One Model to Find them All Deep Learning for Multivariate Time-Series Anomaly Detection in Mobile Network Data

G. García González, S. Martinez Tagliafico, A. Fernández, G. Gómez, J. Acuña*, P. Casas†
IIE–FING, Universidad de la República (Montevideo, Uruguay)
*Telefónica Uruguay & IIE–FING (Montevideo, Uruguay)
†AIT Austrian Institute of Technology (Vienna, Austria)

***Abstract*—Network monitoring data generally consists of hundreds of counters periodically collected in the form of time-series, resulting in a complex-to-analyze multivariate time-series (MTS) process. Traditional time-series anomaly detection methods target univariate time-series analysis, which makes the MTS analysis cumbersome and prohibitively complex. We present *DC-VAE* (Dilated Convolutional - Variational Auto Encoder), a novel approach to anomaly detection in MTS data, leveraging convolutional neural networks (CNNs) and variational autoencoders (VAEs). *DC-VAE* detects anomalies in MTS data through a single model, exploiting temporal information without sacrificing computational and memory resources. In particular, instead of using recursive neural networks, large causal filters, or many layers, *DC-VAE* relies on Dilated Convolutions (DC) to capture long and short-term phenomena in the data. We evaluate *DC-VAE* on the detection of anomalies in the TELCO TELeCOmmunication-networks dataset, a large-scale, multi-dimensional network monitoring dataset collected at an operational mobile Internet Service Provider (ISP), where anomalous events were manually labeled by experts during seven months, at a five-minutes granularity. We benchmark *DC-VAE* against a broad set of traditional time-series anomaly detectors from the signal processing and machine learning domains. We also evaluate *DC-VAE* in open, publicly available datasets, comparing its performance against other multivariate anomaly detectors based on deep learning generative models. Results confirm the advantages of *DC-VAE*, both in terms of MTS data modeling, as well as for anomaly detection. For the sake of reproducibility and as an additional contribution, we make the TELCO dataset publicly available to the community and openly release the code implementing *DC-VAE*.**

***Index Terms*—Anomaly Detection, Deep Learning, Multivariate Time-Series, Variational Auto Encoder, Dilated Convolution, TELCO Open Dataset**

I. Introduction

Network monitoring data often consists of hundreds or thousands of variables periodically measured and analyzed in the form of time-series, resulting in a complex-to-analyze multivariate time-series (MTS) process. Real-time anomaly detection in such MTS processes is a key ingredient for network management, particularly to detect performance degradation and service disruption events that might strongly impact end customers or failures impacting the network's health. There is a vast literature on the problem of anomaly detection in time-series using traditional statistical models [1]–[5]; due to the non-stationary, non-linear, and high-noise characteristics of network monitoring data, traditional models have difficulty predicting these time-series with high precision. Hence, modern approaches to time-series anomaly detection based on deep learning technology have flourished in recent years [6]. 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.

Despite the broad literature, detecting anomalies in time-series data through machine-learning remains a highly arduous task [1], [6], and it has re-gained strong attention in recent years. Some of the ever-present challenges to deal with include the lack of labels and the contamination of normal operation data with anomalies, the high imbalance between normal and anomalous data, and the occurrence of so-called concept drifts [7]–[9], referring to changes in the underlying statistical properties of the analyzed data and prediction targets. By definition, anomalies are rare and sporadic-in-time events; thus, there is generally little information on them for deeper characterization and eventual future detection. This lack of insights into anomalies makes it generally difficult to employ supervised techniques fingerprinting different anomalous behaviors. On top of this, the characterization of an anomaly is typically bound to a certain time-period, which might not necessarily represent what could happen in the future. These challenges have led to a rise in the application of unsupervised learning-based approaches to time-series anomaly detection. Unsupervised learning certainly copes with many flagged challenges, although purely unsupervised approaches tend to realize significantly high false-alarm rates. The availability of partially labeled anomalies can alleviate this problem, enabling weakly-supervised anomaly detection approaches [6], to the detriment of inaccurate or imprecise ground-truth, which introduces a further challenge for model generalization.

Another challenge we deem relevant is the benchmarking of time-series anomaly detection approaches through open datasets of questionable quality. Most of the recent papers in the topic test on one or more of a handful of popular benchmark datasets, including Yahoo [10], Numenta [11], [12], NASA [13], and more. Recent studies [14] have shown that these datasets have flaws that make them unsuitable

for evaluating anomaly detection algorithms, making it even harder to assess the goodness of recent contributions in the domain.

In this paper, we conceive a novel approach for MTS anomaly detection, tackling many of the aforementioned challenges. We introduce *DC-VAE*, a deep-learning-based, unsupervised, and multivariate approach to real-time anomaly detection in MTS, based on popular Variational Auto-Encoders (VAEs) [15]. VAEs are a generative version of classical auto-encoders, with the advantage of producing 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. Using VAEs as an underlying approach allows the user to visualize the region of normal behavior in an interpretable way, enabling fine-grained, per univariate time-series anomaly detection.

To exploit the temporal dependencies and characteristics of time-series data in a fast and efficient manner, we take a Dilated Convolutional (DC) Neural Network (NN) as the VAE's encoder and decoder architecture. DCNNs have shown excellent performance for processing sequential data in a causal manner [16], i.e., without relying on recursive architectures, which are generally less time-efficient and more difficult to train (e.g., gradient exploding/vanishing problems). 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 enabling training – and detection – on longer-in-the-past temporal sequences.

The main properties and contributions of *DC-VAE* can be summarized as follows: (i) **single model for MTS analysis:** *DC-VAE* learns the behavior of the complete MTS process within a single model parametrization, avoiding per-time-series learning and fitting, and further exploiting the richness of the multidimensional process; (ii) **real-time operation:** the model architecture is fully causal, and provides instantaneous predictions for each independent time-series at each new time-step, using a sliding window of past measurements; (iii) **efficient temporal-memory representation:** the VAE encoder/decoder architecture based on dilated convolutions permits to efficiently process temporal sequences of longer length, making detection more robust; (iv) **self-supervised baseline modeling:** by conception, auto-encoders are self-supervised models, because the model trains itself to learn the main features of the input from the very same input samples, and ground-truth labels are only needed for tighter calibration of detection thresholds – nevertheless, in the absence of ground-truth, *DC-VAE* still estimates a normal operation region, indirectly providing a detection threshold; (v) **compact deep-learning architecture:** the structure and number of layers in *DC-VAE*'s architecture is defined by a single parameter T, representing the length of the temporal sliding-window of past measurements used as input; (vi) **independent, per time-series detection:** VAEs provide an estimation of the expected value and its associated standard deviation for each independent time-series, which provides further flexibility and detail to the monitoring process; (vii) **detection results are visually interpretable:** predictions provided by *DC-VAE* define a continual and dynamically adapted normality region, independently for each time-series, making it visually easy to interpret the occurrence of an anomaly.

We apply *DC-VAE* to a MTS dataset arising from the monitoring of an operational mobile ISP, detecting anomalies of very different structural properties. Referred to as the TELCO dataset [17], this *large-scale* – about 750 thousand samples, *long time-span* – seven months' worth of measurements collected at a five-minutes scale, *multi-dimensional* – twelve different metrics (time-series), network monitoring dataset includes ground-truth labels for anomalous events at each individual time-series, manually labeled by the experts of the network operation center (NOC) managing the mobile ISP. We benchmark *DC-VAE* against a broad set of 18 different time-series anomaly detectors coming from the signal processing and machine learning domains, individually testing on each time-series – to keep the scope of the comparative analysis, 15 of these traditional models are combined into a powerful ensemble detector. In addition, we evaluate *DC-VAE* in an open, publicly available dataset commonly used in the literature – the SWaT dataset [18], and compare its performance against other MTS anomaly detectors based on deep learning generative models, which have become very popular in recent years. For the sake of reproducibility and as an additional contribution, we make the TELCO dataset publicly available to the community, and openly release the *DC-VAE*'s code (https://github.com/GastonGarciaGonzalez/DC-VAE).

We note that this work is an extension of a recently published study [19]; the novel contributions of current paper with respect to [19] are as follows: (i) a comprehensive state of the art in the problem of machine learning for time-series anomaly detection; (ii) a more elaborated presentation of the theoretical foundation behind *DC-VAE*; (iii) a more exhaustive performance benchmarking against a much broader set of state-of-the-art detectors, as well as against newer deep-learning-based MTS detectors leveraging Generative Adversarial Networks (GANs) [20]–[22]; (iv) the evaluation of *DC-VAE* in the open SWaT dataset; (v) a deeper analysis of *DC-VAE*'s operation through controlled tests; (vi) the description and release of the TELCO mobile ISP dataset.

The remainder of the paper is organized as follows: Section II presents a comprehensive overview of the related work. In Section III we describe the *DC-VAE* model in detail. Section IV presents the TELCO mobile ISP dataset collected for evaluation, and briefly describes the SWaT dataset. Section V reports the results obtained with *DC-VAE* on the detection of anomalies in TELCO, additionally benchmarking its performance against other approaches in both TELCO and SWaT. Through the testing on synthetic anomalies, Section VI presents a deeper analysis of *DC-VAE*'s response when confronted with different temporal and spatial behaviors such as concept drifts, multidimensional anomalies, and strong outliers. Finally, Section VII concludes the paper.

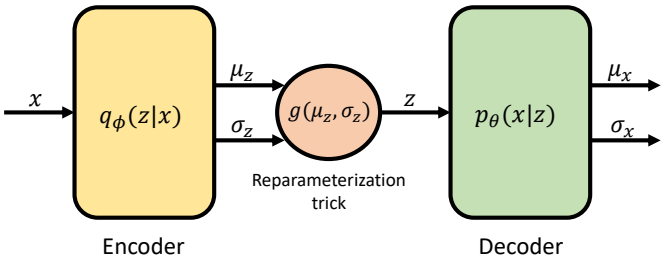


Fig. 1. Variational autoencoder and the re-parameterization trick.

II. Related Work

There are multiple surveys on general-domain anomaly detection techniques [1]–[3] as well as on network anomaly detection [4], [5]. The diversity of data characteristics and types of anomalies results in a lack of universal anomaly detection models. The temporal nature of a very large spectrum of data problems has led to a strong development of the particular field of time-series anomaly detection [1], [23]. It is common to find open libraries implementing the most traditional approaches in the literature – a notable example we use in this study is the python ADTK open library (https://github.com/arundo/adtk/). As noted in [1], most of the methods for unsupervised anomaly detection in univariate and multivariate time-series consist of predicting an expected value based on past information and finding a decision threshold to decide whether the prediction matches the observation. The automatic and adaptive computation of detection thresholds remains an open research problem.

Modern approaches to time-series anomaly detection based on deep learning technology have flourished in recent years [6], [24], [25]. Due to their data-driven nature and achieved performance in multiple domains, generative models such as VAEs [15], and GANs [26] have gained relevance in the anomaly detection field [20], [21], [27]–[31]. VAEs [15], [32], [33] represent a powerful and widely-used class of models to learn complex data distributions. Unlike GANs, a potential limitation of VAEs is the prior assumption that latent sample representations are independent and identically distributed. While this is the most common assumption followed in the literature, there is ongoing research on the benefits of accounting for covariances between samples in time and between time-series to improve model performance [34]–[37]. For example, while the original work [15] assumes that the prior over the parameters and latent variables are centered isotropic Gaussian and the true posteriors are approximately Gaussian with approximately diagonal covariance, [36] proposes an approximation capturing temporal correlations, by considering a Gaussian process prior in the latent space.

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 [38], VRNN [39], Omni-Anomaly [40], and Bi-LSTM [41] among others. Convolutional layers with dilation have also been incorporated into some of these approaches [42]–[44], allowing to speed up the training process based on the possibilities of parallelization offered by these architectures. Transformers [45] is another popular architecture recently showing great performance in sequential data processing; previous work on anomaly detection using transformers and VAEs [46] improves training speed as compared to the state of the art, additionally outperforming standard baseline methods. In particular, the paper improves over [40], considered a reference work in the area. Transformer-based anomaly detection in MTS data is indeed a promising research direction.

Few papers on deep learning-based detectors have addressed the problem of real-time detection. In [47], authors consider the alert delay in detecting so-called *range-anomalies* – i.e., contiguous anomaly segments, and evaluate their models based both on $F1$ scores and on average alert delay. The idea of range-anomaly detection is appealing in practice; in real-world applications, the operator generally does not care about point-wise anomalies, and it is acceptable for an algorithm to trigger an alert for any sample in a contiguous anomaly segment, as far as the detection delay is bounded to a certain max-delay threshold. The work in [48] generalizes the classic measures of Recall, Precision, and F1-score for range-anomalies. We consider these extended performance metrics when evaluating *DC-VAE* in TELCO.

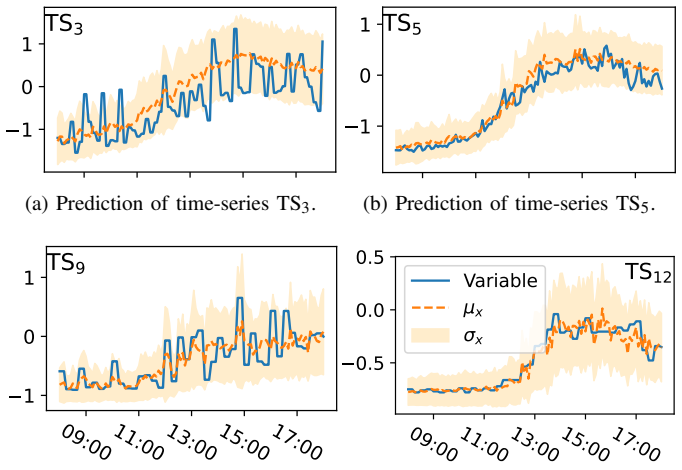
The last topic we overview relates to evaluating and benchmarking model performance through in-the-wild data time-series, using expert domain knowledge for data labeling. Most proposals in the literature have been analyzed on public datasets, such as the well-known Yahoo [10], Numenta [11], [12], NASA [13], or others, where operating conditions are unrealistic, anomalies might be trivial, and labels are poorly assigned in the labeling process [14]. Getting access to datasets labeled by domain experts in an operational environment is irreplaceable for the realistic evaluation of algorithms.

This work has its origins in our previous paper on generative models for network anomaly detection in MTS data [21], where we conceived Net-GAN, an architecture based on GANs and RNNs, where Long Short-Term Memory networks (LSTMs) were employed as both generator and discriminator models to capture temporal dependencies in the data.

III. Anomaly Detection with *DC-VAE*

Sequential data such as time-series is generally processed through sliding windows, condensing the information of the most recent T measurements. Let us define $\boldsymbol{x}$ as a matrix in $\mathbb{R}^{M\times T}$, where M is the number of variables in the MTS process, i.e., defines the dimension of the problem. We also define $x(t) \in \mathbb{R}^{M\times 1}$ as an M-dimensional vector, representing the MTS at a certain time t, and $x_m(t)$, with $m \in \{1, \ldots, M\}$, as the value of the m-th time-series at time t.

As depicted in Figure 1, 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 parametrization 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



(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}.

Fig. 2. Example of time-series analysis through *DC-VAE*, for the TELCO dataset. The normal-operation region is defined by $\boldsymbol{\mu_x}$ and $\boldsymbol{\sigma_x}$.

$\boldsymbol{\mu}_x$ 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 the region determined by the predicted mean and standard deviation. For reference, Figure 2 presents the main ideas behind the usage of VAEs for time-series anomaly detection, in this case portraying the results obtained in the analysis of the TELCO dataset, which is fully described in Section IV. For each of the displayed time-series TS_i – the TELCO dataset corresponds to twelve time-series TS_1 to TS_{12}, its real value x_i, along with the outputs of the VAE μ_{x_i} and σ_{x_i}, are reported.

In the VAE model, observations $\boldsymbol{x}$ are assumed to depend on a random variable $\boldsymbol{z}$ that comes from a lower-dimensional *latent* space. The objective is to maximize $P(\boldsymbol{x})$, the probability of the observations through the model. Similar to $\boldsymbol{x}$, $\boldsymbol{z}$ will also be a sequence of length T, but with a smaller number of dimensions $J < M$, $\boldsymbol{z} \in \mathbb{R}^{J \times T}$. In formal terms, given an input sample $\boldsymbol{x}$ characterized by an unknown probability distribution $P(\boldsymbol{x})$, the objective is to model or approximate the data's true distribution P using a parametrized distribution p_θ with parameters θ. Let $\boldsymbol{z}$ be a random vector jointly-distributed with $\boldsymbol{x}$, representing the latent encoding of $\boldsymbol{x}$. We can express $p_\theta(\boldsymbol{x})$ as:

$$p_\theta(\boldsymbol{x}) = \int_z p_\theta(\boldsymbol{x}, \boldsymbol{z})\, d\boldsymbol{z}, \tag{1}$$

where $p_\theta(\boldsymbol{x}, \boldsymbol{z})$ represents the joint distribution under p_θ of the observable data $\boldsymbol{x}$ and its latent representation or encoding $\boldsymbol{z}$. According to the chain rule, the equation can be rewritten as:

$$p_\theta(\boldsymbol{x}) = \int_z p_\theta(\boldsymbol{x}|\boldsymbol{z}) p_\theta(\boldsymbol{z})\, d\boldsymbol{z} \tag{2}$$

In the vanilla VAE, $p_\theta(\boldsymbol{x}|\boldsymbol{z})$ is considered a Gaussian distribution, and therefore, $p_\theta(\boldsymbol{x})$ is a mixture of Gaussian distributions. The computation of $p_\theta(\boldsymbol{x})$ is very expensive and, in most cases, even intractable. To speed up training and make it feasible, it is necessary to introduce a further function to approximate the posterior distribution $p_\theta(\boldsymbol{z}|\boldsymbol{x})$, in the form of $q_\phi(\boldsymbol{z}|\boldsymbol{x}) \approx p_\theta(\boldsymbol{z}|\boldsymbol{x})$. In this way, the overall problem can be easily translated into the autoencoder domain, in which the conditional likelihood distribution $p_\theta(\boldsymbol{x}|\boldsymbol{z})$ is performed by the *probabilistic* decoder. In contrast, the approximated posterior distribution $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ is computed by the *probabilistic* encoder, cf. Figure 1.

As in every deep-learning problem, it is necessary to define a differentiable loss function to update the network weights through backpropagation. In VAEs, the idea is to jointly optimize the generative model parameters θ to reduce the reconstruction error between the input and the output of the network and the parameters ϕ of the approximated posterior distribution to have $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ as close as possible to the real posterior $p_\theta(\boldsymbol{z}|\boldsymbol{x})$. The Evidence Lower Bound Loss (ELBO) loss function is generally considered for this task. In the case of VAEs, the ELBO loss function $L_{\theta,\phi}$ can be written as follows:

$$\begin{aligned} L_{\theta,\phi} &= -\log(p_\theta(\boldsymbol{x})) + D_{KL}\left(q_\phi(\boldsymbol{z}|\boldsymbol{x}) \parallel p_\theta(\boldsymbol{z}|\boldsymbol{x})\right) \\ &= -\mathbb{E}_{\boldsymbol{z}\sim q_\phi(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})\right] + D_{KL}\left(q_\phi(\boldsymbol{z}|\boldsymbol{x}) \parallel p_\theta(\boldsymbol{z})\right) \end{aligned} \tag{3}$$

where D_{KL} is the Kullback-Leibler divergence, which here basically measures the information loss when using q to approximate p. To train the autoencoder and make the application of backpropagation feasible, a so-called *reparameterization trick* is generally introduced. The main assumption on the latent space is that it can be considered as a set of multivariate Gaussian distributions, and therefore, $\boldsymbol{z} \sim q_\phi(\boldsymbol{z}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{\mu_z}, \boldsymbol{\sigma_z}^2)$. Given a random matrix $\boldsymbol{\varepsilon} \sim \mathcal{N}(0, \boldsymbol{I})$ and $\odot$ defined as the element-wise product, the reparameterization trick permits to explicitly define $\boldsymbol{z} = g(\boldsymbol{\mu_z}, \boldsymbol{\sigma_z}) = \boldsymbol{\mu_z} + \boldsymbol{\sigma_z} \odot \boldsymbol{\varepsilon}$. Thanks to this transformation, the variational autoencoder is trainable. The probabilistic encoder has to learn how to map a compressed representation of the input into the two latent vectors $\boldsymbol{\mu_z}$ and $\boldsymbol{\sigma_z}$. At the same time, the stochasticity remains excluded from the updating process and is injected in the latent space as an external input through $\boldsymbol{\varepsilon}$. Under the Gaussian assumption, the ELBO loss function $L_{\theta,\phi}$ can be explicitly re-written as:

$$\begin{aligned} L_{\theta,\phi} = \frac{1}{2 \times T \times N} \sum_{n=1}^{N} \sum_{t=1}^{T} \Bigg[& \\ & \sum_{m=1}^{M} \left(\frac{\left(x_m(t)^{(n)} - \mu_{x_m}(t)^{(n)}\right)^2}{(\sigma_{x_m}(t)^{(n)})^2} + \log(\sigma_{x_m}(t)^{(n)})^2 \right) \\ & - \sum_{j=1}^{J} \left(1 + \log(\sigma_{z_j}(t)^{(n)})^2 - (\mu_{z_j}(t)^{(n)})^2 - (\sigma_{z_j}(t)^{(n)})^2 \right) \Bigg] \end{aligned} \tag{4}$$

At each iteration, the loss is calculated for a batch of size N; recall that m indicates the variable (time-series) in the space of $\boldsymbol{x}$, and j the variable in the space of $\boldsymbol{z}$, whereas t represents the specific time instant.

To exploit the temporal dimension of the input time-series, we proposed an encoder/decoder architecture based on popular CNNs, using Dilated Convolutions (DCs) [16]. DC is a

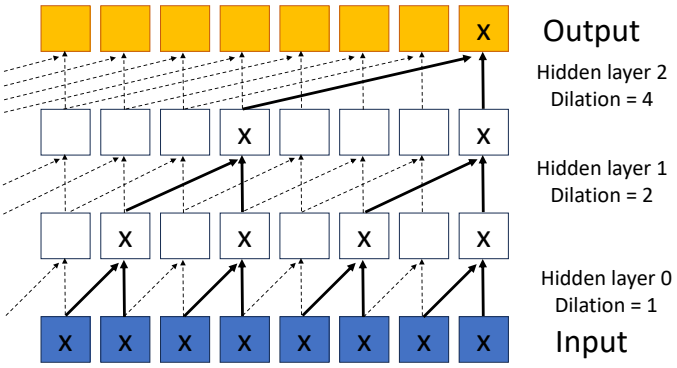


Fig. 3. Using CNNs with causal filters requires large filters or many layers to learn from long sequences. Dilated convolutions improve time-series modeling by increasing the receptive field of the neural network, reducing computational and memory requirements, enabling training on long sequences.

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 3 explains the basic idea behind DCs. The convolutions must be causal, so that detection can be implemented in real-time. Because such architectures do not have recurrent connections, they are often much faster to train than RNNs and do not suffer from complex-to-tame gradient exploding/vanishing problems. Using DCs instead of standard convolutions has several advantages for real-time analysis: (i) they increase the so-called receptive field, meaning that longer-in-the-past information can be fed into the detection; (ii) DCs are computationally more efficient, as they provide larger coverage at the same computation cost; (iii) by using DC, the pooling steps are omitted, thus resulting in lesser memory consumption; (iv) finally, for the same temporary receptive field, the resulting network architecture is much more compact.

Figure 4 depicts the encoder 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.

Note that the dilation process allows doubling T with each added layer. Consequently, a large temporal receptive field of past measurements can be achieved without further deepening the network. The encoder and decoder are symmetric in architecture, both in the number of filters and applied dilations. In the encoder model, the idea is to reduce or maintain layer output dimensions with network depth. The opposite for the decoder is increasing or maintaining the dimension

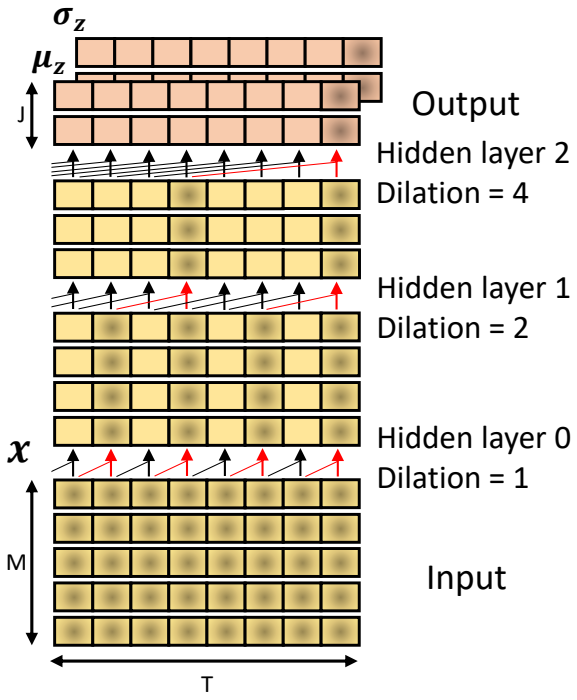


Fig. 4. Encoder architecture using causal dilated convolutions, implemented through a stack of 1D convolutional layers.

until reaching the observations' dimension. In both cases, the sequence length T is always maintained.

Model training is conducted 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. At each time t, the *DC-VAE* model takes as input the matrix $\boldsymbol{x} \in \mathbb{R}^{M \times T}$, constructed out of the last T samples observed in the MTS, and produces as output matrices $\boldsymbol{\mu_x}$ and $\boldsymbol{\sigma_x}$ – for notation brevity, we define $\boldsymbol{\mu} = \boldsymbol{\mu_x}$ and $\boldsymbol{\sigma} = \boldsymbol{\sigma_x}$. From these two output matrices, the anomaly detection only considers their values at time t, corresponding to two vectors $\mu(t)$ and $\sigma(t)$. For each of the univariate time-series m, an anomaly is detected at time t if its value $x_m(t)$ falls outside the normal-operation region, defined by $\mu_m(t)$ and $\sigma_m(t)$. More precisely, an anomaly in time-series m is declared at time t if:

$$|x_m(t) - \mu_m(t)| > \alpha_m \times \sigma_m(t), \tag{5}$$

where $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_m, \ldots, \alpha_M)$ is a vector of M detection sensitivity threshold, where each α_m can be set independently for each time-series, allowing for fine-grained, per time-series calibration of the detection process.

Regarding the calibration of $\boldsymbol{\alpha}$, and despite being *DC-VAE* an unsupervised system, we acknowledge that these thresholds are set relying on annotated anomalies. Inevitably in any anomaly detection problem, it is necessary to set an operating point. This must be set by an expert operator in the system, who knows the behavior of the data and the cost of false detections, both positive and negative. In all sets for anomaly detection, this knowledge is in the labels provided by the experts. There are different techniques to

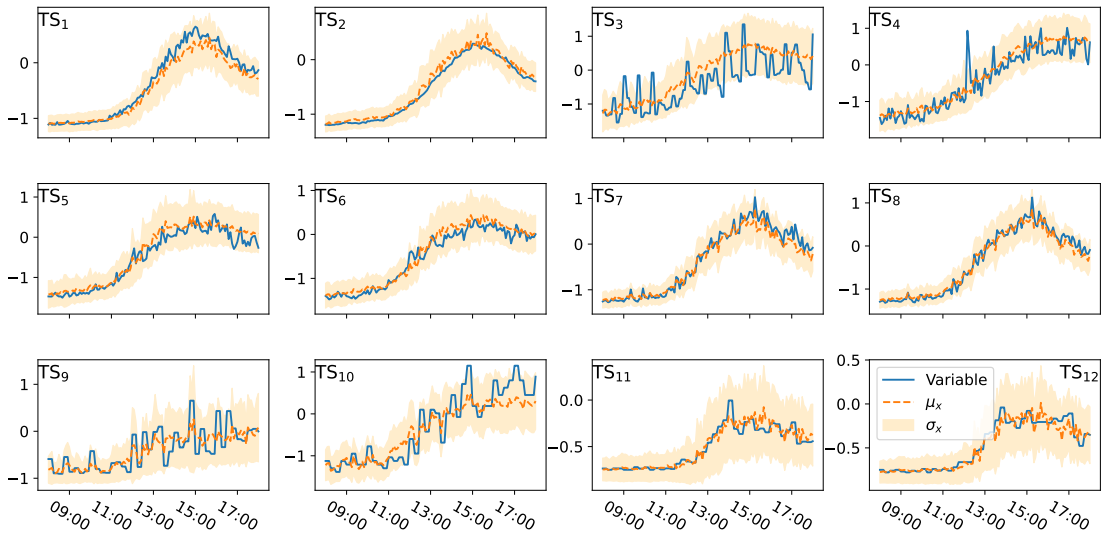


Fig. 5. Snapshots of the TELCO MTS. For each time-series, the region of normal operation is depicted, as estimated from *DC-VAE* predictions $\boldsymbol{\mu}_x$ and $\boldsymbol{\sigma}_x$.

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

dataset	# samples	duration	# anomalous samples
training	310,974	3 months	5,672 (**1.8%**)
validation	103,680	1 month	385 (**0.4%**)
testing	317,953	3 months	3,080 (**1.0%**)
total	732,607	7 months	9,137 (**1.2%**)

define thresholds automatically from the data [49], but all are applicable for the detection of outliers (i.e., values far from normal behavior). In the problem we are dealing with, the interest is to detect anomalies which are often difficult to differentiate from normal behavior, so the calibration stage inevitably must be supervised.

IV. THE TELCO MOBILE ISP DATASET

A. *TELCO – A New Open Dataset Released to the Community*

A recent study [14] alerts on the limitations of evaluating anomaly detection algorithms on popular time-series datasets such as Yahoo, Numenta, or NASA, among others. In particular, these datasets are noted to suffer from known flaws such as trivial anomalies, unrealistic anomaly density, mislabeled ground truth, and run-to-failure bias. For this reason, we decided to evaluate *DC-VAE* in a proprietary MTS dataset, corresponding to real measurements collected at an operation mobile ISP – note that we are publicly releasing this dataset to the community (https://iie.fing.edu.uy/investigacion/grupos/anomalias/en/telco-dataset-2/downloads/). The TELCO dataset [17] corresponds to twelve different time-series, 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. This temporal length is seldom available in other publicly available datasets of this nature and is highly relevant and useful to allow for long-term seasonal behavior analysis.

Each time-series corresponds to aggregated data from different sources; to keep business confidentiality, we do not specify the exact data type reflected by each time-series. The twelve time-series are typical data monitored in a mobile ISP, including the number and amount of prepaid data transfer fees, number and cost of calls, the volume of data traffic, number of SMS, and more.

Figure 5 depicts daily snapshots of the complete TELCO MTS. For each time-series, the region of normal operation is depicted, as estimated from *DC-VAE* predictions $\boldsymbol{\mu}_x$ and $\boldsymbol{\sigma}_x$. Different time-series expose different behaviors, e.g., some of them are noisier (TS_3), others have lower dynamic ranges (TS_{11}), and some others show a smoother evolution (TS_2). To appreciate the strong seasonality component of the time-series, Figure 6 depicts the TELCO MTS for a period of four days, covering weekdays and weekends.

Table I presents the main details of the dataset. Note in particular, how strongly imbalanced the dataset is in terms of normal-operation and anomalous samples, which is the typical case for real network 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 purposes. For the sake of completeness, Table II reports normal-operation and anomalous samples per individual time-series, for the training, validation, and testing sub-sets. The share of anomaly samples is low and significantly different for some of the time-series, adding richness and complexity to the dataset; for example, time series TS_1, TS_4, TS_9, and

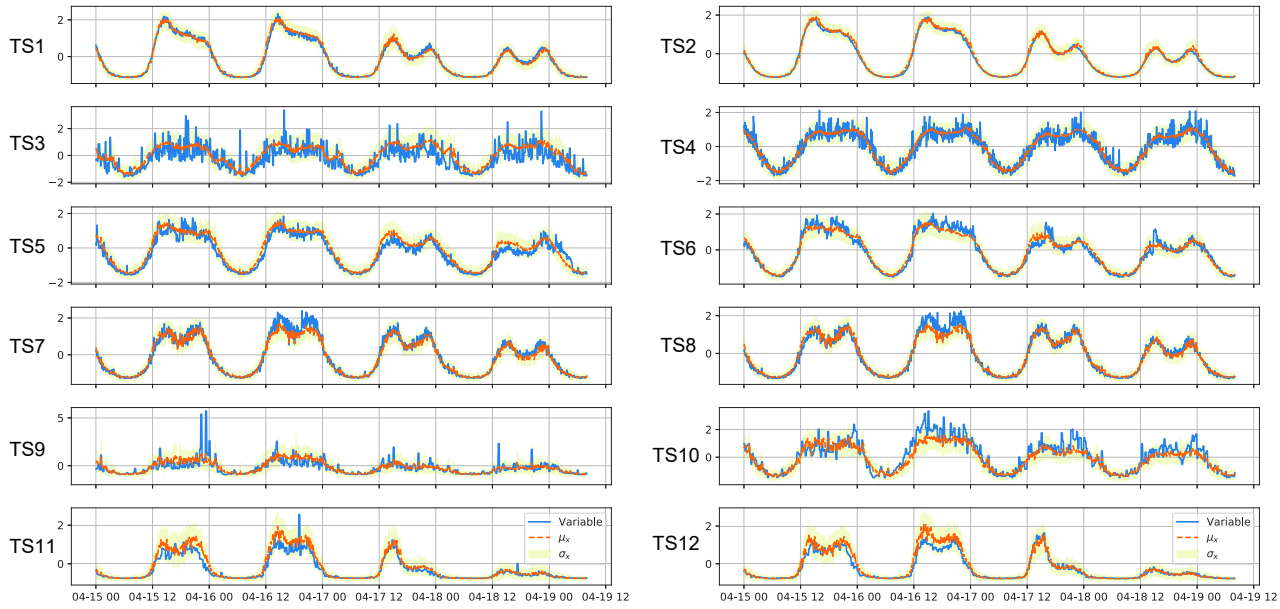


Fig. 6. TELCO dataset time-series, for four days, along with the corresponding *DC-VAE* estimations. The temporary receptive field – i.e., length of the rolling time-window, is $T = 512$ samples, spanning about two days of past measurements.

TABLE II
DISTRIBUTION OF ANOMALY SAMPLES IN THE TELCO DATASET, PER TIME-SERIES AND PER TRAINING, VALIDATION, AND TESTING SUB-SETS. THE SHARE OF ANOMALY SAMPLES IS LOW, AND SIGNIFICANTLY DIFFERENT FOR SOME OF THE TIME-SERIES.

	training			**validation**			**testing**			**total**		
TS ID	normal	anomalous	%	normal	anomalous	%	normal	anomalous	%	normal	anomalous	%
TS_1	24,731	1,183	4,6%	8,628	12	0,14%	26,084	412	1,6%	59443	1607	**2,6%**
TS_2	25,713	201	0,8%	8,629	11	0,13%	25,995	501	1,9%	60,337	713	**1,2%**
TS_3	25,784	130	0,5%	8,636	4	0,05%	26,358	138	0,5%	60,778	272	**0,4%**
TS_4	24,464	1,450	5,6%	8,636	4	0,05%	26,317	179	0,7%	59,417	1,633	**2,7%**
TS_5	25,840	74	0,3%	8,637	3	0,03%	26,390	106	0,4%	60,867	183	**0,3%**
TS_6	25,850	64	0,2%	8,639	1	0,01%	26,390	107	0,4%	60,879	172	**0,3%**
TS_7	25,793	127	0,5%	8,638	2	0,02%	26,227	269	1,0%	60658	398	**0,7%**
TS_8	25,787	127	0,5%	8,640	0	–	26,229	267	1,0%	60,656	394	**0,6%**
TS_9	25,287	627	2,4%	8,508	132	1,53%	25,932	564	2,1%	59,727	1,323	**2,2%**
TS_{10}	24,558	1,356	5,2%	8,463	177	2,05%	25,995	501	1,9%	59016	2,034	**3,3%**
TS_{11}	25,725	189	0,7%	8,601	39	0,45%	26,475	21	0,1%	60,801	249	**0,4%**
TS_{12}	25,770	144	0,6%	8,640	0	–	26,481	15	0,1%	60,891	159	**0,3%**

TS_{10} have a total share of anomaly samples above 2% or 3%.

While the TELCO dataset used in this paper and released to the community has a seven-month time span, we acknowledge that the complete dataset we have collected has almost two years of duration. We have decided to work only on these seven months because it corresponds to the the data for which expert operator annotated labels are available. Although *DC-VAE* trains in a self-supervised fashion, a fair comparison with supervised methods as the one we do in the evaluations requires that all methods share the same training, validation, and test sets.

Nevertheless, and for the sake of completeness, we investigated the impact on *DC-VAE*'s baseline modeling performance when training with longer time-spans, without labels. Figure 7 reports the average log-likelihood $\mathbb{E}_{\boldsymbol{z}\sim q_\phi(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})\right]$ in the reconstruction of TELCO in the testing dataset, using different temporal spans for self-supervised model training. Interestingly, improvements are rather marginal when considering up to 18 months of training data, suggesting that manually labeling a longer time-span for TELCO might not actually provide a richer dataset. In any case, we are working with the expert annotators to release a newer version of TELCO in the near future, covering more than one year of labeled time-series.

B. The SWaT Open Dataset for Cybersecurity Analysis

While the core of the evaluations and benchmarking is conducted on the TELCO dataset, we also evaluate *DC-VAE* in the Secure Water Treatment (SWaT) dataset [18], an open, publicly available dataset commonly used in the literature for cybersecurity analysis. The SWaT dataset consists of 51 time-

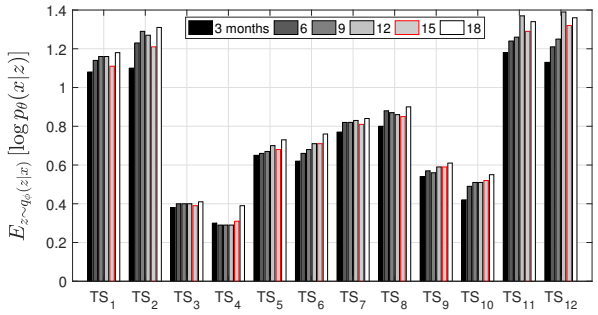


Fig. 7. Average log-likelihood $\mathbb{E}_{\boldsymbol{z}\sim q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\right]$ in the reconstruction of TELCO in the testing dataset, using different temporal spans (3 to 18 months) for self-supervised model training.

series of data collected over eleven days in 2015-2016, on a water treatment operational test-bed, which represents a small-scale version of a large modern cyber-physical system. The dataset contains two sub-sets temporally split; the first week is anomaly free and is considered as the training dataset, whereas the last four days of data contain 36 attacks of different nature and duration (from a few minutes to an hour), and is meant for testing purposes. The total number of anomaly samples accounts for about 5.8% of the total measurements. As an example of the kind of patterns observed in the SWaT MTS, Figure 8 depicts four of the time-series under normal operation. Different from TELCO, which represents a real operational network and anomaly labels are provided by manual inspection on individual time-series, anomaly labels in SWaT correspond to temporal ranges in which the attacks were executed under a controlled environment.

We acknowledge that the SWaT dataset is far from representing a real cyber-physical system and is not perfect as benchmark for anomaly detection, presenting significant trivial anomalies and unrealistic anomaly density, as well as some mislabeled ground truth and missed anomalies in the data (https://mlad.kaspersky.com/swat-testbed/). Nevertheless, there are two main reasons for testing *DC-VAE* in SWaT: (i) firstly, despite its deficiencies, the SWaT dataset is widely used in the state of the art as benchmark for multivariate time-series anomaly detection, and this allows us showing that *DC-VAE* provides similar, or even better performance, than other similar systems in a well-known dataset; (ii) secondly, using SWaT lets us testing the modeling capabilities of *DC-VAE* in a dataset with a broader variety of variables – 51 time series in this case.

V. *DC-VAE* Evaluation and Benchmarking

A. DC-VAE Architecture Calibration

The first step before evaluation of *DC-VAE* is to calibrate the model. As explained in Section III, the length T of the sliding window plays a major role in the architecture of *DC-VAE*. Given the usage of the dilated convolutions, T determines the number of encoder and decoder layers (cf. Figure 4). The dimension J of the latent space is the other relevant parameter to set; while it must be smaller than the MTS dimension M, it must also be big enough to capture the most relevant information of the MTS process. We test different values for the sequence length T to show how this affects the performance of the model. In particular, we test $T = 1, 8, 16, 32, 64, 128, 256, 512, 1024$ samples, considering the average of the mean absolute error (MAE) between x_m and μ_{x_m}, for each time-series m. Sequence length $T = 1$ corresponds to a standard VAE model with only snapshot-like inputs; to avoid an excessively compressed model for this sequence length, we consider here an architecture with three fully-connected layers.

TABLE III
Grid of hyperparameters used in the model calibration.

Hyperparameter	Grid	Best
T	$\{8, 16, 32, 64, 128, 256, 512, 1024\}$	512
J	$\{1, 2, 4, 8\}$	4
γ	$\{10^{-3}, 10^{-4}\}$	10^{-3}
m	$\{32, 64\}$	32
f	$\{8, 16, 32\}$	16

Besides the reconstruction MAE, we also compute the so-called explained variance or variance score $\text{Var}_{\text{score}}$, which compares the variance of the reconstruction error and the variance of the input signal:

$$\text{Var}_{\text{score}}\left(\boldsymbol{x}(t), \boldsymbol{\mu}_{\boldsymbol{x}}(t)\right) \quad = \quad 1 - \frac{\text{Var}\left(\boldsymbol{x}(t) - \boldsymbol{\mu}_{\boldsymbol{x}}(t)\right)}{\text{Var}(\boldsymbol{x}(t))} \qquad (6)$$

The value of $\text{Var}_{\text{score}}$ is between $[0, 1]$, where 1 represents the ideal case. Figure 9 reports the (a) MAE and (b) $\text{Var}_{\text{score}}$ for each sequence length T and corresponding model architecture, in both cases obtained as the average value across all the time-series, for the TELCO validation set. Latent space dimensions $J = 4$, and $J = 8$ are considered in the analysis. The MAE varies considerably for the proposed range, with $T = 512$ providing the smallest reconstruction error, almost identical for both latent space dimensions. Similarly, for the $\text{Var}_{\text{score}}$, $T = 512$ results in the highest score, for both latent space dimensions.

Another relevant hyperparameter is the number of filters f for each hidden convolutional layer, which together with the number of layers and the input and output dimensions define the size of the architecture in terms of the number of trainable parameters. Also, hyperparameters typical of the training stage, such as the learning rate γ and the mini-batch size m, are key to find the optimal solution. To find the best combination of these hyperparameters, we use the Tree-structured Parzen Estimator (TPE) approach [50]. In total, 50 attempts were tested on the grid shown in table III, where the hyperparameters for which the model showed the smallest validation loss are reported in the last column.

The hyperparameter search stage for a deep learning model is one of the most important and most expensive steps, since it involves training many models until the optimal values are found. Therefore, lowering the times of this stage is paramount. To evaluate the time gained by using a fully parallelizable compact architecture such as the one proposed in *DC-VAE*, as compared to traditional recurrent architectures, we created another architecture by replacing all layers with RNNs.

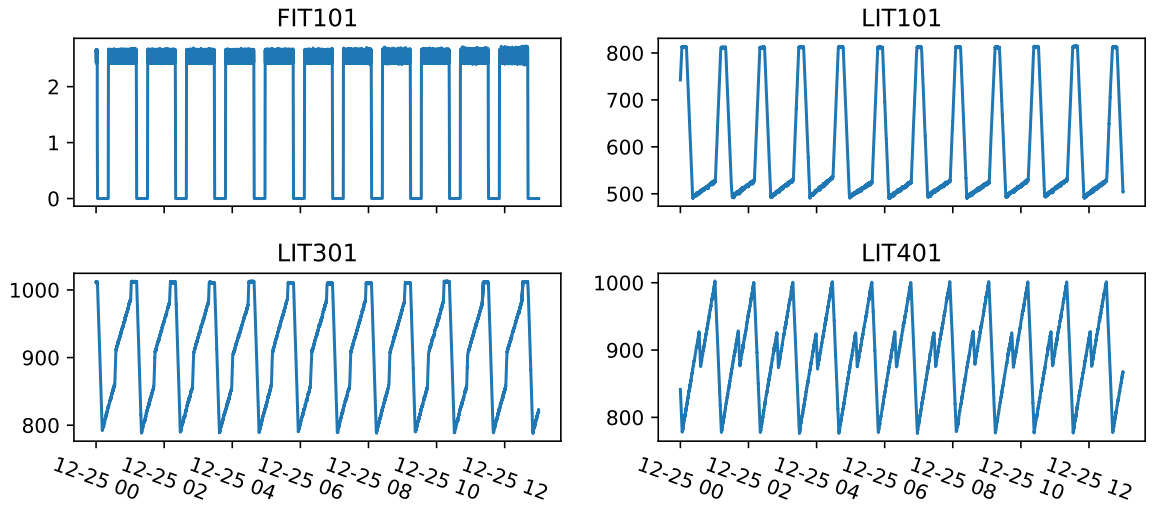
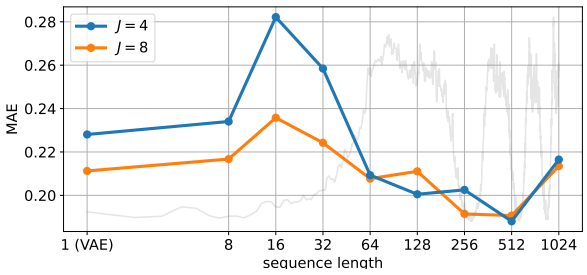
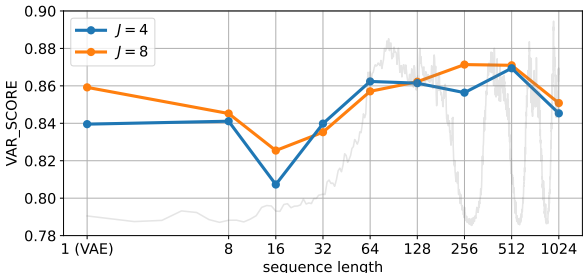


Fig. 8. SWaT – the four time-series represent normal operation. Anomaly labels in SWaT correspond to 36 temporal ranges when attacks were executed.



(a) Reconstruction error.



(b) Variance score.

Fig. 9. Calibration of *DC-VAE* in TELCO. $T = 512$ provides the smallest reconstruction error and the highest variance score.

To search for the hyperparameters, we define another grid that includes the previous one, adding the number of hidden layers: $h = \{2, 4\}$. It is worth remembering that for *DC-VAE*, defining the length of the T sequences automatically sets the number of layers, and thus, this value varies between $[3, 10]$. Gated Recurrent Units (GRU) were the type of layer used in the RNNs, as they showed the highest convergence stability in terms of vanilla RNN and LSTM models.

Table IV reports the comparative times taken for hyperparameter search and model training for both architectures, i.e., *DC-VAE* and the RNN-based one. Tests are performed on standard GPU hardware, using a Nvidia GTX 1060 GPU. The fully causal architecture proposed by *DC-VAE* is more compact and can be optimized and trained much faster than traditional, recursive architectures. In particular, hyperparameter search takes less than half the time, and model training is at least 33% faster.

TABLE IV
TEMPORAL COMPLEXITY FOR ARCHITECTURE OPTIMIZATION AND MODEL TRAINING (HARDWARE REFERENCE: GPU NVIDIA GTX 1060).

	DC-VAE	**RNN**
Hyperparameter search (hours)	**15**	37
Training best model (minutes)	**10**	15

B. Anomaly Detection Results in TELCO

We go back to Figure 6 to show *DC-VAE* in action, using a sliding-window of length $T = 512$ samples. *DC-VAE* can properly track different types of behavior in the time-series, including the strong seasonal daily component, but also the operation during weekdays and weekends, clearly visible in TS_2 and TS_{11}, among others. 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 α_m to address the underlying nature of each monitored metric. Note in addition how different periods of time-series variability result in more or less tight normal-operation regions estimated by *DC-VAE*, as defined by $\sigma(t)$. Figure 10 extends the predictions of *DC-VAE* to a longer time-span, considering two weeks of measurements, for time-series TS_2 and TS_{11}. While both time-series have a strong seasonal component, with marked differences in behavior on weekdays and weekends, TS_{11} has a decreasing trend on the second week, which can be properly tracked by *DC-VAE*.

To apply *DC-VAE* for anomaly detection, we have to calibrate the sensitivity thresholds α, which is usually done in a supervised manner, relying on the labeled anomalies available in the training and validation datasets. This step is the only one that requires a certain level of "supervision" (in the sense of ground-truth availability), but could also be done

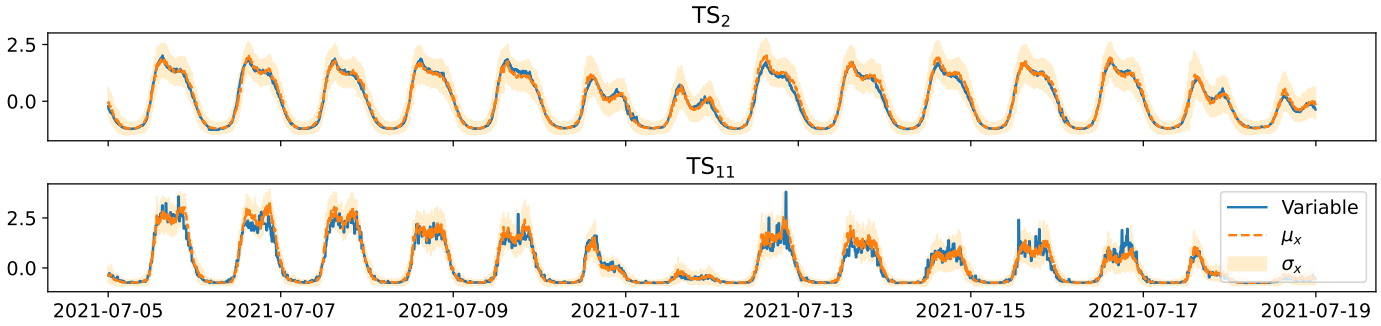
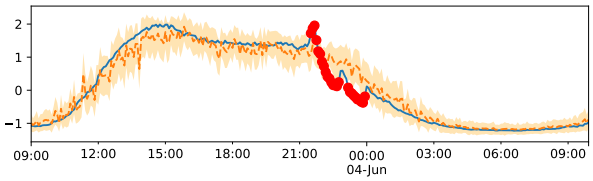
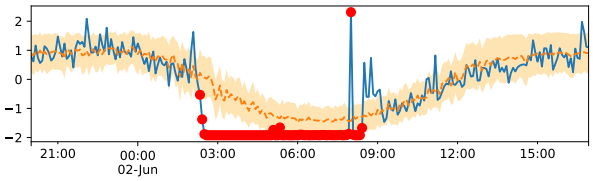


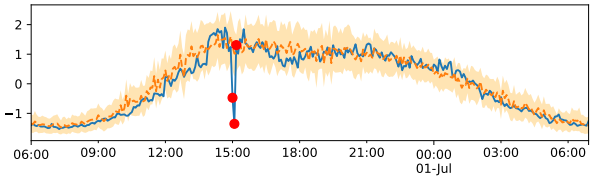
Fig. 10. *DC-VAE* operation for time-series with stationary behavior. Weekly seasonality is identified, with variations between weekdays and weekends.



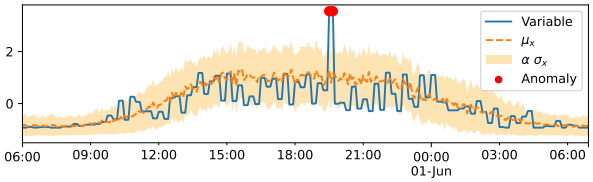
(a) Example of real anomalies in TS_2.



(b) Example of real anomalies in TS_4.



(c) Example of real anomalies in TS_6.



(d) Example of real anomalies in TS_9.

Fig. 11. Examples of real anomalies present in the analyzed dataset, and their identification by *DC-VAE*.

TABLE V
SET OF BENCHMARK TIME-SERIES ANOMALY DETECTORS USED IN TELCO AGAINST *DC-VAE*.

ENS-15	Local Outlier Factor (LOF)
	Isolation Forest (IF)
	Double Roll. Aggregate with Interquartile Range (DRA-IR)
	Quantile Detector (QQ)
	Interquartile Range Detector (IR)
	Generalized Extreme Studentized Deviate Test (G-ESDT)
	DRA with Single Change-Point Detection (DRA-CP)
	Level Shift Detector (LS)
	Volatility Shift Detector (VS)
	Seasonal Decomposition with Exp. Smoothing (SD-ETS)
	Time-Series Seasonality Detector (TSS)
	Autoregressive Detector (AR)
	Linear Regression Detector (LR)
	PCA Detector (PCA)
	K-means Clustering Detector (K-means)
S-EXPS	Seasonal Exponential Smoothing
ARIMA	Auto Regressive Integrated Moving Average
S-VAE	Standard vanilla VAE, equivalent to *DC-VAE* with $T = 1$

in a self-supervised manner, by labeling anomalies through outlier detection techniques. In our specific problem, each sensitivity threshold α_m is calibrated on a per time-series basis, by maximizing the $F1$ score over the training and validation datasets, doing a grid-search of integer values from 1 to 5. In a nutshell, we decide how many standard deviations σ_m shall be considered as tolerance for the normal-operation variability of the data.

Figure 11 reports some examples of real (i.e., labeled) anomalies present in the TELCO dataset, in particular for time-series TS_2, TS_4 TS_6 and TS_9, along with their corresponding identification by *DC-VAE*, where sensitivity thresholds $\boldsymbol{\alpha}$ were calibrated as mentioned before. *DC-VAE* can detect different types of anomalies present in the data, of a more transient and spiky nature in the case of TS_6 and TS_9, or on a more structural basis in the case of TS_2 and TS_4. Note also how some of the actual measurements fall significantly outside the normal-operation region – e.g. in Figure 11(c), but still these were not labeled as anomalous by the expert operator. Whether this is a false-positive produced by *DC-VAE*, or a non-labeled anomaly missed by the expert operator is difficult to know. It is important to note that anomalies in real, operational measurements, as labeled by the expert operator, do not always translate into clear outliers in the data; the contrary is also true, meaning that typical outliers in the data might not correspond to actual anomalies in the eyes of the expert operator. Manual data labeling by experts is prone to human error, many times due to a lack of conclusive information for the operator to take a proper decision. These observations are paramount when evaluating anomaly detectors with real, in-the-wild data.

We run a quantitative performance analysis of *DC-VAE* in the testing dataset (cf. Table I), benchmarking its performance against a broad set of more traditional detectors. As performance metrics, we consider an elaborated version of the traditionally used, per-sample evaluation metrics, to consider a

TABLE VI
ANOMALY DETECTION PERFORMANCE BENCHMARKING IN TELCO, COMPARING *DC-VAE* AGAINST S-EXPS, ARIMA, S-VAE, AND AN ENSEMBLE OF 15 TRADITIONAL DETECTORS (ENS-15). FIRST AND SECOND HIGHEST $F1$ SCORES ARE MARKED IN RED AND BLUE, RESPECTIVELY.

	ENS-15			**S-EXPS**			**ARIMA**			**S-VAE**			***DC-VAE***		
TS ID	R_r	P_r	$F1_r$	R_r	P_r	$F1_r$	R_r	P_r	$F1_r$	R_r	P_r	$F1_r$	R_r	P_r	$F1_r$
TS_1	45%	50%	48%	45%	88%	60%	64%	92%	**75%**	23%	56%	32%	58%	71%	**64%**
TS_2	37%	100%	54%	70%	96%	**81%**	59%	95%	**73%**	16%	92%	27%	74%	20%	**67%**
TS_3	78%	33%	47%	78%	58%	**67%**	78%	46%	58%	71%	50%	59%	86%	47%	**60%**
TS_4	75%	59%	**66%**	67%	41%	**51%**	58%	38%	46%	63%	25%	36%	63%	21%	**32%**
TS_5	73%	73%	**73%**	45%	63%	53%	64%	64%	**64%**	50%	20%	29%	75%	50%	60%
TS_6	88%	62%	**72%**	63%	63%	63%	75%	50%	60%	14%	100%	25%	57%	83%	68%
TS_7	77%	63%	**69%**	69%	53%	60%	69%	46%	56%	45%	100%	63%	72%	90%	**80%**
TS_8	67%	44%	53%	56%	36%	43%	56%	56%	**56%**	57%	35%	43%	44%	80%	**57%**
TS_9	10%	17%	**12%**	5%	5%	5%	19%	9%	12%	6%	4%	4%	17%	11%	**13%**
TS_{10}	8%	18%	11%	48%	44%	46%	48%	38%	42%	39%	81%	**52%**	52%	59%	**55%**
TS_{11}	58%	21%	31%	50%	32%	**39%**	67%	26%	37%	67%	17%	27%	100%	25%	**40%**
TS_{12}	0%	0%	0%	100%	67%	**80%**	100%	24%	**38%**	0%	0%	0%	100%	11%	22%
mean	51%	45%	45%	58%	54%	**54%**	63%	49%	51%	38%	48%	33%	67%	47%	**52%**
median	63%	47%	51%	60%	55%	**57%**	64%	46%	56%	42%	43%	31%	68%	49%	**59%**

TABLE VII
ANOMALY DETECTION PERFORMANCE BENCHMARKING AGAINST DEEP-LEARNING GENERATIVE MODELS IN SWAT.

Detector	R	P	$F1$
Auto Encoder	53%	73%	61%
EGAN	68%	41%	51%
NET-GAN-(G)enerator	65%	98%	**78%**
NET-GAN-(D)iscriminator	65%	29%	40%
MAD-GAN-P (best precision)	55%	100%	70%
MAD-GAN-R (best recall)	100%	12%	22%
MAD-GAN-F1 (best F1 score)	64%	99%	**77%**%
DC-VAE	67%	94%	**78%**

more natural and practical approach for real anomaly detection applications, evaluating detection performance in the form of anomaly temporal-ranges. Traditional metrics can make sense for point anomalies where a true positive corresponds to a correct detection at the precise point in time. However, as shown for example in Figure 11(b), many anomalies occur in the form of multiple, consecutive point anomalies, defining an anomaly range. In such scenarios, it could be already enough to have a partial overlap between the real anomaly range and the predicted anomaly interval to consider a correct detection. Previous papers have considered these observations [11], [47], [48], defining new metrics which prioritize early or delayed detection, or focusing mainly on range anomalies. Therefore, we take the extended definitions of recall and precision as defined in [48] to generalize for ranges of anomalies, considering a correct detection if at least one of the samples between the start and the end of the actual anomaly is flagged by the model. We refer to these extended, range-based metrics as R_r, P_r, and $F1_r$, for recall, precision, and F1-score, respectively. More precisely, given a set of λ Real Anomaly ranges $RA = RA_1 \dots RA_\lambda$ and a set of δ Predicted Anomaly ranges $PA = PA_1 \dots PA_\delta$:

$$R_r(RA, PA) = \frac{\sum_{j=1}^{\lambda} R_r(RA_i, PA)}{\lambda} \quad (7)$$

$$P_r(RA, PA) = \frac{\sum_{j=1}^{\delta} P_r(RA, PA_i)}{\delta} \quad (8)$$

$$F1_r = 2 \times \frac{R_r \times P_r}{R_r + P_r} \quad (9)$$

In a nutshell, an intersection between an anomaly interval and the whole set of predictions is enough to set $R_r(RA_i, PA)$ to one. $P_r(RA, PA_i)$ is determined in its dual form. To consider the manual labeling uncertainty in the real anomaly location [51], we run a preprocessing on the real anomaly regions, convolving the series with a rectangular window, to obtain better-defined anomaly ranges.

Table V summarizes the different anomaly detection approaches considered in the benchmark against *DC-VAE*. Most of these approaches correspond to univariate detection methods (except S-VAE), largely studied in the signal processing domain. A broad set of 15 univariate detectors are integrated into a single ensemble detector, referred to as ENS-15. The ensemble includes regression models, change-point detectors, outliers detectors, dimensionality reduction, clustering, and more. The aggregation corresponds to a majority voting strategy, where each detector is independently calibrated in the training and validation datasets, and a voting threshold maximizing $F1$ validation scores is computed. In TELCO, ENS-15 detects an anomaly if at least four ensemble models detect it. We also consider well-established time-series detectors, such as Seasonal Exponential Smoothing (S-EXPS) and the standard Auto-Regressive Integrated Moving Average (ARIMA) model. These approaches base the detection on the prediction of μ_x and σ_x for each time instant, making them particularly interesting to compare against *DC-VAE*. To

show the advantages of *DC-VAE* as compared to the usage of standard, vanilla VAEs for anomaly detection in time-series, we define the Standard-VAE (S-VAE) as a snapshot-input-based anomaly detection model, where the encoder/decoder architecture is based on a standard 3-layers, fully connected feed-forward neural network, and the input corresponds to the MTS at the specific time of detection – i.e., $T = 1$ in S-VAE. The comparison against S-VAE serves to demonstrate the advantages of *DC-VAE* temporal-aware architecture, through the dilated convolutions. Finally, evaluations are reported independently for each to the twelve time-series TS_m in the TELCO dataset.

Table VI reports the corresponding results in the testing dataset, independently for each time-series, and as an average value. The first observation is that achieved results are in general rather poor, achieving $F1_r$ scores around 60% for eight out of the twelve time-series, and below for the rest. This is highly in contrast with the high $F1$ scores usually reported in the literature, when dealing with simulated or flawed datasets [14]. Indeed, as we explained before, dealing with in-the-wild measurements and human-labeled, highly-imbalanced datasets is more complex than what the results in the literature usually report – real, in practice MTS anomaly detection is highly complex. Performance is significantly different for some of the time-series, which corresponds to the different nature and underlying behavior (cf. Figure 6) and the fraction of anomalies (cf. Table II). While *DC-VAE*'s performance as compared to S-VAE is outstanding, results show that no single approach is superior to the rest in all the time-series. *DC-VAE*'s performance is similar, on average, to S-EXPS and ARIMA. Still, among those already mentioned, the main advantage of *DC-VAE* remains its multivariate operation and the overall MTS modeling within a single learning step.

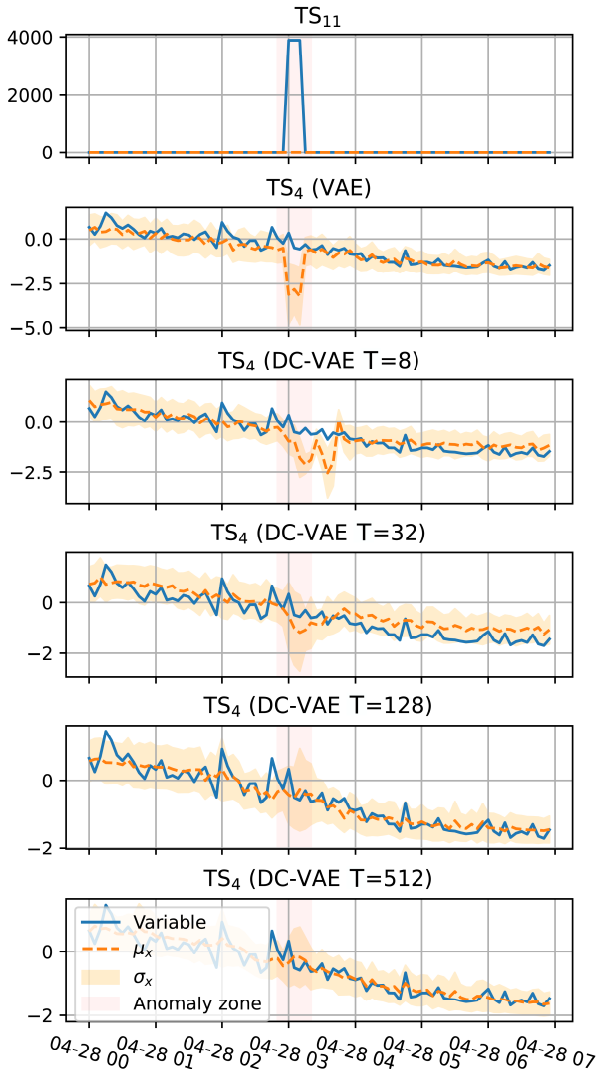


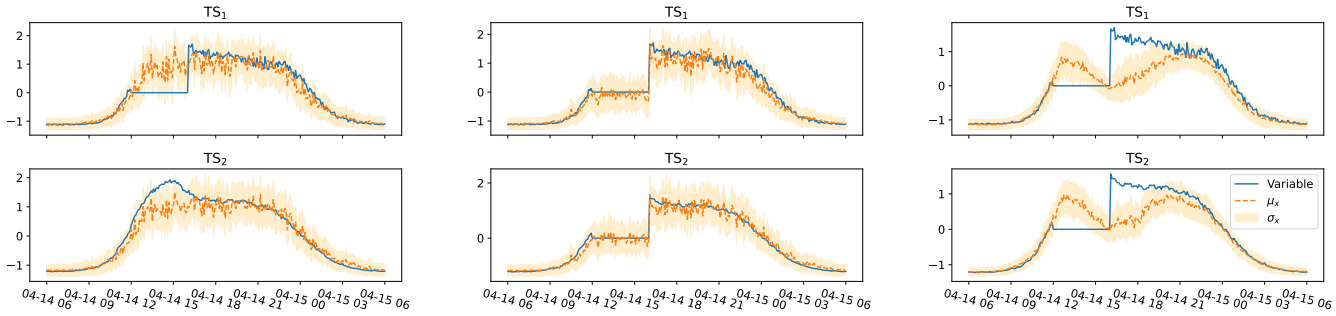
Fig. 12. A strong outlier in TS_{11} results in poor prediction for TS_4, with sequence length $T = 32$. This effect is mitigated with longer lengths T.

C. Benchmarking DC-VAE in the SWaT Open Dataset

For the sake of completeness and to provide a stronger and more comprehensive benchmarking, we compare *DC-VAE* against other deep-learning-based MTS anomaly detectors in SWaT. As discussed in the related work, GAN-based MTS detectors are very popular in the literature, given their flexibility to model a complex MTS process without making any assumptions on the underlying distributions. GANs are a powerful approach to learning the underlying distributions of data samples, in a purely data-driven, model-agnostic manner. Such models can be used in the practice to construct better normal-operation baselines, improving the identification of instances that deviate from this baseline. We, therefore, compare *DC-VAE* against three GAN-based detectors proposed in recent years, including EGAN [22], MAD-GAN [20], and our previous work on GAN-based MTS anomaly detection, referred to as NET-GAN [21].

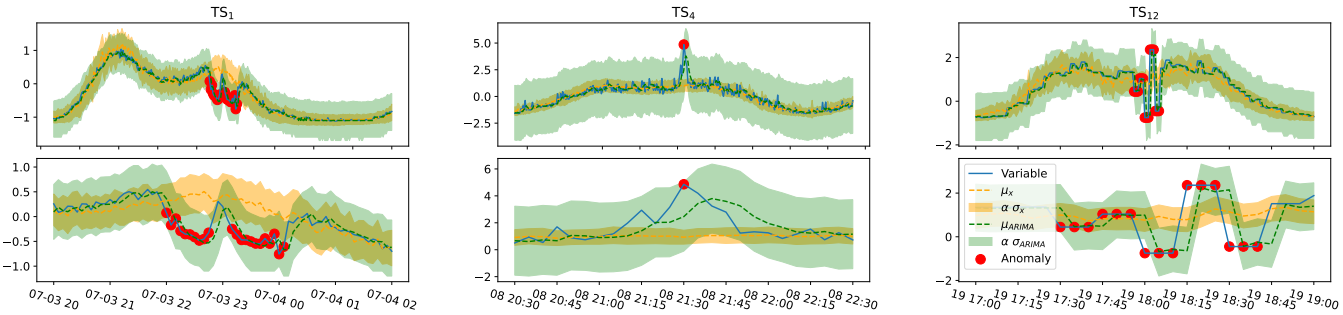
To train *DC-VAE* in SWaT, we take an architecture using $J = 16$ as the dimension of the latent space, and a sequence length $T = 128$, both parameters calibrated in the same way we did it in TELCO (cf. Figure 9). We train both *DC-VAE* and NET-GAN in the SWaT training dataset, using a small share of samples from the attacks for calibration. Regarding EGAN and MAD-GAN, we decided to report here the results obtained by the authors in [20], which would generally correspond to the best performance which could be achieved by these methods. Finally, we also include a standard Auto Encoder (AE) model as the simplest approach comparable to *DC-VAE*.

Table VII reports the results obtained in the testing dataset in terms of recall, precision, and $F1$ scores. We fall back to the standard evaluation on point anomalies instead of range anomalies, to be consistent with the results obtained in SWaT as reported in the literature. We consider two variations of NET-GAN detectors [21], one using the generator function (NET-GAN-G), and the other one the discriminator function (NET-GAN-D). We also consider three different variations of MAD-GAN, optimized for best precision (MAD-GAN-P), recall (MAD-GAN-R), and $F1$ score (MAD-GAN-F1). *DC-VAE* results are comparable to those obtained with NET-GAN-G and MAD-GAN-F1, and significantly better than EGAN or the AE model. In addition, absolute results are also significantly better than those obtained in TELCO, helping us demonstrate that anomaly detection in real data as the one in TELCO,



(a) S-VAE under univariate anomalies. (b) S-VAE under multivariate anomalies. (c) *DC-VAE* under multivariate anomalies.

Fig. 13. S-VAE and *DC-VAE* response to univariate and multivariate anomalies. The simultaneous modeling of the full MTS process adds regularity and stability to the detection.



(a) Range anomaly. (b) Point anomaly. (c) Fluctuating anomaly.

Fig. 14. *DC-VAE* and ARIMA response to range and point anomalies. The lower image is always a close-up view of the upper one. Being univariate and with a small temporal window makes ARIMA less robust for MTS anomaly detection, and missing anomalies.

dealing with the error-prone process of human labeling, is much more complex than what the literature usually reports on such benchmarks. To sum-up, we can claim that *DC-VAE* realizes state-of-the-art detection performance, while again, flagging its underlying advantages.

VI. Temporal and Spatial Response of *DC-VAE*

The visual exploration of *DC-VAE* predictions and detections in TELCO revealed certain behaviors of the model when confronted with different temporal and/or spatial patterns which are worth studying. In particular, the impact of the sequence length T on the reaction of the model to certain phenomena is relevant. Next, we present different prototypical examples of simulated anomalies and their impact on *DC-VAE* predictions, using S-VAE and the ARIMA models for comparison, when applicable.

*1) **Impact of strong outliers**:* the processing of the complete MTS simultaneously has evidenced, and in particular for simpler versions of the model with shorter sequence lengths T, that coarse outliers affecting a single time-series can affect the predictions for other time-series, generating false detections. Figure 12 shows how a major outlier in TS_{11} strongly perturbates predictions for TS_4, especially for sequence length below 32 in this example. This effect can be partially mitigated by taking longer sequences at the input. As a lesson learned, using longer sequences improves the filtering of strong outliers from the data.

*2) **Multivariate model properties**:* besides being more scalable in production, having a single model for the analysis of the complete MTS also improves detection. Figure 13(a) shows S-VAE model predictions for two highly correlated time-series, TS_1 and TS_2. An artificial univariate anomaly in TS_1, emulating a period where the time-series is constant (e.g., no incoming measurements), has a contained impact on the rest of the time-series predictions, as reflected in the predictions of $\boldsymbol{\mu}_x$ and $\boldsymbol{\sigma}_x$ for TS_2. As the S-VAE model has no temporal information (i.e., $T = 1$), predictions are influenced by the fact that the rest of the time-series remained unchanged. Nevertheless, in this example, the anomaly introduced in TS_1 would be clearly detected.

*3) **Temporal model properties**:* we now apply the previous anomaly to all the time-series in the same period and verify how the VAE-based models exploit temporal correlations among time-series. Figure 13(b) shows that this time, the S-VAE model predictions perfectly follow the anomaly, making it go completely undetected. The result is totally different for *DC-VAE*; as shown in Figure 13(c), the predictions of a *DC-VAE* model with a sequence length of $T = 512$ tend to follow the past behavior, and take longer to track the anomaly pattern, effectively detecting it.

Similar to *DC-VAE*, the ARIMA detection model enables the visualization of the normal-operation region. However, as we show in Figure 14, being univariate and with a small temporal window makes ARIMA less robust for MTS anomaly

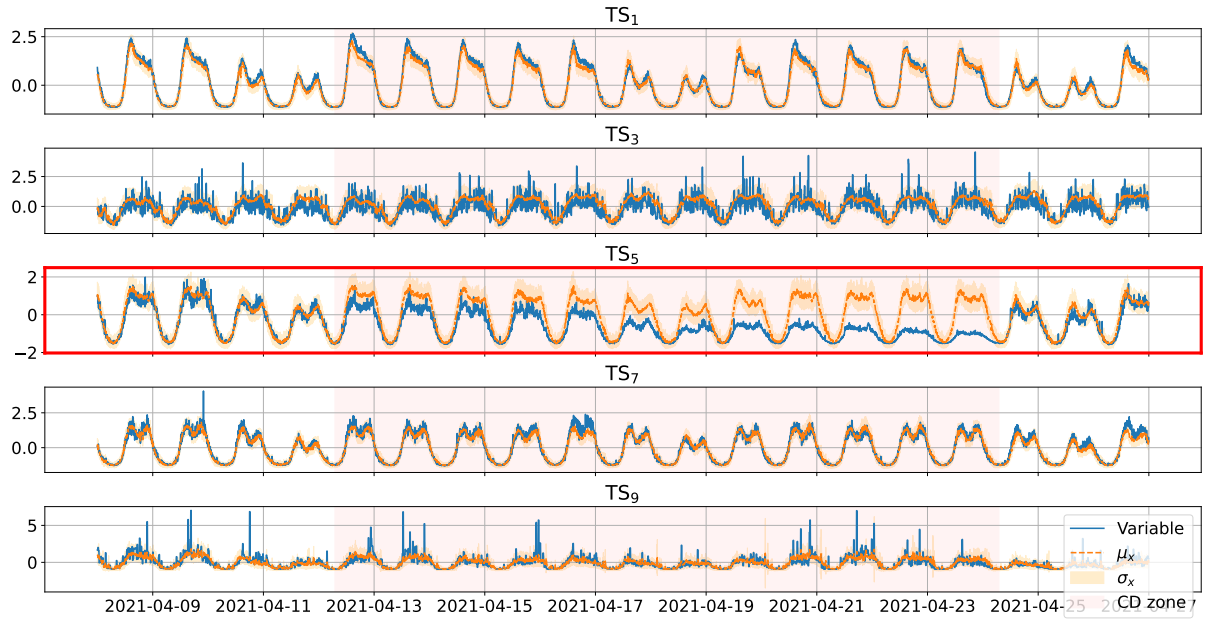


Fig. 15. *DC-VAE* response to univariate concept-drift: a gradual linear fall of the values during the day without affecting night behavior. While the drift does not affect the predictions on the other time-series, it becomes easily detectable at the corresponding time-series.

detection. In the figure, model predictions are depicted in green for ARIMA and in orange for *DC-VAE*, and red dots indicate real (i.e., labeled) anomalies. Figures 14(a) and 14(b) show that the value of σ_x for the ARIMA model is constant over time, but dynamically adapts in *DC-VAE*, providing a better, more accurate normal-operation region. This is a strong advantage of *DC-VAE*, since it adapts to the noise variations that these time-series generally present.

The same happens to the estimations of μ_x. While the estimation of the signal through the ARIMA model closely follows the time-series, even in the occurrence of real anomalies – and thus the model misses detection, the estimation provided by *DC-VAE* maintains a normal behavior in the face of the anomalies, allowing to properly detect them. The bigger spatial (M) and temporal (T) ranges of *DC-VAE* add robustness to the anomaly detection process.

4) ***Concept drift response****:* the ability to detect Concept Drift (CD) in time-series data is a paramount property [8]. The CD can manifest itself as a shift in the mean, an increase or decrease in the variance, or both changes simultaneously, which may be imperceptible for many methods [9]. These CD changes may be related to important trends in the data, requiring proper detection. We simulate a univariate CD in one of the time-series, and check the outputs of *DC-VAE*. Figure 15 shows an example of CD, where a gradual change in the interval indicated as the CD zone is simulated in TS_5. The daily values of the time-series are reduced linearly, starting at 80% (beginning of the CD zone) up to 40% (end of the CD zone). This change does not only affect the mean value of the time-series, but also its variance. Interestingly, predictions of the *DC-VAE* follow the past behavior learned as normal, allowing the CD event to be detected.

VII. Concluding Remarks

DC-VAE is a novel approach to anomaly detection in multivariate time-series, leveraging dilated convolutional neural networks and variational autoencoders. *DC-VAE* detects anomalies in multivariate time-series, exploiting temporal information without sacrificing computational and memory resources. In particular, instead of using recursive neural networks, large causal filters, or many layers, *DC-VAE* relies on dilated convolutions to capture long and short-term phenomena in the data, avoiding complex and less-efficient deep architectures, simplifying learning. Applying *DC-VAE* to real measurements collected at a mobile ISP showed that its underlying architecture is better than traditional, vanilla VAEs regarding time-series anomaly detection. The parameterization of *DC-VAE*'s architecture is defined by a single parameter, namely the length of the sliding window used for temporal analysis, and the normal operation region can be easily adapted on a per time-series basis by adjusting a single integer value, all of these important advantages in practice.

The performance analysis shows that *DC-VAE* has good properties for its implementation in production: scalability, easy adjustment of the normal-operation region, robustness against anomalies in other time-series, as well as against concept drift, which can also be detected. The application of *DC-VAE* in the TELCO and SWaT datasets shows the complementarity with other detection methods and the on-par performance with state-of-the-art MTS anomaly detectors in the literature. The quantitative and qualitative advantages of *DC-VAE* concerning S-VAE evidenced the contribution of the convolutional layers in capturing a longer time horizon.

The open release of the TELCO dataset offers a real, more representative environment to assess and benchmark anomaly detectors, providing a solid contribution to advance the domain.

ACKNOWLEDGMENT

This work has been partially supported 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*, by the CSIC I+D project with reference 22520220100371UD *Anomaly Detection in Time Series: Generalization and Domain Change Adaptation*, by Telefónica, and by the Austrian FFG ICT-of-the-Future project *DynAISEC – Adaptive AI/ML for Dynamic Cybersecurity Systems* – project ID 887504. Gastón García was supported by the ANII scholarship POS-FMV-2020-1-1009239, as well as by CSIC, under program *Movilidad e Intercambios Académicos 2022*. We thank anonymous reviewers as well as associate editors for their constructive feedback and comments, which helped improving our work.

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. [Online]. Available: https://doi.org/10.1145/3444690

[2] 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.

[3] 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

[4] 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.

[5] 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.

[6] 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. [Online]. Available: https://doi.org/10.1145/3439950

[7] T. Hoens, R. Polikar, and N. Chawla, "Learning from streaming data with concept drift and imbalance: an overview," *Progress in Artificial Intelligence*, vol. 1, 2012.

[8] J. a. Gama, I. Žliobaitundefined, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *Association for Computing Machinery*, vol. 46, 2014.

[9] G. H. F. M. Oliveira, R. C. Cavalcante, G. G. Cabral, L. L. Minku, and A. L. I. Oliveira, "Time series forecasting in the presence of concept drift: A pso-based approach," in *2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI)*, 2017, pp. 239–246.

[10] N. Laptev, S. Amizadeh, and Y. Billawala, "S5 - a labeled anomaly detection dataset," 2015. [Online]. Available: https://webscope.sandbox.yahoo.com/catalog.php?datatype=s

[11] A. Lavin and S. Ahmad, "Evaluating real-time anomaly detection algorithms–the numenta anomaly benchmark," in *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2015, pp. 38–44.

[12] S. Ahmad, A. Lavin, S. Purdy, and Z. Agha, "Unsupervised Real-time Anomaly Detection for Streaming Data," *Neurocomputing*, vol. 262, pp. 134–147, 2017.

[13] K. Hundman, V. Constantinou, C. Laporte, I. Colwell, and T. Soderstrom, "Detecting spacecraft anomalies using lstms and non-parametric dynamic thresholding," in *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2018, pp. 387–395.

[14] R. Wu and E. Keogh, "Current Time Series Anomaly Detection Benchmarks are Flawed and are Creating the Illusion of Progress," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–1, 2021.

[15] D. P. Kingma and M. Welling, "Auto-encoding Variational Bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[16] 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.

[17] G. García González, S. Martínez Tagliafico, A. Fernández, G. Gómez, J. Acuña, and P. Casas, "TELCO – a new Multivariate Time-Series Dataset for Anomaly Detection in Mobile Networks," 2023. [Online]. Available: https://dx.doi.org/10.21227/skpg-0539

[18] 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.

[19] 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 *Proceedings of the 7th Workshop on Traffic Measurements for Cybersecurity (WTMC)*, 2022 (to appear), pp. 1–6.

[20] 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.

[21] 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

[22] H. Zenati, C. S. Foo, B. Lecouat, G. Manek, and V. R. Chandrasekhar, "Efficient GAN-Based Anomaly Detection," *CoRR*, vol. abs/1802.06222, 2018.

[23] M. Braei and S. Wagner, "Anomaly Detection in Univariate Time-series: A Survey on the State-of-the-Art," *CoRR*, vol. abs/2004.00433, 2020. [Online]. Available: https://arxiv.org/abs/2004.00433

[24] D. Baessler, T. Kortus, and G. Guehring, "Unsupervised anomaly detection in multivariate time series with online evolving spiking neural networks," *Machine Learning*, vol. 111, p. 1377–1408, 2022.

[25] K. Choi, J. Yi, C. Park, and S. Yoon, "Deep Learning for Anomaly Detection in Time-Series Data: Review, Analysis, and Guidelines," *IEEE Access*, vol. 9, 2021.

[26] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Nets," in *Advances in Neural Information Processing Systems*, vol. 27, 2014.

[27] 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.

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

[29] 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.

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

[31] 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.

[32] C. Doersch, "Tutorial on variational autoencoders," *arXiv preprint arXiv:1606.05908*, 2016.

[33] D. P. Kingma and M. Welling, "An introduction to variational autoencoders," *arXiv preprint arXiv:1906.02691*, 2019.

[34] F. P. Casale, A. V. Dalca, L. Saglietti, J. Listgarten, and N. Fusi, "Gaussian Process Prior Variational Autoencoders," in *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, 2018, pp. 10 390–10 401.

[35] L. Girin, F. Roche, T. Hueber, and S. Leglaive, "Notes on the use of variational autoencoders for speech and audio spectrogram modeling," in *DAFx 2019-22nd International Conference on Digital Audio Effects*, 2019, pp. 1–8.

[36] V. Fortuin, D. Baranchuk, G. Rätsch, and S. Mandt, "GP-VAE: Deep Probabilistic Time Series Imputation," in *International conference on artificial intelligence and statistics*. PMLR, 2020, pp. 1651–1661.

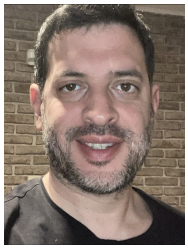
[37] S. Ramchandran, G. Tikhonov, K. Kujanpää, M. Koskinen, and H. Lähdesmäki, "Longitudinal variational autoencoder," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2021, pp. 3898–3906.

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

[39] 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.

[40] Y. Su, Y. Zhao, C. Niu, R. Liu, W. Sun, and D. Pei, "Robust anomaly detection for multivariate time series through stochastic recurrent neural network," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 2828–2837.
[41] S. Shabanian, D. Arpit, A. Trischler, and Y. Bengio, "Variational bi-LSTMs," *arXiv preprint arXiv:1711.05717*, 2017.
[42] 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.
[43] 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.
[44] C. Meng, X. S. Jiang, X. M. Wei, and T. Wei, "A time convolutional network based outlier detection for multidimensional time series in cyber-physical-social systems," *IEEE Access*, vol. 8, pp. 74 933–74 942, 2020.
[45] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.
[46] H. Zhang, Y. Xia, T. Yan, and G. Liu, "Unsupervised anomaly detection in multivariate time series through transformer-based variational autoencoder," in *2021 33rd Chinese Control and Decision Conference (CCDC)*. IEEE, 2021, pp. 281–286.
[47] H. Xu, W. Chen, N. Zhao, Z. Li, J. Bu, Z. Li, Y. Liu, Y. Zhao, D. Pei, Y. Feng *et al.*, "Unsupervised anomaly detection via variational auto-encoder for seasonal KPIs in web applications," in *Proceedings of the 2018 World Wide Web Conference*, 2018, pp. 187–196.
[48] N. Tatbul, T. J. Lee, S. Zdonik, M. Alam, and J. Gottschlich, "Precision and Recall for Time Series," *32nd Conference on Neural Information Processing Systems (NeurIPS)*, 2018.
[49] J. Beirlant, Y. Goegebeur, J. Segers, and J. L. Teugels, "Statistics of extremes: theory and applications," vol. 558, 2004.
[50] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," *Advances in neural information processing systems*, vol. 24, 2011.
[51] A. Shahid, G. White, J. Diuwe, A. Agapitos, and O. O'Brien, "SLMAD: Statistical Learning-Based Metric Anomaly Detection," in *International Conference on Service-Oriented Computing*. Springer, 2020, pp. 252–263.

Gastón García González received the B.Sc. and M.Sc. from Universidad de la República, Uruguay, in 2018 and 2020 respectively. He is currently a Ph.D. candidate and holds a position as Teaching Assistant/Lecturer in the Signal Processing Department at the Electrical Engineering Institute (IIE), Universidad de la República. His main research areas include signal processing, anomaly detection, and machine learning.



Sergio Martínez is an Assistant Professor at the Engineering Faculty of the Universidad de la República, Uruguay, working on signal processing and data science. He holds a degree in Electrical Engineering since July 2012 and a Specialization Diploma in Telecommunications since 2013 from the Universidad de la República.

Alicia Fernandez is a Professor of Signal Processing at the Electrical Engineering Institute (IIE), Universidad de la República, Uruguay. Since 1989, she works at the IIE, in telecommunication and signal processing areas. Her main research interests are signal processing and pattern recognition with focus in biomedical image analysis, biometric identification, anomaly detection and big data analysis.

Gabriel Gómez Sena is an Associate Professor at the Engineering School of the Universidad de la República, Uruguay. He holds a degree in Industrial Engineering (speciality Electronics) since December 1987. He received his M.Sc. in Electrical Engineering from Universidad de la República in 2011, with a thesis concerning statistical methods for traffic classification. His current research interests are related to networking protocols, software defined networking, and anomaly detection in telecommunication networks.

José Acuña is an Assistant Professor at the Engineering Faculty of the Universidad de la República, Uruguay. He holds a degree in Electrical Engineering since July 1994. He received his Ph.D. in Signal Theory and Communications from Universidad de Vigo, Spain, in 2013 with a thesis concerning efficiency of OFDM systems on urban radio channels. His current research interests are related to telecommunication networks and anomaly detection.

Pedro Casas (Member, IEEE) received the Electrical Engineering degree from the Universidad de la República, Uruguay, in 2005, and the Ph.D. degree in computer science from Télécom Bretagne, in 2010. He is currently a Senior Scientist in AI/ML for Networking, with the AIT Austrian Institute of Technology, Vienna. He was a Postdoctoral Researcher with the LAAS-CNRS, Toulouse, from 2010 to 2011, and a Senior Researcher with the Telecommunications Research Center Vienna, from 2011 to 2015. His work focuses on machine learning-based approaches for networking, big data analytics and platforms, Internet network measurements, network security, and anomaly detection, as well as Internet QoE monitoring. He has published more than 200 networking research papers in major international conferences and journals, and received 17 awards for his work, including eight Best Paper Awards. He is the General Chair for different actions in network measurement and analysis, including the IEEE ComSoc ITC Special Interest Group on Network Measurements and Analytics.