_O/(N_O + N_M). \tag{17}$$

At first, the Cluster-Analysis Measure merges two data sets into one and the whole data are clustered with a fixed number of resulting clusters or groups (for example by k-means). Afterward the cluster measure U_C is calculated

$$U_C = \frac{1}{G}\sum_{j=1}^{G} w_j \left[\frac{n_{jO}}{n_j} - c\right]^2, \tag{18}$$

where G is the fixed number of clusters, n_j is the number of observations in the jth cluster, n_{jO} (n_{jM}) is the number of observations from the original (or masked) data in the jth cluster, w_j is the weight assigned to the jth cluster to reflect the importance of particular clusters, and c is the percentage of the observations in each cluster that belong to group N_O (original data). Large values of U_c indicate differences in the cluster memberships, which suggest differences in the distributions of the original and masked data.

According to Woo et al. (2009) this measure has several drawbacks. First, the number of clusters G is a hyperparameter, which needs to be determined through hyperparameter studies. Second, U_c does not reflect if data masking is useful, since the measure gives the ratio of cluster memberships but does not account for the separation or distance between clusters. Third, the measure does not handle the similarity of data points, e.g. when they are close to each other (distance) but assigned to different clusters. In their empirical evaluation, the propensity measure is shown to be more promising than the cluster-based measure.

4.1.4. Memorization and disclosure of original data

A common characteristic of memorization detection is the usage of distance measures to detect data points or attributes in the synthetic data that have a strong similarity or a low distance to data points from the real-world data set that were used to train the model.

Memorization-informed Frechet inception distance. The benchmark study of Bai et al. (2021) proposes a modified variant of the FID, where a memorization penalty is added. The penalty uses cosine similarity to compare images, where images that are closer to original ones than a threshold value are penalized. Their approach is specialized on the image domain, where pre-trained image classifiers are used (Szegedy et al., 2014) that cannot be adopted to the NetFlow domain.

Attribute- and membership disclosure. Choi et al. (2017) examine two variants of disclosure, presence disclosure, and attribute disclosure. Membership disclosure means that synthetic data contain a data point that is quite similar to or a duplicate of a real-world data point of patient data used for training the model. In their experiments, the Hamming Distance is used to determine the similarity of two datapoints. The term attribute disclosure describes the possibility of inferring sensitive attributes of real-world patient data by analyzing a subset of attributes known by an attacker. Original attribute information can be derived from synthetic data points that are closely related (e.g. via distance measures). Given these resemblances, it is possible to reveal attributes through majority vote. This allows an attacker to reveal private information about patients if the generative models memorize the real-world data set. The handling of distance measures for data points with mixed-type features (numerical and categorical) that occur in NetFlows requires a study on its own, in order to apply the membership disclosure and attribute disclosure approaches suggested by Choi et al. (2017).

4.2. Domain metrics

The second type of evaluation metrics is more specific to the domain of the data. These metrics can exploit domain knowledge to determine the utility of synthetic data.

4.2.1. Rule based

Some domain specific data are constrained by certain rules, which can be used to check the syntactical correctness of data. Syntax checks are a rule-based measurement of data quality, where hand-crafted sanity checks are applied to synthetic data in order to evaluate if a generator is able to, for example, model simple distribution value ranges of attributes or the correct number of attributes per NetFlow in general. Ring et al. (2018) use domain knowledge checks, which are a set of hand-crafted rules to check the realism of generated NetFlows. The checks range from simple rules e.g. UDP-NetFlows that should not contain TCP-Flags or Network-specific rules e.g. the netbios messages should be sent by specific internal IP-addresses.

4.2.2. Domain expert evaluation

Domain experts are familiar with the domain and its data. Therefore, they are able to evaluate samples based on prior knowledge. Ehrhart et al. (2022) evaluate samples of time-series sensor data via domain experts. Choi et al. (2017) evaluates synthetic medical data via expert evaluation from a medical doctor. Ring et al. (2018) uses violin plots to represent data distributions that can be visually compared for evaluation purposes. Charlier et al. (2019) visualizes distributions via histograms for the visual comparison of real and synthetic data. Schlör (2022) visualizes distributions via histograms and correlations via heatmaps.

4.2.3. Application based

The quality of synthetic data can be evaluated by using them in domain-specific applications, e.g. in classification settings. Under the assumption that the generated data are labeled, classifiers can be trained, tested, and evaluated.

There exist two main settings, which are Train on Synthetic, Test on Real (TSTR), and Train on Real Test on Synthetic (TRTS) (Goncalves et al., 2020; Dankar et al., 2022). Both classifiers provide classification results, which can be compared against a classifier which is purely trained and evaluated on real data, via confusion matrix and standard classification metrics like accuracy, precision, recall, false positive rate, etc.

As shown by Yin et al. (2022), this metric can be applied in the domain of NetFlow. Although, they applied TSTR exclusively, the TRTS setting is applicable as well.

A more specific use case for synthetic NetFlow data is data augmentation in terms of oversampling (Guo et al., 2021; Liu et al., 2021).

Most network traffic consists of normal traffic, while different types of attacks are special abnormal behavior in a network representing minority classes. Many classifiers for multi-class classification require balanced class distributions to learn properly. A classifier is trained on the original data with the skewed distribution and the over-sampled data, augmented via synthetic data. The performance gain of the classifier allows conclusions to be drawn about the quality of the synthetic data.

4.3. Anomaly detection and classification models

The NetFlow-Benchmark data sets are labeled and therefore can be used to train intrusion detection systems. Our benchmarks apply an anomaly detection method, in the settings TRTS, which has already been successfully applied in the NetFlow domain (Yin et al., 2022) and additionally TSTR. This setting requires an established and reliable model with a low training time. Isolation Forest, One-Class Support Vector Machine and XGBoost meet these criteria and are used for TSTR and TRTS in our benchmarks. A short description of each model is given in the following.

Isolation Forest (IF) (Liu et al., 2008) is a well-performing unsupervised anomaly detection method with a low training time on a recent large-scale anomaly detection benchmark (Han et al., 2022). This model-based anomaly detection method isolates anomalies explicitly via an ensemble of Isolation Trees (iTrees), where anomalies are detected via the shortest average path lengths on the iTrees. IF features a linear time complexity with a low memory requirement, and works well with high dimensional data and training data which do not contain any anomalies.

The One-Class Support Vector Machine (OCSVM) (Scholkopf et al., 1999) is another established unsupervised anomaly detection method. The algorithm estimates the support vectors of a hyper-plane that separates all the data points from the origin in the feature space of a single class. When unseen data points (test data) are below the hyperplane and closer to origin, they are classified as outliers. The OCSVM employs kernels to differentiate between data distributions of various shapes and, considering the noise inherent in real-world data, utilizes soft margins. Soft margins allow some misclassified data points, in order to prevent overfitting to noise.

XGBoost (Chen and Guestrin, 2016) is a tree boosting algorithm for supervised classification that handles sparse data and scales to billions of data points while maintaining low resource requirements. XGBoost has been successfully applied in many experiments and provides state-of-the-art results (Chen and Guestrin, 2016; Han et al., 2022) on various data sets.

5. A benchmarking methodology for NetFlow data

This section describes the methodology behind our proposed benchmarking process. At first, suitable metrics for our benchmark are selected based on our metric review in the previous Section 4. Next, we construct summarizing metrics, that aggregate multiple metrics based on our metrics hierarchy (see Fig. 2). Finally, we define our overall benchmarking process.

5.1. Benchmarking metric selection

According to Goncalves et al. (2020) there is no one-size-fits-all approach to synthetic data evaluation, and evaluation approaches must be tailored to the domains. NetFlow data contain attributes of numerical and categorical types. There are either methods that will process both types of attributes, or there are different methods for each attribute type. Based on the metrics hierarchy in Fig. 2, we select metrics for each subgroup of metrics (bottom row in the diagram) to cover the different focus of each comparison metric of the respective category. In our selection process, we favor metrics that have been successfully applied to NetFlows in previous works; otherwise, we check the suitability of metrics from other data domains. Our selection focuses on objective metrics for various aspects of similarity with value ranges from $[0, 1]$ that can be easily aggregated by averaging multiple metrics of various aspects. Table 2 summarizes which metrics will be used in the benchmark.

Single attribute. We will apply the Jensen–Shannon divergence because of its value range from $[0, 1]$ and has been applied in the NetFlow domain before by Yin et al. (2022). The Earth Mover's Distance (Wasserstein-1 distance) is not used since this measure is part of the WGAN, where this distance is applied in the loss function and the Earth-Movers distance does not range from $[0, 1]$, which complicates the application among different data sets, which can result in different value ranges. The Euclidean distance has to be modified to handle mixed-type data points like NetFlows and therefore requires fundamental research in application on this domain. The Support Coverage and Hellinger-Distance are metrics evaluated in other data domains to compare distributions, but are neglected in favor of the established Jensen–Shannon divergence.

Multiple attributes. The metrics Pearson's correlation coefficient, uncertainty coefficient, and correlation ratio calculate data correlations, but differ in the type of data they can process. None of the correlation metrics have been applied to the NetFlow domain for synthetic data evaluation previously, but they appear relevant since they measure correlations in the underlying multivariate data. For example, in real-world data, the TCP protocol and Flags typically appear together, whereas UDP and Flags do not. To measure the similarity of data in terms of correlations between attributes, all three measures are required for mixed type NetFlow data. In order to aggregate the correlation matrices to a single value, the Pairwise Correlation Distance or Mean Absolute Error can be used. Our metric selection favors the Mean Absolute Error because of its value-range from $[0, 1]$.

Population. There exist multiple model-based ways to evaluate the similarity of two populations. The simplest approach is a discriminator that is trained to separate the two populations. The Classifier Two-Sample Test and Propensity Score are closely related to the discriminator model. In our benchmark, we use two discriminator models, namely the Isolation Forest and the One-Class Support Vector Machine, which are evaluated based on the False Positive Rate. Data set one is labeled as class normal, while data set two is labeled as class anomaly. The models are fitted on data set one and tested on data set two. Unrelated data would be perfectly separated, resulting in zero false positives (FP). If the data are very similar, they shall not be separated perfectly by the discriminator model, resulting in many FPs. Based on these assumptions, the discriminator IF is evaluated via the commonly used false positive rate (FPR) metric, where a higher FPR indicates a higher similarity of two data sets. The Discriminator addresses the fundamental dependency of the Classifier Two-Sample Test and the Propensity Score, which are not considered in favor of the fundamental Discriminator evaluation using FPR. Ring et al. (2018) applied the Euclidian distance to measure the distance of a synthetic data distribution and the real data distribution, but a detailed mathematical motivation is missing, that is required since the Euclidean distance is defined for data points exclusively but not for probability distributions. Moreover, this metric is not restricted to a range of $[0, 1]$ and is therefore less favored for our use case.

The Cluster-Analysis measure is based on unsupervised clustering of data, but has several drawbacks in terms of hyperparameter tuning and overall reliability (Woo et al., 2009) and is therefore not considered as a population-based similarity measure.

Table 2
Overview of evaluation metrics used for NetFlow evaluation in this work.

Category	Scope	Metric
Data	Single attribute Multiple attributes Population	Jensen–Shannon divergence Pearson correlation coefficient, Correlation ratio, Uncertainty coefficient Discriminator (IF, OCSVM)
Domain	Task application Rule based	TSTR, TRTS (IF, OCSVM, XGBoost) Syntax-Checks for NetFlows

Table 3
NetFlow syntax check definitions.

Syntax check	Definition
IP-Address	Check if the IP address contains four octettes, where each octettes value ranges from 0 to 255.
Port	Check if the port range is in the range from 0 to 2^{32}.
Label	Check if the labels is 1 or 0.
TCP-Flags	Check if UDP-NetFlows do not contain TCP-Flags.
Positive values	Check if (float) value is larger or equal to 0.
In and Out sum	Check if the sum of in and out values is greater 0 per NetFlow entry e.g. for in and out bytes

Table 4
The NetFlow attributes with the applied syntax checks.

Attribute	Syntax checks
IPV4_SRC_ADDR	IP-Address
L4_SRC_PORT	Port
IPV4_DST_ADDR	IP-Address
L4_DST_PORT	Port
PROTOCOL	Positive values
IN_BYTES	Positive values, In and Out sum
OUT_BYTES	Positive values, In and Out sum
IN_PKTS	Positive values, In and Out sum
OUT_PKTS	Positive values, In and Out sum
TCP_FLAGS	Positive values, TCP-Flags
FLOW_DURATION_MILLISEC.	Positive values
Label	Label

Memorization. The Memorization-Informed Frechet Inception Distance is not considered due to the missing InceptionNet Model in the NetFlow domain.

The Hamming Distance calculates the distance between two categorical sequences. The NetFlows contain mixed types attributes and therefore, the Hamming Distance cannot be applied to NetFlow.

Since both metrics have preconditions that are not met in the domain of NetFlows, they are both discarded in our benchmark.

Application based. The application based evaluation of the TRTS and TSTR setting uses the models IF, OCSVM and XGBoost. The NetFlow data sets contain labels that indicate normal behavior or attacks. The models are trained on the (normal)[1] data from one data set (e.g synthetic) and tested on the other data set to evaluate the data in the TSTR and TRTS setting. In this setting, the model's performance is evaluated using F1-Scores, with higher values (approaching 1.0) indicating that the synthetic data adequately capture task-specific properties, thereby enabling the training of an effective classification model. This measure is an indicator for the practical applicability of the generated labeled data.

Specific applications such as data augmentation for oversampling (Guo et al., 2021; Liu et al., 2021) are out of scope for our benchmark. Data augmentation for oversampling requires the application of special generative models that focus on the generation of minority classes, instead of generating the complete data distribution.

Rule based. The Syntax Checks for NetFlows are based on the technical constraints which are defined by the NetFlow-Format. These rules are fundamental for the application of NetFlow data and are used in our benchmark.

The applied syntax checks for each NetFlow feature in Table 4 are defined in Table 3.

The visual inspection of generated data and its distributions by domain experts is challenging due to the limited availability of such experts and the fact that subjective evaluations cannot provide objective measurements. The evaluation of raw data by domain experts is therefore discarded in our benchmark.

5.2. Constructing joint metrics for netflow data

In our benchmark, we apply the similarity metrics in Table 2. The metric selection compiled a setup of 14 metrics which evaluate diverse aspects of similarity, but they are hard to use in a setting where one wants to compare the similarity of two data samples. If we have a sample of real data R and samples of synthetic data S_A from generator A and samples S_B from synthetic data generator B, it is inconvenient to compare 14 different metrics individually and determine if S_A or S_B more similar to R. Especially if one wants to use the metrics as an optimization criterion, e.g. in the setting of hyperparameter-tuning for deep generative models, the evaluation of 14 metrics is inconvenient. The following proposes our methodology to aggregate the metrics based on our metrics hierarchy in Fig. 2, where we aggregate the metrics based on the two distinctive subgroups *Data Metrics* and *Domain Metrics*. While the *Data Metrics* can be applied independent of the underlying domain and can be applied to other domains, the *Domain Metrics* exploit characteristics of the underlying domain, e.g. the syntax of the data.

Data metrics aggregation. The Eqs. (19) to (23) define how the Data Dissimilarity Score ($DSim_{\text{data}}$) is calculated by averaging selected data similarity metrics.

$$DSim_{\text{data}}(R,S) = \frac{Sim_{\text{att}}(R,S) + Sim_{\text{cor}}(R,S) + Sim_{\text{pop}}(R,S)}{n_{\text{dataMetrics}}} \tag{19}$$

The single attribute metric Eq. (20) is applied to each attribute individually (pairwise for each) and then averaged.

$$Sim_{\text{att}}(R,S) = \overline{d}_{JS}(R,S) \tag{20}$$

The correlation metrics Eq. (21) are calculated for each data set, and two data set correlations are compared afterward. The difference in the resulting correlation matrices is aggregated using the Mean Absolute Error Eq. (22).

$$cor(X,Y) = \begin{cases} cor_P(X,Y) & \text{if } X \text{ and } Y \text{ numerical} \\ cor_\eta(X,Y) & \text{if } X \text{ cat. and } Y \text{ num.} \\ cor_U(X,Y) & \text{if } X \text{ and } Y \text{ categorical} \end{cases} \tag{21}$$

$$Sim_{cor}(R,S) = \frac{1}{n^2}\sum_i^n \sum_j^n \left| cor(R_i,R_j) - cor(S_i,S_j) \right| \tag{22}$$

The similarity in terms of the data population Eq. (23) is evaluated via a discriminator model that distinguishes two data sets from each other. The classification results are evaluated using the false positive rate.

$$Sim_{\text{pop}}(R,S) = \sum_{D \in \{\text{IF},\text{OCSVM}\}} FPR_D(R,S) + FPR_D(S,R) \tag{23}$$

[1] For IF and OCSVM.

Domain metrics aggregation. The Domain Dissimilarity Score ($DSim_{\text{domain}}$) in Eq. (24) is the average value of the selected domain-specific metrics. The R and S value in each equation refers to the first (or real) and the second (or synthetic) data sets that are evaluated.

$$DSim_{\text{domain}}(R,S) = \frac{Sim_{\text{task}}(R,S) + N_{\text{SyntaxCheckErrors}}(S)}{n_{\text{domainMetrics}}} \tag{24}$$

The synthetic data are filtered based on syntax checks (Table 3), since syntactical correctness is required for the application of similarity metrics. The ratio of syntactically wrong NetFlows $N_{\text{SyntaxCheckErrors}}$ counts the number of NetFlows which contain at least one syntax error. NetFlows with more than one NetFlow syntax error are counted as one syntactically wrong Netflow.

The task-specific application of the data is evaluated via an anomaly detection task Eq. (25), where the normal and attack labels of the NetFlow data are leveraged.

$$Sim_{\text{task}}(R,S) = \sum_{T\in\{\text{IF,OCSVM,XGB}\}} (1 - F1_T(R,S) + 1 - F1_T(S,R)) \tag{25}$$

The label distribution of the data sets is unbalanced. Therefore, metrics such as accuracy are inadequate since they do not consider imbalanced distributions (Han et al., 2011). The F1-Score is the harmonic mean of precision and recall with a value range between 0 and 1, where 1 is the best value. In our setting the anomaly detection model is evaluated via the F1-Score metric with the averaging methods of macro, micro and weighted. The micro-averaging counts the total true positives, false negatives, and false positives to calculate the metric globally. The macro-averaging calculates the unweighted mean for each label and does not consider label imbalance. The weighted averaging calculates the label average weighted by the number of true instances for each label. This is similar to macro-averaging, but considers imbalanced labels. Since the NetFlow data sets are highly imbalanced in terms of normal and malicious traffic, we focus on the weighted F1-Score.

5.3. Process of benchmarking synthetic NetFlow data

The benchmark process depicted in Fig. 3 is executed in two phases. In a pre-study, the application of the selected similarity metrics is evaluated on published benchmark data sets. From each data set 30 subsets, consisting of 10,000 random samples each, are drawn. On the basis of these samples, the similarity of data from the sample distribution (*intra* data sets) can be calculated. The similarity of samples from a different distribution (*inter* data set) is calculated as well.

The values of the intra and inter data set can be used as guidelines to evaluate the similarity of synthetic data in the study. Next, we apply similarity metrics to synthetic data that were generated by our generative models. Here, the metrics calculate the similarity of a sample of the real-world data set and the synthetic data set. With the intra and inter data set similarities, an upper- and lower bound for the similarity is set, which is used in the assessment of the synthetic data.

6. Metric selection validation on real data

This section describes the application of the metrics to evaluate the selected similarity metrics for NetFlow data. It explains the general setup, the used data sets, and their preprocessing. Next, data processing for the generative models and the processing scheme for the synthetic data are given.

6.1. Setup

Data. For the evaluation of similarity metrics, several commonly used benchmark data sets, which are designed for Network Intrusion Detection Systems, are used. In order to achieve a comparable setup regarding the data sets, the benchmark utilizes the data sets of Sarhan et al. (2021), who transformed established benchmark data sets into a common NetFlow format. In the benchmark, we use the transformed data sets: NF-CSE-CIC-IDS2018, NF-ToN-IoT and NF-UNSW-NB15.

NF-CSE-CIC-IDS2018 (originally published in Sharafaldin et al. (2018a)) was released by the Communications Security Establishment (CSE) and the Canadian Institute for Cybersecurity (CIC). It is based on a company network consisting of multiple departments. Normal traffic is generated via realistic network events generated by abstract human interactions. The attacks are executed from clients outside the company network. The data set contains 8,392,401 NetFlows, with 87.86% normal and 12.14% attack NetFlows.

NF-ToN-IoT (originally published in Booij et al. (2022)) was released by the University of New South Wales (UNSW) Sydney. It consists of Internet of Things (IoT) and Industrial IoT traffic generated by services and devices. The data set contains 1,379,274 NetFlows, with 19.6% normal and 80.4% attack NetFlows.

NF-UNSW-NB15 (originally published in Moustafa and Slay (2015)) was released by the Cyber Range Lab of the Australian Centre for Cybersecurity (ACCS). IT contains a mixture of real normal traffic and synthetic attack traffic. The data set contains 1,623,118 NetFlows, with 95.54% normal and 4.46% attack NetFlows.

Given that benchmark data sets comprise millions of NetFlows, obtaining a complete data distribution of the real data source, as well as fully understanding a complex generator model and its distribution, is not straightforward. Therefore, similarity metrics will be evaluated based on samples of 10,000 data points (NetFlow entries). In our setup, 30 subsets of 10,000 sampled data points will be drawn randomly from each real data set. The similarity metrics are applied pairwise to each sample of the real data sets. The intra data set distance evaluates the distance of data points from a common distribution. We hypothesize that these samples should be very similar. The inter data set distance evaluates the distance of data points from different distributions. Therefore, these samples are expected to be less similar to each other. The intra and inter data set distance allows defining value ranges (upper- and lower bounds), which can be applied in the evaluation of synthetic data sets.

Preprocessing of data. The IF and OCSVM are used for two different similarity metrics in our setting. First, they are applied as Discriminator, which shall distinguish two data sets from each other, e.g. synthetic and real. Second, the Isolation Forest, OCSVM and XGBoost are used in a task-specific setting.

The Isolation-Forest implementation from Scikit-Learn (Pedregosa et al., 2011) supports numerical attributes exclusively. The NetFlow data contain various categorical attributes that need to be transformed into a numerical representation. An overview of the attributes, data types, and encoding styles is given in Table 5. The encoding style ✗ indicates that this attribute is not considered in the encoded data. While some attributes like labels can be transformed via one-hot encoding, others like IP addresses require a different approach. The use of a one-hot encoding for attributes with many unique categorical values would result in large sparse vectors. Ring et al. (2018) suggest encoding IPs and ports in a binary vector, where each digit is a single vector dimension. This approach is applied in our benchmark, since it requires no prior training of embeddings, but features more dense vectors than one-hot encoding. The numerical attributes like duration, bytes, and packets are binned and one-hot encoded afterward.

6.2. Validation on real data results

In this section, we demonstrate that the selected metrics effectively distinguish samples from different NetFlow data sets. In order to analyze the metrics and their interconnections further, the Pearson correlation among the samples of all data sets are analyzed. The heatmap in Fig. 4 depicts the Pearson Correlations between the different metrics. The plot shows a correlation between data-specific metrics or domain-specific metrics.

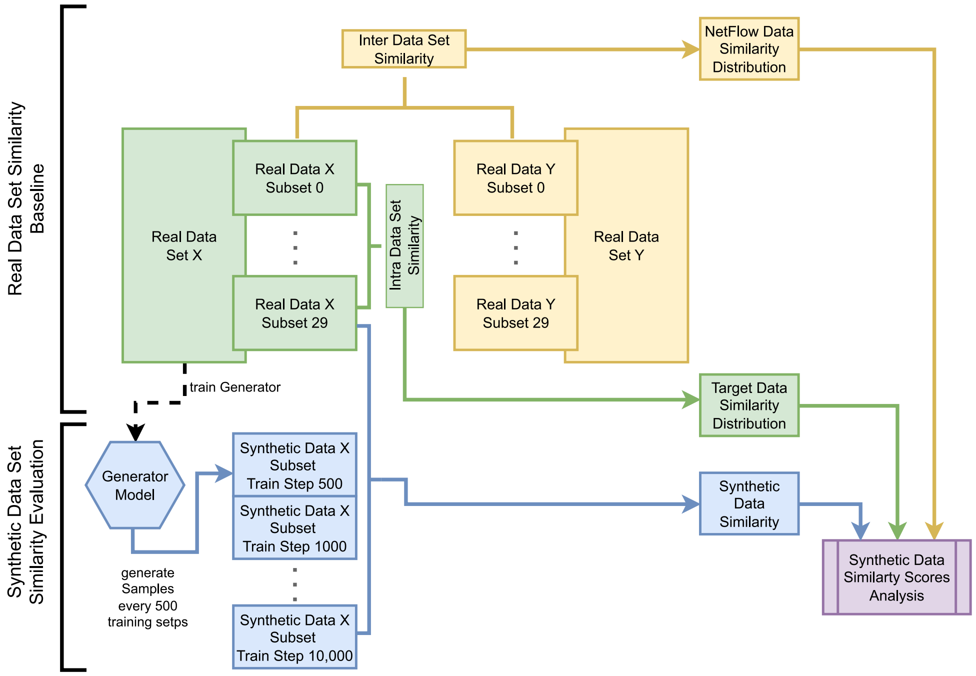

Real Data Set Similarity Baseline
Inter Data Set Similarity
NetFlow Data Similarity Distribution
Real Data Set X
Real Data X Subset 0
Real Data X Subset 29
Intra Data Set Similarity
Real Data Y Subset 0
Real Data Y Subset 29
Real Data Set Y
train Generator
Target Data Similarity Distribution
Synthetic Data Set Similarity Evaluation
Generator Model
Synthetic Data X Subset Train Step 500
Synthetic Data X Subset Train Step 1000
Synthetic Data X Subset Train Step 10,000
generate Samples every 500 training setps
Synthetic Data Similarity
Synthetic Data Similarty Scores Analysis

Fig. 3. A structured overview of benchmarking process. We obtain baseline distance values for intra data set scores (green) and inter data set scores (yellow) for reference of the synthetic data set scores (blue). The intra data set scores are calculated from samples of the same data distribution. The inter data set scores are calculated by using samples from the target data set X and samples from another NetFlow data set Y. The synthetic data set scores are calculated by using samples from the training data set X and samples of generated data . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 5
The NetFlow V1 attributes with data types and encoding style.

Attribute	Data type	Encoding
IPV4_SRC_ADDR	Categorical	Split + binarize
L4_SRC_PORT	Categorical	Binarize
IPV4_DST_ADDR	Categorical	Split + binarize
L4_DST_PORT	Categorical	Binarize
PROTOCOL	Categorical	One-hot
L7_PROTO	Categorical	✗
IN_BYTES	Numerical	Bin + one-hot
OUT_BYTES	Numerical	Bin + one-hot
IN_PKTS	Numerical	Bin + one-hot
OUT_PKTS	Numerical	Bin + one-hot
TCP_FLAGS	Numerical	Binarize
FLOW_DURATION_MILLISEC.	Numerical	Bin + one-hot
Label	Categorical	Original value
Attack	Categorical	✗

There is greater correlation in general in the area of data metrics (JSD Mean - OCSVM Discriminator), which shows a coherence among the data driven similarity measures. The domain-based metrics do not show a general correlation among all metrics, especially the IF-Task and the OCSVM-Task show differences towards the TSTR and the TRTS setting, which highlights the significance to incorporate both metrics. The XGBoost task metrics show a stronger correlation towards the data-driven metrics than the domain-driven metrics.

The correlation analysis of the metrics show some correlation between metrics and unrelated metrics as well, which emphasizes the application of multiple metrics to assess the data similarity in a wholesome manner.

The box plots of the Data Dissimilarity Score Fig. 5(a) and the Domain Dissimilarity Score Fig. 5(b) show the mean values of all metrics across all data set samples used in our benchmark. All metrics are in the interval of $[0, 1]$ and most metrics follow the rule of “lower value is better” except the F1-Score which is adapted by calculating $1 - \text{F1-Score}$. The plots show that the average metrics for the data-based and domain-based metrics are able to differentiate between data sets, with little distributional variance for each pairwise data set comparison.

Interestingly, the Data Dissimilarity and the Domain Dissimilarity indicate a greater separation of the TI data set in comparison to the CC and UN.

The application of the metrics to real NetFlow samples demonstrates that these metrics can effectively distinguish samples from different sources. Consequently, these metrics are validated as an effective method for determining the similarity between samples as they can be used to assess the capability of a generator in creating data that ranges from perfectly resembling a given target distribution (the lower bound) to still representing valid NetFlow data from a potentially different target distribution (upper bound of acceptable data quality) detailed in the following section.

7. Case study: Benchmarking of synthetic data

After evaluation the general application of the similarity metrics on real data, we apply them for the evaluation of synthetic data in this section. The general setup including the data sets and the data preprocessing is identical to the previous section. At first, we describe the two generative models WGAN and GPT-2 which we use to generate synthetic NetFlow data. Second, we apply the data and domain dissimilarity and apply it to the generated data. Since the pure dissimilarity metric values are hard to interpret, we use the results from the real data set samples from the pre-study to define upper and lower bounds of the value. The lower bound consist of metric values calculated from samples of the target data set exclusively (intra data dissimilarity). Synthetic samples close to these values can be considered as a good proxy of the target data set. The upper bound consist of metric values

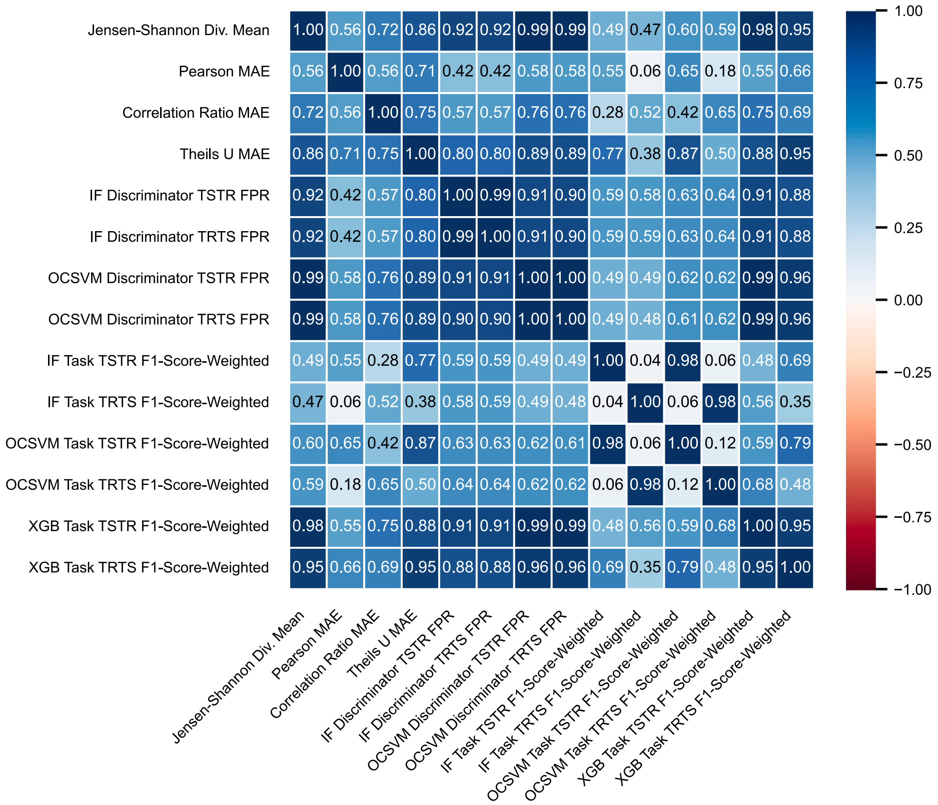
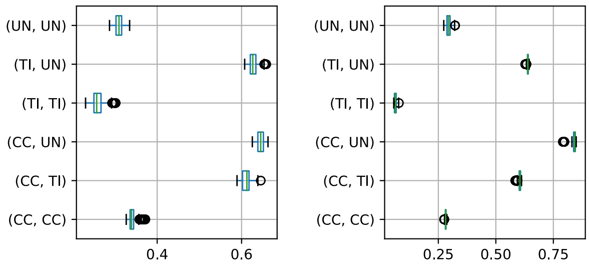


Fig. 4. Correlation of the metrics based on all data set samples. MAE = Mean Absolute Error, IF = Discriminator, OCSVM = One Class Support Vector Machine, XGB = XGBoost, Discriminator = Train model to separate real and synthetic data, Task = application of a model in a classification task using the data labels.



(a) Data Dissimilarity Score

(b) Domain Dissimilarity Score

Fig. 5. Distributions of the data and domain dissimilarity pairwise grouped by data sets: NF-CSE-CIC-IDS2018 = CC, NF-ToN-IoT = TI, NF-UNSW-NB15 = UN.

calculated from metric values between samples from the target data set and samples from other NetFlow data set (inter data set dissimilarity). Synthetic samples close to this bound are considered to be NetFlows in a general sense, but they are a less good proxy for the target data.

7.1. Generator models

WGAN. The WGAN model cannot process categorical information, which requires the encoding of categorical values into numerical representations. Our work adopts the binary encoding from Ring et al. (2018), but in contrast to the original scheme, where the protocol is one-hot encoded, we encode the protocol in the style of binary encoding as well because our data contain more than the three protocols UDP, TCP and ICMP, e.g. OSPFIGP, SCTP and others. Moreover, we also generate value encoded labels, where 0 indicates normal traffic and 1 indicates attacks.

GPT-2. In our setting, the model is trained with NetFlow data in the format of comma-separated values (csv) as raw-text input. Afterward, the generated data are filtered, where structurally wrong NetFlows are removed. Structurally incorrect NetFlows include entries with an incorrect number of features, such as too few or too many commas per line or completely empty lines that are unusable for further analysis.

7.2. Benchmarking of synthetic data results

In the following, we present a case study to show how our proposed evaluation framework can be applied to evaluate synthetically generated NetFlow data. In this section, the Data and Domain Dissimilarity are applied to the synthetic NetFlows that are generated by GPT-2 and WGAN-binary every 500 training steps, to evaluate the training process of each model. The output data of the generative models are subject to various degrees of freedom, e.g. some generate numerical vectors which are transformed to NetFlows and other models generate raw (structured) text. Due to the different output forms, the output is not necessarily structured in a valid NetFlow syntax. Syntax checks serve as a filter for the synthetic data, ensuring that only plausible NetFlow values are considered and that the encoding scheme, e.g. for machine learning models, is appropriately applied. Additionally, numeric features, e.g. the number of bytes or packets, need to be checked, since only numerical values can be evaluated correctly by certain similarity metrics like the Pearson correlation coefficient.

The plots in Fig. 6 show the training history of the models evaluated via the Data Dissimilarity Score and Domain Dissimilarity Score for the three data sets in our experiments. The scores alone would be difficult

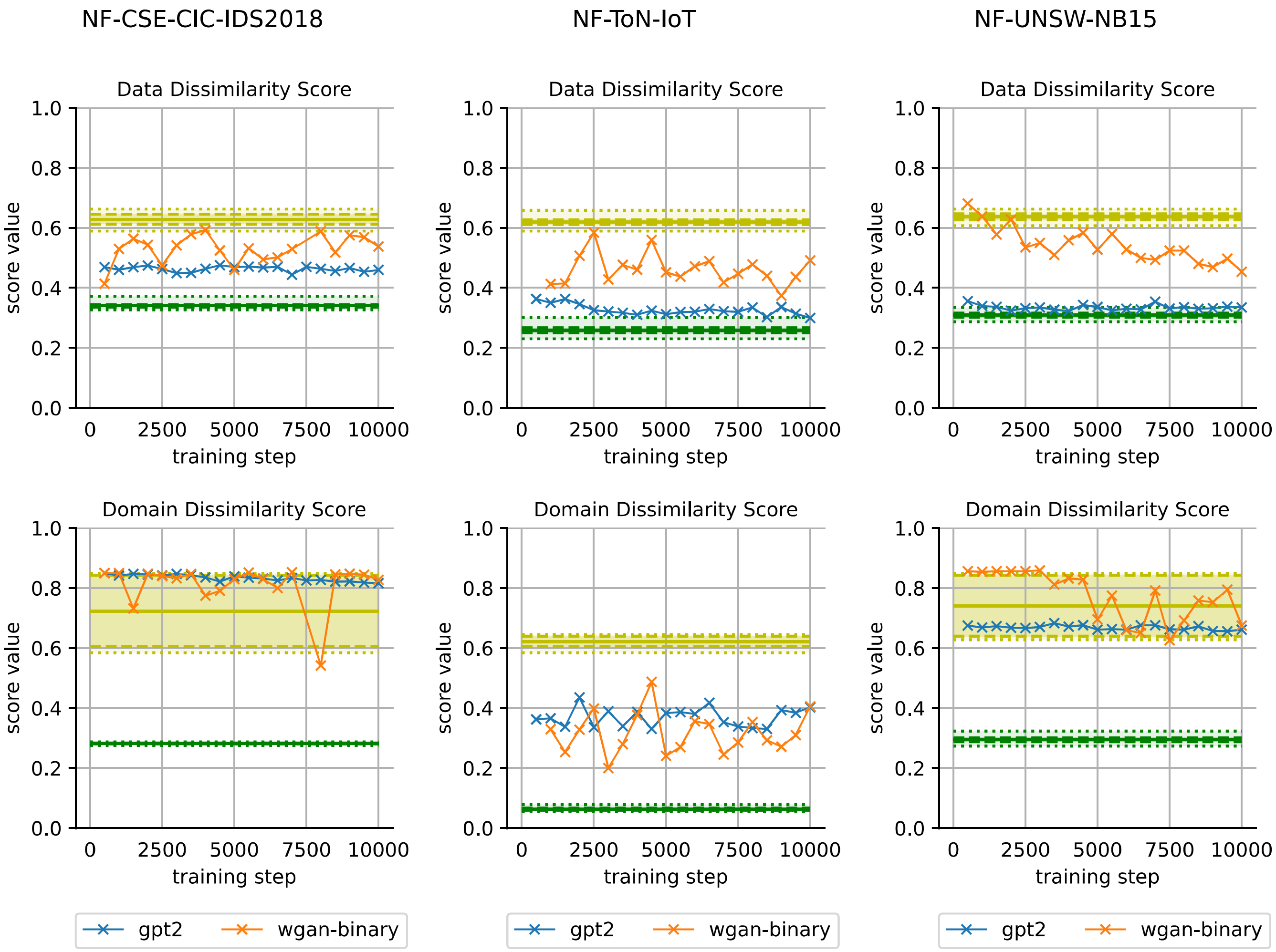


Fig. 6. Dissimilarity metrics for every 500 training steps, comparing different dissimilarity scores (rows) per synthetic data set and model (columns); score values for the synthetic data sample (y-axis) for every training step sample (x-axis). The highlighted areas, show the intra data set similarity in light-green and green The inter data similarity is highlighted in light yellow and yellow. The value ranges of light colored yellow and green areas indicate the range of the minimum and the maximum values of real data set comparisons, borders are indicated via dotted lines. The darker yellow and green areas indicate the 25% and the 75% quantiles of the distributions and are separated via dashed lines. The median value is indicated by a continuous line . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

to interpret, as a viewer might not understand whether the underlying absolute similarity value is favorable or unfavorable. Essentially, it is unclear whether the value, though different, still represents valid data similar to the original. Based on the application of the metrics to the real data set, we add baseline values for reference as upper and lower bounds: The intra data set scores are based on samples of the target data set exclusively and therefore reflect the scores of data from the same target distribution. Inter data set scores are calculated from samples originating from different data sets. Although these data sets are generally similar, such as both being real NetFlow data, the samples may not exhibit similar behaviors or accurately represent the same target distribution. There are highlighted value ranges in Fig. 6, to show the intra data set similarity in light-green and green, and the inter data similarity in light yellow and yellow. The value ranges of light-colored variants indicate the range of the minimum and the maximum values across the real data set comparisons, and the borders are indicated via dotted lines. The darker variants of green and yellow indicate the 25% and the 75% quantiles of the distributions and are indicated via dashed lines. The median value of the distributions is represented by a continuous line.

The plots indicate a more stable training behavior of the GPT-2 model, where the model's data are well-fitted at 1000 training steps and do not show significant improvement afterward. The Data Dissimilarity Score in the first row shows that the generative models are able to generate data with a low dissimilarity score, where the score value is close to the intra data distribution (green area). The Domain Similarity Score in the second row demonstrates that the application of the synthetic data resembles scores close to the inter data set distributions (yellow area) rather than the intra data set distributions for the NF-CSE-CIC2018 and the NF-UNSW-NB15. The synthetic NF-ToN-IoT shows dissimilarity scores between the inter and intra data set baselines. Although the data successfully capture NetFlow properties (yellow area) suitable for training general anomaly detection systems, it appears to lack characteristics unique to the target data set (green area). This suggests that the generative models capture data driven properties of the training data well and can therefore be used as proxy data in terms of data driven properties. The generated data show differences when they are applied in their domain e.g. in anomaly detection tasks in comparison to the original training data, which means that the generated data cannot be regarded as ideal proxy data for the original training data in terms of domain application.

Overall, the plots demonstrate the importance of specific data and domain measures to evaluate synthetic data in a comprehensive way. Although the models effectively capture and replicate the distributions of the training data, utilizing the generated data in practical applications reveals a different outcome. Our findings show that data with similar distributions do not necessarily excel in domain-specific applications. The Data Dissimilarity shows that almost all models fit the target data well and achieve scores similar to the target distribution during the training process. The Domain Dissimilarity Scores indicate that both models produce data that behave differently from the target data in domain-specific applications, without visible improvement throughout the training process. This underscores the necessity to employ both evaluation perspectives for a comprehensive benchmarking procedure.

8. Conclusion

The objective of this work was to close the gap of a missing standardized evaluation setup for synthetically generated NetFlow data from machine learning models such as WGAN and GPT-2. In this work, we therefore presented a standardized benchmark framework for synthetic NetFlow data evaluation. Initially, we compiled a range of similarity metrics from the literature. Then, we selected metrics specifically designed to assess different aspects of similarity, focusing on both data and domain-based similarities. To validate the metrics, their values were calculated using 90 subsets derived from three real-world data sets. The results demonstrate that the selected metrics are sensitive to the data set, emphasizing their appropriateness as bounds for a differentiated evaluation. Lastly, the benchmarking of synthetic data samples was carried out as a case study by using the inter- and intra-similarity scores from the real-world data sets as upper and lower bounds. Utilizing publicly available benchmark data sets allows the similarity ranges of these data sets to serve as benchmark values for evaluating future generators trained on the same data sets.

In general, combining data-specific and domain-specific metrics offers a comprehensive approach to evaluate not only the distributional properties, but also the practical applicability of generated data with various metrics to better reflect the diverse characteristics of data sets. The Data and Domain Dissimilarity scores enable the quantification of data quality from synthetic generation through objective numeric values and facilitate a model-agnostic assessment of data quality. The proposed line plots serve as an intuitive way to visualize the evaluation results in a compact format by indicating reasonable ranges of the target data set and NetFlow data in general.

By publicly releasing our benchmark framework and benchmark data,[2] we aim to contribute to the establishment of a more standardized and comparable evaluation of real-world and synthetic NetFlow data sets and associated generative models.

For future work, we plan to further enhance this framework by incorporating additional metrics and applying it to a broader range of data sets and generative models.

9. Advantages and limitations

Our proposed framework for benchmarking synthetic data has several benefits, but is also subject to limitations which must be considered. Aggregating multiple metrics into two distinct categories, data-driven and domain-driven, enables quicker comparisons of data generation approaches by reducing the analysis from 14 individual metrics to two composite metrics. These scores, the data dissimilarity and domain dissimilarity, can serve as numerical objectives for optimizing deep learning models, particularly through hyperparameter tuning. Finally, we would like to refer to the system performance, in particular the runtime for calculating the Data and Domain Dissimilarity Scores. The framework requires about 3 min to calculate the dissimilarity metrics of two samples (10,000 NetFlows each) on a device with the following specifications: 11th Gen Intel(R) Core(TM) i7-1165G7 featuring 2.80 GHz and 32,0 GB RAM. The low performance requirements enable researches to apply the evaluation framework directly on low spec machines like laptops, which emphasizes the accessibility of the framework.

Currently, the proposed setup has some limitations that need to be considered. First, our approach requires evaluations with sample sizes of at least 10,000 data points, which may not always be available, particularly for generating rare events of specific attacks where only a few thousand NetFlows exist. Second, duplicates of original NetFlow that are part of the generated data are not considered directly, which can be addressed by incorporating suitable memorization metrics. Third, although the NetFlow features used in this case study are common in most public data sets, additional features available from NetFlow exporters are not yet included in our benchmark framework. Moreover, the domain-specific metrics in our setup are tailored to NetFlow data only. While data-driven metrics can be universally applied to other domains as well, domain-specific metrics, such as syntax checks, require adaptation when applied to different data domains.

CRediT authorship contribution statement

Maximilian Wolf: Writing – review & editing, Writing – original draft, Visualization, Software, Data curation, Conceptualization. **Julian Tritscher:** Writing – review & editing, Writing – original draft. **Dieter Landes:** Writing – review & editing, Supervision. **Andreas Hotho:** Writing – review & editing, Supervision. **Daniel Schlör:** Writing – review & editing, Writing – original draft, Supervision, Conceptualization.

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.

Acknowledgment

This work is funded by the Bavarian Ministry of Economic Affairs, Regional Development and Energy through the GENESIS project (grant no. DIK-2110-0035//DIK0422/03).

Appendix. Netflow data samples

This subsection lists example NetFlow samples from the NF-CSE-CIC-IDS2018 data set for visual comparison of real and generated data. Real data samples from the original data set are displayed in Table A.6. Synthetic data samples from GPT-2 are in Table A.7, samples from the WGAN are in Table A.8.

[2] https://github.com/maxwolf-code/netflow_data_similarity_framework.

Table A.6
Real NetFlow data sampled from NF-CSE-CIC-IDS2018.

SRC_ADDR	SRC_PT	DST_ADDR	DST_PT	PROTO.	I_BY	O_BY	I_PK	O_PK	FLAGS	DUR.	Label
172.31.69.25	21	18.221.219.4	60 538	6	40	0	1	0	20	0	0
172.31.65.78	50 821	169.254.169.254	80	6	360	455	5	5	219	4 294 966	0
77.72.82.96	57 701	172.31.69.24	7 073	6	40	0	1	0	2	0	0
172.31.68.21	54 424	172.31.0.2	53	17	209	326	3	3	0	0	0
172.31.66.54	51 451	172.31.0.2	53	17	62	126	1	1	0	4 294 966	0
172.31.69.6	52 500	18.219.211.138	8 080	6	538	341	5	5	219	4 294 955	1
172.31.69.24	49 910	172.31.69.16	55 600	6	44	0	1	0	2	0	0
172.31.64.35	49 282	172.31.0.2	53	17	150	270	2	2	0	4 294 936	0
172.31.65.59	60 555	172.31.0.2	53	17	71	87	1	1	0	4 294 966	0
172.31.64.51	64 092	172.31.0.2	53	17	62	270	1	1	0	4 294 964	0

Table A.7
Synthetic NetFlow data based on NF-CSE-CIC-IDS2018 from GPT-2 after 10,000 training iterations.

SRC_ADDR	SRC_PT	DST_ADDR	DST_PT	PROTO.	I_BY	O_BY	I_PK	O_PK	FLAGS	DUR.	Label
172.31.65.43	65 111	172.31.0.2	53	17	136	334	2	2	0	4 294 927	0
172.31.65.43	49 362	172.31.0.2	53	17	70	86	1	1	0	4 294 967	0
5.101.40.43	61 297	172.31.65.43	3389	6	1460	1873	8	7	222	4 293 292	0
172.31.65.43	50 965	54.240.190.208	443	6	126	126	2	2	24	4 294 889	0
212.92.116.36	62 196	172.31.67.41	3389	6	1380	1721	8	7	222	4 292 999	0
5.101.40.105	58 950	172.31.67.41	3389	6	1364	1721	8	7	222	4 293 293	0
212.92.116.36	62 180	172.31.67.41	3389	6	1380	1721	8	7	222	4 292 870	0
172.31.65.43	49 671	52.165.231.192	443	6	154	166	2	1	24	4 294 889	1
37.46.246.28	56 249	172.31.65.43	3389	6	1592	3086	11	8	30	4 291 872	0
165.132.169.106	63 880	172.31.65.43	3389	6	215	92	4	2	30	4 284 607	0

Table A.8
Synthetic NetFlow data based on NF-CSE-CIC-IDS2018 from WGAN after 10,000 training iterations.

SRC_ADDR	SRC_PT	DST_ADDR	DST_PT	PROTO.	I_BY	O_BY	I_PK	O_PK	FLAGS	DUR.	Label
60.23.201.202	59 732	172.31.67.13	3389	6	168	0	0	2	22	3 847 352	0
172.31.64.37	49 262	172.31.0.2	37	17	17	134	0	3	0	4 294 967	0
81.68.232.171	64 832	172.31.67.45	3357	6	1508	2041	12	15	223	4 290 608	0
172.31.65.39	51 322	172.31.0.2	37	17	120	134	0	2	0	4 294 966	0
172.31.66.37	49 418	195.236.236.249	345	6	21	184	5	17	155	2 531 948	0
172.31.66.53	50 619	209.240.252.235	4600	6	7	281	1	17	157	2 126 444	0
172.31.65.55	49 863	172.21.0.2	177	16	71	255	0	3	0	4 291 810	0
172.23.65.198	63 612	172.31.64.6	53	17	124	134	0	2	0	4 294 967	0
172.31.64.37	49 230	172.159.192.10	33	4	21	0	1	3	0	4 269 598	0
16.68.217.202	63 829	172.31.67.13	3389	6	1516	1881	8	6	94	4 293 868	0

References

Arjovsky, M., Chintala, S., Bottou, L., 2017. Wasserstein GAN. arXiv arXiv:1701.07875.

Bai, C.Y., Lin, H.-T., Raffel, C., Kan, W., 2021. On training sample memorization: Lessons from benchmarking generative modeling with a large-scale competition. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining.

Booij, T.M., Chiscop, I., Meeuwissen, E., Moustafa, N., den Hartog, F.T.H., 2022. ToN-IoT the role of heterogeneity and the need for standardization of features and attack types in IoT network intrusion data sets. IEEE Internet Things J. 9 (1), 485–496. http://dx.doi.org/10.1109/JIOT.2021.3085194.

Borji, A., 2021. Pros and cons of GAN evaluation measures: New developments. Comput. Vis. Image Underst. 215, 103329.

Charlier, J., Singh, A., Ormazabal, G., State, R., Schulzrinne, H., 2019. SynGAN: Towards generating synthetic network attacks using GANs. arXiv abs/1908.09899.

Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P., 2002. SMOTE: Synthetic minority over-sampling technique. J. Artificial Intelligence Res. 16, 321–357. http://dx.doi.org/10.1613/jair.953.

Chen, T., Guestrin, C., 2016. XGBoost: A scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

Chen, Y., Keogh, E., Hu, B., Begum, N., Bagnall, A., Mueen, A., Batista, G., 2015. UCR time series classification archive. URL: www.cs.ucr.edu/~eamonn/timeseriesdata/.

Choi, E., Biswal, S., Malin, B.A., Duke, J.D., Stewart, W.F., Sun, J., 2017. Generating multi-label discrete patient records using generative adversarial networks. In: Machine Learning in Health Care.

Claise, B., 2004. Cisco systems NetFlow services export version 9. RFC 3954, 1–33.

Dankar, F.K., Ibrahim, M.K., Ismail, L., 2022. A multi-dimensional evaluation of synthetic data generators. IEEE Access PP, 1.

Ehrhart, M., Resch, B., Havas, C., Niederseer, D., 2022. A conditional GAN for generating time series data for stress detection in wearable physiological sensor data. Sens. (Basel Switz.) 22 (16), http://dx.doi.org/10.3390/s22165969.

Fisher, R.A., 1992. Statistical Methods for Research Workers. Springer New York, New York, NY, http://dx.doi.org/10.1007/978-1-4612-4380-9_6.

Goncalves, A., Ray, P., Soper, B., Stevens, J., Coyle, L., Sales, A.P., 2020. Generation and evaluation of synthetic patient data. BMC Med. Res. Methodol. 20 (1), 108. http://dx.doi.org/10.1186/s12874-020-00977-1.

Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A.C., Bengio, Y., 2014. Generative adversarial nets. In: Neural Information Processing Systems. NIPS.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A., 2017. Improved training of wasserstein GANs. arXiv arXiv:1704.00028.

Guo, Y., Xiong, G., Li, Z., Shi, J., Cui, M., Gou, G., 2021. Combating imbalance in network traffic classification using GAN based oversampling. In: 2021 IFIP Networking Conference. IFIP Networking, pp. 1–9.

Han, S., Hu, X., Huang, H., Jiang, M., Zhao, Y., 2022. ADBench: Anomaly detection benchmark. arXiv abs/2206.09426.

Han, J., Pei, J., Tong, H., 2011. Data Mining: Concepts and Techniques. Morgan kaufmann.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S., 2017. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In: Neural Information Processing Systems. NIPS.

Hu, J., 2018. Bayesian estimation of attribute and identification disclosure risks in synthetic data. Trans. Data Priv. 12, 61–89.

Karras, T., Laine, S., Aila, T., 2018. A style-based generator architecture for generative adversarial networks. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition. CVPR, pp. 4396–4405.

Kholgh, D.K., Kostakos, P., 2023. PAC-GPT: A novel approach to generating synthetic network traffic with GPT-3. IEEE Access 11, 114936–114951. http://dx.doi.org/10.1109/ACCESS.2023.3325727.

Koochali, A., Walch, M., Thota, S., Schichtel, P., Dengel, A.R., Ahmed, S., 2022. Quantifying quality of class-conditional generative models in time-series domain. arXiv abs/2210.07617.

Kullback, S., Leibler, R.A., 1951. On information and sufficiency. Ann. Math. Stat. 22 (1), 79–86. http://dx.doi.org/10.1214/aoms/1177729694.

Liu, X., Li, T., Zhang, R., Wu, D., Liu, Y., Yang, Z., 2021. A GAN and feature selection-based oversampling technique for intrusion detection. Secur. Commun. Netw. 2021, 9947059:1–9947059:15.

Liu, F.T., Ting, K.M., Zhou, Z.-H., 2008. Isolation forest. In: 2008 Eighth IEEE International Conference on Data Mining. pp. 413–422.

Lopez-Paz, D., Oquab, M., 2016. Revisiting classifier two-sample tests. arXiv arXiv:1610.06545.

Manocchio, L.D., Layeghy, S., Portmann, M., 2021. FlowGAN - Synthetic network flow generation using generative adversarial networks. In: 2021 IEEE 24th International Conference on Computational Science and Engineering. CSE, pp. 168–176.

Moustafa, N., Slay, J., 2015. UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In: 2015 Military Communications and Information Systems Conference. MilCIS, pp. 1–6. http://dx.doi.org/10.1109/MilCIS.2015.7348942.

Nekvi, R.I., Saha, S., Mtawa, Y.A., Haque, A., 2023. Examining generative adversarial network for smart home DDoS traffic generation. In: 2023 International Symposium on Networks, Computers and Communications. ISNCC, IEEE, Doha, Qatar, pp. 1–6. http://dx.doi.org/10.1109/ISNCC58260.2023.10323616, URL: https://ieeexplore.ieee.org/document/10323616/.

Park, N., Mohammadi, M., Gorde, K., Jajodia, S., Park, H., Kim, Y., 2018. Data synthesis based on generative adversarial networks. Proc. VLDB Endow. 11, 1071–1083.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., 2011. Scikit-learn: Machine learning in python. J. Mach. Learn. Res. 12, 2825–2830.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al., 2019. Language models are unsupervised multitask learners. OpenAI Blog 1 (8), 9.

Ring, M., Schlör, D., Landes, D., Hotho, A., 2018. Flow-based network traffic generation using generative adversarial networks. Comput. Secur. 82, 156–172.

Ring, M., Wunderlich, S., Scheuring, D., Landes, D., Hotho, A., 2019. A survey of network-based intrusion detection data sets. Comput. Secur. 86, 147–167.

Ruiz, N., Muralidhar, K., Domingo-Ferrer, J., 2018. On the privacy guarantees of synthetic data: A reassessment from the maximum-knowledge attacker perspective. In: Privacy in Statistical Databases.

Sajjadi, M.S.M., Bachem, O., Lucic, M., Bousquet, O., Gelly, S., 2018. Assessing generative models via precision and recall. arXiv abs/1806.00035.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X., 2016. Improved techniques for training gans. Adv. Neural Inf. Process. Syst. 29.

Sarhan, M., Layeghy, S., Moustafa, N., Portmann, M., 2021. NetFlow datasets for machine learning-based network intrusion detection systems. In: Deze, Z., Huang, H., Hou, R., Rho, S., Chilamkurti, N. (Eds.), Big Data Technologies and Applications. Springer International Publishing, Cham, pp. 117–135.

Schlör, D., 2022. Detecting Anomalies in Transaction Data (Doctoral thesis). Universität Würzburg.

Scholkopf, B., Williamson, R.C., Smola, A., Shawe-Taylor, J., Platt, J.C., 1999. Support vector method for novelty detection. In: Neural Information Processing Systems.

Scott, D.W., 2015. Multivariate Density Estimation: Theory, Practice, and Visualization. John Wiley & Sons.

Sharafaldin, I., Habibi Lashkari, A., Ghorbani, A.A., 2018a. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In: Proceedings of the 4th International Conference on Information Systems Security and Privacy. ICISSP-INSTICC, SciTePress, pp. 108–116. http://dx.doi.org/10.5220/0006639801080116.

Sharafaldin, I., Lashkari, A.H., Ghorbani, A.A., 2018b. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In: International Conference on Information Systems Security and Privacy.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S.E., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., 2014. Going deeper with convolutions. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition. CVPR, pp. 1–9.

Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A.A., 2009. A detailed analysis of the KDD CUP 99 data set. In: 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications. pp. 1–6.

Theil, H., 1970. On the estimation of relationships involving qualitative variables. Am. J. Sociol. 76 (1), 103–154.

Woo, M.-J., Reiter, J.P., Oganian, A., Karr, A.F., 2009. Global measures of data utility for microdata masked for disclosure limitation. J. Priv. Confid. 1.

Yin, Y., Lin, Z., Jin, M., Fanti, G., Sekar, V., 2022. Practical GAN-based synthetic IP header trace generation using NetShare. In: Kuipers, F., Orda, A. (Eds.), Proceedings of the ACM SIGCOMM 2022 Conference. ACM, New York, NY, USA, pp. 458–472. http://dx.doi.org/10.1145/3544216.3544251.

Maximilian Wolf is a Research Associate at the Coburg University of Applied Sciences and Arts. He earned a master's degree in Computer Science from Coburg. He has also worked as a Lecturer in Data Mining and Reinforcement Learning at Coburg. His research interests include the generation of flow-based network data via generative modeling and cyber-security intrusion detection via data-mining.

Julian Tritscher is a Research Associate at the University of Wuerzburg. He holds a master's degree in Computer Science from the University of Wuerzburg. His research focus lies in the combination of anomaly detection and explainable artificial intelligence (XAI) for the detection of occupational fraud and intrusion attacks.

Dieter Landes is with Coburg University of Applied Sciences and Arts since 1999. Currently, he is a full professor of artificial intelligence and machine learning. He holds a diploma in informatics from the University of Erlangen–Nuremberg, and a doctorate in knowledge-based systems from the University of Karlsruhe. He has published around 100 papers in journals, books, and at conferences. His research interests include applications of artificial intelligence in various domains, such as cybersecurity, predictive maintenance, or learning analytics, as well as software engineering and requirements engineering.

Andreas Hotho is a professor at the University of Würzburg and holds the Chair of Data Science. Science 2020, he is the spokesman of the Centre for Artificial Intelligence and Data Science (CAIDAS) at the JMU Würzburg. He holds a Ph.D. from the University of Karlsruhe, where he worked from 1999 to 2004 at the Institute for Applied Informatics and Formal Description Methods (AIFB) in the areas of text, data and web mining, semantic web and information retrieval. From 2004 to 2009 he was a senior researcher at the University of Kassel and from 2011 to 2018 a member of L3S in Hannover. Since 2005 he has been leading the development of the social bookmarking and publication sharing platform BibSonomy. For more than 10 years, his research group has been working on topics related to data science with a focus on ranking, recommendation, semantics, knowledge graphs and most importantly deep learning in the last years. Researching new data science and machine learning methods for very large amounts of data, the group has gained experience in handling data from various social media platforms with different degrees of structure in text analysis. We have a focus on analyzing historical novels in cooperation with Digital Humanities, and on corporate data for recommendation or anomaly detection. Lately, there is also increasing interest in research on the processing of sensor data for the analysis of air pollution and bee behavior, as well as the development of new machine learning models for local climate models in cooperation with Geography. Andreas Hotho has published over 200 papers in journals and conferences, co-edited special issues and books, and co-chaired workshops. The leading European conference in the field of machine learning and data mining, ECML PKDD, was successfully organized by him and colleagues in Würzburg in 2019. He received the SWSA Ten-Year Award at the International Semantic Web Conference 2018 for his work on semantic extraction and the Best Paper Award at the Web Conference 2015 for his analysis of user behavior on the web using the HypTrails method. In recent years, his research has been supported by funding from the EU, DFG and BMBF, as well as industrial collaborations.

Daniel Schlör is a Research Associate at the University of Wuerzburg. He holds a Ph.D. from the University of Wuerzburg. He has also worked as a Lecturer in Machine Learning for Anomaly Detection. His main research interests are deep learning and anomaly detection in the fields of cybersecurity and fraud detection.