# DC-VAE, Fine-grained Anomaly Detection in Multivariate Time-Series with Dilated Convolutions and Variational Auto Encoders


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

Sergio Martinez Tagliafico
*IIE–FING*
*Universidad de la República*
*Montevideo, Uruguay*
*sematag@fing.edu.uy*

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

Gabriel Gómez
*IIE–FING*
*Universidad de la República*
*Montevideo, Uruguay*
*ggomez@fing.edu.uy*

José Acuña
*Universidad de la República*
*& Telefónica Uruguay*
*Montevideo, Uruguay*
*acuna@fing.edu.uy*

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



***Abstract*—Due to its unsupervised nature, anomaly detection plays a central role in cybersecurity, in particular on the detection of unknown attacks. A major source of cybersecurity data comes in the form of multivariate time-series (MTS), representing the temporal evolution of multiple, usually correlated measurements. Despite the many approaches available in the literature for time-series anomaly detection, the automatic detection of abnormal events in MTS remains a complex problem. In this paper we introduce *DC-VAE*, a novel approach to anomaly detection in MTS, leveraging convolutional neural networks (CNNs) and variational auto encoders (VAEs). DC-VAE detects anomalies in time-series data, 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, avoiding complex and less-efficient deep architectures, simplifying learning. We evaluate DC-VAE on the detection of anomalies on a large-scale, multi-dimensional network monitoring dataset collected at an operational mobile Internet Service Provider (ISP), where anomalous events were manually labeled during a time span of 7-months, at a five-minutes granularity. Results show the main properties and advantages introduced by VAEs for time-series anomaly detection, as well as the out-performance of dilated convolutions as compared to standard VAEs for time-series modeling.**



***Index Terms*—Anomaly Detection, Deep Learning, Multivariate Time-Series, Dilated Convolution, VAE**


## 1. Introduction

Cybersecurity 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 cybersecurity, in particular to detect 0-day attacks or threats never seen before. 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 cybersecurity time-series data, these traditional models have difficulty predicting them 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.

In this paper we introduce *DC-VAE*, an unsupervised and multivariate approach to anomaly detection in time-series, based on popular Variational Auto-Encoders (VAEs). VAEs are a generative version of classical auto-encoders, with the particularity of having, by conception, continuous latent spaces; as such, VAEs map the input variables into a multivariate latent distribution, which enables a generative process. A VAE provides 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. One of the key advantages of VAEs for anomaly detection is that, for a given input, they produce as output prediction (i.e., reconstruction) 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 underlying approach allows the user to visualize the region of normal behavior in a simple and appealing way, enabling fine-grained, per univariate time-

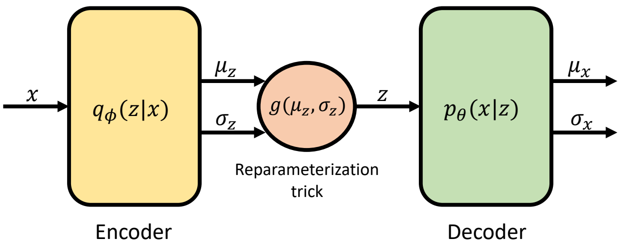
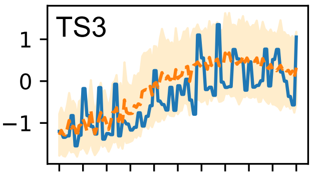
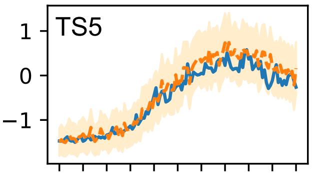


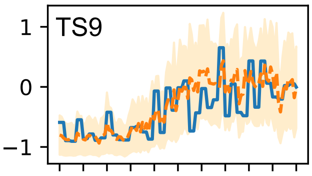

Figure 1. Variational autoencoder and the reparameterization trick.



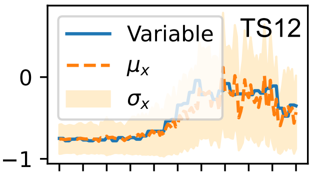

(a) Prediction of time-series $\mathrm{TS}_3$.




(b) Prediction of time-series $\mathrm{TS}_5$.




(c) Prediction of time-series $\mathrm{TS}_9$.




(d) Prediction of time-series $\mathrm{TS}_{12}$.

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

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 [7], 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 most importantly, enabling training – and detection – on longer-in-the-past temporal sequences.

We apply DC-VAE to MTS arising from the monitoring of an operational mobile ISP, detecting anomalies of very different structural properties. We compare DC-VAE against a traditional VAE model for snapshot-input-based anomaly detection, where the encoder/decoder architecture is based on standard, fully connected feed-forward neural networks, and the input corresponds to the MTS at the specific time of detection. We shall refer to this model as Standard-VAE (S-VAE).

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 in detail; Section 4 presents the mobile ISP dataset collected for evaluation, and reports the results obtained with DC-VAE in the detection of anomalies, additionally benchmarking its performance against S-VAE. Finally, Section 5 concludes the paper.

## 2. 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. Modern approaches to time-series anomaly detection based on deep learning technology have flourished in recent years [6]. 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 [8]–[14].

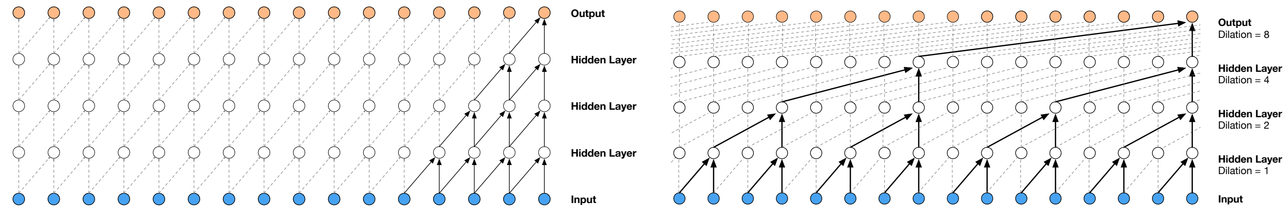
Modeling data sequences through a combination of variational inference and deep learning architectures has been vastly researched in other domains in recent years, mostly by extending VAEs to Recurrent Neural Networks (RNNs), with architectures such as STORN [15], VRNN [16], and Bi-LSTM [17] among others. Convolutional layers with dilation have been also incorporated into some of these approaches [18], [19], allowing to speed up the training process based on the possibilities of parallelization offered by these architectures.

This work has as basis our previous work on generative models for network anomaly detection in mutivariate time-series [14], 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.

## 3. 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 $\boldsymbol{\mu}_x$ 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, used in this paper for evaluation purposes (see Section 4). For each of the displayed time-series $\mathrm{TS}_i$ – the TELCO dataset corresponds to tweleve time-series $\mathrm{TS}_1$ to $\mathrm{TS}_{12}$, its real value $x_i$, along with the outputs of the VAE $\mu_{x_i}$ and $\sigma_{x_i}$, are reported.




(a) Normal convolution. (b) Dilated convolution.

Figure 3. (*)Figure taken from the original WaveNet paper [7]. 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.

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 $p_\theta$ with parameters $\theta$. 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:

$$p_\theta(\boldsymbol{x}) \quad = \quad \int_{\boldsymbol{z}} p_\theta(\boldsymbol{x}, \boldsymbol{z})\, d\boldsymbol{z}, \qquad (1)$$

where $p_\theta(\boldsymbol{x}, \boldsymbol{z})$ represents the joint distribution under $p_\theta$ 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:

$$p_\theta(\boldsymbol{x}) \quad = \quad \int_{\boldsymbol{z}} p_\theta(\boldsymbol{x}|\boldsymbol{z}) p_\theta(\boldsymbol{z})\, d\boldsymbol{z} \qquad (2)$$

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, while the approximated posterior distribution $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ is computed by the *probabilistic* encoder, cf. Figure 1.

To train this 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 and the probabilistic encoder has to learn how to map a compressed representation of the input into the two latent vectors $\boldsymbol{\mu_z}$ and $\boldsymbol{\sigma_z}$, while the stochasticity remains excluded from the updating process and is injected in the latent space as an external input through $\boldsymbol{\varepsilon}$.

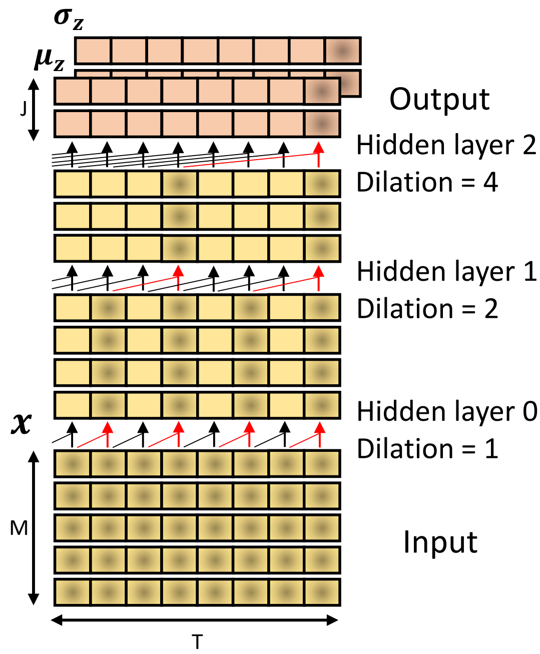



Figure 4. Encoder architecture using causal dilated convolutions, implemented through a stack of 1D convolutional layers.

To exploit the temporal dimension of the input time-series, we proposed an encoder/decoder architecture based on popular CNNs, using Dilated Convolutions (DCs) [7]. 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 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.

| 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 WERE MANUALLY LABELED, FOR TWELVE DIFFERENT METRICS.

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.

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:

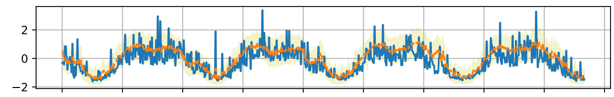
$$|x_m(t) - \mu_m(t)| > \alpha_m \times \sigma_m(t), \quad (3)$$

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

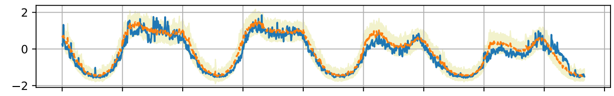
# 4. DC-VAE Evaluation and Benchmarking

## 4.1. The TELCO Dataset

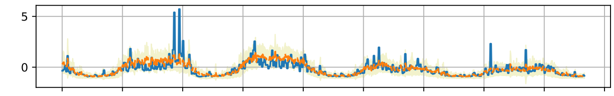
A recent study [20] alerts on the limitations of evaluating anomaly detection algorithms on popular time-series datasets such as Yahoo, Numenta, or NASA among others. In particular, these datasets are noted to suffer from known flaws such as trivial anomalies, unrealistic anomaly density, mislabeled 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 – we are currently working on the possible public release of this dataset to the community, but as of today, the dataset is private. The TELCO dataset 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. 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 purposes.



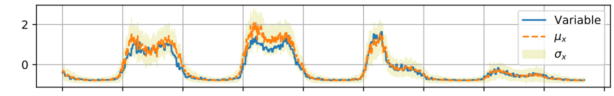

(a) Prediction of time-series $TS_3$.




(b) Prediction of time-series $TS_5$.




(c) Prediction of time-series $TS_9$.



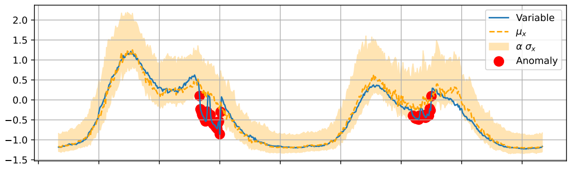

(d) Prediction of time-series $TS_{12}$.

Figure 5. Example of time-series analysis through DC-VAE, for the TELCO dataset, using $T = 512$ samples – almost 2 days of temporary receptive field in the past.

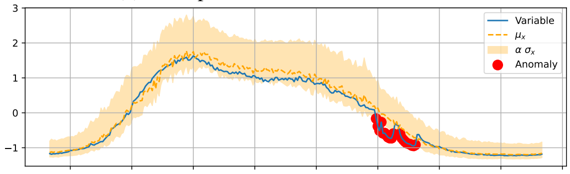
## 4.2. Evaluation Results

Figure 5 shows DC-VAE in action, using a sliding-window of length $T = 512$ samples, corresponding to roughly two days of past measurements. This length of time-window is the one providing better validation results in the TELCO dataset. We take the same time-series depicted in Figure 2 as reference, but now considering a longer time span of four days. 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, e.g., visible in Figure 5(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 $\alpha_m$ 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 $\sigma(t)$.

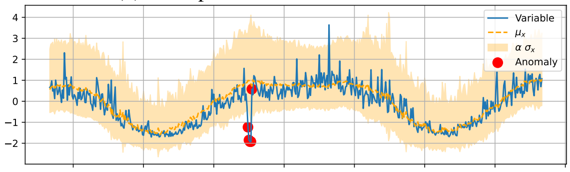
To apply DC-VAE for anomaly detection, we have to calibrate the sensitivity thresholds $\boldsymbol{\alpha}$, which is usually done in a supervised manner, relying on the labeled anomalies available in the training and validation datasets. This step is the only one which requires certain level of



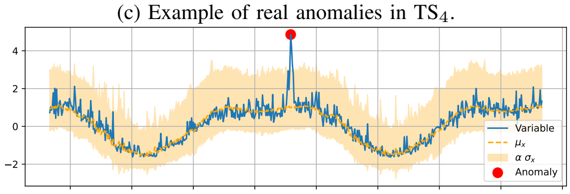

(a) Example of real anomalies in $\mathrm{TS}_2$.




(b) Example of real anomalies in $\mathrm{TS}_2$.




(c) Example of real anomalies in $\mathrm{TS}_4$.




(d) Example of real anomalies in $\mathrm{TS}_4$.

Figure 6. Examples of real anomalies present in the analyzed dataset, and their identification by DC-VAE.

"supervision" (in the sense of ground-truth availability), but could also be done in a self-supervised manner, by labeling anomalies through outlier detection techniques. In our specific problem, each sensitivity threshold $\alpha_m$ 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 $\sigma_m$ shall be considered as tolerance for the normal-operation variability of the data.

Figure 6 reports some examples of real (i.e., labeled) anomalies present in the TELCO dataset, in particular for time-series $\mathrm{TS}_2$ and $\mathrm{TS}_4$, 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 $\mathrm{TS}_4$, or on a more structural basis in the case of $\mathrm{TS}_2$. Note also how some of the actual measurements fall significantly outside the normal-operation region – e.g. in Figure 6(c), but still these were not labeled as anomalous by the expert operator. Whether or not this is a false-positive produced by DC-VAE, or a non-labeled anomaly missed by the expert operator is difficult to know.

Here it is important to note that anomalies in the wild data, as reported and 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, at least in the eyes of the expert operator. Manual data labeling by experts is prone to human error, many times due to lack of conclusive information available to the operator to take a proper decision. These observations are actually critical to consider when evaluating anomaly detectors with real, in the wild data. As a relevant note in this direction, such complexities in the process of properly labeling data, and its interaction with the actual performance of the AI/ML data-driven model, have originated a pretty novel discipline referred to as *Data-centric AI* (DCAI) [21], which studies the problem of systematically engineering high-quality datasets to train machine learning models.

We also run a quantitative performance analysis of DC-VAE in the testing dataset (cf. Table 1). As performance metrics, we consider an elaborated version of the traditionally used, per-sample evaluation metrics, to consider a 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 6(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 work have considered these observations [22]–[24], defining new metrics which prioritize early or delayed detection, or focusing mainly on range anomalies. We therefore take the extended definitions of recall and precision as defined in [24] 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 are 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. Finally, evaluations are reported independently for each to the twelve time-series $\mathrm{TS}_m$ in the TELCO dataset.

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. Table 2 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 [20]. 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 5). Nevertheless, the outperformance of DC-VAE as compared to S-VAE is outstanding, largely improving both detection of anomalies (i.e., $R_r$) as well as overall performance (i.e., $F1_r$), by almost a factor of two on average.

| TS ID | S-VAE $R_r$ | S-VAE $P_r$ | S-VAE $F1_r$ | DC-VAE $R_r$ | DC-VAE $P_r$ | DC-VAE $F1_r$ |
|---|---|---|---|---|---|---|
| $TS_1$ | 23% | 56% | 32% | 58% | 71% | **64%** |
| $TS_2$ | 16% | 92% | 27% | 74% | 20% | **67%** |
| $TS_3$ | 71% | 50% | 59% | 86% | 47% | **60%** |
| $TS_4$ | 63% | 25% | **36%** | 63% | 21% | 32% |
| $TS_5$ | 50% | 20% | 29% | 75% | 50% | **60%** |
| $TS_6$ | 14% | 100% | 25% | 57% | 83% | **68%** |
| $TS_7$ | 45% | 100% | 63% | 72% | 90% | **80%** |
| $TS_8$ | 57% | 35% | 43% | 44% | 80% | **57%** |
| $TS_9$ | 6% | 4% | 4% | 17% | 11% | **13%** |
| $TS_{10}$ | 39% | 81% | 52% | 52% | 59% | **55%** |
| $TS_{11}$ | 67% | 17% | 27% | 100% | 25% | **40%** |
| $TS_{12}$ | 0% | 0% | 0% | 100% | 11% | **22%** |
| mean | 38% | 48% | 33% | **67%** | 47% | **52%** |
| median | 42% | 43% | 31% | **68%** | 49% | **59%** |

TABLE 2. ANOMALY DETECTION PERFORMANCE WITH DC-VAE AND S-VAE.

A preliminary assessment on the low performance obtained for some of the time-series reveals issues linked to poor labeling in some cases, as well as lack of sensitivity in some others (i.e., finer-grained $\alpha$ values might be needed). Still, DC-VAE results in terms of its modeling and tracking capabilities for multivariate time-series data are promising, and its application to real measurements additionally permits to evidence the difficulties behind a broadly studied, yet unsolved problem. A deeper evaluation of DC-VAE in the TELCO dataset is part of our ongoing work, including the benchmarking of other anomaly detection approaches in this dataset.

## 5. Concluding Remarks

DC-VAE is a novel approach to anomaly detection in multivariate time-series, leveraging dilated convolutional neural networks and variational auto encoders. 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. The application of DC-VAE to real measurements collected at a mobile ISP showed that its underlying architecture is better than traditional, vanilla VAEs when it comes to time-series anomaly detection, showing as such promising results. The parametrization of DC-VAE's architecture is basically 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. We are currently evaluating DC-VAE on top of publicly available datasets recently put into question by the research community, from what we expect to realize state-of-the-art detection performance; this should help us demonstrating that anomaly detection in real data as the one considered in this paper, dealing with the error-prone process of human labeling, is actually much more complex than what the literature usually reports on such benchmarks. The exploration of the data-centric AI domain and its application to the problem of anomaly detection in in-the-wild multivariate time-series data looks like a promising venue to improve the field. We are working together with the mobile ISP originating the TELCO dataset to release it to the community in the short future.

## 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 Telefónica, and by the Austrian FFG ICT-of-the-Future project *DynAISEC – Adaptive AI/ML for Dynamic Cybersecurity Systems*. 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. [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] 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.

[8] 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.

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

[10] 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.

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

[12] 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.

[13] 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.

[14] 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. 4952, may 2021. [Online]. Available: https://doi.org/10.1145/3466826.3466843

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

[16] 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.

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

[18] 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.

[19] 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.

[20] 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.

[21] A. Ng, "Unbiggen AI," *IEEE Spectrum*, 2022. [Online]. Available: https://spectrum.ieee.org/andrew-ng-data-centric-ai

[22] 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.

[23] 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.

[24] 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.