On the Quest for Foundation Generative-AI Models for Anomaly Detection in Time-Series Data

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

Pedro Casas
AIT - Austrian Institute of Technology
Vienna, Austria
pedro.casas@ait.ac.at

Emilio Martínez
IIE–FING, Universidad de la República
Montevideo, Uruguay
emartinez@fing.edu.uy

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

***Abstract*—Network security 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. We investigate a novel approach to time-series modeling, inspired by the successes of large pre-trained foundation models. We introduce FAE (Foundation Auto-Encoders), a foundation generative-AI model for anomaly detection in time-series data, based on Variational Auto-Encoders (VAEs). By foundation, we mean a model pre-trained on massive amounts of time-series data which can learn complex temporal patterns useful for accurate modeling, forecasting, and detection of anomalies on previously unseen datasets. Based on the DC-VAE architecture originally designed for multivariate anomaly detection, FAE leverages VAEs and Dilated Convolutional Neural Networks (DCNNs) to build a generic model for univariate time-series modeling, which could eventually perform properly in out-of-the-box, zero-shot anomaly detection applications. We introduce the main concepts and ideas of this foundation model, and present some preliminary results in a multi-dimensional network monitoring dataset, collected from an operational mobile Internet Service Provider (ISP). This work represents a significant step forward in the development of foundation generative-AI models for anomaly detection in time-series analysis, with applications spanning cybersecurity, network management, and beyond.**

***Index Terms*—Multivariate Time-Series Data, Anomaly Detection, Generative AI, VAE, Foundation Models**

1. Introduction

Network security and monitoring data typically comprises hundreds or even thousands of variables that are regularly measured and analyzed as time-series data, resulting in complex multivariate time-series (MTS) processes. Detecting anomalies in real-time within such MTS processes is crucial for effective network security, in particular to detect unknown attacks. While the literature offers a plethora of traditional statistical models for anomaly detection in time-series data, they often struggle with the non-stationary, non-linear, and noisy nature of network monitoring data, leading to suboptimal predictions. In recent years, there has been a surge in the adoption of modern deep learning-based approaches for time-series anomaly detection [1], owing to their ability to handle complex dependencies and generate realistic data sequences. Generative AI methodologies, in particular, have gained attention for their performance in time-series modeling [2]–[4].

In this paper, we focus on devising a Generative AI model capable of matching or even surpassing the performance of conventional time-series modeling methods without the need for training on the specific target dataset - a concept known as Zero-Shot Learning (ZSL). ZSL is a problem setup in deep learning where, at test time, a learner observes samples from classes which were not observed during training, and needs to predict the class that they belong to. The ZSL concept is powerful and appealing for network security applications, and such a foundation model could be utilized with limited, or even without specific fine-tuning on the downstream data typically used by other models. The zero-shot approach offers several inherent advantages: firstly, it simplifies the application of the model for time-series modeling, eliminating the requirement for specialized knowledge of fine-tuning techniques and the significant computational resources associated with them; secondly, it naturally aligns with scenarios characterized by limited data availability, where training or fine-tuning data is limited; lastly, by harnessing the comprehensive pattern extrapolation capabilities of extensively pre-trained models, it circumvents the substantial time, effort, and domain-specific expertise typically demanded for crafting dedicated time-series models.

We therefore investigate if a model pre-trained on multiple time-series data can learn temporal patterns useful for accurate forecasting on previously unseen time-series. For doing so, we use as starting point our former DC-VAE model [5], a deep-learning-based, unsupervised, and multivariate approach to real-time anomaly detection in MTS data, based on popular Variational Auto-Encoders (VAEs) [6]. VAEs are generative AI models that learn the underlying distribution of the data and can generate

new samples from this distribution. In the context of time-series data, VAEs can capture latent representations of temporal patterns and generate sequences that exhibit similar characteristics, making them powerful for generalization and ZSL. VAEs learn a low-dimensional latent space representation of the input data, which captures the underlying structure of the data in a compressed form. By learning meaningful representations, VAEs can generalize well to unseen data points that lie within the same distribution as the training data, supporting generalization to new instances. As generative models, VAEs can generate new samples from the learned latent space distribution. potentially enabling ZSL, as the model can produce samples that belong to unseen classes or categories without explicitly training on them. By sampling from the latent space, VAEs can generate diverse and realistic data points even for classes not present in the training set.

We introduce and investigate FAE (Foundation Auto-Encoders), a foundation generative-AI model for anomaly detection in time-series data, based on VAEs. FAE uses DC-VAE's network architecture [5], originally designed for multivariate anomaly detection. In particular, it leverages VAEs and Dilated Convolutional Neural Networks (DCNNs) to build a generic model for *univariate time-series modeling*, which could eventually perform properly in out-of-the-box, zero-shot anomaly detection applications. The reasons for moving from multivariate to univariate time-series analysis are twofold: from an architectural point of view, we become independent of the spatial dimensionality of a MTS dataset – i.e., we fix the spatial input dimensionality to one – and can therefore apply exactly the same architecture without any modifications; from an analytics perspective, while a univariate model is at a disadvantage compared to a multivariate model – i.e., it loses access to cross-correlational information, which we have shown might be critical for better data modeling [5], [7] – the univariate version puts the focus exclusively on the temporal behavior of the data, which is exactly the target of the generalization we are looking for – we want a model that generalizes across the temporal dimension and not necessarily the spatial one, which varies among different problems.

We introduce the main concepts and ideas behind FAE, along with its network architecture, and present some preliminary results in the analysis a multi-dimensional network monitoring dataset – TELCO [8], collected from an operational mobile Internet Service Provider (ISP), which we have recently released openly to the research community. The remainder of the paper is organized as follows: Section 2 presents an overview of the related work. In Section 3 we describe the FAE model and its underlying network architecture. Section 4 reports the preliminary results obtained with FAE on the analysis of time-series from TELCO, with a particular focus on generalization and zero-shot modeling. Discussion on the potential of FAE, along with its limitations, is presented in Section 5. Finally, Section 6 concludes the paper.

2. Related Work

There are multiple surveys on general-domain anomaly detection techniques [9]–[11] as well as on network anomaly detection [12], [13]. 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 [1]. Due to their data-driven nature and achieved performance in multiple domains, generative models such as VAEs and Generative Adversarial Networks (GANs) [14] have gained relevance in the anomaly detection field [4], [15]–[20].

VAEs [6], [21], [22] 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 to improve model performance [23]–[26]. For example, while the original work [6] 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, [25] 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 [27], VRNN [28], and Bi-LSTM [29] among others. Convolutional layers with dilation have been also incorporated into some of these approaches [2], [30], allowing to speed up the training process based on the possibilities of parallelization offered by these architectures. One of these approaches using Dilated Convolutional Neural Networks as the encoder-decoder architecture for VAEs is our DC-VAE model [5], [31].

Transformer-based models [32] are gaining popularity in recent years for time-series analysis, given their remarkable performance in large-scale settings, such as long sequence time-series forecasting (LSTF). LSTF requires capturing long-range dependencies between input and output efficiently. Earlier examples include the TFT interpretable model [33] and the MQTransformer model [34]. The Informer model [35] introduced Transformers for long sequence forecasting through sparse self-attention mechanisms. This concept has since been further refined through various forms of inductive bias and attention mechanisms in models like the Autoformer [36] and the FEDformer [37].

Finally, there is a recent surge in papers targeting the conception of foundation models for time-series data, capable of generating accurate predictions for diverse datasets not seen during training. The underlying concept of these models is to rely on highly expressive, large-scale architectures which are trained on millions or billions of time-series data points, coming from very diverse domains and having high heterogeneity in terms of temporal behaviors and characteristics. TimeGPT-1 [38], PromptCast [39], LLMTime [40], TimesFM [41], Lag-Llama [42], and Time-LLM [43] are all examples of novel foundation models for time-series forecasting, which target a ZSL application. FAE follows exactly this concept, but using a much smaller and simpler architecture. While this adds

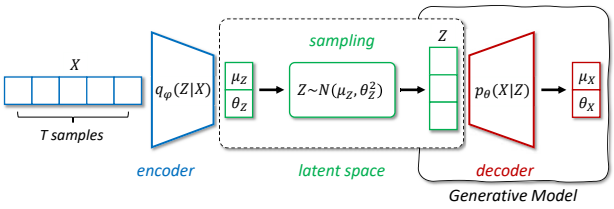


Figure 1. FAE's variational autoencoder. The decoding function enables a generative AI process, by sampling the latent space distribution.

certain limitations in terms of expressiveness and therefore generalization capabilities, it also opens the door to the exploration of other venues, such as the combined utilization of smaller foundation models in the form of ensembles, in combination with domain detection strategies.

3. FAE Model and Network Architecture

Time-series are generally processed through sliding windows, condensing the information of the most recent T measurements. We define $\boldsymbol{X}$ as the input vector in $\mathbb{R}^{1\times T}$. As depicted in Figure 1, for a given input $\boldsymbol{X}$, the trained VAE model produces two different predictions, $\boldsymbol{\mu_X}$ and $\boldsymbol{\sigma_X}$ – vectors in $\mathbb{R}^{1\times T}$, corresponding to the parameterization of the probability distribution which better represents the given input. If the VAE model was trained (mainly) with data describing the normal behavior of the monitored system, then the output for a non-anomalous input would not deviate from the mean $\boldsymbol{\mu_X}$ more than a specific integer $\boldsymbol{\alpha}$ times the standard deviation $\boldsymbol{\sigma_X}$. On the contrary, if the input presents an anomaly, the output would not belong to this normality region. The main goal of the VAE model is to learn a compressed representation of $\boldsymbol{X}$ in an unsupervised manner. This compressed representation $\boldsymbol{Z}$ is referred to as a latent variable, and it is learned by training the VAE to generate data that is similar to the input data. VAEs learn a probabilistic mapping between the input data and its latent variable, which allows to generate new data by sampling from the learned latent variable distribution.

Figure 2 depicts the encoder/decoder architecture used in FAE, which is an adaptation of DC-VAE's architecture, for the case of univariate time-series analysis. The FAE model functions as a univariate model trained on various series within a system simultaneously, treating them as distinct classes of series. Similar to the original DC-VAE version, FAE allows for monitoring of all time-series within a MTS process using a single model, albeit analyzing one time-series at a time. The architecture, based on dilated convolutional neural networks (DCNNs), is capable to exploit the temporal dependence of values for longer sequences. The main difference with DC-VAE is that the new architecture has to accommodate univariate input samples $\boldsymbol{X} \in \mathbb{R}^{1\times T}$, rather than multivariate ones. To maintain the concept of compression – i.e., the dimension of the latent space $\boldsymbol{Z}$ has to be lower than the input dimension of $\boldsymbol{X}$ – the latent space in FAE reduces dimensionality along the temporal dimension; in DC-VAE, the dimensionality reduction operates in the spatial dimension.

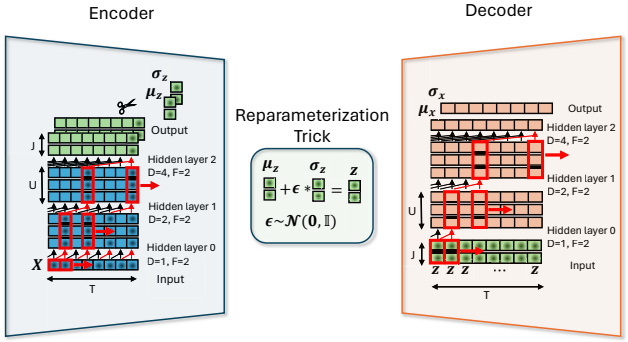


Figure 2. FAE's encoder/decoder architecture using causal dilated convolutions, implemented through a stack of 1D convolutional layers.

Using DCNNs forces to keep the sequence length T at the output of each hidden layer – referred to as $\boldsymbol{H}$ – thus $\boldsymbol{H} \in \mathbb{R}^{U\times T}$, where U is the number of filters in the layer. As shown in Figure 2, the dilation of each layer at the encoder increases exponentially as the network deepens, ensuring that each time t of the output at the final layer has information from all the previous times of $\boldsymbol{X}$ up to t, i.e., $[X_0, X_1, ..., X_t]$. This means that the last sample of the output at position $T-1$ contains information from the entire input sequence $\boldsymbol{X}$. Then, two layers in parallel with J filters of size one are applied at the output of the last hidden layer, bringing the output to a J-dimensional latent space. The output of the encoder results from keeping only the values at time $T-1$ at the output of these filters, resulting in vectors $\boldsymbol{\mu_Z}, \boldsymbol{\sigma_Z} \in \mathbb{R}^{J\times 1}$, which define the distribution in the latent space of the inputs $\boldsymbol{X}$, where $J < T$. Using the reparameterization trick [6], the latent vector $\boldsymbol{Z} \in \mathbb{R}^{J\times 1}$ is generated, corresponding to the encoding of observation $\boldsymbol{X}$, and is then fed into the decoder. The decoder remains the same as in DC-VAE, which is symmetric with respect to the encoder. However, given that its input requires a sequence of T values and not a single one, the input to the decoder is generated by repeating the vector $\boldsymbol{Z}$ for a total of T times, obtaining an input sample $\in \mathbb{R}^{J\times T}$. As a result, FAE's decoder extracts information from the same latent vector $\boldsymbol{Z}$ for each time t, to generate the output parameters $\boldsymbol{\mu_X}, \boldsymbol{\sigma_X} \in \mathbb{R}^{1\times T}$, which are used to evaluate deviations from the input observation $\boldsymbol{X}$. If the FAE model was trained (mainly) with data describing the normal behavior of the analyzed time-series, then the value of a non-anomalous sample $\boldsymbol{X}_t$ at time t would not deviate from the predicted mean $\boldsymbol{\mu_{X_t}}$ more than a specific integer α times the standard deviation $\boldsymbol{\sigma_{X_t}}$. On the contrary, if the sample is anomalous, it would not belong to the region determined by the predicted mean and standard deviation.

In terms of size of the architecture, an interesting characteristic of FAE is that its structure and number of layers is defined by the length T of the sliding window. In particular, the number of hidden layers N and the length of filters F are related through the dilation factor $d = F^h$ of the DCNNs, which grows exponentially with the layer depth $n \in [0, N-1]$. Subsequently, N is the minimum value that verifies: $T \leq 2 * F^{N-1}$. In the architectural example (cf. Figure 2), the window length is $T = 8$ and the filter length is $F = 2$, and the target is achieved by taking $N = 3$ hidden layers. This direct relationship between

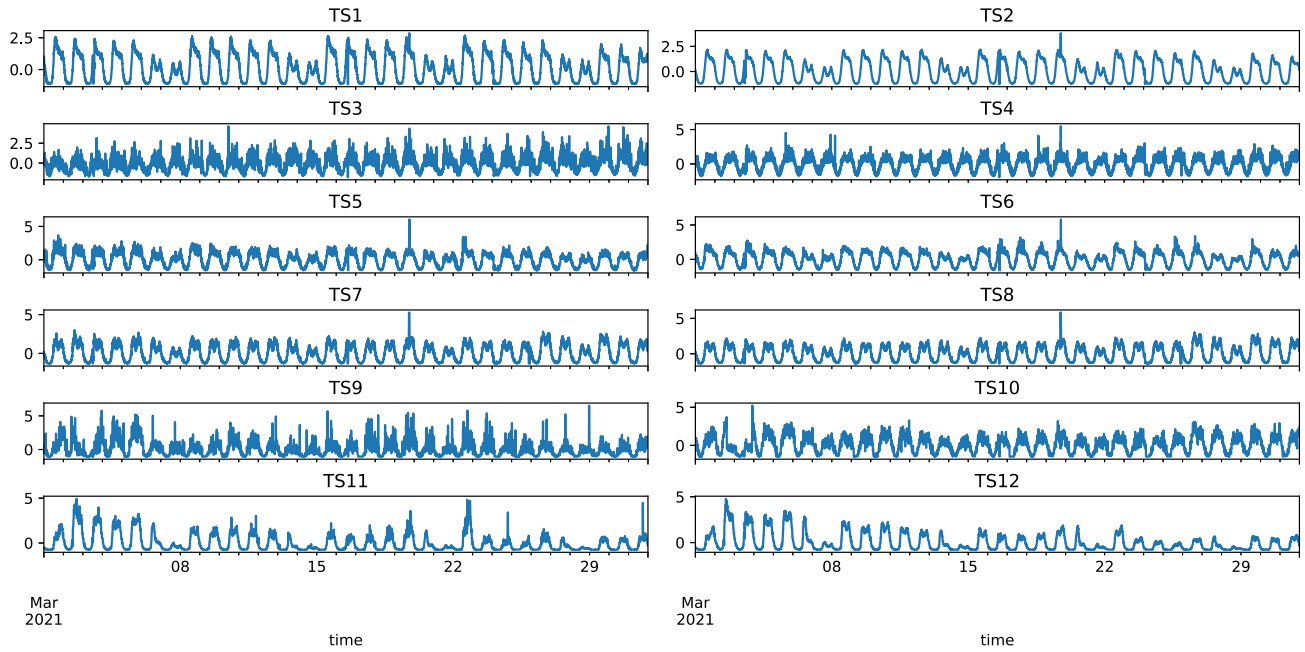


Figure 3. TELCO time-series, for one month worth of data (March 2021), sampled at a five minutes rate.

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. As a final reference of architectural complexity, for a relatively small FAE architecture, using $T = 256$ samples (less than one day of samples, at a 5' sampling-rate), the network exposes roughly half a million free parameters to train. Training FAE with a sufficiently large and heterogeneous training set, comprising multiple time-series of different characteristics, enhances its capability to generalize to unseen data and eventually to different domains.

4. FAE Time-Series Prediction in the Practice

We experiment with FAE in the analysis of an open MTS dataset arising from the monitoring of an operational mobile ISP, consisting of time-series with different structural properties. Referred to as the TELCO dataset [8], this *large-scale* – about 750 thousand samples, *long time-span* – seven months' worth of measurements (January 1st to July 31st, 2021) collected at a five-minutes scale, *multi-dimensional* – twelve different 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. The twelve time-series are typical data monitored in a mobile ISP, including the volume of data traffic, number of SMS messages, number and amount of prepaid data transfer fees, number and cost of calls, etc.

In this paper we focus on a more qualitative analysis of FAE's performance, focusing on its ability to properly track and reconstruct the different TELCO time-series. Figure 3 depicts a one-month example from the complete TELCO MTS dataset. Different time-series expose different behaviors, e.g., some of them are noisier (TS_3 and TS_9), others have lower dynamic ranges (TS_1), and some others show a smoother evolution (TS_2). All time-series exhibit daily seasonality, but some behave differently on weekends compared to workdays, while others show monthly trends either ascending or descending.

TABLE 1. GRID OF HYPERPARAMETERS USED IN THE MODEL CALIBRATION.

Hyperparameter	Grid Search Ranges	Best
T - sequence length	$\{128 - 512\}$, step=32	256
J - latent dimension	$\{16 - T/4\}$, step=16	48
γ - learning rate	$\{1e^{-5} - 5e^{-4}\}$	$6e^{-5}$
m - mini-batch size	$\{16 - 96\}$, step=16	32
U - number of filters	$\{16 - 128\}$, step=16	128

4.1. Hyperparameter Search and Training

One of the most important aspects when working with deep learning models is the search of model and training hyperparameters, along with the subsequent training of the model. Table 1 shows the grid used for the hyperparameter search, as well as the best values (smallest validation loss), identified by Tree-structured Parzen Estimator (TPE) search [44]. In total, 50 attempts were tested on the grid. In the table, T corresponds to the sequence length, and J is the dimensionality of the latent space. Training hyperparameters include the learning rate γ and the mini-batch size m. Finally, U is the number of filters 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 p. Considering the five minutes sampling rate of the time-series, the selected sequence length of $T = 256$ samples corresponds to a time window of 21hs and 20 minutes. The exact number of trainable parameters in the identified architecture is $p = 483.840$.

We split the full, 7-months dataset in three independent, time-ordered sub-sets, using measurements from

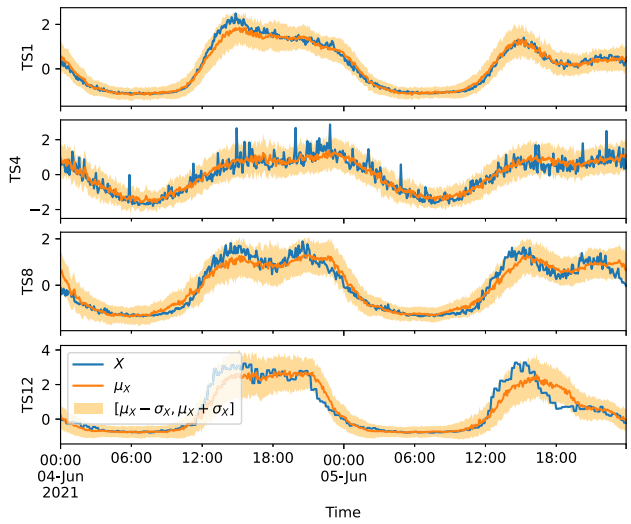


Figure 4. Predictions with fully-trained FAE (12 time-series) in two days of testing samples from June 2021 (Friday and Saturday).

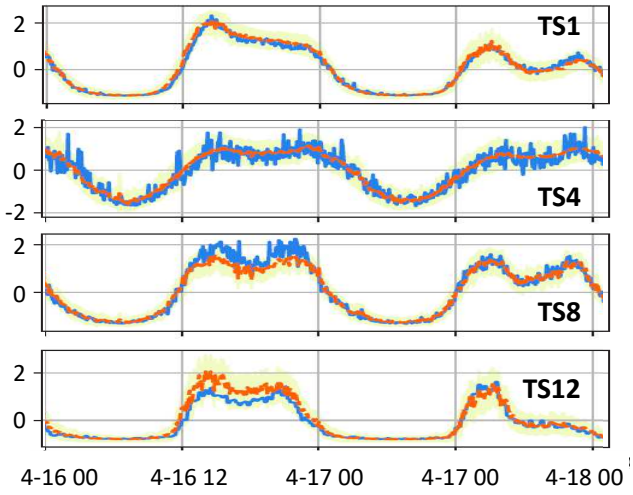


Figure 5. Predictions with DC-VAE in two days of validation samples from April 2021 (Friday and Saturday).

January to March for model training (3 months), April for model validation (1 month), and May to July for testing purposes (3 months). One of the disadvantages of the FAE model as compared to the multivariate DC-VAE model is the training time, and hence the time required for hyper-parameter search. The FAE model requires approximately five times more training time than DC-VAE.

4.2. FAE Modeling Performance

We evaluate the prediction performance of FAE in samples from the testing set, considering training on the full three months of data, for the 12 time-series, i.e., more than 300.000 samples. Figure 4 depicts the resulting predictions $\boldsymbol{\mu_{X_t}}$ and $\boldsymbol{\sigma_{X_t}}$ for two days of testing samples $\boldsymbol{X}_t$ from June 2021, for four representative time-series, including TS_1, TS_4, TS_8, and TS_{12}. To add more variability, we consider a working day (Friday 4th) and a weekend day (Saturday 5th). FAE can properly track different types of behavior in the time-series, including the strong seasonal daily component, but also the operation during workdays and weekends, clearly visible in TS_{12}. Interesting to note is how different periods of time-series variability result in more or less tight normal-operation regions estimated by FAE, as defined by $\boldsymbol{\sigma_{X_t}}$.

For the sake of completeness and comparison, Figure 5 depicts the predictions obtained by the former DC-VAE multivariate model in the same four time-series, using a different time period in April – in this case from the validation test – from Friday 16th till Saturday 17th. Results are similar, but in particular for TS_{12}, DC-VAE can better capture the drop observed on Saturday evening, exploiting the strong spatial correlation observed on Saturday between TS_{12}, TS_{11}, and both TS_1 and TS_2 (cf. Figure 3). Nevertheless, note that FAE predictions are slightly better than DC-VAE's for TS_8 and TS_{12} on Friday.

To better understand the modeling capabilities of FAE, we focus on the analysis of the latent space $\boldsymbol{Z}$, for the different time-series and the different times of the analysis. Recall that the latent space set for FAE in this analysis is $J = 48$; to easily visualize $\boldsymbol{Z}$ as a two- or

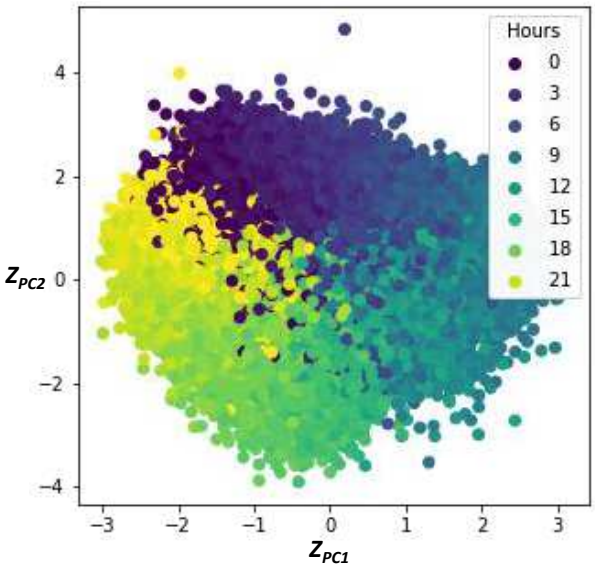
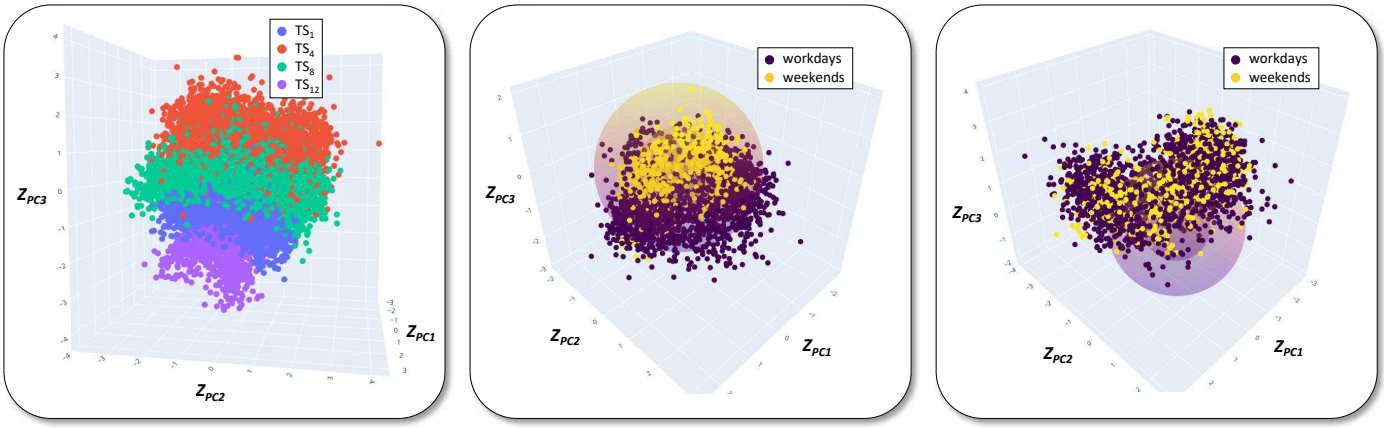


Figure 6. Latent space representation – temporal evolution.

three-dimensional space, we apply standard PCA analysis, and study the top-two and top-three principal components $\boldsymbol{Z_{PCi,...i=1,2,3}}$. Figure 6 shows the latent representation of each sample $\boldsymbol{X}_t$ for a single day, for all the 12 time-series, depicting the encoded samples in colors, each color representing a different three-hour period of the day. Plotting the first two principal components shows that each hour period maps to a certain position in the latent space, and interestingly, the direction mirrors the progression of hours on a clock, ordered continuously by hour of the day.

Figure 7(a) depicts now the first three principal components, displaying the previously analyzed four time-series TS_1, TS_4, TS_8, and TS_{12}, independently. FAE maps each time-series to a different region of the latent space, showing it can properly differentiate among different time-series characteristics. Closely located samples from different time-series exhibit similar behaviors in the time-series space. For example, time-series TS_1 and TS_{12} are closer



(a) Latent representations per time-series. (b) Workdays vs weekends, TS_1. (c) Workdays vs weekends, TS_4.

Figure 7. Latent space representation, (a) per different time-series (TS_1, TS_4, TS_8, TS_{12}), and (b,c) specifically for TS_1 and TS_4 in a temporal basis, considering workdays (purple) and weekends (yellow).

to the center of the latent space, and both exhibit a similar behavior (cf. Figure 3), with marked differences between weekends and workdays. On the other hand, time-series TS_4 and TS_8 have a similar temporal behavior without marked variations between workdays and weekends, and are located together and farther from the center of Z.

The difference between time-series for workdays and weekends is further explored in Figures 7(b,c), where workday samples are displayed in purple color, and weekends in yellow, for (b) time-series TS_1 and (c) time-series TS_4. For reference, the plots include two spheres with radius one and two, reflecting the expected Gaussian distribution of the latent space. The difference between workdays and weekends are clear for TS_1, with weekends clustering closer to the center (smaller dynamic range, cf. Figure 3), and workdays located closer at the sphere borders (bigger dynamic range). This difference between workdays and weekends is not observed for TS_4.

Finally, note in Figure 3 how time-series TS_{11} and TS_{12} exhibit a downtrend behavior during the month, which can also be observed in the latent-space. Figure 8 depicts the latent representation of time-series TS_{12}, where days of the month are differentiated by color, from day 1 in purple to day 31 in yellow. As the month goes by, the representation in the latent space moves from the outside borders closer towards the center.

To wrap-up these preliminary evaluations, we observe how FAE can properly capture and differentiate among the different temporal behaviors present in the time-series used for training, suggesting a sufficiently expressive model and architecture to model a large and heterogeneous dataset of time-series. In addition, the visual analysis of the latent representations in FAE evidences how VAEs – despite their generative nature – are rather transparent in their operation and behavior, making interpretation and analysis simpler and more human-friendly. This is indeed a strong advantage of VAEs as a powerful yet explainable generative AI model, as compared to modern generative AI approaches, which operate in a more black-box manner.

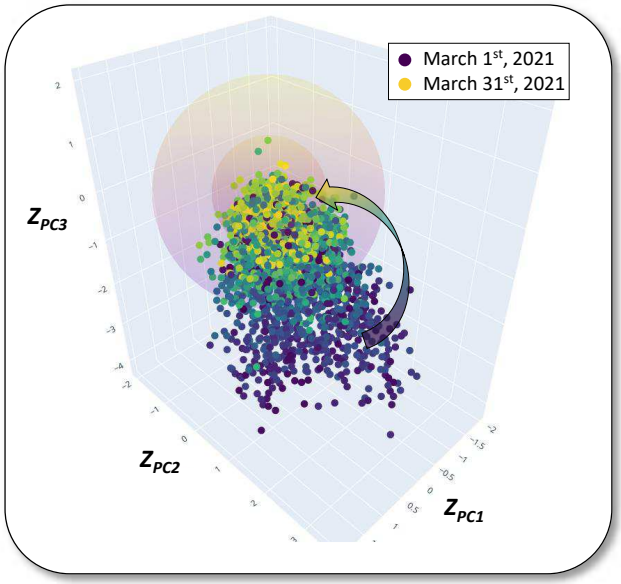
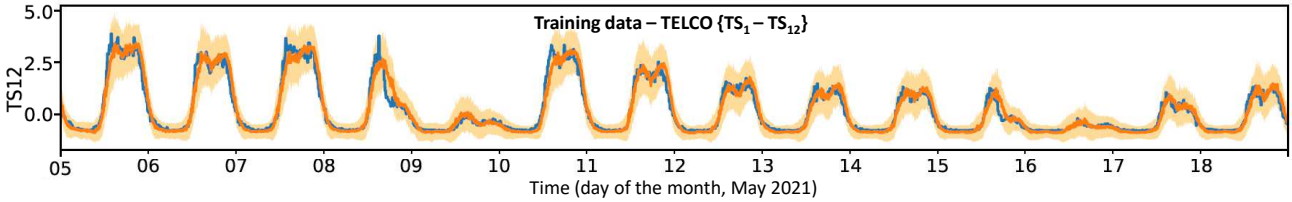


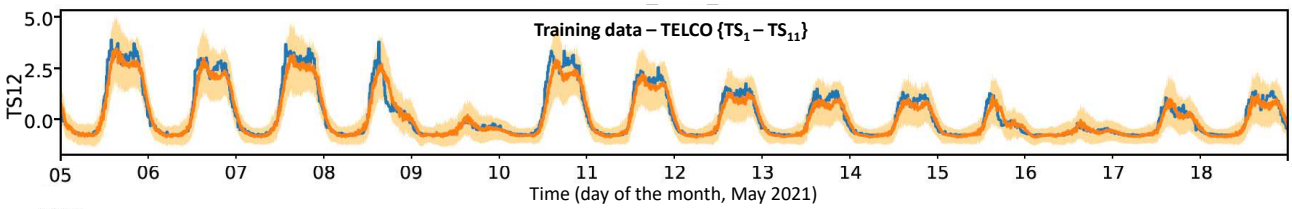
Figure 8. Latent space representation for TS_{12}, in a daily basis – from day 1 in purple to day 31 in yellow, for the full month of March 2021.

4.3. Zero-shot Modeling Behavior

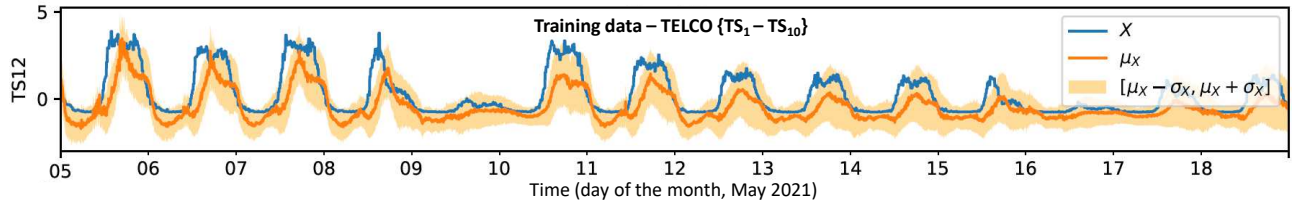
We investigate now the performance of FAE in a zero-shot setting, testing the model for time-series not seen at training time. We focus the analysis on TS_{12}, due to its combined seasonality and particular temporal trend, as well as its strong correlated behavior to TS_{11} (cf. Figure 3). We train FAE on three different training datasets from TELCO, using the same 3/1/3-months temporal split for training/validation/testing, but considering a different number of time-series TS_i. The first model uses all 12 time-series – we refer to it as *full-FAE*; the second model considers a zero-shot setting for TS_{12}, with a training dataset which includes time-series TS_1 to TS_{11}, leaving out all samples from TS_{12}; given the strong temporal correlation between TS_{12} to TS_{11}, we also train a third model leaving out all samples from TS_{11} and TS_{12}, i.e., training on time-series TS_1 to TS_{10}. The full FAE model mimics a situation where we pre-train with a sufficiently



(a) FAE predictions for TS_{12}, with full-FAE (12 time-series).



(b) FAE predictions for TS_{12}, with FAE trained without TS_{12}.



(c) FAE predictions for TS_{12}, with FAE trained without TS_{11} and TS_{12}.

Figure 9. Zero-shot modeling experimentation, predicting TS_{12} for two weeks in the testing dataset (May 2021). (a) FAE is trained on the full, 12 time-series training set – modeling performance is optimal. (b) FAE is trained on 11 time-series, leaving out TS_{12} – performance remains almost unchanged. (c) FAE is trained on 10 time-series, leaving out TS_{11} and TS_{12} – modeling performance is impacted.

large and heterogeneous dataset which covers the statistical behavior of the downstream data – i.e., a model that *has seen it all*. The other two models mimic two different levels of zero-shot learning: the former represents a pure zero-shot setting for TS_{12}, where the pre-trained model has nevertheless observed a similar statistical behavior in a different time-series, i.e., TS_{11} – in particular, it has seen both the seasonality and the monthly trend behaviors; the latter represents a more challenging setting, where the pre-trained model has not seen the monthly trend behavior, which is not present in TS_1 to TS_{10}.

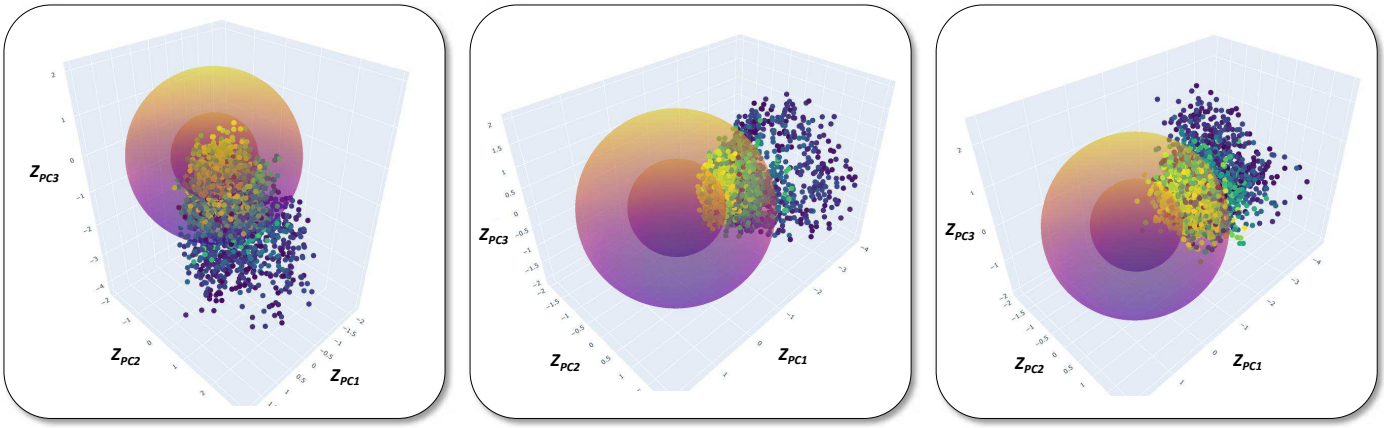
Figure 9 presents the prediction performance of the three models, when applied to two weeks of TS_{12} samples, from May 5 to May 19, 2021. In Figure 9(a), the modeling performance for full-FAE is optimal, as it can properly track the different behaviors and patterns in the time-series, similarly to Figure 4. A similar performance is observed in Figure 9(b) for the second model, which learns the characteristics of TS_{12} at training time, from TS_{11}. Not surprisingly, the performance of the third model in Figure 9(c) is significantly worse than for the other two models, given the lack of a similar temporal pattern in the training data. To some extent, there is an identification with the patterns observed in time-series TS_1 – note how the daily sharp peaks are exacerbated – which is coherent with their close representations in the latent space (cf. Figure 7(a)). Nevertheless, it somehow manages to capture and track the monthly downtrend, even without previous evidence of it.

To conclude, Figure 10 shows the latent representation of the TS_{12} test samples for the three FAE pre-trained models, where colors represent the different days of the analysis window, going from day 5 in purple to day 19 in yellow. Full-FAE encoded samples form a sort of cone in the latent space in Figure 10(a), where the base (purple and blue) represents the first days of the month and the tip – pointing towards the center of the latent space – represents the days towards end of the month. Figure 10(b) shows a similar cone-shape for the samples encoded by the second pre-trained model, but this time, the tip of the cone has moved away from the center. Finally, while Figure 10(c) shows a similar distribution of samples, with yellow and clearer colors closer to the center and darker ones at the periphery of the central sphere, the regular cone-shape observed before is no longer well-defined, evidencing a different mapping behavior of the model.

5. Discussion and Limitations of FAE

Selecting VAEs for our foundation model exploration has both benefits and limitations, which we briefly discuss next. On the benefits side, we have shown how easy it is to explore and interpret the functioning of the encoding and the behavior of the encoded time-series in the latent space, making the model transparent and easy to tame, particularly for training. While VAEs may struggle with capturing long-range dependencies in the data, we have shown how the integration of DCNNs as part of the encoding/decoding networks enables tracking multiple different temporal behaviors in the time-series, from seasonality to long-term trends.

FAE shows potential to be a strong foundation model for time-series analysis, but so far, we have only trained and tested the model with a single, large-scale dataset



(a) Full-FAE (12 time-series). (b) FAE trained without TS_{12}. (c) FAE trained without TS_{11} and TS_{12}.

Figure 10. Latent space representation for TS_{12}, with different FAE pre-trained models. Colors represent the different days of the analysis window, going from day 5 in purple to day 19 in yellow, for the full month of March 2021.

from a particular domain, and thus still require further assessment. Indeed, thorough testing and validation would be necessary to assess FAE's performance and generalization capabilities in different scenarios. Ultimately, the effectiveness of FAE as a foundation model would depend on its performance in various real-world scenarios and its ability to generalize to different datasets and different domains.

While powerful, VAEs may have limitations in terms of expressiveness when tasked with learning and mapping a large-scale number of highly heterogeneous time-series data. VAEs operate under the assumption of a latent variable space with a simple distribution, which may not always capture the intricate and diverse characteristics of more complex and diverse time-series data.

We note that the performance of FAE in generalization and zero-shot learning tasks can be affected by factors such as the complexity of the data, the dimensionality of the latent space, the choice of the encoder/decoder architecture, and the quality and diversity of the training data. Additionally, while VAEs can capture global structure in the data distribution, they may not always capture fine-grained details or handle complex data distributions as effectively as other generative models. Finally, in terms of scalability, FAE's performance may degrade with extremely large datasets or highly heterogeneous data, as the model complexity may need to increase significantly.

6. Concluding Remarks

We have introduced FAE, a novel approach for time-series modeling, motivated by the performance realized by large pre-trained foundation models in different domains. FAE targets the detection of anomalies in univariate time-series data, leveraging VAEs and DCNNs to pre-train on large-scale, heterogeneous time-series datasets, potentially enabling to properly model and track a baseline for normal operation, even on unseen datasets.

The preliminary assessment of FAE's performance has shown promising results. In particular, we have provided evidence of FAE's capabilities to capture and distinguish various temporal behaviors within the training time-series, demonstrating a promising capacity to model large and heterogeneous datasets effectively. The interpretability of FAE's latent representations showcased VAEs' transparency in operation, facilitating simpler analysis and interpretation compared to black-box generative AI models.

Our exploration extended to the zero-shot learning scenario, where FAE's performance on unseen time-series was assessed. We tested FAE in three settings, ranging from optimal modeling performance to more challenging scenarios. While FAE performs properly in tracking different behaviors and patterns in the time-series, even in the absence of previous evidence, there is room for further improvement, especially in capturing previously unseen temporal trends.

These initial findings underscore FAE's potential as a feasible foundation model for time-series analysis. However, it is essential to note that our evaluation was limited to a single large-scale dataset within a specific domain. As part of our ongoing work, we are focusing on comprehensively testing FAE's performance and generalization capabilities across diverse scenarios, using much larger and heterogeneous time-series datasets for training, considering other domains beyond networking.

Acknowledgment

This work has been partially supported by the Austrian FFG ICT-of-the-Future project *DynAISEC – Adaptive AI/ML for Dynamic Cybersecurity Systems* – ID 887504, and by the Uruguayan CSIC project with reference CSIC-I+D-22520220100371UD *Generalization and Domain Adaptation in Time-Series Anomaly Detection*.

References

[1] 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.

[2] 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.

[3] J. Yoon, D. Jarrett, and M. van der Schaar, "Time-series Generative Adversarial Networks," in *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[4] 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

[5] G. García González, S. Martinez Tagliafico, A. Fernández, G. Gómez, J. Acuña, and P. Casas, "One Model to Find Them All – Deep Learning for Multivariate Time-Series Anomaly Detection in Mobile Network Data," *IEEE Transactions on Network and Service Management*, pp. 1–1, 2023.

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

[7] G. García González, P. Casas, and A. Fernández, "Fake it till you Detect it: Continual Anomaly Detection in Multivariate Time-Series using Generative AI," in *2023 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, 2023, pp. 558–566.

[8] 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

[9] 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.

[10] 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.

[11] 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

[12] 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.

[13] 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.

[14] 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.

[15] 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.

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

[17] 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.

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

[19] 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.

[20] 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.

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

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

[23] 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*, 2018.

[24] 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 Dig Audio Effects*, 2019, pp. 1–8.

[25] 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.

[26] 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.

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

[28] 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.

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

[30] 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.

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

[32] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. Kaiser, and I. Polosukhin, "Attention is All you Need," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[33] B. Lim, S. Arik, N. Loeff, and T. Pfister, "Temporal Fusion Transformers for Interpretable Multi-horizon Time Series Forecasting," *International Journal of Forecasting*, vol. 37, no. 4, pp. 1748–1764, 2021.

[34] K. C. Chen, L. Dicker, C. Eisenach, and D. Madeka, "MQTransformer: Multi-horizon Forecasts with Context Dependent Attention and Optimal Bregman Volatility," in *KDD 2022 Workshop on Mining and Learning from Time Series – Deep Forecasting: Models, Interpretability, and Applications*, 2022.

[35] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, "Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting," in *Thirty-Fifth AAAI Conference on Artificial Intelligence*. AAAI Press, 2021, pp. 11 106–11 115. [Online]. Available: https://doi.org/10.1609/aaai.v35i12.17325

[36] H. Wu, J. Xu, J. Wang, and M. Long, "Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting," *CoRR*, vol. abs/2106.13008, 2021. [Online]. Available: https://arxiv.org/abs/2106.13008

[37] T. Zhou, Z. Ma, Q. Wen, X. Wang, L. Sun, and R. Jin, "FEDformer: Frequency Enhanced Decomposed Transformer for Long-term Series Forecasting," *CoRR*, vol. abs/2201.12740, 2022. [Online]. Available: https://arxiv.org/abs/2201.12740

[38] A. Garza and M. Mergenthaler-Canseco, "TimeGPT-1," 2023.

[39] H. Xue and F. D. Salim, "PromptCast: A New Prompt-Based Learning Paradigm for Time Series Forecasting," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–14, 2023.

[40] N. Gruver, M. Finzi, S. Qiu, and A. G. Wilson, "Large Language Models Are Zero-Shot Time Series Forecasters," 2023.

[41] A. Das, W. Kong, R. Sen, and Y. Zhou, "A Decoder-only Foundation Model for Time-series Forecasting," 2024.

[42] K. Rasul, A. Ashok, A. R. Williams, H. Ghonia, R. Bhagwatkar, A. Khorasani, M. J. D. Bayazi, G. Adamopoulos, R. Riachi, N. Hassen, M. Biloš, S. Garg, A. Schneider, N. Chapados, A. Drouin, V. Zantedeschi, Y. Nevmyvaka, and I. Rish, "Lag-Llama: Towards Foundation Models for Probabilistic Time Series Forecasting," 2024.

[43] M. Jin, S. Wang, L. Ma, Z. Chu, J. Y. Zhang, X. Shi, P.-Y. Chen, Y. Liang, Y.-F. Li, S. Pan, and Q. Wen, "Time-LLM: Time series forecasting by reprogramming large language models," in *International Conference on Learning Representations (ICLR)*, 2024.

[44] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for Hyper-parameter Optimization," *Advances in neural information processing systems*, vol. 24, 2011.