Fake it till you Detect it: Continual Anomaly Detection in Multivariate Time-Series using Generative AI

Gastón García González
IIE–FING
Universidad de la República
Montevideo, Uruguay
gastong@fing.edu.uy

Pedro Casas
Digital Safety & Security
Austrian Institute of Technology
Vienna, Austria
pedro.casas@ait.ac.at

Alicia Fernández
IIE–FING
Universidad de la República
Montevideo, Uruguay
alicia@fing.edu.uy

***Abstract*—Anomaly detection in Multivariate Time-Series (MTS) data plays an important role in multiple domains, especially in cybersecurity, for the detection of unknown attacks. *DC-VAE* is a recent approach we have proposed for anomaly detection in network measurement multivariate data, which uses Variational Auto Encoders (VAEs) and Dilated Convolutional Neural Networks (DCNNs) to model complex and high-dimensional MTS data. However, detecting anomalies using VAEs can result in performance degradation and even catastrophic forgetting when trained on dynamic and evolving network measurements, particularly in the event of concept drifts. We extend *DC-VAE* to a continual learning setup, leveraging the generative AI properties of the underlying models to deal with continually evolving data. We introduce GenDeX, an approach to Generative AI-based anomaly detection which compresses the patterns extracted from past measurements into a generative model that can synthesize MTS data out of input Gaussian noise, mimicking the characteristics of the MTS data used for training. GenDeX relies on a Deep Generative Replay paradigm to realize continual learning, combining synthesized past MTS measurements with new observations to update the detection model. Using a large-scale, multi-dimensional network monitoring dataset collected from an operational mobile Internet Service Provider (ISP), we showcase the functionality of *DC-VAE* in the event of concept drifts, and study in-depth its generative characteristics, assessing GenDeX synthetically generated MTS examples. GenDeX enables *DC-VAE* adapting to continually evolving data, overcoming the limitations of catastrophic forgetting.**

***Index Terms*—Anomaly Detection, Generative AI, VAE, Multivariate Time-Series, GenDeX**

1. Introduction

Time-series analysis is an essential approach to cybersecurity, in particular to profile temporal data behaviors and to detect anomalies in real-time. While time-series based anomaly detection has a long standing literature associated to signal processing techniques [1], modern approaches to time-series anomaly detection based on deep learning technology have flourished in recent years [2]. Most approaches in the literature address the problem by either focusing on univariate time-series modeling and analysis – running an independent detector for each time-series, or by considering multi-dimensional input data with short-term memory analysis, to avoid the scalability limitations introduced by very deep architectures, or the complexities and delays introduced by recurrent topologies. To address these limitations, we have recently introduced *DC-VAE* [3], a deep-learning based approach to *unsupervised anomaly detection* in multivariate time-series (MTS), based on Variational Auto-Encoders (VAEs) [4]. VAEs are a generative version of classical auto-encoders, with the particularity of having, by conception, a probabilistic manner to describe an observation in the latent space. Thus, rather than training an encoder which outputs a single value describing each latent state attribute, the encoder is formulated to describe a probability distribution for each latent attribute. For a given input, VAEs produce as output prediction not only an expected value, but also the associated standard deviation, corresponding to the distribution the model understands (i.e., has learned) generated the corresponding input. This automatically defines a *normality region* for each independent time-series, which can then be easily exploited for detecting deviations beyond this region. To exploit the temporal dependencies and characteristics of time-series data in a fast and efficient manner, *DC-VAE* uses Dilated Convolutional Neural Networks (DCNNs) as the VAE's encoder and decoder architecture. Compared to normal convolutions, dilated convolutions improve time-series modeling by increasing the receptive field of the neural network, reducing computational and memory requirements, and most importantly, enabling training – and detection – on longer-in-the-past temporal sequences.

One of the main limitations faced by *DC-VAE*, and in general by AI/ML-driven approaches for anomaly detection, is their inability to deal with so-called *concept drifts*. Concept drifts correspond to events where the statistical properties of the target variable or the relationships between the input features and the target variable change over time. As such, the patterns and rules that an AI/ML model learned from historical data may no longer hold in the current data, and the model may need to be updated to adapt to the changes. Concept drifts are intrinsically related to another phenomenon which impacts and degrades the performance of AI/ML models, referred to as *catastrophic forgetting*. Catastrophic forgetting is a different but related problem, which occurs when an AI/ML model trained on a set of tasks or data samples forgets previously learned information when learning new tasks or samples. Under catastrophic forgetting, the

performance of the model on the old tasks deteriorates significantly after learning new tasks, even if the old tasks and new tasks are related. Concept drift and catastrophic forgetting are strongly related because they both involve changes in the data distribution that can cause the AI/ML model to become outdated or inaccurate. Both problems require methods to adapt to changing data distributions, by retraining the underlying models. In its simplest and most effective form, retraining an AI/ML model with newly acquired data (post concept drift or for new similar tasks) typically requires all previously used training data as well. This traditional method of retraining is hence limited by the availability of past data, as well as by the amount of memory and computational resources.

We resort to the *continual learning* paradigm [5] to address the continual model adaptation and retraining of *DC-VAE*. Continual learning enables a model to learn from a stream of evolving data, without forgetting previously learned knowledge. It involves updating the model's parameters and architecture as new data arrives, while also preserving knowledge learned from previous data, representing a promising approach to deal with concept drift. We extend *DC-VAE* to a continual learning setup, leveraging the generative AI properties of the underlying VAE model to remember past data. By conception, once the encoder-decoder VAE model has been trained, the decoding function is capable to synthesize new "fake" data mimicking the characteristics of the MTS training datasets, using as input only Gaussian noise. As such, the decoder acts as a lossy compression of the data used for training. We combine *DC-VAE* and its generative decoder into GenDex, an approach to continual learning for anomaly detection in MTS network measurements. In a nutshell, when *DC-VAE* is confronted with concept drifts, or is applied to a new MTS dataset – e.g., measurements collected at a different network or representing a different process – GenDeX uses the previously trained decoder to synthesize past MTS measurements, and combines them with the new MTS data to retrain the underlying VAE model. GenDeX follows a *Deep Generative Replay* [6] paradigm for continual learning, where a generative model produces synthetic data which replays old memories during training, augmenting the heterogeneity and expressiveness of the retraining. The rationale behind GenDeX is that *DC-VAE* continually improves its tracking and baselining capabilities as it processes new measurements with different underlying statistical characteristics, improving as such its generalization and anomaly detection capabilities with time.

In this paper we study in depth the generative capabilities of *DC-VAE*, describing the underlying architecture and its adaptation to make it operate as a synthetic MTS generator. We showcase *DC-VAE* operation in different MTS datasets, including measurements collected at an operational mobile ISP. We investigate the characteristics of the resulting latent space, taming it for fine-grained, temporal data generation. Finally, we demonstrate how the trained decoder generates synthetic time-series out of Gaussian noise, which track each of the individual time-series of the MTS process, despite their different nature. The reminder of the paper is organized as follows: Section 2 briefly overviews the related work; in Section 3 we describe the *DC-VAE* model and architecture in detail, explaining its operation through application in an operational mobile ISP dataset; Section 4 shows *DC-VAE* when confronted with different types of concept drifts, introducing GenDeX; Section 5 reports the results obtained in the synthetic generation of MTS data tracking the temporal evolution of the different mobile ISP time-series. Finally, Section 6 concludes the paper.

2. Related Work

There are multiple surveys on general-domain anomaly detection techniques [1], [7], [8] as well as on network anomaly detection [9], [10]. The diversity of data characteristics and types of anomalies results in a lack of universal anomaly detection models. Modern approaches to time-series anomaly detection based on deep learning technology have flourished in recent years [2]. Due to their data-driven nature and achieved performance in multiple domains, generative models such as VAEs and Generative Adversarial Networks (GANs) have gained relevance in the anomaly detection field [11]–[17].

Modeling data sequences through a combination of variational inference and deep learning architectures has been vastly researched in other domains in recent years, mostly by extending VAEs to Recurrent Neural Networks (RNNs), with architectures such as STORN [18], VRNN [19], and Bi-LSTM [20] among others. Convolutional layers with dilation have been also incorporated into some of these approaches [21], [22], allowing to speed up the training process based on the possibilities of parallelization offered by these architectures. One of these approaches using Dilated Convolutional Neural Networks as the encoder-decoder architecture for VAEs is our *DC-VAE* model [3].

There are various approaches to continual learning, including *regularization techniques* [5], *generative replay* [6], and *dynamic architecture* [23]. Regularization techniques involve penalizing the model's parameters to reduce the impact of new data on previously learned knowledge. One of such techniques is Elastic Weight Consolidation (EWC) [5], which uses a quadratic penalty term to constrain the neural network's weights during training to protect important parameters from forgetting. Generative replay involves generating synthetic data that is similar to previously observed data to reinforce old memories. Deep Generative Replay (DGR) [6] is an example of this approach, which uses a generative model to produce synthetic data that is similar to previously observed data. The synthetic data is used to replay old memories during training to prevent forgetting. Similar to DGR, BooVAE [24] generates new data to augment the training set. However, unlike generative replay, BooVAE generates new samples by perturbing the existing data rather than directly generating new samples from scratch, (in theory) preserving the statistical properties of the original data distribution. Dynamic architecture involves expanding or shrinking the model's architecture to accommodate new knowledge or discard outdated knowledge. Progressive Neural Networks (PNN) [23] is a notable approach to dynamic architectures, which dynamically expands the neural network architecture to incorporate new knowledge while retaining previous knowledge. PNN can achieve

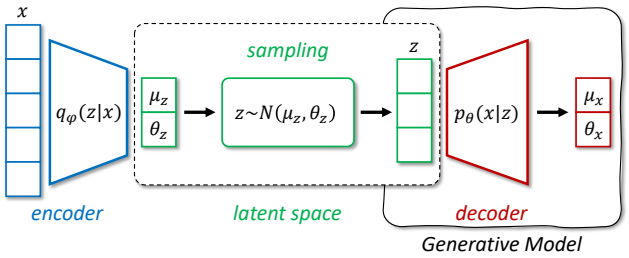


Figure 1. *DC-VAE*'s variational autoencoder. The encoder/decoder architecture is based on Dilated CNNs.

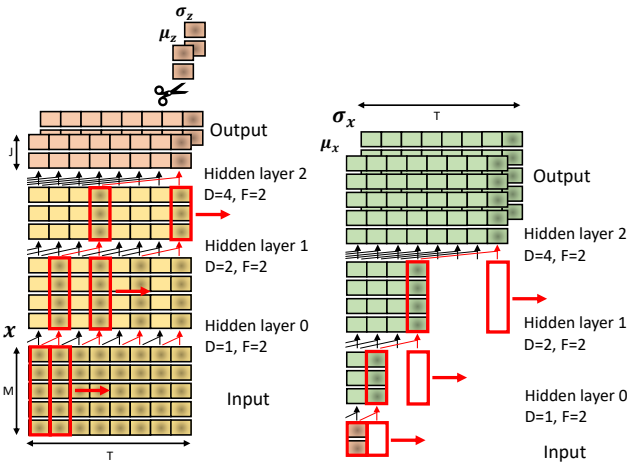
high accuracy on sequential learning tasks without forgetting previously learned knowledge. This paper is a continuation of our initial work on continual learning for anomaly detection in multivariate time-series data [25] (extended abstract/poster presentation).

3. *DC-VAE* Model and Architecture

MTS data is generally processed through sliding windows, condensing the information of the most recent T measurements. We define $\boldsymbol{x}$ as a matrix in $\mathbb{R}^{M\times T}$, where M is the number of variables in the MTS process. As depicted in Figures 1 and 2, for a given input $\boldsymbol{x}$, the trained VAE model produces two different predictions, $\boldsymbol{\mu}_x$ and $\boldsymbol{\sigma}_x$ – matrices in $\mathbb{R}^{M\times T}$, corresponding to the parameterization of the probability distribution which better represents the given input. If the VAE model was trained (mainly) with data describing the normal behavior of the monitored system, then the output for a non-anomalous input would not deviate from the mean $\boldsymbol{\mu}_x$ more than a specific integer $\boldsymbol{\alpha}$ times the standard deviation $\boldsymbol{\sigma}_x$. On the contrary, if the input presents an anomaly, the output would not belong to this normality region.

The main goal of the VAE model is to learn a compressed representation of $\boldsymbol{x}$ in an unsupervised manner. This compressed representation $\boldsymbol{z}$ is referred to as a latent variable, and it is learned by training the VAE to generate data that is similar to the input data. Similar to $\boldsymbol{x}$, $\boldsymbol{z}$ will also be a sequence of length T, but with a smaller number of dimensions $J < M$, $\boldsymbol{z} \in \mathbb{R}^{J\times T}$. VAEs learn a probabilistic mapping between the input data and its latent variable, which allows to generate new data by sampling from the learned latent variable distribution. The *probabilistic* encoder computes the approximated posterior distribution $p_\theta(\boldsymbol{z}|\boldsymbol{x})$ in the form of $q_\phi(\boldsymbol{z}|\boldsymbol{x}) \approx p_\theta(\boldsymbol{z}|\boldsymbol{x})$, whereas the conditional likelihood distribution $p_\theta(\boldsymbol{x}|\boldsymbol{z})$ is realized by the *probabilistic* decoder, both distributions parameterized by ϕ and θ, respectively. In the vanilla VAE, the latent space is assumed to be a set of multivariate Gaussian distributions, and therefore, $\boldsymbol{z} \sim q_\phi(\boldsymbol{z}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{\mu_z}, {\boldsymbol{\sigma_z}}^2)$.

The latent vector $\boldsymbol{z}$ is sampled from the encoder-generated distribution before feeding it to the decoder. This random sampling makes it difficult for backpropagation to happen for the encoder, as errors cannot be propagated. VAE uses a re-parameterization trick to model the sampling process, which makes it possible for the errors to propagate through the network. The latent vector $\boldsymbol{z}$ is explicitly represented as a function of the encoder's output $\{\boldsymbol{\mu}_z, \boldsymbol{\sigma}_z\}$, in the form $\boldsymbol{z} = \boldsymbol{\mu_z} + \boldsymbol{\sigma_z}\boldsymbol{\varepsilon}$, with $\boldsymbol{\varepsilon} \sim \mathcal{N}(0, \boldsymbol{I})$.



(a) Encoder network. (b) Decoder network.

Figure 2. *DC-VAE* encoder/decoder architecture using causal dilated convolutions, implemented through a stack of 1D convolutional layers. The architecture is slightly modified for GenDeX synthetic MTS generation, trimming $\boldsymbol{z}$ to keep an easier to analyze latent space.

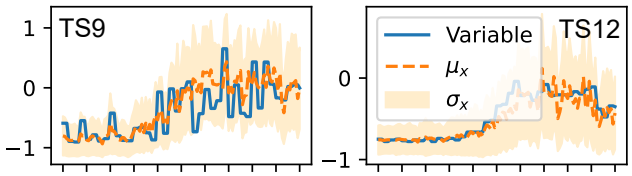
To exploit the temporal dimension of the input time-series, *DC-VAE* encoder/decoder architecture is based on popular CNNs, using Dilated Convolutions (DCs) [26]. DC is a technique that expands the input by inserting gaps between its consecutive samples. In simpler terms, it is the same as a normal convolution, but it involves skipping samples, so as to cover a larger area of the input.

Figure 2 depicts the encoder/decoder architecture used in *DC-VAE*. The network architecture must be such that the output values depend on all previous input values. The length T of the sliding window plays a key role here, as it must ensure that the output at t depends on the input at that time and at $\{t-1, t-2, \ldots, t-T+1\}$. The simplest way to achieve this is to use filters of length $F = 2$ and DCs with dilatation factor $d = F^h$, which grow exponentially with the layer depth $h \in [0, H-1]$, where H is the number of layers of the network. Subsequently, H is the minimum value that verifies: $T \leq 2 * F^{H-1}$. In the example, the window length is $T = 8$, and the target is achieved by taking $H = 3$ layers. This direct relationship between T and the network architecture has a strong practical impact, making it easy to construct the encoder/decoder, based on the desired temporal-depth of the analysis. The original architecture of *DC-VAE* is slightly modified for GenDeX synthetic MTS generation, trimming $\boldsymbol{z} \in \mathbb{R}^{J\times T}$ to keep a more compact and easier to analyze latent space. As shown in Figure 2, we only keep the last dimension of the latent variable $\boldsymbol{z}$ at T, resulting in a vector $\boldsymbol{z} \in \mathbb{R}^J$. We analyze the resulting latent space in Section 5.

Training the VAE involves minimizing the standard ELBO loss function [4], consisting of a *reconstruction loss*, which measures the difference between the original input data and its reconstruction, and a *regularization term*, which forces the learned latent variable distribution to be close to a standard normal distribution. This term ensures that the learned latent variables are meaningful and useful for generating new data samples. Model training is fully unsupervised and on top of normal-operation data, to capture the baseline for anomaly detection. Once trained, the detection process runs continually, rolling the sliding window of length T by a unitary-time step.

dataset	# samples	duration	# anomalous samples
training	310,980	3 months	5,407 (**1.7%**)
validation	103,680	1 month	385 (**0.4%**)
testing	317,952	3 months	7754 (**2.4%**)
total	732,612	7 months	13,546 (**1.8%**)

TABLE 1. TELCO DATASET. SEVEN-MONTHS WORTH OF MEASUREMENTS MANUALLY LABELED, FOR 12 DIFFERENT METRICS.



(a) Prediction of TS_9. (b) Prediction of TS_{12}.

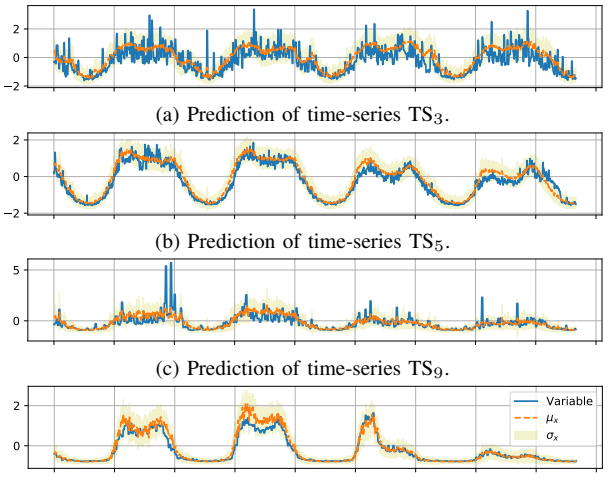
Figure 3. Example of time-series analysis through *DC-VAE*. Normal-operation is defined by $\boldsymbol{\mu_x}$ and $\boldsymbol{\sigma_x}$.

We evaluate *DC-VAE* in a proprietary MTS dataset, corresponding to real measurements collected at an operation mobile ISP. The TELCO dataset corresponds to twelve different time-series TS_1 to TS_{12}, with a temporal granularity of five minutes per sample, collected and manually labeled for a period of seven months, between January 1 and July 31, 2021. Table 1 presents the main details of the dataset. Note in particular how strongly imbalanced is the dataset in terms of normal-operation and anomalous samples, which is the typical case for real cybersecurity measurements in operational deployments. By definition, anomalies are rare events. We split the full dataset in three independent, time-ordered sub-sets, using measurements from January to March for model training, April for model validation, and May to July for testing.

For reference and to better understand *DC-VAE*'s operation, Figures 3, 4, and 5 present *DC-VAE* predictions, using a sliding-window of length $T = 512$ samples, corresponding to roughly two days of past measurements. For each of the displayed time-series TS_i, its real value x_i, along with the outputs of the VAE μ_{x_i} and σ_{x_i}, are reported. *DC-VAE* properly tracks different and individual types of behavior in the time-series, including the strong seasonal daily component, but also the operation during weekdays and weekends, e.g., visible in Figure 4(d). In this example, time-series TS_3 and TS_9 are noisier than time-series TS_5 and TS_{12}, which justifies the need for different sensitivity thresholds $\boldsymbol{\alpha} = \{\alpha_i\}$ to address the underlying nature of each monitored metric. Indeed, in *DC-VAE*, each α_i can be set individually, for each time-series. Note in addition how different periods of time-series variability result in more or less tight normal-operation regions estimated by *DC-VAE*. The detection of anomalies with different nature is depicted in Figure 5.

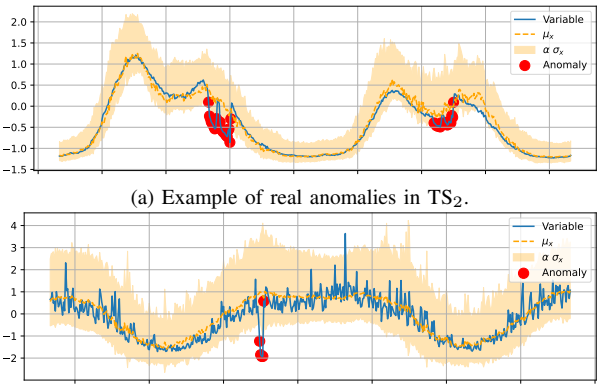
4. GenDeX - Continual Learning for *DC-VAE*

A Concept Drift (CD) can manifest itself as a shift in the mean, an increase or decrease in the variance, or even as complete data modifications. Such changes may be related to important trends in the data or to measurements collected in a different setup, requiring proper detection and retraining. Figure 6 shows an example of *DC-VAE*



(a) Prediction of time-series TS_3.
(b) Prediction of time-series TS_5.
(c) Prediction of time-series TS_9.
(d) Prediction of time-series TS_{12}.

Figure 4. Example of time-series analysis through DC-VAE, using $T = 512$ samples – almost 2 days of temporary receptive field in the past.



(a) Example of real anomalies in TS_2.
(b) Example of real anomalies in TS_4.

Figure 5. Examples of real anomalies present in the analyzed dataset, and their identification by *DC-VAE*.

operation under a concept drift, where a gradual change in the interval indicated as the CD zone is simulated in a single time-series (TS_5), leaving the other series untouched. *DC-VAE* is not capable to track this individual drift, given its multivariate nature – the complete MTS process introduces an hysteresis effect in the reaction of the model. Note in particular how the model can perfectly track the non-modified time-series, and how the estimation for TS_5 follows the pre-CD pattern. Once the induced drift is over, and the MTS process returns to previous statistical behavior, *DC-VAE*'s tracking for TS_5 becomes again accurate. Figure 7 shows *DC-VAE* under a more drastic concept drift, in this case considering data from different years (2015 and 2017) from the open SWaT dataset [27] – commonly used for detection of cyber-attacks in cyber-physical systems. Figure 7(a) shows the tracking of *DC-VAE* in (top) the 2015 normal operation dataset used for training, (middle) the 2015 attack dataset used for testing, and (bottom) the 2017 dataset. *DC-VAE* performs accurately in the testing dataset, as the underlying empirical distributions of both training and testing datasets significantly overlap, as evidenced in Figure 7(b). However, the model totally fails to capture the SWaT dataset in 2017, as the underlying distributions of the corresponding data are significantly different.

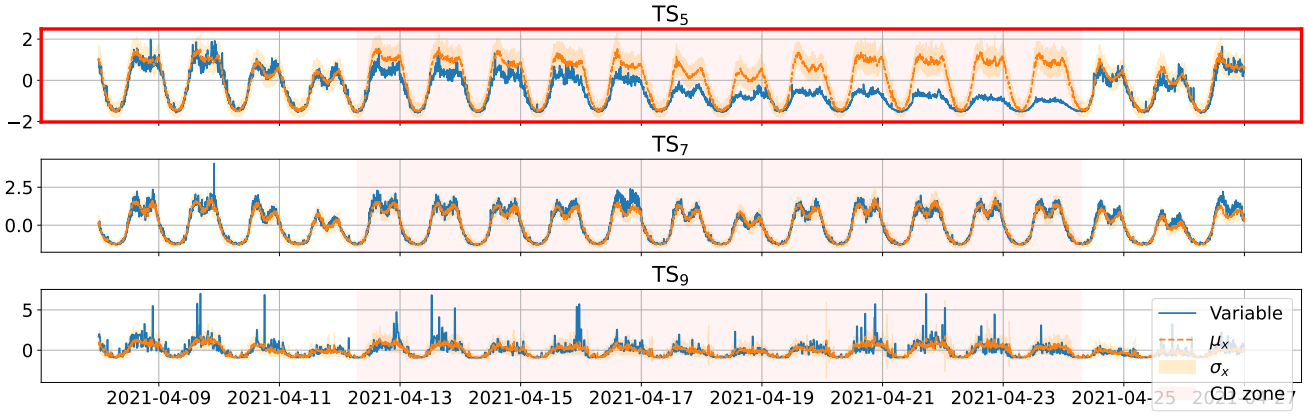
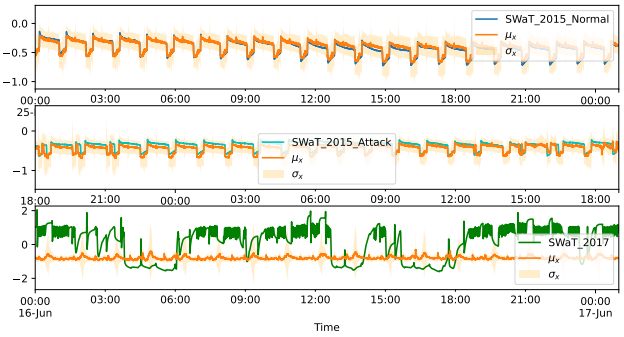
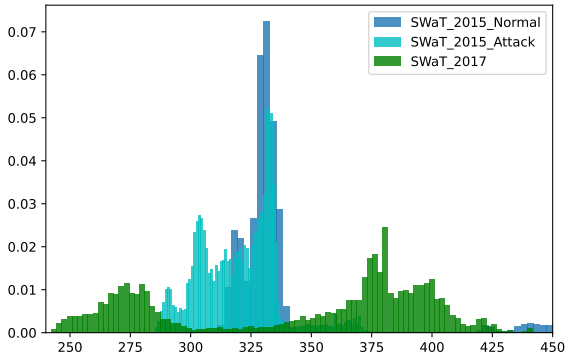


Figure 6. *DC-VAE* response to univariate concept-drift: a gradual linear fall of the values during the day.



(a) *DC-VAE* under strong concept drift.



(b) Subsets S_{2015-N}, S_{2015-A}, and S_{2017}.

Figure 7. Strong subset changes requires retraining.

We therefore explore an approach to cope with the described concept drifts, in particular exploiting the generative nature of the *DC-VAE* model for continual learning. In a continual learning framework, we assume a continually evolving stream of data, represented as a sequence of subsets S_j, each characterized by a specific underlying distribution. We define a sequence of λ_∞ subsets $S_1, \ldots, S_{\lambda_\infty}$ sequentially arriving, and assume access to only the data in current subset S_t, with $t \leq \lambda_\infty$. We consider a CD occurring at time t, and thus, assume that the underlying distributions of $S_1, \ldots, S_{t-1}$ are similar among them, but significantly different from S_t. An initial *DC-VAE* model is trained using S_1 data, which performs accurately till time t. We refer to this model as *DC-VAE*$_0 = \{q^0_\phi, p^0_\theta\} = \{E^0_\phi, D^0_\theta\}$, where E and D represent the encoding and decoding functions, respectively.

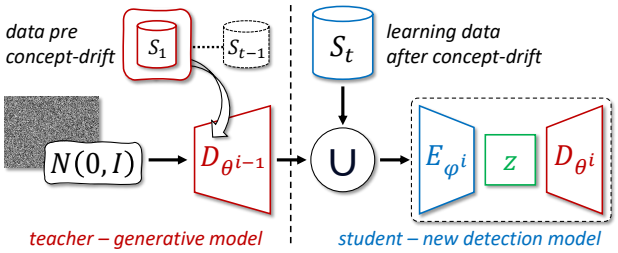
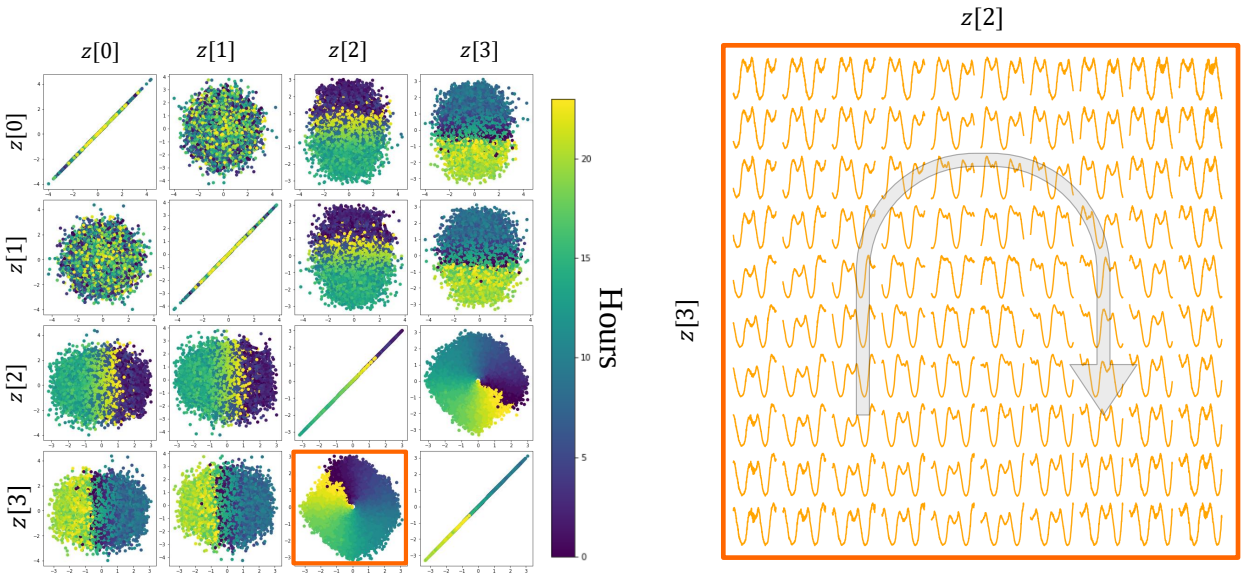


Figure 8. The GenDeX generative replay approach. At time t, a concept drift significantly modifying the underlying distribution of S_t triggers a model retraining event i.

GenDeX follows the principles behind Deep Generative Replay (DGR) [6] to adapt *DC-VAE*$_0$ to the new data S_t, without forgetting the parameterization learned from S_1, valid for $S_1, \ldots, S_{t-1}$. Figure 8 explains the GenDeX approach. The decoding function D^0_θ acts as generator, and it is used to synthesize a new dataset $F_{1\to(t-1)}$ out of Gaussian noise, which mimics former training examples in S_1 and its underlying distribution. We say D^0_θ acts as the *teacher* model. Then, the new *student* model *DC-VAE*$_1$ is trained on joint synthetic data F and new data S_t. This approach is conceptually simple, model-agnostic and overcomes catastrophic forgetting, as the updated model *DC-VAE*$_1$ is now capable to handle pre- and post-concept drift data distributions. The challenging part in GenDeX is to tame the latent space of *DC-VAE* to actually generate an MTS process which reliably reproduces the data initially used for training, in a fully controllable manner.

Recall that the latent space $\boldsymbol{z} \in \mathbb{R}^{J\times T}$ can be potentially huge, e.g., in the examples we showed in Section 3, $J = 4$ and $T = 512$, so we have to deal with a 2048-dimensional space, and thus, sampling Gaussian noise of such dimensionality might not generate the desired outcome. Therefore, as mentioned before and as reflected by the architecture of *DC-VAE* in Figure 2, we trim the latent space dimensionality and focus exclusively on $\boldsymbol{z}$ at T, resulting in a vector $\boldsymbol{z} \in \mathbb{R}^J$. Realizing a latent space where the sample distribution approaches a zero-one normal distribution, as the VAE hypothesis states, helps the generative part of the VAE model, i.e., the decoder, to generate samples that resemble the real ones, by simply drawing inputs from such a Gaussian distribution. Next, we demonstrate how to realize the generating function in



(a) Latent space representation.

(b) Synthetic time-series generation.

Figure 9. GenDeX latent space representation. Latent space $\boldsymbol{z}$ with $J = 4$. The colors correspond to the hours of the day. Grid of samples generated from uniform sampling on dimensions $z[2]$ and $z[3]$ of the $\boldsymbol{z}$ latent space. If the figure is traversed clockwise, it is possible to see how the generated time-series evolve over time.

the practice, exploring the latent space and reporting the results obtained in the synthetic generation of MTS data from the TELCO time-series dataset.

5. Exploring GenDeX Generative AI

We now focus on the generative properties of *DC-VAE*, firstly by analyzing the latent space generated by the encoding function E_ϕ, and then by exploring the generative capabilities of the generative model as represented by the trained decoding function D_θ. The dimension of the latent space in a VAE model is one of the hyper-parameters to define during model evaluation. These dimensions are restricted by the dimensions of the input samples $\boldsymbol{x}$ space, as for the model to only capture the relevant information or energy of the samples, there must be a dimension reduction. By conception and hypothesis, the distribution of the samples $\boldsymbol{z}$ living in the latent space must be a normal distribution with zero mean and an identity covariance matrix. This is enforced during training with the second term of the ELBO loss function.

To evaluate the behavior of the encoder E_ϕ, a representation of the latent space is shown for a trained *DC-VAE* model, using TELCO data. We take $J = 4$, resulting in $\boldsymbol{z} = \{z[0], z[1], z[2], z[3]\}$. Figure 9(a) depicts the resulting latent representation, projecting on each bi-dimensional combination of dimensions $z[i]$. Each point in the figure corresponds to the projection of a sample from the validation set. The first observation to highlight is that the distribution of samples in $\boldsymbol{z}$ does look very close to a zero-one normal distribution. It is certainly centered at zero, and the highest concentration of points is in the range $[-3, 3]$. As we explained before, while this is enforced by the ELBO target loss function the VAE was trained for, it is not always properly realized, due to usual problems of so-called *degeneration learning* in generative models. VAEs in particular suffer from the problem of

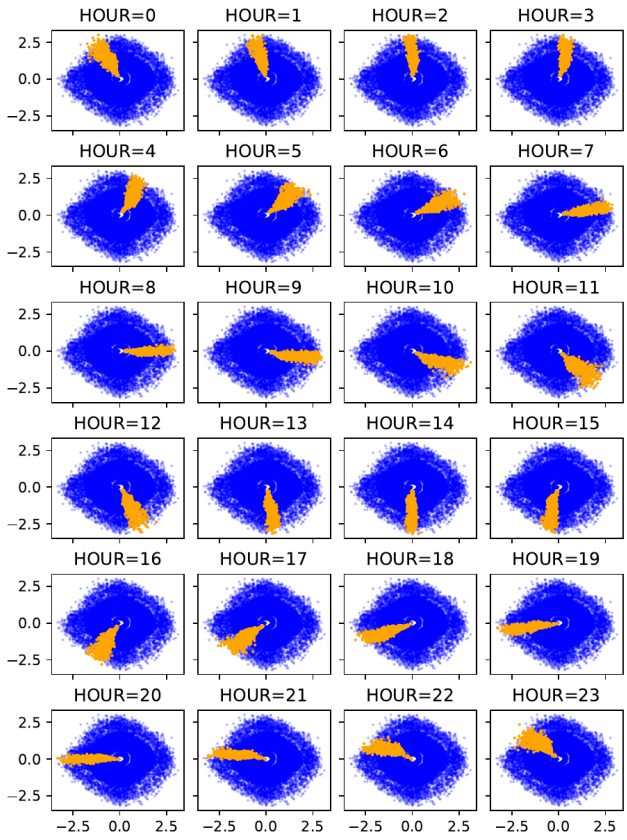


Figure 10. *DC-VAE* latent space representation, in an hourly basis. Sampling the latent space at different angles results in different times of the day in the generated time-series.

degeneration when using deep architectures [28], which seriously weakens the correlation between the input and the corresponding latent variables, failing in both latent representation and generation. The usage of dilated con-

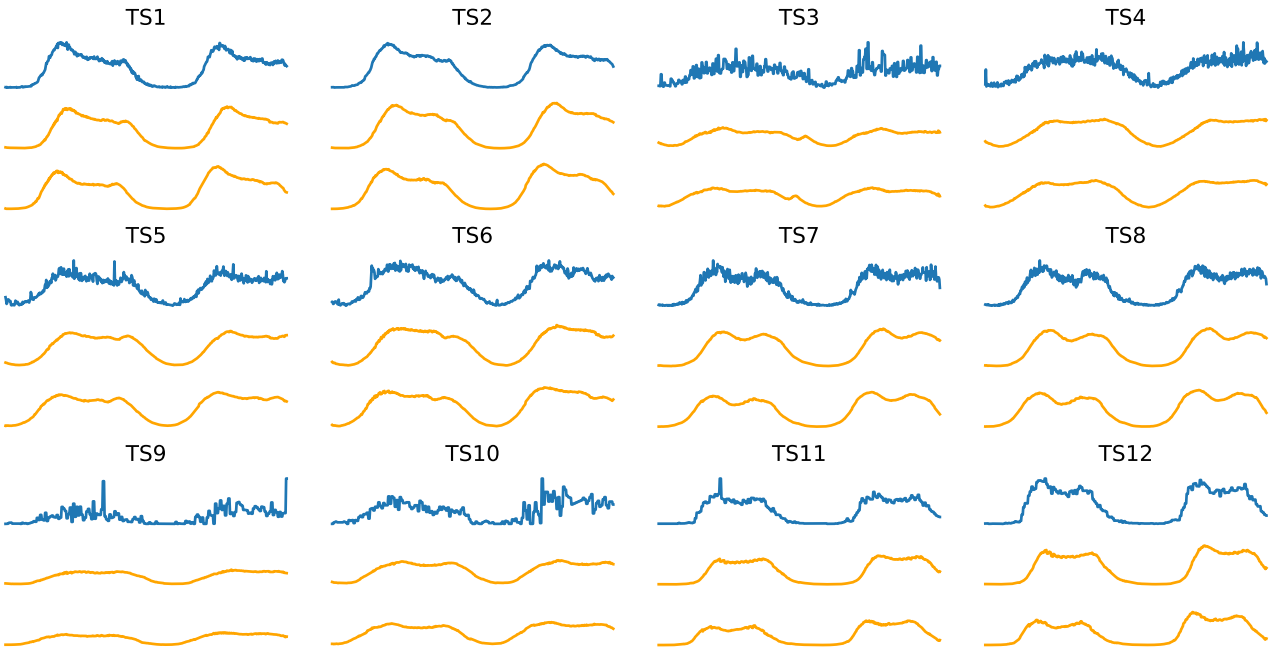


Figure 11. Synthetic MTS data generated through GenDeX. For each time-series in TELCO, two examples of time-series generated from noise are depicted. The trend of the twelve time-series is perfectly captured by the synthetically generated examples.

volutions helps *DC-VAE* stabilizing the latent space, as skipping connections in a VAE deep architecture is known to enable the preservation of information between inputs and latent variables [28].

The Gaussian property of the latent space distribution is essential for the MTS generation process, as there are no input samples $\boldsymbol{x}$ in GenDeX to use as reference, thus samples need to be generated from input noise. Besides the shape of the realized distribution, and to reflect the temporal dimension of the MTS data, Figure 9(a) depicts the coded samples in colors, each color representing a different hour of the day. More specifically, each sample color corresponds to the discretized hourly values of the newest sample-value within the input sequence, at time t. If we consider the bi-dimensional latent space $\{z[2], z[3]\}$, we observe how each hour of the day maps to a different angular area in the data distribution. To appreciate this effect better, Figure 10 shows the same encoding, but this time highlighting the $\{z[2], z[3]\}$ values for each hour. Interestingly, each hour has a particular range of angles, and these are sequentially arranged, ordered continuously by hour of the day. Under this setup, it is enough to feed the decoder D_θ with samples drawn from a zero-one normal distribution to generate synthetic MTS samples out of noise. Figure 9(b) shows a series of synthetically generated time-series, by uniformly sampling on dimensions $z[2]$ and $z[3]$. If the figure is traversed clockwise, it is possible to appreciate how the generated time-series evolve over time.

We now move on to the generation of synthetic MTS data, for the twelve time-series in TELCO, using D_θ. Figure 11 shows two examples per time-series generated out of noise, along with real time-series included in the original validation set, for two days worth of time series duration. The trend of the twelve time-series is perfectly captured by the synthetically generated examples, with the paramount advantage of these being synthetically generated by D_θ. The twelve time-series are properly generated, despite having different types of behavior and variability.

To evaluate the generative power of GenDeX more broadly, we generate the same number of samples as those in the validation set for each time-series, and compare them with the real time-series values in the validation set. Figure 12 reports, for each time-series, the distribution of the generated and real values, in the form of a histogram. Each pair of distributions have strong overlapping, especially for non-spiky values. Time-series TS_3, TS_9, and TS_{10} show a rather variable behavior, with values strongly deviating from the baseline, which cannot be tracked by the generated baseline values, as shown in the corresponding histograms. Recall that we are using GenDeX to track the form and trends of the time-series, by generating μ_x, which would naturally not capture spiky behaviors. Indeed, we are interested in adapting the baselines for anomaly detection, to enable a proper detection of deviations from these baselines.

6. Concluding Remarks

DC-VAE is a promising approach for anomaly detection in network measurement multivariate data, but similarly to other learning-based approaches, it requires retraining when confronted to concept drifts and new detection tasks. We have extended *DC-VAE* to a continual learning setup, leveraging the generative AI properties of the underlying models to deal with continually evolving data. Through GenDeX, *DC-VAE* can be easily retrained without requiring access to past MTS data, maintaining modeling performance without forgetting previously learned knowledge. The rationale behind GenDeX is that *DC-VAE* can continually improve its tracking and baselining capabilities as it processes new measurements with different underlying statistical characteristics, improving

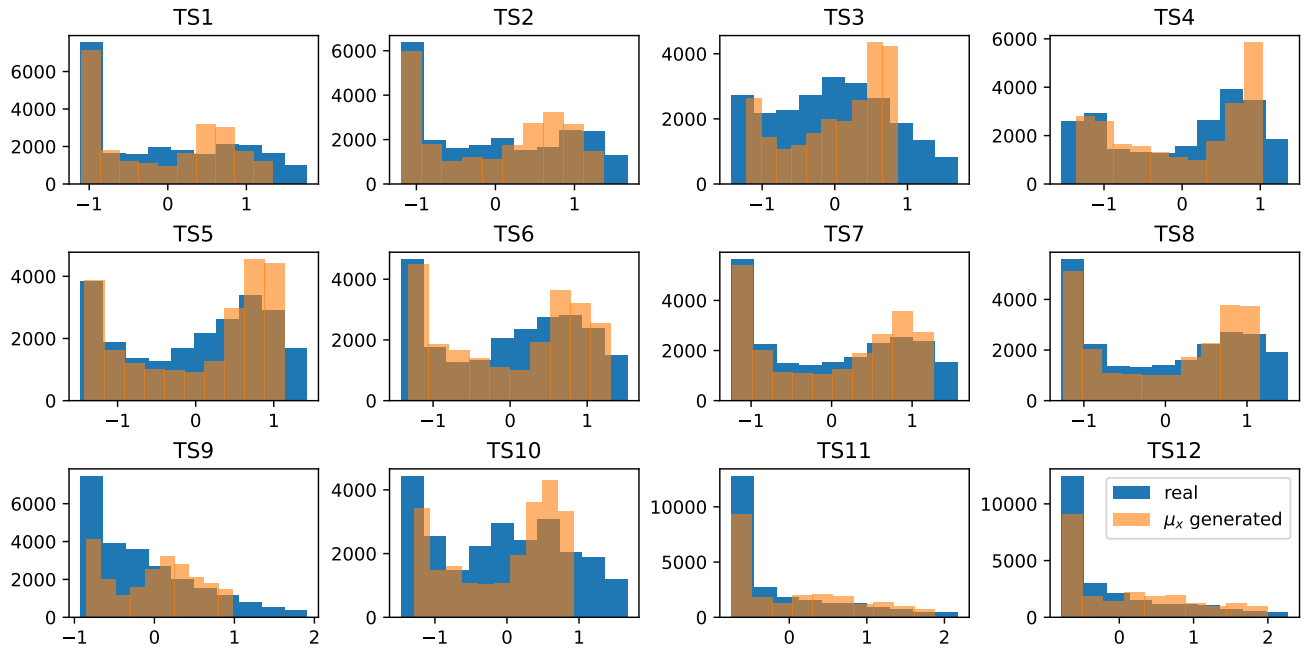


Figure 12. Synthetic MTS data generated through GenDeX. Histograms of samples (μ_x) generated from noise for each time-series of the TELCO dataset. The same number of samples as those in the validation set are generated for each time-series.

as such its generalization and anomaly detection capabilities with time. In this paper we have investigated the generative AI capabilities of *DC-VAE*, exploring the latent space and demonstrating how to tame it to enable a generative process of MTS synthetic data. Using real ISP measurements, we have shown how GenDeX synthetically generated MTS samples perfectly track the behavior and trends of the real data originally used for model training, drawing these samples from simple Gaussian noise.

While not reported in this paper, our initial results on the application of GenDeX for improved anomaly detection under emulated concept drift scenarios and different datasets show that it is possible to improve *DC-VAE* baseline modeling and anomaly detection as more data is added, despite having different underlying distributions. We are working on a large-scale benchmarking to demonstrate the performance gain introduced by GenDeX.

Finally, throughout the paper we have assumed that GenDeX is applied once a concept drift takes place, but in the practice, such concept drifts must be automatically detected to trigger the proposed retraining process. There is abundant literature in the problem of on-line and off-line concept drift detection, and thus assume it out of the scope of this paper; nevertheless, a practical deployment of GenDeX requires the instantiation of such a concept drift detection step. GenDeX enables *DC-VAE* dealing with concept drifts and adapting to continually evolving data, overcoming the limitations of catastrophic forgetting, retraining without the need of past data.

Acknowledgment

This work has been partially supported by the Austrian FFG ICT-of-the-Future project *DynAISEC – Adaptive AI/ML for Dynamic Cybersecurity Systems*, and by the ANII-FMV project with reference FMV-1-2019-1-155850 *Anomaly Detection with Continual and Streaming Machine Learning on Big Data Telecommunications Networks*. Gastón García was supported by the ANII scholarship POS-FMV-2020-1-1009239, and by CSIC, under program *Movilidad e Intercambios Académicos 2022*.

References

[1] A. Blázquez-García, A. Conde, U. Mori, and J. A. Lozano, "A review on outlier/anomaly detection in time series data," *ACM Comput. Surv.*, vol. 54, no. 3, Apr. 2021.

[2] G. Pang, C. Shen, L. Cao, and A. V. D. Hengel, "Deep learning for anomaly detection: A review," *ACM Comput. Surv.*, vol. 54, no. 2, Mar. 2021.

[3] G. García González, S. Martinez Tagliafico, A. Fernández, G. Gómez, J. Acuña, and P. Casas, "DC-VAE, Fine-grained Anomaly Detection in Multivariate Time-Series with Dilated Convolutions and Variational Auto Encoders," in *2022 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, 2022, pp. 287–293.

[4] D. P. Kingma and M. Welling, "Auto-encoding Variational Bayes," *CoRR*, vol. abs/1312.6114, 2013. [Online]. Available: https://arxiv.org/abs/1312.6114

[5] J. Kirkpatrick, R. Pascanu, N. C. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell, "Overcoming Catastrophic Forgetting in Neural Networks," *CoRR*, vol. abs/1612.00796, 2016. [Online]. Available: http://arxiv.org/abs/1612.00796

[6] H. Shin, J. K. Lee, J. Kim, and J. Kim, "Continual Learning with Deep Generative Replay," *CoRR*, vol. abs/1705.08690, 2017. [Online]. Available: http://arxiv.org/abs/1705.08690

[7] M. Gupta, J. Gao, C. Aggarwal, and J. Han, "Outlier detection for temporal data," *Synthesis Lectures on Data Mining and Knowledge Discovery*, vol. 5, no. 1, pp. 1–129, 2014.

[8] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, Jul. 2009. [Online]. Available: https://doi.org/10.1145/1541880.1541882

[9] M. Ahmed, A. N. Mahmood, and J. Hu, "A survey of network anomaly detection techniques," *Journal of Network and Computer Applications*, vol. 60, pp. 19–31, 2016.

[10] W. Zhang, Q. Yang, and Y. Geng, "A survey of anomaly detection methods in networks," in *2009 International Symposium on Computer Network and Multimedia Technology*. IEEE, 2009, pp. 1–3.

[11] S. Zavrak and M. Iskefiyeli, "Anomaly-based intrusion detection from network flow features using variational autoencoder," *IEEE Access*, vol. 8, pp. 108 346–108 358, 2020.

[12] H. Zenati, C. S. Foo, B. Lecouat, G. Manek, and V. R. Chandrasekhar, "Efficient GAN-based anomaly detection," *arXiv preprint arXiv:1802.06222*, 2018.

[13] R.-Q. Chen, G.-H. Shi, W. Zhao, and C.-H. Liang, "A joint model for IT operation series prediction and anomaly detection," *Neurocomputing*, vol. 448, pp. 130–139, 2021.

[14] J. Donahue, P. Krähenbühl, and T. Darrell, "Adversarial feature learning," *arXiv preprint arXiv:1605.09782*, 2016.

[15] D. Li, D. Chen, B. Jin, L. Shi, J. Goh, and S.-K. Ng, "MAD-GAN: Multivariate anomaly detection for time series data with generative adversarial networks," in *International Conference on Artificial Neural Networks*. Springer, 2019, pp. 703–716.

[16] A. Geiger, D. Liu, S. Alnegheimish, A. Cuesta-Infante, and K. Veeramachaneni, "TadGAN: Time series anomaly detection using generative adversarial networks," in *2020 IEEE International Conference on Big Data (Big Data)*. IEEE, 2020, pp. 33–43.

[17] G. García González, P. Casas, A. Fernández, and G. Gómez, "On the usage of generative models for network anomaly detection in multivariate time-series," *SIGMETRICS Perform. Eval. Rev.*, vol. 48, no. 4, p. 49–52, may 2021. [Online]. Available: https://doi.org/10.1145/3466826.3466843

[18] J. Bayer and C. Osendorfer, "Learning stochastic recurrent networks," *arXiv preprint arXiv:1411.7610*, 2014.

[19] J. Chung, K. Kastner, L. Dinh, K. Goel, A. C. Courville, and Y. Bengio, "A recurrent latent variable model for sequential data," *Advances in neural information processing systems*, vol. 28, 2015.

[20] S. Shabanian, D. Arpit, A. Trischler, and Y. Bengio, "Variational bi-LSTMs," *arXiv preprint arXiv:1711.05717*, 2017.

[21] Z. Yang, Z. Hu, R. Salakhutdinov, and T. Berg-Kirkpatrick, "Improved variational autoencoders for text modeling using dilated convolutions," in *International conference on machine learning*. PMLR, 2017, pp. 3881–3890.

[22] G. Lai, B. Li, G. Zheng, and Y. Yang, "Stochastic wavenet: A generative latent variable model for sequential data," *arXiv preprint arXiv:1806.06116*, 2018.

[23] A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell, "Progressive neural networks," *CoRR*, vol. abs/1606.04671, 2016. [Online]. Available: http://arxiv.org/abs/1606.04671

[24] A. Kuzina, E. Egorov, and E. Burnaev, "Boovae: Boosting approach for continual learning of vae," *Advances in Neural Information Processing Systems*, vol. 35, 2021.

[25] G. G. González, P. Casas, A. Fernández, and G. Gómez, "Steps towards Continual Learning in Multivariate Time-Series Anomaly Detection Using Variational Autoencoders," in *Proceedings of the 22nd ACM Internet Measurement Conference*, ser. IMC '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 774–775. [Online]. Available: https://doi.org/10.1145/3517745.3563033

[26] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[27] A. P. Mathur and N. O. Tippenhauer, "SWaT: A Water Treatment Testbed for Research and Training on ICS Security," in *IEEE International Workshop on Cyber-physical Systems for Smart Water Networks (CySWater)*, 2016, pp. 31–36.

[28] H. Zheng, J. Yao, Y. Zhang, and I. W. Tsang, "Degeneration in VAE: in the Light of Fisher Information Loss," 2018.