Generative sparse data augmentation dealing with performance evaluation

Manuel Sánchez-Laguardia[1][0009−0005−0149−2726], Gastón García González[1][0009−0002−6652−7713], Emilio Martinez[1][0009−0005−8418−0006], Sergio Martinez, Alicia Fernández[1][0000−0003−2905−2210], and Gabriel Gómez[1][0000−0002−4213−8791]

Facultad de Ingeniería, Universidad de la República, Uruguay {msanchez, gastong, emartinez, sematag, alicia, ggomez}@fing.edu.uy
https://www.fing.edu.uy/

Abstract. Data augmentation has become a critical strategy for enhancing the generalization ability of deep learning models, particularly in domains characterized by limited or irregular data. In the context of sparse and intermittent demand time series, the lack of extensive datasets makes synthetic data generation especially valuable. Building on our previous work introducing the ASTELCO dataset—an augmented version of real-world e-commerce demand data—this study proposes a set of classical quantitative metrics for assessing the quality of synthetic time series generated by deep generative models.We assess three data augmentation methods using these metrics and make both the code and datasets publicly available to support reproducibility and further research. We also highlight the relevance and interpretability of these metrics in the evaluation of generative performance, particularly in sparsity-aware applications.

Keywords: Sparse Time Series · Generative Models · Data Augmentation · Performance metrics.

1 Introduction

Augmenting time series data has proven valuable for improving the generalization of deep learning models, especially in tasks such as classification, anomaly detection, and forecasting [11, 18]. This is particularly important when working with sparse or irregular data, where limited observations make it difficult to capture temporal dependencies and distributional patterns. In multi-series contexts, having access to datasets with sufficient granularity and history is essential for modeling relationships between individual series and their overall dynamics [11]. Diverse datasets have also played a key role in training foundation models capable of generalizing across domains and performing zero-shot predictions [9].

Despite the recent increase in research on time series analysis, public access to databases derived from monitoring the operation of real systems remains limited, particularly in the case of sparse or intermittent demand series.

Sparse time series are characterized by non-zero values that appear sporadically in time, with the remaining of the values being 0. This inherent property, coupled with the variability in the occurrence patterns across different series, poses significant challenges for forecasting[12]. In anomaly detection, such series present an additional difficulty for detection algorithms, which often exhibit reduced performance compared to more active series[14].

The lack of more research focusing on this type of data is likely due to the limited availability of such data for training and evaluating models.

Previous studies have demonstrated the effectiveness of Generative Adversarial Networks (GANs) and Variational Auto-Encoders (VAEs) in generating synthetic data from real data. For instance, [19] proposed a GAN-based architecture and a comprehensive performance evaluation method, which we will consider as a primary reference for our work. The study evaluated performance using four metrics. Similarly, [3] introduced a VAE-based architecture, which is compared to previous metrics and other architectures[5], showing comparable quantitative performance.

In [15] we addressed the challenge of generating a synthetic sparse dataset through data augmentation techniques, with the aim of providing a novel dataset to the academic community. We contributed with the publication of a database with sparse intermittent demand series, ASTELCO, generated from real data, STELCO. It also included the performance comparison of different generative models with metrics proposed in [19].

This research benefited from the collaboration with the e-commerce division of a mobile Internet Service Provider (ISP), which supplied a real diverse dataset, STELCO, employed in the synthetic generation process.

This work extends the analysis, presented in the 14th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2025)[15], with new evaluation metrics that seek to provide greater consistency between visual results and quantitative results.

The following sections describe the performance metrics proposed. Then, the characteristics of the models used to generate the data augmentation based on TimeGAN and DC-VAE are briefly described. Finally, the experiments, results, and conclusions of the generated databases are presented.

2 Performance evaluation metrics for Sparse Data Augmentation

In this section, we present the evaluation metrics used to assess the performance of the different generative methods used.

2.1 Sparsity

The first is a comparison of the sparsity between the original dataset and the generated ones.

$$sparsity = 100\left(1 - \frac{NZ(A)}{T(A)}\right). \quad (1)$$

The measure presented in Equation (1) was employed to assess the sparsity of the series, where $NZ(A)$ denotes the number of nonzero values in the series A, and $T(A)$ is the total number of values in A, which equals the length of the series.

In Table 1, a comparison is shown between the sparsity of a set of publicly available databases[6] and our STELCO database released to the community in [15]. This dataset comprises records of invoices generated through the ISP's online commerce platform, encompassing various payment methods. Notably, certain payment methods show high levels of activity, whereas others show very little, thereby introducing a diverse range of behaviors to the whole.

Table 1: Comparative table of databases with sparse series. Table from [15].

Database name	Time Interval	Number of series	Total number of samples	Sparsity	Description
Online Retail	1 min - 11 days Mean = 30 min	1	17,914	70.30%	Transactions for an online retail business in the UK.
Car Parts	1 month	2,674	136,374	75.90%	Demand for vehicle spare parts.
Entropy 1	1 day - 15 days	1,200	132,579	35.65 %	Demand for heavy machinery spare parts in China.
Entropy 2	1 month	57	1,938	41.90 %	Demand for parts from a manufacturing company in China.
STELCO (ours)	10 ns to 3 days Mean = 2 min	18	287,734	67.42 %	Invoicing amount in e-commerce platform.

In Table 1, it can be noted that the STELCO dataset has the shortest time interval and the largest number of samples.

2.2 Diversity, Fidelity, Usefulness

The following metrics used were inspired by RCGAN[5] and TimeGAN[19]. These articles present methodologies to evaluate the quality of the generated data based on three criteria: *diversity*: samples should be distributed in such a way that they cover the actual data; *fidelity*: the samples should be indistinguishable from the real data; and *usefulness*: the samples should be as useful as the actual data when used for the same predictive purposes.

Visual analysis: PCA y t-SNE Two visual analysis methods are used: Principal Component Analysis (PCA) and t-distributed Stochastic Neighbor Embedding (t-SNE). These techniques allow for the visualization of the extent to which the distribution of the generated samples resembles that of the original data within a two-dimensional space. This approach facilitates a qualitative assessment of the *diversity* of the generated samples.

Discriminative Score is proposed as a metric to estimate *fidelity*.

It consists in the evaluation of a classifier trained to differentiate between real and generated data sequences. This training is conducted in a supervised manner, with the original and generated data labeled beforehand. Then, the classification error is used as a quantitative evaluation of *fidelity*. The metric defined as Discriminative Score is presented in Equation (2).

$$Discriminative\ Score = |0.5 - accuracy|. \tag{2}$$

The ideal scenario, which would minimize the discriminative score, occurs when the classification accuracy is 0.5. In this case, the classifier would perceive all incoming real data as genuine and all synthetic data also as real. Therefore, half of the data would be accurately classified (the real instances) and the other half would be misclassified (the synthetic instances). This outcome suggests that the synthetic data is indistinguishable from the real data.

Predictive Score is proposed as a metric to estimate *usefulness*.

A sequence prediction model is trained to forecast the time vectors of the next step in each input sequence. Specifically, for a sequence of data ranging from 0 to T, the objective is to predict the value of the series at time T+1. The model is trained on the generated data and evaluated on the original data. Its performance is quantified using the mean absolute error (MAE) as defined in Equation (3), thereby providing a quantitative assessment of *usefulness*.

$$Predictive\ Score = MAE = \sum_{i=1}^{D} |x_i - y_i| \tag{3}$$

where x and y are series of dimension D, corresponding to the predictions and the real data.

2.3 KL, JS, Wasserstein, KS and MMD

However, in [15] we found that discriminative and predictive score metrics depend on how an LSTM network behaves to get a score. Additionally, the predictive score, which involves predicting the next value of a sequence using an LSTM, would give fairly good results just predicting 0, given that our data is sparse. This is why we decided to evaluate our datasets with more well-known metrics: Kullback–Leibler (KL) divergence[2], Jensen–Shannon (JS) divergence[4], Wasserstein distance[16], Kolmogorov-Smirnov (KS) test[13], and Maximum Mean Discrepancy (MMD)[10]. These metrics provide complementary perspectives on the

similarity between the generated and original distributions, offering a more robust evaluation of generative quality.

The Kullback–Leibler divergence measures how much a model distribution Q differs from a true probability distribution P. It is particularly sensitive to discrepancies in the tails of the distribution, making it useful for detecting whether a generative model captures rare events in sparse time series. Mathematically:

$$D_{KL}(P||Q) = \sum_{x \in \mathcal{X}} P(x) \log \frac{P(x)}{Q(x)}. \tag{4}$$

The Jensen–Shannon divergence is based on KL divergence but is symmetric and always finite. Its stability makes it suitable for sparse regions and extreme zeros, providing a more balanced comparison:

$$JSD(P||Q) = \frac{1}{2} D_{KL}(P||M) + \frac{1}{2} D_{KL}(Q||M), \tag{5}$$

where $M = \frac{1}{2}(P + Q)$.

In the experiments, both P and Q were estimated from the normalized histograms of the real and generated data. This procedure was applied consistently for the computation of both the KL and the JS divergences.

The Wasserstein distance, also known as the Earth Mover's Distance, measures the minimal cost of transporting probability mass from the real distribution P_r to the generated distribution P_g. It is defined as the infimum, over all possible couplings γ with marginals P_r and P_g, of the expected distance between paired samples. It is robust to small localized deviations, such as sporadic peaks in sparse time series:

$$W(P_r, P_g) = \inf_{\gamma \sim \Pi(P_r, P_g)} \mathbb{E}_{(x,y)\sim\gamma} \left[\|x - y\| \right]. \tag{6}$$

The Kolmogorov–Smirnov test measures the maximum discrepancy between the empirical cumulative distribution of a sample and the reference distribution. Its interpretable, nonparametric criterion makes it suitable for quick checks of distributional similarity.

Finally, the Maximum Mean Discrepancy (MMD) uses kernel methods to detect global differences between high-dimensional or complex distributions, which is particularly valuable for multivariate sparse time series without requiring density estimation.

3 Data Augmentation Models Evaluated

To generate synthetic data from the utilized dataset, a comprehensive analysis of various existing methods was conducted, with the objective of implementing these techniques in the context of sparse time series[11, 17]. Among the methods employed in this study were TimeGAN and DC-VAE.

3.1 GAN-based generation

TimeGAN[19] is a method rooted in Generative Adversarial Networks (GANs), specifically designed for the generation of time series data. This model comprises a generator tasked with producing new synthetic data, which attempts to deceive a discriminator that functions to distinguish between real and fictitious data. GANs have demonstrated strong performance not only in time series generation [1] but also in other domains, such as image generation.

A distinctive feature of TimeGAN is its incorporation of an additional embedding network that facilitates a reversible mapping between features and latent representations, thereby addressing the challenges posed by the high dimensionality of the GAN's latent space. The model employs three loss functions: one unsupervised loss associated with the GAN, another associated with the embedding network, and a supervised step-wise loss. The supervised step-wise loss utilizes real data as a reference, promoting the model's ability to capture the temporal sequential dynamics inherent in the data. This loss is minimized through the joint training of the generation and embedding networks.

3.2 VAE-based generation

TimeVAE[3] is a method used for synthetic generation of time-series based on Variational Auto-Encoders (VAEs). They propose an interpretable VAE architecture where they present two blocks: Trend and Seasonality, that get added to the decoder in order to add specific temporal structures to the decoding process. Thus, the output from the decoder results in the element-wise summation of the trend block output, seasonality block outputs and the residual base decoder output.

DC-VAE[7] is a method used for anomaly detection in time series, which takes advantage of convolutional neural networks (CNN) and variational auto-encoders (VAE). DC-VAE detects anomalies in time series data by 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. This method is based on the reconstruction of time series and is not used as a generative method like TimeGAN. However, we wanted to test its performance in generative tasks such as this one.

4 Complete time series synthesis

A common factor with the models evaluated is that the generation of data is done on a window-by-window basis. This inhibits the models ability to reconstruct the dynamics of the original series sample by sample. The lack of temporal coherence between windows undermines their concatenation, making it challenging to establish continuity. The correlation between consecutive windows depends on

independently sampled latent vectors, which do not guarantee temporal proximity.

In this case, given the sparsity of the time series analyzed in this study, it would be worthwhile to investigate whether retaining only the last value from each window and concatenating them could yield an entire synthetic time series, that not only matches the input data in length, but also in its temporal dynamics.

This approach would not work for continuous time-series since each window presents a specific dynamic that would not be so easily concatenated. An example of windows from a continuous time series from the TELCO[8] dataset is illustrated in Figure 3a. However, in sparse series such as STELCO, given the low probability of occurrence of peaks, it could be argued that their concatenation could yield an entirely new time series that preserves the original distribution of the data.

5 Experiments and Results

5.1 Dataset Description

The experiments that follow are reported using the STELCO dataset. This dataset comprises transactions recorded on an online commerce platform. Consequently, in cases where no transactions occur, no corresponding records are generated. Due to the transactional nature of the dataset, no null values were present in the actual data.

To address this issue, a resampling procedure, using the mean time difference between samples, was implemented prior to computation of sparsity for datasets exhibiting this characteristic. This approach effectively introduced null values into the dataset, allowing for a more meaningful calculation of sparsity.

To facilitate the subsequent analysis of our data, three groups of series were formed, defining them from the lowest to the highest volume of transactions. The first group (*low*) contains the series with the lowest number of transactions, the second group (*mid*) series with an average volume of transactions and the third group (*high*) series with the highest volume of transactions. To match the number of values in each group with an appropriate sampling frequency, we chose to resample the groups at intervals of 1 hour, 5 minutes, and 1 minute, respectively. Thus, the number of values varied for each group accordingly: 625 values for the *low* dataset, 7,600 values for the *mid* dataset and 38,000 values for the *high* dataset.

All the series in our set were standardized in order to preserve the confidentiality of the data. With these three sub-groups of series, the analyses presented below were carried out. An example of a series from each group is shown in Figure 1.

5.2 Generative Models Configuration and visual results

TimeGAN The initial results using TimeGAN were obtained for the subset designated as *low* employing model parameters of 10,000 epochs and a sequence

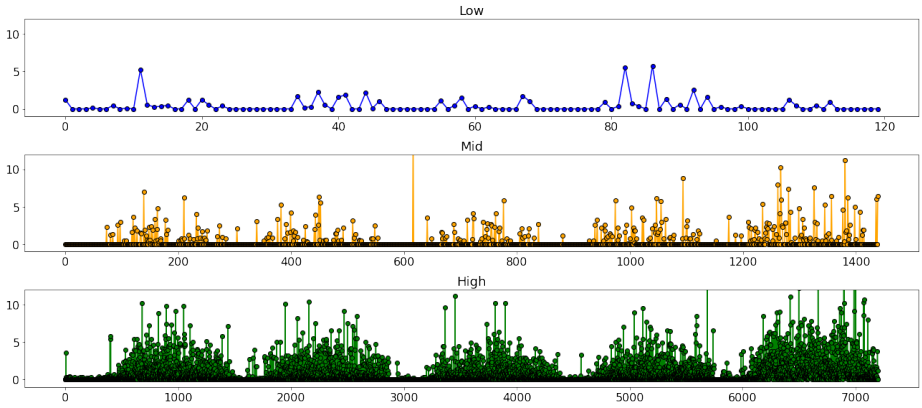


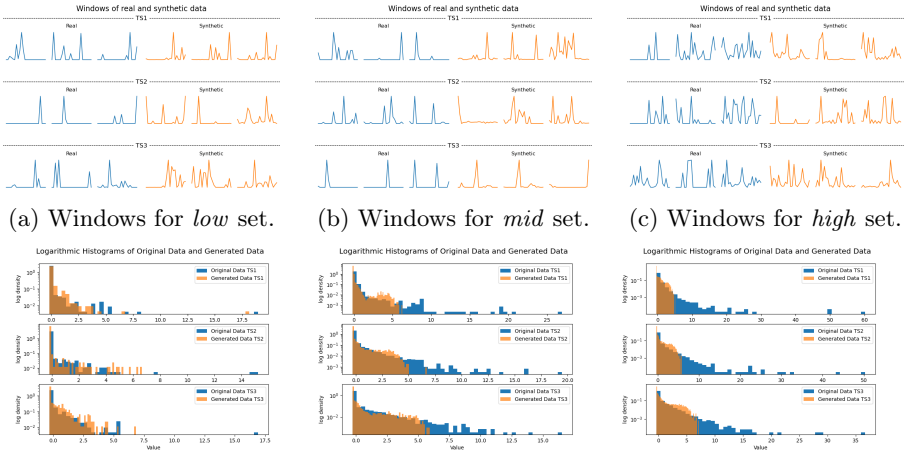
Fig. 1: Examples of five-day segments from distinct time series in our dataset, sampled at different temporal resolutions: 1 hour (Low), 5 minutes (Mid), and 1 minute (High), respectively.

length of 24. Figures 2a and 2d illustrate windows of the time series over a specified period of time, along with histograms depicting the distributions of both the original and synthetic data. The synthetic data demonstrate a distribution that closely resembles that of the real data; however, comprehensive performance evaluations will be conducted below.

The second experiment employing TimeGAN was conducted on the subset designated as *mid*, maintaining a sequence length of 24 while increasing the number of epochs to 30,000, due to the larger volume of input data. Figures 2b and 2e show the plots and histograms of the original and synthetic data, respectively. In this case, it is seen that some of the peaks of higher values are lost and are not generated in the synthetic data.

The third experiment using TimeGAN was conducted on the *high* subset, employing the same sequence length of 24 but 50,000 epochs, given the larger volume of input data compared to the other subsets. The window plots and histograms of both the original and synthetic data are illustrated in Figures 2c and 2f, respectively. Although the generated windows appear to align closely with the original data, the histograms reveal a higher density of non-zero values in the synthetic data than in the original. Furthermore, the largest values are completely absent in the generated dataset. Future investigations could benefit from a hyperparameter search to explore the effects of varying window lengths and the number of iterations on the generation process.

TimeVAE To perform our tests, we used the interpretable TimeVAE architecture with one Trend block, one seasonality block and the base residual decoder. The trend block was selected with 4 trend polynomials ($p = 4$). The seasonality block varied for each dataset: with $m = 7$ and d equal to the duration of a day



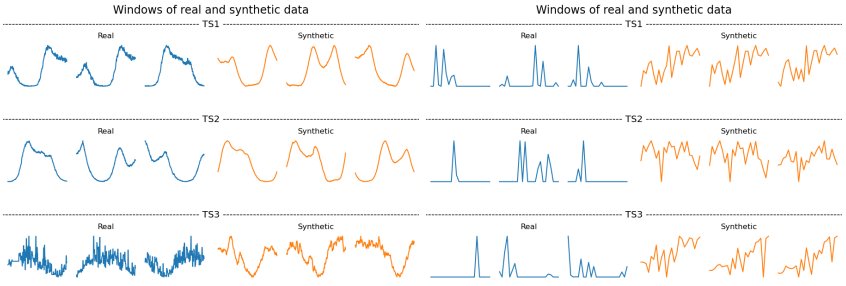
(a) Windows for *low* set. (b) Windows for *mid* set. (c) Windows for *high* set.

(d) Histogram for *low* set. (e) Histogram for *mid* set. (f) Histogram for *high* set.

Fig. 2: TimeGAN: Visual comparison of original and generated data across different subsets (*low*, *mid*, and *high*). Figure from [15].

(24 in the case of the *low* subset, 288 for the *mid* subset, and 1440 for the *high* subset); where m is the number of seasons, and d is the duration of each season.

The results obtained with this configuration are illustrated in Figure 3b, and show a difficulty in capturing the temporal dynamics of our data. This is not in accordance with the good results obtained with continuous data, with daily seasonality like TELCO, as seen in Figure 3a.



(a) TELCO windows (b) STELCO windows

Fig. 3: TimeVAE: Real and synthetic windows for TELCO (left) and STELCO (right). Figure from [15].

DC-VAE In the initial approach, the model was used to reconstruct the input series to evaluate its performance. The reconstruction of the three series of the *low* subgroup are depicted in Figure 4. A window length of 24 points was selected, corresponding to one day of activity. The model and its training process were slightly modified to shift from a multivariate approach to a global one. In this global mode, each input series was processed independently, without utilizing information from the other series for reconstruction. As illustrated in the plots, the model demonstrates a certain difficulty in reconstructing the highest activity peaks, instead primarily reflecting the mean value of each window.

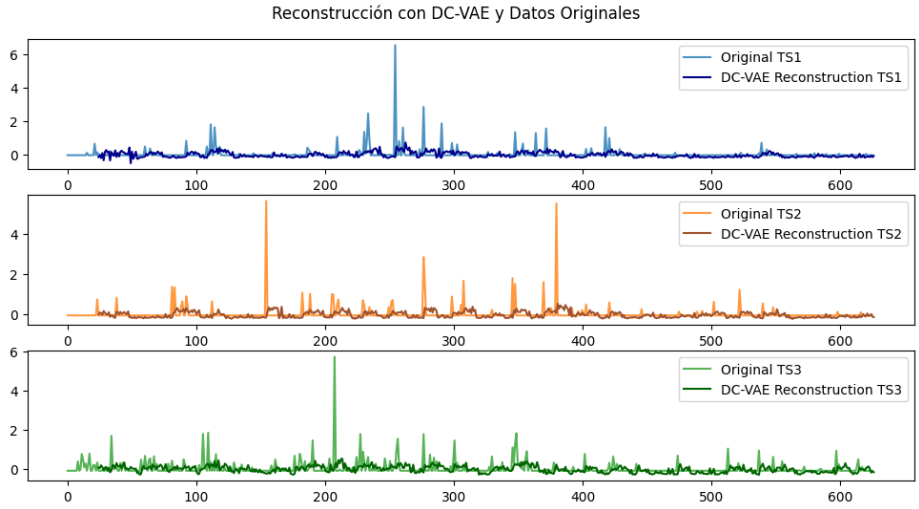


Fig. 4: Reconstruction of the *low* subgroup series with DC-VAE. Figure from [15].

The next step was to try to generate synthetic data from the original data. For this purpose, the already trained model was used, in this case with the *low* subgroup. Vectors with a uniform distribution (0,1), of dimension equal to the dimension of the latency space of the model, were generated and passed through the decoder. Thus, a window of $T = 24$ samples is obtained at the output of the decoder. For each uniform sample of the latency space, a window is generated, which are comparable with the windows of the original series. This procedure was repeated for each subset, and Figure 5 shows the comparison of real and synthetic windows for the subset *low*.

It was observed that the DC-VAE inadequately captured the dynamics of the original data, resulting in generated data that lacked resemblance to the originals and exhibited a certain degree of noise. This issue may arise from several factors. Firstly, the dimensionality reduction inherent in auto-encoders tends to prioritize lower frequency data, which can lead to the loss of higher frequency components.

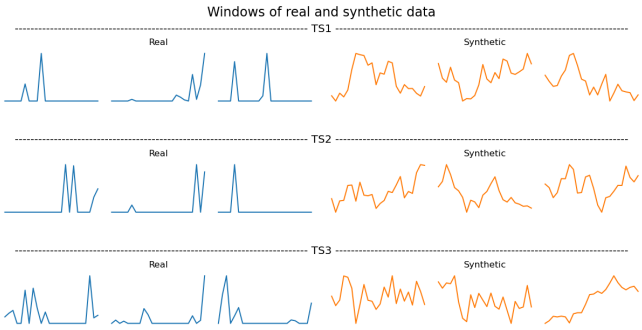


Fig. 5: Comparison of windows between original data (blue color) and synthetic data (orange color), generated from the DC-VAE decoder trained with the *low* subgroup series. Figure from [15].

Consequently, this results in the omission of significant peaks in our sparse data, which are crucial for our analysis.

Finally, a review of both reconstruction (4) and generation (5) results suggests that the DC-VAE is more suited for time series that exhibit higher activity levels and periodic dynamics. This is likely due to the Gaussian distribution of the DC-VAE output, which smooths the reconstruction process. In contrast, the original data does not exhibit such a distribution; rather, its values are predominantly zero, resulting in a distribution that aligns more closely with a Laplacian model. We have discussed the potential for future adaptations of the network to produce an output distribution that better fits the data, although this endeavor will require significant time and resources, and thus will be left for subsequent research.

5.3 Performance evaluation results

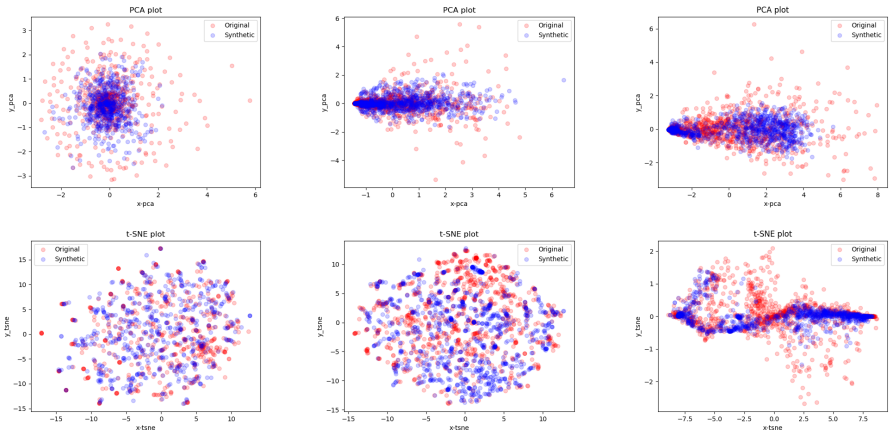
In this section, we present the results obtained from the evaluation metrics used to assess the performance of the different generative methods. First, a comparison of the sparsity between the original dataset and the generated ones is presented. This comparison, observed in Table 2, shows that for both the low and mid datasets, TimeGAN generated the closest dataset in terms of sparsity to the original. For the high dataset, it is the DC-VAE that achieves the closest to the original one, but it is a similar value throughout all datasets, so additional metrics are needed to give an accurate evaluation.

Proceeding to the next category of metrics: the visual indicators. Figure 6 shows the PCA and t-SNE plots for the experiments performed with TimeGAN on the *low*, *mid*, and *high* subsets. The plots illustrate that the generated data (blue) closely resembles the real data (red), as evidenced by the similar spatial distributions in both PCA and t-SNE representations. This similarity is particularly notable for the *low* and *mid* subsets. However, many red points located at

Table 2: Sparsity comparison between original and generated datasets.

Dataset	Original	TimeGAN	TimeVAE	DC-VAE
low	83.8%	82.8%	52.7%	48.6%
mid	77.8%	85.6%	35.0%	50.4%
high	54.8%	65.4%	4.2%	53.1%

the edges of the distributions are not covered by the blue points, which appear to correspond to the uncovered tails in the histograms shown in Figure 2.



(a) *low* (b) *mid* (c) *high*

Fig. 6: TimeGAN: PCA (top) and t-SNE (bottom) plots. Figure from [15].

To estimate the *Discriminative Score*, a time series classification model was developed, based on a 2-layer Long Short-Term Memory (LSTM) recurrent neural network (RNN). This model was trained to distinguish between real and synthetically generated data sequences, allowing us to measure how effectively the generative model produces data that resembles the real sequences.

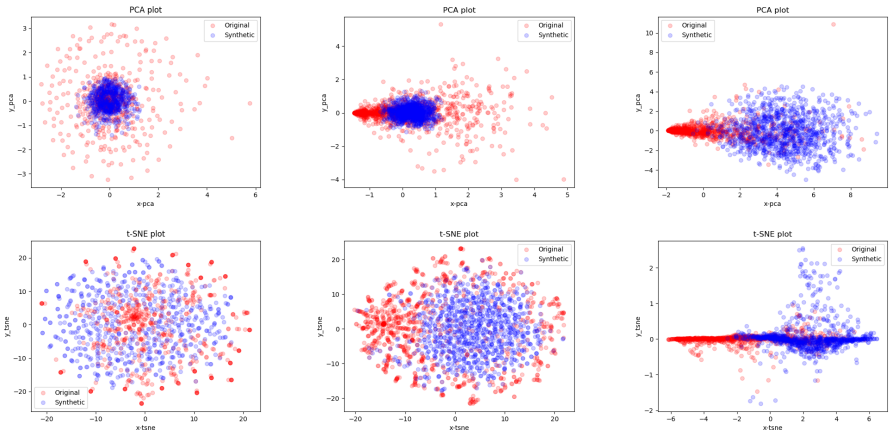
For the estimation of the *Predictive Score*, a separate sequence prediction model was employed, also utilizing a two-layer LSTM architecture. This model was designed to assess how well the generated sequences follow a logical time progression and whether they can be used to accurately predict future values.

In order to ensure robustness and statistical reliability, the entire procedure, encompassing both fidelity and usefulness evaluations, was repeated across 10 independent trials. The final results, reported in Table 3, represent the average performance over these iterations for each corresponding data subset.

Table 3: Comparison of model performance across the three data subsets, evaluated using the first set of metrics. Table from [15].

Metric	Method	Data subset		
		low	mid	high
Discriminative Score	DC-VAE	0.250 ± 0.026	0.264 ± 0.019	0.230 ± 0.024
	TimeGAN	0.268 ± 0.083	0.249 ± 0.131	0.269 ± 0.045
Predictive Score	DC-VAE	0.525 ± 0.006	0.687 ± 0.104	0.554 ± 0.004
	TimeGAN	0.519 ± 0.001	0.520 ± 0.002	0.738 ± 0.004
Time to train and generate	DC-VAE	79 s	678 s	1,634 s
	TimeGAN	5,624 s	30,494 s	65,220 s

As can be seen, the metric results do not reveal a substantial difference between the two generative models, suggesting that both exhibit comparable fidelity and usefulness. Nevertheless, this observation does not fully align with the qualitative inspection of the generated samples shown in Figures 2 and 5, corresponding to TimeGAN and DC-VAE, respectively. This discrepancy becomes more evident when considering the PCA and t-SNE plots in Figures 7 and 6, which highlight that the representations of original and generated data do not fully overlap for DC-VAE, while TimeGAN achieves a closer alignment.



(a) *low* (b) *mid* (c) *high*

Fig. 7: DC-VAE: PCA (top) and t-SNE (bottom) plots. Figure from [15].

Another factor that should be taken into account when comparing methods is the time they take to train and generate the synthetic data. This is where DC-VAE excels, given its VAE architecture is quite small and presents fast computing times. TimeGAN, on the other hand, is quite slow, performing at its worst with

large amounts of data and large window sizes. In Table 3, the elapsed training time is shown for each method using an NVIDIA GeForce RTX 3090 with 24.5GB of GPU memory.

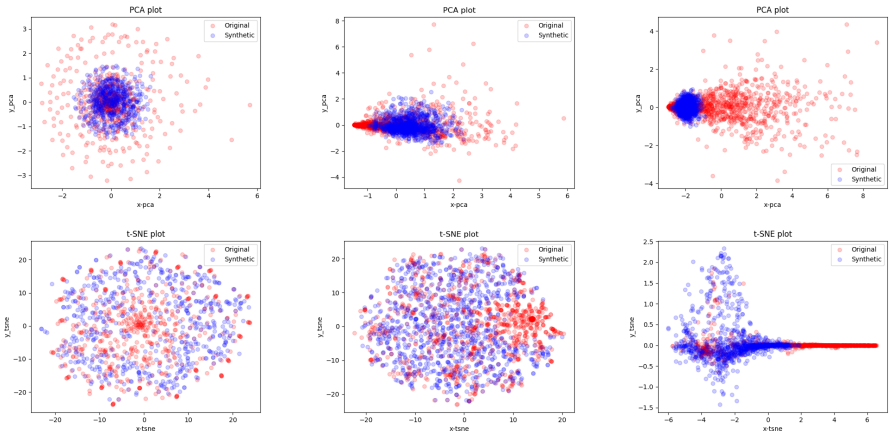
5.4 Complete time series synthesis visual results

In order to assess this aspect, the same performance evaluation procedure was applied to the concatenated windows generated using TimeGAN, with the results presented in Table 4 and Figure 8. These results suggest that it is possible to generate an entire time series when the data are sufficiently sparse, as they are comparable to those in Figure 6 and Table 3. This approach was therefore applied to the low subgroup, yielding the series shown in Figure 9. Nevertheless, an issue remains with the less frequent high-value peaks, which are not adequately represented in the generated series. Future work will focus on adapting the concatenation method to better capture these dynamics and enable the effective generation of complete time series.

Table 4: Comparison of window concatenation performance for the 3 subsets of data evaluated. Table from [15].

Metric	Method	Data subset		
		low	mid	high
Discriminative Score	TimeGAN	0.2678 ± 0.0828	0.2486 ± 0.1307	0.2694 ± 0.0447
	TimeGAN-concat	0.219 ± 0.101	0.2466 ± 0.0654	0.3341 ± 0.1234
Predictive Score	TimeGAN	0.5186 ± 0.001	0.5204 ± 0.0017	0.7377 ± 0.0043
	TimeGAN-concat	0.5239 ± 0.0004	0.5197 ± 0.0005	0.7373 ± 0.0006

As we observed, both for individual windows and for their concatenation, TimeGAN visually demonstrates better performance, with the generated series resembling the original ones more closely than those produced by DC-VAE and TimeVAE. However, the Discriminative and Predictive Score metrics, originally proposed by the authors of TimeGAN and also employed in our previous study [15], indicate that TimeGAN and DC-VAE achieve similar performance. This discrepancy may be due to the fact that these metrics rely on classification and prediction models based on neural networks, which could suffer from underfitting or overfitting, thereby failing to accurately capture the intended differences. It may also reflect the limitations of these metrics when applied to sparse or intermittent series, as shown in Table 2, where the large proportion of zeros introduces imbalance that affects the evaluation of the models. For this reason, and in order to obtain metrics more consistent with the visual comparisons, we conducted an analysis using classical quantitative measures, as presented in the following section.



(a) *low* (b) *mid* (c) *high*

Fig. 8: TimeGAN concatenated windows: PCA (top) and t-SNE (bottom) plots. Figure from [15].

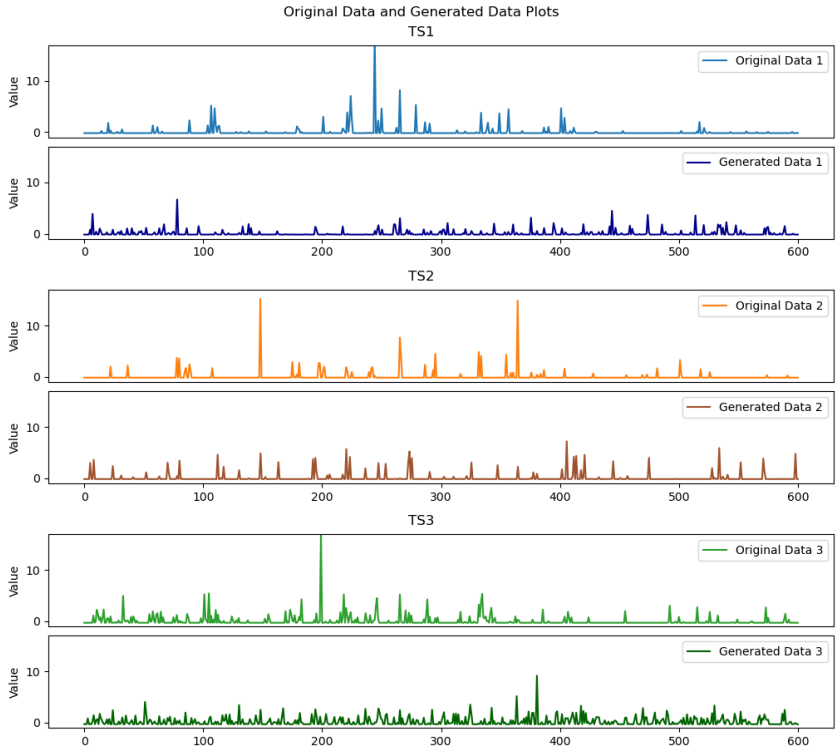


Fig. 9: TimeGAN: concatenated windows for the *low* subgroup. Figure from [15].

5.5 Results of the KL, JS, Wasserstein, KS and MMD metrics

The classical metrics presented in Section 2.3 are well-established and widely used in the literature to compare differences between distributions. In addition, they exhibit characteristics that are particularly useful for our problem. For example, the KL divergence is sensitive to discrepancies in the tails of the distribution—an aspect observable in Figure 2, where generative models struggle to reproduce such patterns. Similarly, the JS divergence provides stability in sparse regions, while MMD has the advantage of not requiring density estimation.

The results of applying these metrics to the data generated by TimeGAN and DC-VAE, compared against the original data, are shown in Table 5. Across all evaluation criteria, TimeGAN consistently demonstrates superior performance, highlighting its ability to better capture both the global structure and the sparse patterns present in the original datasets. These metrics form part of the novel evaluation framework proposed in this work. It is worth noting that the generated windowed sequences were concatenated for evaluation; however, this approach does not introduce any limitations, since the metrics assess the underlying data distributions and are inherently independent of sequence length.

Table 5: Comparison of model performance across the three data subsets, evaluated using the second set of metrics. Bold values indicate the best performance for each metric.

Metric	Method	Data subset		
		low	mid	high
Kullback–Leibler divergence	DC-VAE	0.667	**0.607**	3.329
	TimeGAN	**0.389**	0.953	**0.339**
Jensen-Shannon divergence	DC-VAE	0.148	0.115	0.566
	TimeGAN	**0.019**	**0.063**	**0.147**
Wasserstein distance	DC-VAE	0.569	0.396	3.407
	TimeGAN	**0.077**	**0.247**	**0.505**
Kolmogorov-Smirnov test	DC-VAE	0.408	0.242	0.646
	TimeGAN	**0.030**	**0.183**	**0.290**
Maximum-Mean Discrepancy (MMD)	DC-VAE	0.122	0.054	0.545
	TimeGAN	**0.0001**	**0.015**	**0.058**

6 Conclusions and future work

This work repositions the focus from the generation of synthetic data to the critical assessment of how performance metrics can inform our understanding of data augmentation quality. We introduced a set of quantitative metrics designed to better capture the structural and statistical properties of such data and applied them to three generative approaches.

Our results show that the classical metrics provide a stronger alignment between quantitative evaluation and the visual assessment of the generated time series. Unlike the metrics previously used in [15], which often failed to capture noticeable qualitative differences, the metrics introduced in this work consistently reflect the patterns evident through visual inspection. This represents an important step toward more reliable and interpretable evaluation practices in the generation of sparse time series.

The comparative analysis of different generative models further illustrated the utility of these metrics in revealing specific strengths and weaknesses of each approach. For instance, while some models produced visually convincing patterns, only the proposed metrics were able to robustly quantify these improvements.

Looking forward, future work will aim to refine these evaluation tools further, incorporating domain-specific considerations or downstream task relevance.

Finally, it is important to emphasize that we made the STELCO dataset and the generation procedure for ASTELCO publicly available, along with the accompanying to facilitate the reproducibility of the results.

6.1 Code and datasets

We provide access to all materials utilized for conducting the experiments, including both the real and generated datasets: the code used to run the experiments with TimeGAN[1] and DC-VAE[2], the metrics used to evaluate the performance of the models[3], and both STELCO and ASTELCO datasets[4].

Acknowledgments. This work has been partially supported by the Uruguayan CSIC project with reference *CSIC-I+D-22520220100371UD "Generalization and Domain Adaptation in Time-Series Anomaly Detection"*, and by Telefónica. The authors would also like to thank José Acuña and the Telefonica team for their valuable support and contributions.

References

1. Brophy, E., Wand, Z., She, Q., Ward, T.: Generative adversarial networks in time series: A systematic literature review. In: ACM Computing Surveys, Volume 55, Issue 10. pp. Article No.: 199, Pages 1 – 31 (2023)
2. Cover, T.M.: Elements of information theory. John Wiley & Sons (1999)
3. Desai, A., Freeman, C., Wang, Z., Beaver, I.: Timevae: A variational auto-encoder for multivariate time series generation. arXiv preprint arXiv:2111.08095 (2021)
4. Endres, D., Schindelin, J.: A new metric for probability distributions. IEEE Transactions on Information Theory **49**(7), 1858–1860 (2003). https://doi.org/10.1109/TIT.2003.813506

[1] https://github.com/ydataai/ydata-synthetic
[2] https://github.com/GastonGarciaGonzalez/DC-VAE
[3] https://github.com/manu3z/data-augmentation-evaluation-metrics
[4] https://iie.fing.edu.uy/investigacion/grupos/anomalias/stelco-dataset/

5. Esteban, C., Hyland, S.L., Rätsch, G.: Real-valued (medical) time series generation with recurrent conditional gans (2017), https://arxiv.org/abs/1706.02633
6. Fan, L., Zhang, J., Mao, W., Cao, F.: Unsupervised anomaly detection for intermittent sequences based on multi-granularity abnormal pattern mining. In: Entropy. pp. 25, 123 (2023)
7. García González, G., Martinez Tagliafico, S., Fernández, A., Gómez, G., Acuña, J., Casas, P.: Dc-vae, fine-grained anomaly detection in multivariate time-series with dilated convolutions and variational auto encoders. In: IEEE European Symposium on Security and Privacy Workshops (EuroS&PW). pp. 287–293 (2022)
8. García González, G., Martínez Tagliafico, S., Fernández, A., Gómez, G., Acuña, J., Casas, P.: Telco (2023). https://doi.org/10.21227/skpg-0539, iEEE Dataport. https://dx.doi.org/10.21227/skpg-0539
9. González, G.G., Casas, P., Martínez, E., Fernández, A.: On the quest for foundation generative-ai models for anomaly detection in time-series data. In: 2024 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW). pp. 252–260. IEEE (2024)
10. Gretton, A., Borgwardt, K.M., Rasch, M.J., Schölkopf, B., Smola, A.: A kernel two-sample test. The journal of machine learning research **13**(1), 723–773 (2012)
11. Iglesias, G., Talavera, E., González-Prieto, A., Mozo1, A., Gómez-Canaval1, S.: Data augmentation techniques in time series domain: a survey and taxonomy. In: Neural Computing and Applications. p. 10123–10145 (2023)
12. Makridakis, S., Spiliotis, E., Assimakopoulos, V.: M5 accuracy competition: Results, findings, and conclusions. International Journal of Forecasting **38**(4), 1346–1364 (2022). https://doi.org/https://doi.org/10.1016/j.ijforecast.2021.11.013, https://www.sciencedirect.com/science/article/pii/S0169207021001874, special Issue: M5 competition
13. Massey Jr, F.J.: The kolmogorov-smirnov test for goodness of fit. Journal of the American statistical Association **46**(253), 68–78 (1951)
14. Renz, P., Cutajar, K., Twomey, N., Cheung, G.K.C., Xie, H.: Low-count time series anomaly detection. In: 2023 IEEE 33rd International Workshop on Machine Learning for Signal Processing (MLSP). pp. 1–6 (2023). https://doi.org/10.1109/MLSP55844.2023.10285979
15. Sánchez-Laguardia, M., García González, G., Martinez, E., Martinez, S., Fernández, A., Gómez, G.: Astelco: An augmented sparse time series dataset with generative models. In: Proceedings of the 14th International Conference on Pattern Recognition Applications and Methods - ICPRAM. pp. 283–290. INSTICC, SciTePress (2025). https://doi.org/10.5220/0013185900003905
16. Villani, C., et al.: Optimal transport: old and new, vol. 338. Springer (2008)
17. Wang, C., Wu, K., Zhou, T., Yu, G., Cai, Z.: Tsagen: Synthetic time series generation for kpi anomaly detection. IEEE Transactions on Network and Service Management **19**(1), 130–145 (2022). https://doi.org/10.1109/TNSM.2021.3098784
18. Wen, Q., Sun, L., Yang, F., Song, X., Gao, J., Wang, X., Xu, H.: Time series data augmentation for deep learning: A survey. arXiv preprint arXiv:2002.12478 (2020)
19. Yoon, J., Jarrett, D., van der Schaar, M.: Time-series generative adversarial networks. In: Advances in Neural Information Processing Systems 32 (NeurIPS 2019). pp. 5508–5518 (2019)