Generative Learning for Imputation of Univariate Time Series Using Images and GANs

Mauricio M. Almeida, João D. S. De Almeida, Geraldo B. Junior, Aristófanes C. Silva, and Anselmo C. Paiva

***Abstract*—Time series are widely used because they capture the relationship between events over time. However, when data is missing, this dependency is compromised. Traditional imputation techniques are being replaced by approaches based on neural networks, especially generative adversarial networks (GANs) and conditional generative adversarial networks (cGANs), due to their high capacity for reconstructing complex patterns.In this research, we explore an innovative path: transforming time series into images to recover missing data. We propose a new version of this transformation, incorporating two channels, trend and seasonality. We evaluated four generative architectures (CycleGAN, DCGAN, Pix2Pix, and DiscoGAN) in 240 models, covering different datasets and absence rates from 10% to 40%. We also introduced an unprecedented loss function to reduce the sensitivity of networks to anomalous data. The results show that the proposed two-channel approach produced the most stable imputations. Pix2Pix showed the best average performance: 75% of the Mean Absolute Percentage Error (MAPE) errors were below 0.21 (10% missing) and 2.66 (40% missing). For the Adapted Symmetric Mean Absolute Percentage Error (ASMAPE), the maximum third quartile values were 1.06 and 3.21, respectively. With the proposed loss function, the MAPE error fell from 12.95 to 3.21, and the ASMAPE from 2.86 to 2.66 in the best scenario, with a significant reduction that reinforces the effectiveness of the technique.**

Link to graphical and video abstracts, and to code: https://latamt.ieeer9.org/index.php/transactions/article/view/9963

***Index Terms*—Imputation, cGANs, Image of time series.**

I. Introdution

THE demand for data in various social sectors is constant in today's society. Algorithms and techniques are being sought to provide more precise information on the characteristics of data. This demand is partly due to the knowledge that solid, well-founded information is essential for guiding strategic decisions, whether in Health [1], Finance [2], Education [3] or any other field. Having access to reliable data and accurate analysis is key to understanding trends, identifying patterns and anticipating future scenarios.

One type of useful data is time series data. A Time Series (TS) is a set of observations organized in chronological order, where each observation is associated with an event at a specific time. These observations are used to represent the relational evolution of a phenomenon over time, allowing trends, seasonal patterns and variations in different periods to be analyzed [4].

In the context of TS analysis, algorithms play a crucial role. They make it possible to extract valuable insights from data, revealing and relating hidden information and thus transforming raw data into actionable knowledge, i.e. patterns, correlations or any extracted information that can be used by humans and machines for decision-making.

Through advanced data analysis techniques, such as machine learning and artificial intelligence, it is possible to obtain more accurate predictive models and discover complex correlations between variables or sets of variables at different times. However, it is important to note that the quality of the results obtained depends on both the quantity and quality of the TS available. It is therefore necessary to ensure that the information is properly collected and stored, and to apply pre-processing and cleaning techniques to deal with possible noise and inconsistencies [5], [6].

Before any analysis or application of time series, it is crucial to carry out a thorough assessment of the quality of the data. One of the recurring challenges in this field is dealing with the presence of missing data, which can jeopardize the accuracy and reliability of analyses. The miss data in a time series can have a negative impact on the ability to identify patterns, make accurate forecasts and carry out proper statistical analysis. It is therefore essential to apply appropriate techniques to deal with this missing data in order to minimize bias and ensure reliable results [7, p.3-7].

There are various approaches to dealing with missing data in time series, such as imputing missing values based on statistical methods, interpolating data using smoothing techniques or even carefully excluding observations with missing data, depending on the importance and amount of missing data [8]. The most efficient current techniques in data imputation are machine learning algorithms [9], [10].

In the context of machine learning applied to time series, there is growing interest in Generative Adversarial Networks (GANs) and Conditional Generative Adversarial Networks (cGANs). These networks are applied both to generate synthetic time series [11] and for classification [12], imputation of missing data [13] and imputation of missing data using images [14].

However, these networks are designed for image processing, which requires converting time series into

The associate editor coordinating the review of this manuscript and approving it for publication was Ruth Aguilar (*Corresponding author: Mauricio Morais Almeida*).

This work was supported by CAPES (Finance Code 001), CNPq (Grant 305253/2025-5), and FAPEMA.

Mauricio Morais Almeida, J. D. S. De Almeida, G. B. Junior, A. C. Silva, and Anselmo C. Paiva are with applied computing group, Federal University of Maranhão, São Luís - MA, Brazil (e-mails: mauricio.ma@discente.ufma.br, jdallyson@nca.ufma.br, geraldo@nca.ufma.br, ari@nca.ufma.br, and paiva@nca.ufma.br).

visual representations. There are several approaches to this transformation, each seeking to capture specific patterns. For example, Barra et al. [15] used a conversion based on polar coordinates to forecast financial market series. However, there is no evidence that this technique is effective for more generic applications.

Another approach is from Sezer e Ozbayoglu [16], which uses convolutional networks to transform series into images, however the application in the specific context of financial trading has no generic application, as well as we do not have how to return from image to time series.

However, there is a spatio-temporal transformation proposed by Almeida at al. [14] that transforms time series into images. This transformation encodes the spatial relationship between the data and the temporal relationship in the same 2D matrix and is perfectly invertible, i.e. it returns from image to series without loss.

Thus, in view of the above, this work aims to propose an evolution of the method of transforming time series into [**?**] images, as well as to compare the performance of different cGANs in the task of imputing time series using images. With this, we highlight two main contributions: (1) evaluating the application of generative adversarial models (GANs) in the recovery of time series data and (2) proposing an evolution of the transformation presented by Almeida et al. [14], reformulating the problem of data imputation as an image recovery task. With this approach, we seek to identify the most effective generative model for dealing with gaps in time series, exploring the adaptation of techniques originally designed for image processing.

II. Related Work

Data loss is one of the main factors that compromise the effective use of actionable information derived from processed datasets. The quality of such information is directly tied to the quality of the underlying data, making it necessary not only to recover the missing values but also to ensure a minimum level of reliability in the recovered data.

The choice of technique for imputing missing values can significantly influence the representativeness and overall quality of the reconstructed data. Traditional imputation approaches, such as linear interpolation or constant replacement, as well as more advanced statistical models like ARIMA (*Autoregressive Integrated Moving Average*) [17], have been widely employed. However, just as ARIMA is tailored for stationary or stationarizable time series, the performance of many methods varies according to the characteristics of the dataset [18].

This is reinforced by recent studies which, although based on machine learning an approach that has generally outperformed traditional methods, particularly *Generative Adversarial Networks* (GANs) and *Conditional Generative Adversarial Networks* (cGANs) show that, in certain contexts, statistical models can be more efficient than state-of-the-art models (e.g., [19]). However, like many other studies, this work is limited in the variability of series types considered: it relies on only three different domains. Thus, it underscores the improbability of a single, general model that performs well across diverse types of time series [18].

Brophy et al. [20] investigate GAN architectures, applications, evaluation metrics, and other particularities. According to the authors, GANs have proven effective in a range of applications, such as image generation and the synthesis of time-series data. This efficacy is corroborated in the recent literature, for example, Al-Fakih et al. [21] and Yin et al. [22], where TimeGAN and SeqGAN are used for imputation of geological data, and GANs combined with GRUs are employed to improve imputation in medical datasets, respectively.

In the context of time-series data imputation, Huang et al. [23] propose the TSDIGAN network (Time Series Data Imputation using GAN). The architecture employs a generator that upscales a noise vector conditioned on the data to produce synthetic images of the time series, while the discriminator reduces the dimensionality of those images to distinguish real from generated samples. This approach has proven effective in domains such as vehicular traffic, where multiple series of the same type can be arranged as coherent images. Nonetheless, its applicability is limited because the method requires the existence of numerous time series of the same type in order to form such images. In contrast, Almeida et al. [14] propose a transformation of time series into images that enables the recovery of missing values; in their method time is explicitly encoded in the matrix arrangement of the image, but they do not apply a specific transformation of the series' values using a conditional GAN such as Pix2Pix.

Thus, we propose an extension of the time-series-to-image transformation presented by Almeida et al. [14], treating the temporal spacing of each series as a conditional value. To this end, we evaluate four conditional GAN architectures (CycleGAN, Pix2Pix, DiscoGAN), in which generation is influenced by additional information about temporal spacing, in contrast to traditional GANs such as DCGAN that do not employ this form of explicit conditioning. Unlike prior works that apply GANs to specific domains, our evaluation uses at least five distinct series from different areas, enabling assessment of the models' robustness across varied contexts.

III. Materials and Method

This section introduces the networks and datasets employed, the proposed method, and the metrics used to evaluate the experiments.

A. Conditional Generative Adversarial Networks

The conditional Generative Adversarial Network (cGAN), introduced by Mirza and Osindero [24], is a framework that leverages a game-theoretic approach between two neural networks: the generator $\mathbf{G}$ and the discriminator $\mathbf{D}$. The training objective of the generator is to produce realistic synthetic samples conditioned on specific input information, while the discriminator is trained to differentiate between samples generated by the generator and real samples.

The training process of a cGAN commences with the generator receiving a conditional input, typically a data vector

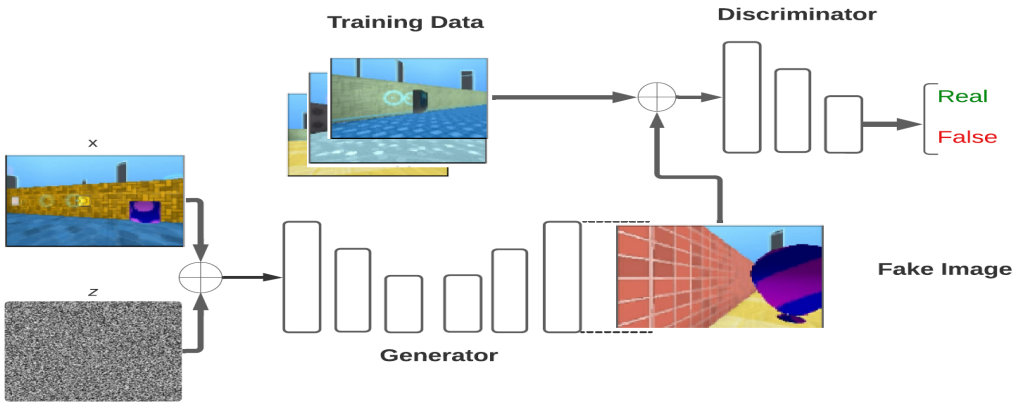


Fig. 1. High-level architecture of the Pix2Pix network, a cGAN network.

x, which encapsulates the information to be conditioned for generation, representing the prior probability. The generator aims to learn a distribution ψ_g over the input data x. Noise, denoted as $\psi_z(z)$, is introduced to the data x, and the generator, conditioned on this noise, generates a synthetic sample $\hat{y}$ approximating the true data y [24, p.2]. The synthetic data $\hat{y}$ is subsequently combined with the true data y and fed to the discriminator. The generator's learning process over $d(z)$ is represented by $\mathbf{G}(z, \theta_g)$, while the discriminator's learning is modeled by $\mathbf{D}(x, \theta_d)$, where θ_g and θ_d denote the parameters of the respective models. Notably, $\mathbf{D}(x, \theta_d)$ outputs a single probability [24, p.2].

Training is formulated as a *min-max* game, where the generator seeks to minimize the function $\log(1 - D(G(z)))$, while the discriminator aims to maximize the probability of correctly identifying real samples by maximizing $\log(D(x))$. In essence, the generator strives to deceive the discriminator, while the discriminator endeavors to distinguish real samples from synthetic ones. The cGAN's cost function is thus defined as shown in Equation 1, as described by (Mirza and Osindero) [24, p.2-3].

$$[H] \min_G \max_D V(D,G) = \mathbb{E}_{x\sim\psi_{\text{data}}(x)} [\log D(x|y)] + \mathbb{E}_{z\sim\psi_z(z)} [\log(1 - D(G(z|y)))] \quad (1)$$

The Pix2Pix cGAN (Image-to-Image Translation with Conditional Adversarial Networks) [25] is a widely adopted architecture for image-to-image translation tasks. The specific implementation used in this study is based on the approach proposed by Almeida et al. [14]. Additionally, the CycleGAN (Cycle-Consistent Adversarial Networks) [26], a variant of Pix2Pix, was employed. The primary distinction between the two lies in CycleGAN's dual architecture, which incorporates two generators and two discriminators.

In CycleGAN, one generator maps an input to a desired output, while the second generator takes the output and attempts to reconstruct the original input. Each mapping process is evaluated by its respective discriminator to assess translation quality.

Both the Pix2Pix cGAN and CycleGAN are extensively utilized for various image-to-image translation tasks, including style transfer, color transformation, and other visual modifications. These networks are particularly effective in scenarios lacking a direct, well-defined mapping between input and output images.

The evaluation also included the DCGAN (*Deep Convolutional Generative Adversarial Network*) [27] and the Discriminative Generative Adversarial Network (DiscoGAN) [28], a convolutional cGAN. Notably, CycleGAN represents an advanced iteration of DiscoGAN. In its initial form, DiscoGAN focuses on cross-domain translation, whereas the enhanced CycleGAN introduces reversibility through content preservation across domains.

B. Datasets

The univariate time series datasets were selected based on the following criteria: they must be publicly available, exhibit a detectable and regular temporal sequence, contain no missing data, have at least 1.024 samples, and be numerical in nature.

Among the selected datasets, the first, *Monthly Sunspots* (SU) [29], includes data on the international sunspot number recorded weekly from 1749 to 1983.

The *Electricity Usage* (EU) dataset [30] comprises residential electricity consumption data in kilowatt-hours over a 3.5-month period, from November 25, 2011, to March 17, 2012. The series includes four variables (*Consumption*, *kWh*, *Off-peak/Mid-peak*, and *On-peak*), with *Consumption* being the variable of interest and thus selected for analysis.

The *Electricity Load Diagrams 2011–2014* (ELD) dataset [31] contains electricity consumption data from 2011 to 2015, collected at 15-second intervals, resulting in a total of 140.256 samples. This dataset consists of 370 independent variables, labeled MT-001 to MT-370. To mitigate sample imbalance, 71 randomly selected series from this dataset were used.

The *Bike Sharing* (BSH) dataset [32] includes two main series: one with hourly samples and another with daily samples of bicycle rentals and returns at different locations. The hourly series was adopted, specifically the *cnt* subseries, which contains 17.389 records of the total number of bicycles rented during the period.

The *Ammonia* (AMM) dataset [33] provides data on ammonia levels in the water flow at a water treatment plant,

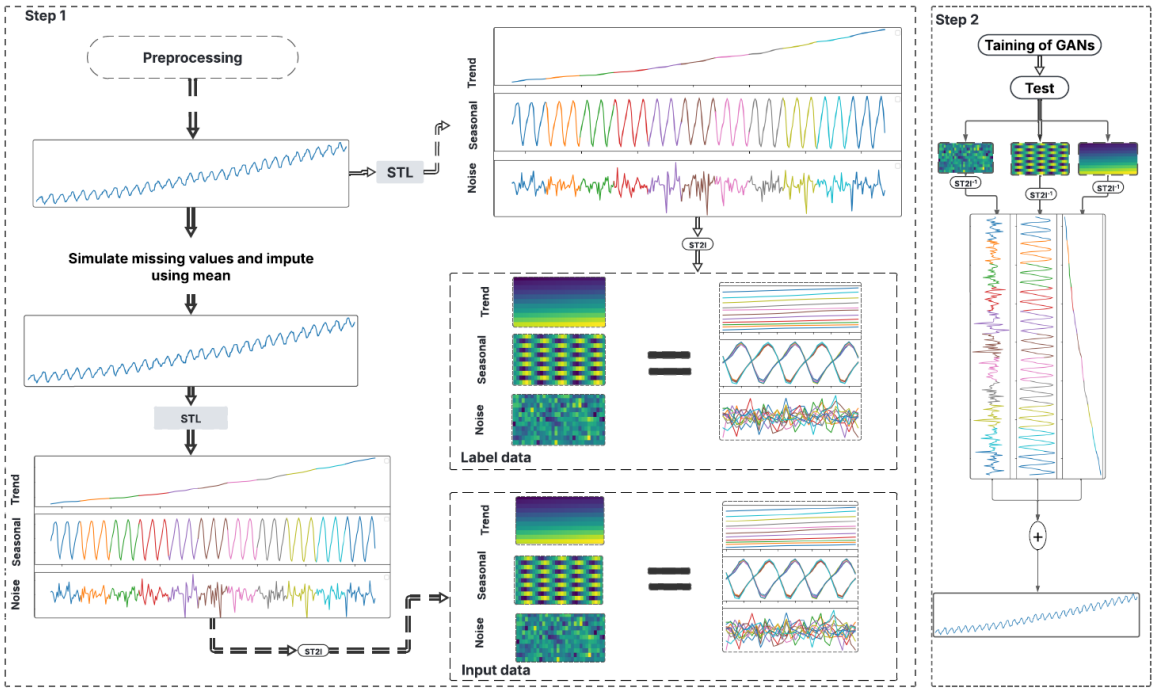


Fig. 2. Proposed methodology workflow

with measurements recorded every six hours. Meanwhile, *Distillate Flowrate* (DF) [34] contains data on the distillate flow rate of a liquid in a distillation tower, and *Daily Minimum Temperatures in Melbourne* (DMTME) [35] includes records of daily minimum temperatures in Melbourne, USA, from 1981 to 1990.

Table I summarizes the datasets used in this study. Each dataset is identified by its name, the number of time series, the source of the data, and the range of values present in the data. These datasets were selected based on the criteria outlined at the beginning of this section and are specific, requiring a minimum size of 1×1024, to meet the research objectives.

TABLE I
SUMMARY DESCRIPTION OF THE DATASETS

Dataset	Number of TS	Source	Value Range
SU	1	[29]	[0−253.8]
EU	1	[30]	[0.12−5.45]
ELD	370	[31]	[0−192.800]
BSH	1	[32]	[1−503.47]
AMM	1	[33]	[9.99−58.74]
DF	1	[34]	[0−1.000]
DMTME	1	[35]	[7−43.3]

C. Proposed Method

This section outlines the sequence of steps undertaken to conduct this research, as illustrated in Fig. 2. The main stages of the proposed method are described below, namely: Data Collection and Preparation, Construction of the Dataset with Transformation of Time Series into Images, and Model Training.

1) Data Collection and Preparation: The initial step in conducting the experiments for this research involves acquiring relevant time series data for the study. During this data collection phase, the quality of the data is assessed, addressing potential issues such as noise, inconsistencies, duplicates, and outliers.

Regarding inconsistent samples, an example can be found in the dataset [31], which describes energy consumption. It was observed that all series contain null values at the beginning, indicating periods without consumption, and these were subsequently removed. To deal with outliers, the interquartile range (IQR) method [36] was used to identify and replace outliers with the last or closest valid value. Values outside the interquartile range (i.e., below $Q1 - 1.5 \cdot \text{IQR}$ or above $Q3 + 1.5 \cdot \text{IQR}$) were classified as outliers for each data window of size 30.

2) Transforming Time Series into Images: The proposed Time Series to Image (ST2I) transformation method is an extended version of the approach developed by Almeida et al. [14](Fig. 3) . It involves arranging a time series into a matrix format such that each row of the resulting image represents a segment of the time series, with each segment consistently starting and ending on the same dates with respect to the frequency.

2011-01-01 00:00:00	2011-01-01 01:00:00	…	2011-01-01 20:00:00
2011-01-02 00:00:00	2011-01-02 01:00:00	…	2011-01-01 20:00:00
⋮	⋮	⋮	⋮
2011-01-20 00:00:00	2011-01-20 01:00:00	…	2011-01-20 20:00:00

Fig. 3. Temporal arrangement example of the ST2I method for generating a 20×20 image.

Fig. 3 illustrates the process of transforming a time series

into an image. Notably, the figure does not depict the actual values but rather the temporal indexing of the values, i.e., the positions that the values corresponding to the indicated dates will occupy. Thus, starting from a time series, a preprocessing step is carried out in which missing values are artificially simulated using the MCAR (Missing Completely At Random) mechanism. The imputation is performed as described in Subsection III-C1. In the proposed approach, in addition to these steps, the STL algorithm [37] is applied to both the preprocessed and the original series in order to extract their trend, seasonality, and residual components. STL is applied to the input data only after the insertion of missing values so as to better approximate real-world scenarios. Subsequently, the ST2I transformation is applied to each of these components, generating the images that comprise the dataset.

3) Training: After constructing the dataset, the Pix2Pix, DCGAN, DiscoGAN, and CycleGAN networks are trained to generate a realistic reconstruction of the time series from a masked image using mean imputation. To force the networks to learn to recover points of interest, a slight Gaussian noise is added to all non-missing points. The masks indicating the missing values are used exclusively for error calculation and noise insertion in order to emphasize the regions of interest.

The models are evaluated based on the reconstruction errors of the time series obtained from the generated images. Reconstruction follows the STL decomposition, considering the sum of the trend, seasonal, and noise components. Since the extracted noise is the same for both the prediction and the label, it can be optionally added to both without altering the final error.

$$\hat{Y} = C_1 + C_2 \quad \text{ou} \quad \hat{Y} = C_1 + C_2 + \varepsilon, \quad \text{with } Y = C_1 + C_2 + \varepsilon$$

Because ε is the same, the difference $Y - \hat{Y}$ is equal in both scenarios.

To reduce the variance in estimates of the loss functions during GAN training, we adopt a James–Stein estimator–based strategy following Gu and Moon [38]. Suppose a vector of individual losses $\mathcal{L} = [\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_n]$, obtained, for example, from the binary cross-entropy loss or the L1 loss with *reduction="none"*.

The mean of the loss vector is defined by Equation (2), the sample variance (uncorrected for bias) by Equation (3), and the centered squared norm by Equation (4).

$$\mu = \frac{1}{n}\sum_{i=1}^{n}\mathcal{L}_i, \quad (2) \qquad S^2 = \frac{1}{n}\sum_{i=1}^{n}(\mathcal{L}_i - \mu)^2, \quad (3)$$

$$\|\mathcal{L} - \mu\|^2 = \sum_{i=1}^{n}(\mathcal{L}_i - \mu)^2. \quad (4)$$

The James-Stein estimator applied to the average loss is then defined as

$$\widehat{\mathcal{L}}_{\text{JS}} = \mu + \left(1 - \frac{(n-2)S^2}{\|\mathcal{L} - \mu\|^2 + \varepsilon}\right)_+ \cdot \left(\frac{1}{n}\sum_{i=1}^{n}(\mathcal{L}_i - \mu)\right), \quad (5)$$

where $(\cdot)_+ = \max(0, \cdot)$ is the lower truncation operation to ensure a non-negative shrinkage factor, and ε is a numerical stability term. This estimator smooths the loss mean, shifting it toward the most robust value, which is particularly useful in adversarial contexts such as GANs, where training is notoriously unstable. In our approach, we use $\widehat{\mathcal{L}}_{\text{JS}}$ instead of the traditional loss average, aiming for a more balanced training to mitigate the dominance of one network over the other. The proof of the James-Stein theorem can be found in Appendix A or in [39].

D. Evaluation Metrics

To evaluate the experiments conducted in this study, the Mean Absolute Percentage Error (MAPE) was selected. The MAPE, as defined in Equation (6), is particularly sensitive to localized errors. This implies that in scenarios with significant discrepancies at specific points, the MAPE tends to amplify the magnitude of the error, making it a valuable metric for detecting such specific variations.

Conversely, for a more balanced global assessment, the *Adapted Symmetric Mean Absolute Percentage Error* (ASMAPE), proposed by Almeida et al. [14], was adopted. The main difference between ASMAPE and traditional sMAPE is the treatment of zero values: in ASMAPE, correct forecasts equal to zero are not penalized, which makes it more robust for cyclical or intermittent series and more faithful to the actual performance of the model when zeros have relevant meaning.

$$\text{MAPE} = \frac{100}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \quad (6)$$

where n is the total number of observations;y_i is the actual value at position i and $\hat{y}_i$ is the predicted value at position i.

The ASMAPE is defined by the following equation:

$$\text{ASMAPE} = \frac{1}{n}\sum_{i=1}^{n}\begin{cases}0, & \text{if } y_i = \hat{y}_i = 0\\ \frac{|y_i - \hat{y}_i|}{|y_i| + |\hat{y}_i|}, & \text{otherwise}\end{cases} \quad (7)$$

where y_i is the actual value, $\hat{y}_i$ is the predicted value, and n is the number of samples.

IV. Results and Discussion

This section presents the synthesized results across all folds for each model, with respect to the ASMAPE and MAPE metrics.

Training of the four networks was conducted equitably across all conditions. All components (trend, seasonality, and noise) were considered, as well as data excluding noise, resulting in images with one, two, and three channels. Additionally, for each channel variation, different rates of missing values were tested, specifically 10%, 20%, 30%, and 40%.

Consequently, 48 experiments were performed, producing 240 models using cross-validation with a 70-15-15 split for cases where more than three images were obtained per series. In cases with fewer than four images, sampling was prioritized for the test set. All models were trained for 1,000 epochs, with Early Stopping applied to monitor the validation loss of the

generators, ensuring a minimum of 100 epochs to promote robust convergence.

To facilitate understanding of the results, this section is organized into three parts: (i) analysis of the impact of the proposed loss (Equation 5) on the performance and stability of the most sensitive model, CycleGAN; (ii) evaluation of improvements in the other models (DCGAN, DiscoGAN, and Pix2Pix), considering stability, presence of outliers, and average performance; and (iii) discussion of patterns observed with the increase in the rate of missing data.

CycleGAN showed the greatest variations and the worst initial performance, with SMAPE of 7.80 ± 12.50 and extremely high MAPE ($2.35 \times 10^{14} \pm 5.10 \times 10^{15}$) in standard loss, Table V. Our proposed loss (Equation 5) promoted a reduction in SMAPE to 5.45 ± 10.31 and in MAPE to $1.87 \times 10^{14} \pm 4.27 \times 10^{15}$ (Table IV), indicating an improvement in stability and accuracy. Despite this significant reduction, MAPE remains high due to its extreme sensitivity to outliers, which distort the overall imputation mean.

However, analysis of the percentage of predictions with an error greater than 100%, presented in row `%>100` of Table IV, reveals that these outliers represent a minority, with an almost irrelevant percentage frequency in the total set. This indicates that most of CycleGAN's predictions are within an acceptable error range. Therefore, despite the visual impact of the high MAPE, the lower SMAPE and the reduced frequency of outliers suggest that the model achieves reasonable performance in most cases, especially after applying the loss based on the James–Stein estimator, which contributes to reducing the variability of the results.

Table II presents a comparison between the use of the standard loss and the proposed loss for the scenario with 2 channels and 10% loss. It can be observed that, although pix2pix already presented the best performance before the application of the new loss, it still showed a consistent improvement in both metrics, SMAPE and MAPE. Models such as DCGAN and DiscoGAN also benefited from the proposed loss, showing clear reductions in errors and greater stability. In particular, the significant reduction in SMAPE and MAPE of CycleGAN highlights the impact of loss in mitigating the effects of outliers.

TABLE II
COMPARISON BETWEEN THE THIRD QUARTILE WITH STANDARD LOSS VS $\widehat{\mathcal{L}}_{\text{JS}}$ (2C, 40% LOSS)

Model	SMAPE (Standard $\rightarrow \widehat{\mathcal{L}}_{\text{JS}}$)	MAPE (Standard $\rightarrow \widehat{\mathcal{L}}_{\text{JS}}$)
CycleGAN	15.83 $\rightarrow$ 6.57	15.61 $\rightarrow$ 5.33
DCGAN	16.1 $\rightarrow$ 6.35	15.52 $\rightarrow$ 5.58
DiscoGAN	15.21 $\rightarrow$ 4.89	14.19 $\rightarrow$ 4.14
Pix2Pix	12.95 $\rightarrow$ **3.21**	12.86 $\rightarrow$ **2.66**

The results indicate that the models show significant differences in performance and stability. Pix2pix (Table VI clearly stands out, with the lowest mean error values and lowest variance, reflecting its superior generalization ability and robustness even without changes in loss. The DCGAN (Table VII and DiscoGAN (Table VIII) models demonstrate good stability, with reduced average errors and less influence from outliers after applying the proposed loss, showing significant gains in the SMAPE and MAPE metrics. In contrast, CycleGAN (Table V) has very high variance, especially in MAPE, suggesting greater sensitivity to extreme cases, which compromises its consistency. These differences highlight that, although the optimized loss contributes to improving overall stability and accuracy, the model architecture remains a determining factor in the final performance.

TABLE III
SUMMARY TABLE OF THE MAIN AVERAGE RESULTS OBTAINED FROM THE AVERAGES OF THE FOLDS, BASED ON SMAPE

Model	SMAPE	MAPE	Configuration
CycleGAN	5.45 $\pm$ 10.31	1.87E+14 $\pm$ 4.27E+15	2C – Rate 10%
DCGAN	4.99 $\pm$ 4.39	3.28 $\pm$ 3.08	2C – Rate 30%
DiscoGAN	3.81 $\pm$ 3.93	1.65 $\pm$ 5.74	1C and 3C– Rate 20%
Pix2Pix	**1.14** $\pm$ **2.89**	**0.24** $\pm$ **1.14**	1C – Rate 20%

In models such as DCGAN, DiscoGAN, and Pix2Pix, it is observed that the error tends to decrease with an increase in the rate of missing data. This may occur because the discriminator faces greater difficulty in distinguishing real samples from generated ones, which leads it to learn more robust representations and prevents it from easily "dominating" the generator. In this way, the generator is encouraged to improve its generalization ability. CycleGAN, on the other hand, did not exhibit this behavior, possibly due to differences in its architecture or training. Thus, the presence of missing data can act as an indirect form of regularization, promoting a healthier balance in the dynamics between generator and discriminator.

It is observed that, in most cases, the performance of models using 1 channel (complete series) and 3 channels (series decomposed into trend, seasonality, and noise) is quite similar, while the use of 2 channels (trend and seasonality) stands out for providing a more consistent improvement. This suggests that removing noise during decomposition helps to focus on the structural components of the series, facilitating learning. In addition, the noise removed in the 2-channel approach can be reincorporated into the forecast later, ensuring that the original variability of the series is preserved without compromising the stability of the training.

In summary, the models demonstrated good overall performance, with emphasis on the 2-channel configuration, where DCGAN, DiscoGAN, and Pix2Pix presented stable and consistent results. Table III shows the best results and their respective configurations. It can be observed that there is no superior model, because although Pix2Pix was the model with the best absolute performance and least sensitive to outliers, DCGAN proved to be more efficient with higher missing data rates. Therefore, determining the best model depends heavily on the application scenario, even warranting the use of more than one model for mixed scenarios.

V. CASE STUDY

To clarify the results presented in the Tables of Section IV, we observed high MAPE error rates exceeding 100, sometimes reaching values around 10^{13}. These large errors

TABLE IV
CYCLEGAN PERFORMANCE IN IMPUTING MISSING DATA WITHOUT USING EQUATION 5

Rate	10			20			30			40		
Channels	*1C*	*2C*	*3C*	*1C*	*2C*	*3C*	*1C*	*2C*	*3C*	*1C*	*2C*	*3C*
Metric	ASMAPE											
mean	18.14	16.47	18.14	16.83	16.78	16.83	16.74	16.17	16.74	16.73	14.96	16.73
std	14.46	12.99	14.46	14.07	12.92	14.07	13.99	12.93	13.99	13.97	13.08	13.97
min	0.89	3.91	0.89	0.88	3.23	0.88	0.89	2.98	0.89	0.88	2.86	0.88
25%	10.69	10.48	10.69	11.09	10.93	11.09	11.22	10.41	11.22	11.28	9.13	11.28
50%	15.89	14.11	15.89	13.84	14.50	13.84	13.72	13.67	13.72	13.67	11.93	13.67
75%	20.12	17.57	20.12	17.01	17.85	17.01	16.87	16.96	16.87	16.84	15.83	16.84
max	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
Metric	MAPE											
mean	6.31E+15	1.76E+14	6.31E+15	1.18E+16	3.78E+14	1.18E+16	1.79E+16	5.49E+14	1.79E+16	2.54E+16	6.59E+14	2.54E+16
std	1.96E+17	4.66E+15	1.96E+17	2.71E+17	1.00E+16	2.71E+17	3.50E+17	1.46E+16	3.50E+17	5.65E+17	1.60E+16	5.65E+17
min	0.16	0.65	0.16	0.33	1.20	0.33	0.54	1.86	0.54	0.69	2.48	0.69
25%	2.36	2.28	2.36	4.79	4.96	4.79	7.25	7.22	7.25	10.05	8.34	10.05
50%	3.65	3.22	3.65	6.28	6.94	6.28	9.16	9.88	9.16	12.92	11.19	12.92
75%	4.98	4.33	4.98	8.30	9.15	8.30	12.27	13.39	12.27	17.32	15.61	17.32
max	8.38E+18	1.76E+17	8.38E+18	1.13E+19	3.76E+17	1.13E+19	1.44E+19	5.29E+17	1.44E+19	2.37E+19	4.97E+17	2.37E+19

TABLE V
CYCLEGAN PERFORMANCE IN IMPUTING MISSING DATA USING EQUATION 5

Channels	1C				2C				3C			
Metric	SMAPE											
Rate	**10**	**20**	**30**	**40**	**10**	**20**	**30**	**40**	**10**	**20**	**30**	**40**
Mean ± std	5.73 ± 9.76	7.26 ± 9.89	7.91 ± 9.95	8.92 ± 10.59	5.45 ± 10.31	6.76 ± 11.84	7.49 ± 12.93	8.17 ± 13.06	5.73 ± 9.76	7.26 ± 9.89	7.91 ± 9.95	8.92 ± 10.59
Max / Min	98.42 / 0.00	98.79 / 0.00	97.43 / 0.00	98.05 / 0.00	98.27 / 0.94	98.84 / 1.16	99.10 / 1.15	98.61 / 1.31	98.42 / 0.00	98.79 / 0.00	97.43 / 0.00	98.05 / 0.00
25%	2.56	3.64	4.27	4.96	2.31	2.85	3.35	3.68	2.56	3.64	4.27	4.96
50%	3.35	4.63	5.23	6.07	2.85	3.62	4.20	4.62	3.35	4.63	5.23	6.07
75%	4.91	6.67	7.20	8.46	4.01	5.32	5.75	6.57	4.91	6.67	7.20	8.46
% >100	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Metric	MAPE											
Mean ± std	2.34E+15 ± 4.54E+16	5.25E+15 ± 1.02E+17	9.98E+15 ± 1.62E+17	1.51E+16 ± 2.97E+17	1.87E+14 ± 4.27E+15	5.09E+14 ± 1.12E+16	9.98E+14 ± 2.38E+16	1.45E+15 ± 3.18E+16	2.34E+15 ± 4.54E+16	5.25E+15 ± 1.02E+17	9.98E+15 ± 1.62E+17	1.51E+16 ± 2.97E+17
Max / Min	1.41E+18 / 0.00	3.91E+18 / 0.00	4.97E+18 / 0.00	9.79E+18 / 0.00	1.38E+17 / 0.19	3.14E+17 / 0.43	8.00E+17 / 0.70	8.22E+17 / 1.05	1.41E+18 / 0.00	3.91E+18 / 0.00	4.97E+18 / 0.00	9.79E+18 / 0.00
25%	0.51	1.46	2.59	3.99	0.46	1.13	2.01	2.96	0.51	1.46	2.59	3.99
50%	0.68	1.90	3.21	4.92	0.58	1.45	2.52	3.70	0.68	1.90	3.21	4.92
75%	1.02	2.76	4.54	6.89	0.83	2.14	3.50	5.33	1.02	2.76	4.54	6.89
% >100	1.24	2.81	3.62	4.59	0.92	1.67	2.16	2.86	1.24	2.81	3.62	4.59

TABLE VI
PIX2PIX PERFORMANCE IN IMPUTING MISSING DATA USING EQUATION 5

Channels	**1C**				**2C**				**3C**			
Metric	**SMAPE**											
Rate	**10**	**20**	**30**	**40**	**10**	**20**	**30**	**40**	**10**	**20**	**30**	**40**
Mean ± std	1.96 ± 3.45	2.96 ± 3.69	3.61 ± 4.01	4.35 ± 4.30	1.14 ± 2.89	1.74 ± 3.45	2.32 ± 3.94	2.84 ± 4.12	1.96 ± 3.45	2.96 ± 3.69	3.61 ± 4.01	4.35 ± 4.30
Max / Min	60.73 / 0.00	49.92 / 0.00	54.68 / 0.00	65.16 / 0.00	66.22 / 0.01	51.82 / 0.01	43.97 / 0.01	41.93 / 0.02	60.73 / 0.00	49.92 / 0.00	54.68 / 0.00	65.16 / 0.00
25%	0.85	1.40	1.77	2.26	0.49	0.79	1.08	1.37	0.85	1.40	1.77	2.26
50%	1.38	2.34	2.98	3.80	0.76	1.22	1.75	2.24	1.38	2.34	2.98	3.80
75%	1.98	3.36	4.16	5.17	1.06	1.79	2.50	3.21	1.98	3.36	4.16	5.17
% >100	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Metric	**MAPE**											
Mean ± std	0.42 ± 0.79	1.36 ± 4.66	2.44 ± 7.35	3.92 ± 11.99	0.24 ± 1.14	0.69 ± 1.51	1.39 ± 2.10	2.30 ± 2.97	0.42 ± 0.79	1.36 ± 4.66	2.44 ± 7.35	3.92 ± 11.99
Max / Min	17.29 / 0.00	189.62 / 0.00	295.24 / 0.00	487.75 / 0.00	45.84 / 0.00	48.52 / 0.00	50.15 / 0.01	65.34 / 0.01	17.29 / 0.00	189.62 / 0.00	295.24 / 0.00	487.75 / 0.00
25%	0.17	0.56	1.06	1.84	0.10	0.31	0.66	1.09	0.17	0.56	1.06	1.84
50%	0.28	0.95	1.83	3.07	0.15	0.49	1.06	1.84	0.28	0.95	1.83	3.07
75%	0.42	1.42	2.62	4.27	0.21	0.73	1.54	2.66	0.42	1.42	2.62	4.27
% >100	0.00	0.05	0.05	0.05	0.00	0.00	0.00	0.00	0.00	0.05	0.05	0.05

TABLE VII
DCGAN PERFORMANCE IN IMPUTING MISSING DATA USING EQUATION 5

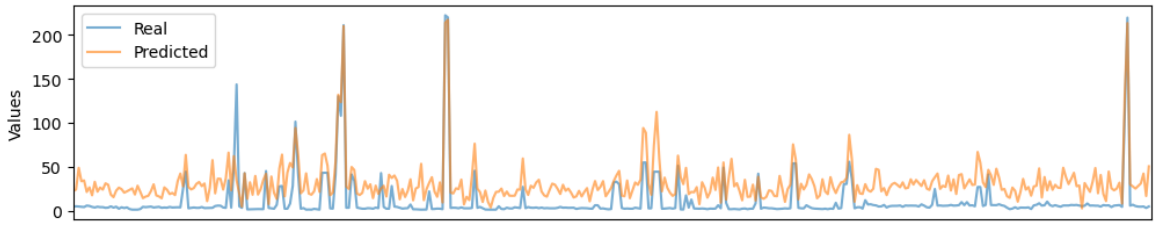
Channels	**1C**				**2C**				**3C**			
Metric	**SMAPE**											
Rate	**10**	**20**	**30**	**40**	**10**	**20**	**30**	**40**	**10**	**20**	**30**	**40**
Mean ± std	6.52 ± 5.04	6.52 ± 5.04	6.52 ± 5.01	6.52 ± 5.03	5.03 ± 4.39	5.06 ± 4.38	4.99 ± 4.39	5.03 ± 4.39	6.52 ± 5.04	6.52 ± 5.04	6.52 ± 5.01	6.52 ± 5.03
Max / Min	81.72 / 0.00	82.55 / 0.00	80.99 / 0.00	82.21 / 0.00	51.53 / 0.01	52.20 / 0.00	52.01 / 0.01	51.95 / 0.00	81.72 / 0.00	82.55 / 0.00	80.99 / 0.00	82.21 / 0.00
25%	4.38	4.34	4.38	4.44	2.47	2.48	2.44	2.45	4.38	4.34	4.38	4.44
50%	6.16	6.12	6.16	6.18	4.63	4.67	4.57	4.66	6.16	6.12	6.16	6.18
75%	8.14	8.12	8.05	8.05	6.41	6.44	6.29	6.35	8.14	8.12	8.05	8.05
% >100	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Metric	MAPE											
Mean ± std	1.53 ± 3.62	3.09 ± 8.87	4.64 ± 12.65	6.18 ± 15.98	1.10 ± 1.07	2.22 ± 2.06	3.28 ± 3.08	4.42 ± 4.16	1.53 ± 3.62	3.09 ± 8.87	4.64 ± 12.65	6.18 ± 15.98
Max / Min	141.46 / 0.00	362.75 / 0.00	513.49 / 0.00	643.18 / 0.00	23.52 / 0.00	42.55 / 0.00	64.10 / 0.00	89.23 / 0.00	141.46 / 0.00	362.75 / 0.00	513.49 / 0.00	643.18 / 0.00
25%	0.87	1.79	2.70	3.60	0.49	1.02	1.50	2.03	0.87	1.79	2.70	3.60
50%	1.29	2.59	3.91	5.28	0.97	1.98	2.93	3.97	1.29	2.59	3.91	5.28
75%	1.72	3.42	5.09	6.84	1.39	2.83	4.12	5.58	1.72	3.42	5.09	6.84
% >100	0.05	0.05	0.05	0.11	0.00	0.00	0.00	0.00	0.05	0.05	0.05	0.11

are caused by several factors: first, the model was trained on a dataset comprising time series with varying scales, as shown in Table I. Additionally, the dataset includes different types of series, such as energy consumption, ammonia concentration, and sunspot counts. Another important factor is the models' sensitivity, which, although mitigated, remains susceptible to data with different scales and to outliers.

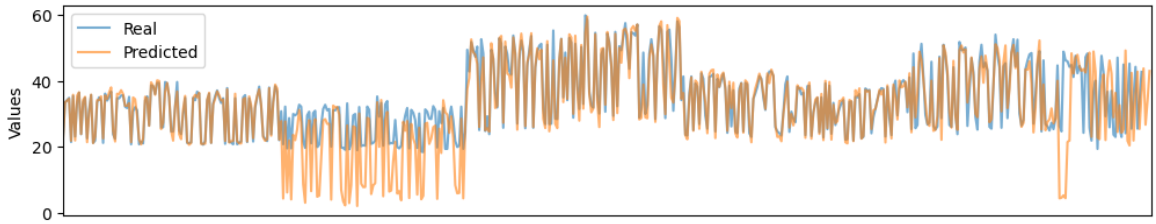
Regarding outliers, we observed that even after their

TABLE VIII
PERFORMANCE OF DICOGAN IN IMPUTING MISSING DATA USING EQUATION 5

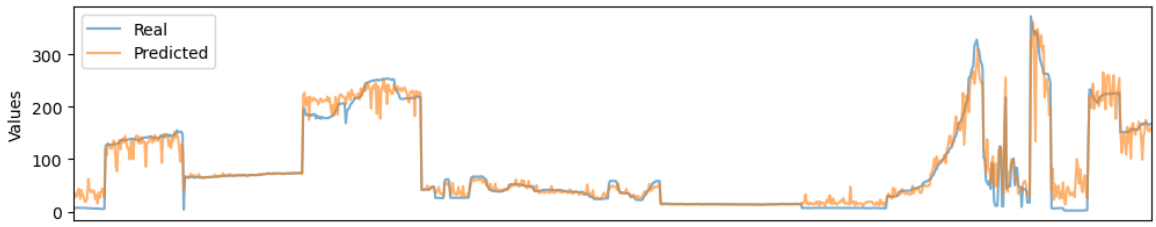
Channels	1C				2C				3C			
Metric	SMAPE											
Rate	10	20	30	40	10	20	30	40	10	20	30	40
Mean ± std	6.68 ± 6.89	3.81 ± 3.93	3.85 ± 3.89	4.00 ± 4.06	4.78 ± 4.71	3.97 ± 4.34	4.13 ± 4.60	3.86 ± 4.53	6.68 ± 6.89	3.81 ± 3.93	3.85 ± 3.89	4.00 ± 4.06
Max / Min	77.50 / 0.00	74.57 / 0.00	79.55 / 0.00	82.00 / 0.00	46.51 / 0.02	46.36 / 0.02	43.65 / 0.02	44.14 / 0.02	77.50 / 0.00	74.57 / 0.00	79.55 / 0.00	82.00 / 0.00
25%	2.25	2.04	2.23	2.21	2.41	1.90	1.84	1.69	2.25	2.04	2.23	2.21
50%	4.80	3.08	3.36	3.43	3.70	3.21	3.12	2.77	4.80	3.08	3.36	3.43
75%	8.52	4.52	4.43	4.68	5.93	4.83	4.94	4.89	8.52	4.52	4.43	4.68
% >100	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Metric	MAPE											
Mean ± std	1.29 ± 2.22	1.65 ± 5.74	2.60 ± 10.37	3.74 ± 16.40	1.00 ± 1.00	1.67 ± 1.76	2.58 ± 2.78	3.21 ± 3.57	1.29 ± 2.22	1.65 ± 5.74	2.60 ± 10.37	3.74 ± 16.40
Max / Min	77.46 / 0.00	238.32 / 0.00	431.66 / 0.00	683.58 / 0.00	19.60 / 0.00	34.02 / 0.01	50.19 / 0.01	64.06 / 0.01	77.46 / 0.00	238.32 / 0.00	431.66 / 0.00	683.58 / 0.00
25%	0.43	0.81	1.34	1.75	0.47	0.76	1.11	1.34	0.43	0.81	1.34	1.75
50%	0.91	1.23	2.01	2.76	0.75	1.33	1.91	2.23	0.91	1.23	2.01	2.76
75%	1.59	1.80	2.69	3.87	1.24	2.05	3.14	4.14	1.59	1.80	2.69	3.87
% >100	0.00	0.05	0.05	0.11	0.00	0.00	0.00	0.00	0.00	0.05	0.05	0.11



(a) series: ELD(MT_188)



(b) series: ELD(MT_093)



(c) series: ELD(MT_093)

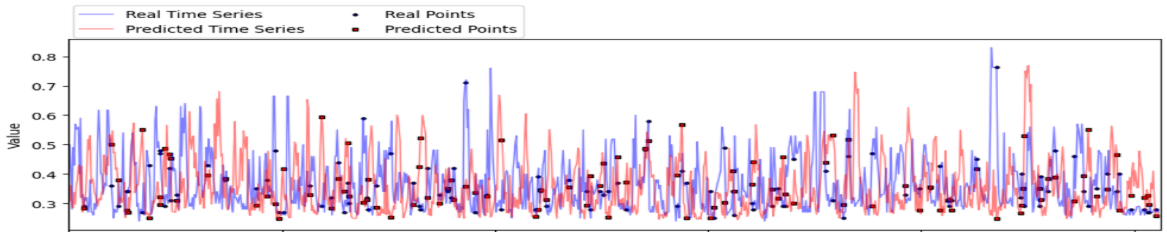
Fig. 4. Comparison of prediction results between CoroJA_RetinaNet and the baseline model.

removal, there are still behaviors resembling isolated outliers, which correspond to peak energy consumption data (Fig. 4a). Other patterns include mixed anomalies, featuring both individual anomalous points and entire anomalous intervals (Fig. 4b). These anomalous intervals are discussed in Almeida et al. [18], where such segments are considered sub-patterns of the series that may or may not repeat.

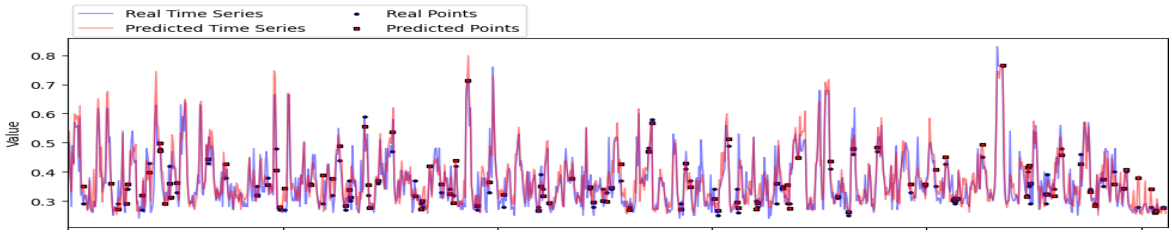
In a third example, in a series of energy consumption, we have multiscale patterns (Fig. 4) where there are several sections with zeros, zero intervals, others with high consumption, and others with oscillation.

Although these occurrences can cause high standard deviation and increase average error, they are a minority and do not compromise the robustness of the adversarial network methods used. Our approach with these networks reveals interesting findings on how imputing data with diverse sources, through more generalist models, can uncover valuable data characteristics, paving the way for new perspectives in developing more robust imputation techniques.

Finally, an example with the CycleGAN network, which initially showed very high MAPE values in imputation, is presented. By maintaining the entire pipeline but removing only the gradient correction based on the loss proposed in Equation 5, Fig. 5a shows one of the best predictions by the original model, with anomalous values in the real series complicating correct imputation. In contrast, Fig. 5b, presenting the corrected example using our approach, demonstrates significantly mitigated difficulty and more



(a) Prediction without the use of $\widehat{\mathcal{L}}_{\text{JS}}$



(b) Prediction using $\widehat{\mathcal{L}}_{\text{JS}}$

Fig. 5. Examples of poor predictions in anomalous series with CycleGAN

accurate imputation.

VI. Conclusion

TThis study advances the method of transforming time series into images, building on Almeida et al. [14] by adding temporal spacing as an explicit conditioning factor. We evaluated four generative networks for missing data imputation, training 240 models, and reframed time series recovery as an image reconstruction task.

Transforming series into images via separation of trend, seasonality, and noise produced strong results, especially in the 2C version. Training revealed a performance imbalance favoring the discriminator; however, higher missing-data rates improved generator performance as the discriminator's advantage diminished. The explicitly conditioned networks Pix2Pix and DiscoGAN proved to be more robust, especially when the series structure was preserved through component separation. Furthermore, by evaluating the 240 models per image window (i.e., data segments), we confirmed that the time series exhibit local patterns 18, which explains the high MAPE errors in some data windows. Despite these high values, our proposal to circumvent this problem had the desired effects, improving the stability of the models and, consequently, the quality of the imputation.

Finally, this research paves the way for data imputation in time series by framing it as an image reconstruction problem, particularly in the 2C version, where noise is discarded during the imputation process, allowing the model to focus on the core information. Subsequently, the noise is reintegrated without interfering with the imputation and without requiring the models to expend additional effort to learn or ignore it. This strategy enhances network performance by directing the learning process toward the essential patterns of the time series.

For future research, it is suggested to explore a wider range of models and GANs, and to combine the transformation proposed by almeida at al. [14] with Complete Ensemble Empirical Mode Decomposition with Adaptive Noise (CEEMDAN) to optimize channel expansion.Researchers wishing to apply our loss function should note that it was developed specifically for our problem. Although it may be effective in other contexts, it is crucial that researchers evaluate its suitability for the task at hand.

Acknowledgments

The authors acknowledge the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), Brazil - Finance Code 001, Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq)(Grant number 305253/2025-5), Brazil, and Fundação de Amparo à Pesquisa Desenvolvimento Científico e Tecnológico do Maranhão (FAPEMA) for the financial support.

Appendix A
Proof of the Main Theorem of the James–Stein Estimator

Theorem 1 (James and Stein 1961)**.** *Suppose that* $X_k \overset{ind}{\sim} \mathcal{N}(\mu_k, 1)$ *for* $k = 1, \ldots, d$*. Let* $\mu = (\mu_1, \ldots, \mu_d)$ *and* $\bar{X} = (X_1, \ldots, X_d)$ *be the sample mean. Let* $\hat{\mu}_{js}$ *be the James-Stein estimator. If* $d > 3$*, then:*

$$\mathbb{E}[\|\mu - \hat{\mu}_{js}\|^2] < \mathbb{E}[\|\mu - \bar{X}\|^2] \quad \forall \mu \in \mathbb{R}^d.$$

To prove this theorem , we will briefly use three main steps:

- Stein's lemma: Characterization of the Gaussian distribution.
- Stein's risk estimator: Unbiased estimator for the risk function.
- James-Stein estimator: The importance of shrinkage

Lemma 1 (Stein's lemma)**.** *Suppose that* $X \sim \mathcal{N}(\mu, \delta^2)$ *and let* $h: \mathbb{R} \to \mathbb{R}$ *be differentiable with* $\mathbb{E}[|h'(X)|] < \infty$*. Then:*

$$\mathbb{E}[(X - \mu)h(X) = \delta^2 \mathbb{E}[h'(X)].$$

Proof. Let $\mu = 0$ and $\delta^2 = 1$. The equation holds for $h(x)+c$, so take $h(0) = 0$. From this it follows that:

$$\begin{aligned}\int_0^\infty xh(x)e^{-\frac{x^2}{2}}\,dx &= \int_0^\infty x\left[\int_0^x h'(y)\,dy\right]e^{-\frac{x^2}{2}}\,dx \\ &= \int_0^\infty\int_0^\infty \mathbb{1}_{[y<x]}\,xh'(y)e^{-\frac{x^2}{2}}\,dy\,dx \\ &= \int_0^\infty h'(y)\left[\int_y^\infty xe^{-\frac{x^2}{2}}\,dx\right]dy \\ &= \int_0^\infty h'(y)e^{-\frac{y^2}{2}}\,dy.\end{aligned}$$

The change in the order of integration is justified by $\mathbb{E}[|h'(X)|] < \infty$. Similarly, and because it is symmetric, it is trivial that:

$$\int_{-\infty}^0 xh(x)e^{-\frac{x^2}{2}}\,dx = \int_{-\infty}^0 h'(y)e^{-\frac{y^2}{2}}\,dy$$

Therefore, if $X \sim \mathcal{N}(0,1)$, then:

$$\mathbb{E}[Xh(X)] = \mathbb{E}[h'(X)].$$

In general, $X = \mu + Z\delta \sim \mathcal{N}(\mu, \delta^2)$, from which it follows that $\mathbb{E}[(X-\mu)h(X) = \delta\mathbb{E}[Z\cdot h(\mu+\delta Z)] = \delta^2\mathbb{E}[\cdot h'(\mu+\delta)] = \delta^2\mathbb{E}[h'(X)]$.

□

Lemma 2 (Stein's lemma, multivariate case)**.** *Let $X_1,\ldots,X_d$ be such that $X_k \overset{ind}{\sim} \mathcal{N}(\mu_k,1)$ for $k = 1,\ldots,d$. Then $X = (X_1,\ldots,X_d) \sim \mathcal{N}(\mu, I_d)$ with $\mu = (\mu_1,\ldots,\mu_d) \in \mathbb{R}^d$. If $h : \mathbb{R}^d \to \mathbb{R}^d$ with $\mathbb{E}[||\nabla h(x)||] < \infty$, then the following holds:*

$$\mathbb{E}[(X-\mu)^T h(X)] = \mathbb{E}[tr(\nabla h(X)])$$

Proof. Using Lemma 1, we have that:

$$\begin{aligned}&\mathbb{E}[(X_k-\mu_k)h(X)] = \\ &\mathbb{E}\Big[\mathbb{E}[(X_k-\mu_k)h(X) \mid X_1,\ldots,X_{k-1},X_{k+1},\ldots,X_d]\Big]. \quad (8)\end{aligned}$$

By the condition of the lemma, this conditional expectation is equal to:

$$\mathbb{E}\left[\mathbb{E}\left[\frac{\partial h_k(X)}{\partial X_k}\,\middle|\, X_1,\ldots,X_{k-1},X_{k+1},\ldots,X_d\right]\right] = \mathbb{E}\left[\frac{\partial h_k(X)}{\partial X_k}\right].$$

Therefore, the unbiased risk estimate is

Adding for each $k = 1,\ldots,d$, we obtain:

$$\mathbb{E}[(X-\mu)^\top h(X)] = \mathbb{E}[\mathrm{tr}(\nabla h(X))].$$

□

Lemma 3 (Stein Risk Estimator)**.** *Suppose that $X_1,\ldots,X_d$ are independent with $X_k \sim \mathcal{N}(\mu_k,1)$ and that we estimate $\mu = (\mu_1,\ldots,\mu_d)$ via $\hat{\mu}(X) = X - h(X)$, where h is differentiable with $\mathbb{E}[\|\nabla h(X)\|] < \infty$. Define*

$$\hat{R}(X) = d + \|h(X)\|^2 - 2\,\mathrm{tr}(\nabla h(X)),$$

then the statistic $\hat{R}(X)$ is unbiased for the quadratic risk, that is,

$$\mathbb{E}[\hat{R}(X)] = \mathbb{E}[\|\hat{\mu}(X)-\mu\|^2] = R(\mu,\hat{\mu}).$$

Proof. By the multidimensional version of Stein's Lemma (Lemma 2),

$$\begin{aligned}R(\mu,\hat{\mu}) &= \mathbb{E}\left[\sum_{k=1}^d (X_k-\mu_k-h_k(X))^2\right] \\ &= \mathbb{E}\Bigg[\sum_{k=1}^d (X_k-\mu_k)^2 + \sum_{k=1}^d h_k^2(X) \\ &\quad - 2\sum_{k=1}^d (X_k-\mu_k)h_k(X)\Bigg] \\ &= d + \mathbb{E}[\|h(X)\|^2] - 2\mathbb{E}[(X-\mu)^\top h(X)] \\ &= d + \mathbb{E}[\|h(X)\|^2] - 2\mathbb{E}[\mathrm{tr}(\nabla h(X))] \\ &= \mathbb{E}[\hat{R}(X)]\end{aligned}$$

□

Finally, we will prove Theorem 1 using Lemma 3:

Proof. First,

$$R(\mu,\mathbf{X}) = \mathbb{E}[\|\mathbf{X}-\mu\|^2] = \sum_{k=1}^d \mathbb{E}[(X_k-\mu_k)^2] = \sum_{k=1}^d 1 = d.$$

For $\hat{\mu}_{\text{JS}}(\mathbf{X})$,

$$h(\mathbf{X}) = \frac{d-2}{\|\mathbf{X}\|^2}\mathbf{X}, \quad h_k(\mathbf{X}) = \frac{(d-2)X_k}{X_1^2+\cdots+X_d^2}.$$

And therefore,

$$\begin{aligned}\frac{\partial h_k(\mathbf{X})}{\partial X_k} &= \frac{d-2}{X_1^2+\cdots+X_d^2} - \frac{(d-2)X_k(2X_k)}{(X_1^2+\cdots+X_d^2)^2} = \\ &= \frac{d-2}{\|\mathbf{X}\|^2} - \frac{2(d-2)X_k^2}{\|\mathbf{X}\|_2^4},\end{aligned}$$

that is,

$$\mathrm{tr}(\nabla h(\mathbf{X})) = \frac{d(d-2)}{\|\mathbf{X}\|^2} - \frac{2(d-2)\sum_{k=1}^d X_k^2}{\|\mathbf{X}\|_2^4} = \frac{(d-2)^2}{\|\mathbf{X}\|^2}.$$

Furthermore, note that

$$\|h(\mathbf{X})\|^2 = \sum_{k=1}^d \left(\frac{(d-2)X_k}{\|\mathbf{X}\|^2}\right)^2 = \frac{(d-2)^2}{\|\mathbf{X}\|_2^4}\sum_{k=1}^d X_k^2 = \frac{(d-2)^2}{\|\mathbf{X}\|^2}.$$

.Therefore, the unbiased risk estimate is

$$\hat{R}(\mathbf{X}) = d + \frac{(d-2)^2}{\|\mathbf{X}\|^2} - 2\cdot\frac{(d-2)^2}{\|\mathbf{X}\|^2} = d - \frac{(d-2)^2}{\|\mathbf{X}\|^2}.$$

Finally, applying Stein's Risk Estimator, the following holds for all $\mu \in \mathbb{R}^d$,

$$R(\mu,\hat{\mu}_{\text{JS}}) = \mathbb{E}[\hat{R}(\mathbf{X})] = \mathbb{E}\left[d - \frac{(d-2)^2}{\|\mathbf{X}\|^2}\right] < d = R(\mu,\mathbf{X}).$$

□

Thus, in scenarios where the loss function is sensitive to outliers and produces unstable gradients that may occur due to the dominance of one network over another, the use of the James-Stein estimator is justified because it reduces the expected quadratic risk more robustly than the sample mean, mitigating the influence of extreme values.

REFERENCES

[1] L. A. Lynn, "Artificial intelligence systems for complex decision-making in acute care medicine: a review," *Patient safety in Surgery*, vol. 13, no. 1, p. 6, 2019. DOI: https://doi.org/10.1186/s13037-019-0188-2.

[2] O. B. Sezer, M. U. Gudelek, and A. M. Ozbayoglu, "Financial time series forecasting with deep learning : A systematic literature review: 2005–2019," *Applied Soft Computing*, vol. 90, p. 106181, 2020. DOI:https://doi.org/10.1016/j.asoc.2020.106181.

[3] S. Yang, H.-C. Chen, W.-C. Chen, and C.-H. Yang, "Student enrollment and teacher statistics forecasting based on time-series analysis," *Computational intelligence and neuroscience*, vol. 2020, 2020. DOI: https://doi.org/10.1155/2020/1246920.

[4] M. I. Habadi and C. P. Tsokos, "Statistical forecasting models of atmospheric carbon dioxide and temperature in the middle east," *Journal of Geoscience and Environment Protection*, 2017. DOI: https://doi.org/10.4236/gep.2017.510002.

[5] Y. Su, Y. Zhao, C. Niu, R. Liu, W. Sun, and D. Pei, "Robust anomaly detection for multivariate time series through stochastic recurrent neural network," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 2828–2837, 2019. DOI: https://doi.org/10.1145/3292500.3330672.

[6] A. A. Cook, G. Mısırlı, and Z. Fan, "Anomaly detection for iot time-series data: A survey," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6481–6494, 2019. DOI: https://doi.org/0.1109/JIOT.2019.2958185.

[7] R. J. Little and D. B. Rubin, *Statistical analysis with missing data*. John Wiley & Sons, 2019. DOI: https://doi.org/10.1002/9781119482260.

[8] H. Ahn, K. Sun, and K. Kim, "Comparison of missing data imputation methods in time series forecasting," *Computers, Materials & Continua*, vol. 70, no. 1, pp. 767–779, 2022. DOI: https://doi.org/10.32604/cmc.2022.019369.

[9] A. Y. Yıldız, E. Koç, and A. Koç, "Multivariate time series imputation with transformers," *IEEE Signal Processing Letters*, vol. 29, pp. 2517–2521, 2022. DOI: https://doi.org/10.1109/LSP.2022.3224880.

[10] J. Park, J. Müller, B. Arora, B. Faybishenko, G. Pastorello, C. Varadharajan, R. Sahu, and D. Agarwal, "Long-term missing value imputation for time series data using deep neural networks," *Neural Computing and Applications*, vol. 35, no. 12, pp. 9071–9091, 2023. DOI: https://doi.org/10.1007/s00521-022-08165-6.

[11] Z. Yang, Y. Li, and G. Zhou, "Ts-gan: Time-series gan for sensor-based health data augmentation," *ACM Transactions on Computing for Healthcare*, vol. 4, no. 2, pp. 1–21, 2023. DOI:https://doi.org/10.1145/3583593.

[12] F. Huang and Y. Deng, "Tcgan: Convolutional generative adversarial network for time series classification and clustering," *Neural Networks*, 2023. DOI:https://doi.org/10.1016/j.neunet.2023.06.033.

[13] L. Xu, L. Xu, and J. Yu, "Time series imputation with gan inversion and decay connection," *Information Sciences*, p. 119234, 2023. DOI:https://doi.org/10.1016/j.ins.2023.119234.

[14] M. M. Almeida, J. D. S. de Almeida, G. B. Junior, A. C. Silva, and A. C. de Paiva, "Univariate time series missing data imputation using pix2pix gan," *IEEE Latin America Transactions*, vol. 21, no. 3, pp. 505–512, 2023. DOI:https://doi.org/10.1109/TLA.2023.10068853.

[15] S. Barra, S. M. Carta, A. Corriga, A. S. Podda, and D. R. Recupero, "Deep learning and time series-to-image encoding for financial forecasting," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 3, pp. 683–692, 2020. DOI:https://doi.org/10.1109/JAS.2020.1003132.

[16] O. B. Sezer and A. M. Ozbayoglu, "Algorithmic financial trading with deep convolutional neural networks: Time series to image conversion approach," *Applied Soft Computing*, vol. 70, pp. 525–538, 2018. DOI:https://doi.org/10.1016/j.asoc.2018.04.024.

[17] J. Kihoro, K. Athiany, W. KH, *et al.*, "Imputation of incomplete nonstationary seasonal time series data," *Mathematical Theory and Modeling*, vol. 3, no. 12, pp. 142–154, 2013.

[18] M. M. Almeida, J. D. S. Almeida, D. B. P. Quintanilha, G. B. Júnior, and A. C. Silva, "A meta-learning based neural network and lstm for univariate time series missing data imputation," *Applied Soft Computing*, vol. 172, p. 112845, 2025. DOI:https://doi.org/10.1016/j.asoc.2025.112845.

[19] B. Sanwouo, C. Quinton, and R. Rouvoy, "Ts-pothole: automated imputation of missing values in univariate time series," *Neural Computing and Applications*, vol. 36, no. 36, pp. 22923–22955, 2024. DOI:https://doi.org/10.1007/s00521-024-10391-z.

[20] E. Brophy, Z. Wang, Q. She, and T. Ward, "Generative adversarial networks in time series: A systematic literature review," *ACM Computing Surveys*, vol. 55, no. 10, pp. 1–31, 2023. DOI:https://doi.org/10.1145/3559540.

[21] A. Al-Fakih, A. Koeshidayatullah, T. Mukerji, S. Al-Azani, and S. I. Kaka, "Well log data generation and imputation using sequence based generative adversarial networks," *Scientific Reports*, vol. 15, no. 1, 2025. DOI:https://doi.org/10.1038/s41598-025-95709-0.

[22] Y. Yin, Z. Yuan, I. M. Tanvir, and X. Bao, "Electronic medical records imputation by temporal generative adversarial network," *BioData Mining*, vol. 17, no. 1, p. 19, 2024. DOI:https://doi.org/10.1186/s13040-024-00372-2.

[23] T. Huang, P. Chakraborty, and A. Sharma, "Deep convolutional generative adversarial networks for traffic data imputation encoding time series as images," *International Journal of Transportation Science and Technology*, vol. 12, no. 1, pp. 1–18, 2023. DOI:https://doi.org/10.1016/j.ijtst.2021.10.007.

[24] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *CoRR*, vol. abs/1411.1784, 2014. DOI:https://doi.org/10.48550/arXiv.1411.1784 .

[25] P. Isola, J. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," *CoRR*, vol. abs/1611.07004, 2016. DOI:https://doi.org/10.48550/arXiv.1611.07004.

[26] J. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," *CoRR*, vol. abs/1703.10593, 2017. DOI:https://doi.org/10.48550/arXiv.1703.10593.

[27] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015. DOI: https://doi.org/10.48550/arXiv.1511.06434.

[28] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim, "Learning to discover cross-domain relations with generative adversarial networks," *CoRR*, vol. abs/1703.05192, 2017. DOI:https://doi.org/10.48550/arXiv.1703.05192.

[29] Andrews and Herzberg, "monthly-sunspots." Dataset on GitHub, 1985. Available in: https://raw.githubusercontent.com/jbrownlee/Datasets/master/monthly-sunspots.csv.

[30] K. Dunn, "Electricityusage." OpenMV.net Datasets, 2012. Available in: https://openmv.net/.

[31] A. Trindade, "ElectricityLoadDiagrams20112014." UCI Machine Learning Repository, 2015. DOI:https://doi.org/10.24432/C58C86.

[32] H. Fanaee-T, "Bike Sharing Dataset." UCI Machine Learning Repository, 2013. DOI: https://doi.org/10.24432/C5W894.

[33] K. Dunn, "Ammonia concentration." OpenMV.net Datasets, 2018. Available in: https://openmv.net/.

[34] K. Dunn, "Distillate flowrate." OpenMV.net Datasets, 2014. Available in: https://openmv.net/.

[35] U. Bandara, "Daily minimum temperatures in melbourne." Dataset on GitHub, 2023. Available in: https://github.com/upul/WhiteBoard/tree/master/data.

[36] J. Tukey, "Exploratory data analysis," 1977. DOI:https://doi.org/10.1007/978-3-319-43742-2_15.

[37] R. B. Cleveland, W. S. Cleveland, J. E. McRae, I. Terpenning, *et al.*, "Stl: A seasonal-trend decomposition," *J. off. Stat*, vol. 6, no. 1, pp. 3–73, 1990.

[38] J. Gu and B. Moon, "James-stein gradient combiner for inverse monte carlo rendering," in *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pp. 1–10, 2025. DOI:https://doi.org/10.1145/3721238.3730714.

[39] P. Orenstein, "Melhor que a média: ideias em estatística matemática." https://impa.br/palestras-de-divulgacao-35o-cbm/, 2024. Palestra apresentada no 35º Colóquio Brasileiro de Matemática, IMPA.

Mauricio M. Almeida holds a bachelor's degree in Mathematics from the Federal Institute of Education, Science, and Technology of Maranhão (IFMA). He has a master's degree and is pursuing a doctorate in Computer Science at the Federal University of Maranhão (UFMA). His research interests include the application of artificial intelligence in time series imputation and forecasting, the removal of coherent noise in geological data, and applications in medical imaging.

João D. S. de Almeida holds a degree in Computer Science from the Federal University of Maranhão (UFMA) (2007), a master's degree in Electrical Engineering from UFMA (2010), and a Ph.D. in Electrical Engineering from UFMA (2013). He is currently an Associate Professor I at UFMA. He coordinates the Vision and Image Processing Laboratory (VipLab-UFMA). He has experience in Computer Science, working mainly on the following topics: image processing, machine learning, ophthalmic medical images, and time series.

Geraldo B. Junior holds an undergraduate degree in Computer Science, a Master's degree in Electrical Engineering with emphasis on Computer Science, and PhD in Electrical Engineering with emphasis on Computer Science, all held at the Federal University of Maranhão (UFMA). He is an Associate Professor I at UFMA, a permanent member of the Post-graduation Programs of Master in Computer Science (PPGCC/UFMA) and Ph.D. in Computer Science / Association UFMA-UFPI. Has experience in Computer Science, working mainly on the following topics: computer vision, machine learning, deep learning, and medical image processing.

Aristófanes C. Silva holds a bachelor's degree in Computer Science, a master's degree in Electrical Engineering from the Federal University of Maranhão (UFMA), and a PhD in Computer Science from the Pontifical Catholic University of Rio de Janeiro. He is currently a Full Professor at UFMA. He has experience in Computer Science, with emphasis on Graphic Processing (Graphics), working mainly on the following topics: medical imaging and artificial intelligence.

Anselmo C. Paiva holds a BSc in civil engineering from Maranhão State Univeristy -Brazil in 1990; an MSc in civil engineering-Structures in 1993; and a PhD in Informatics from the Pontiphical Catholic University of Rio de Janeiro – Brazil in 2002. He is currently a Full Professor at the Informatics Department at the Federal University of Maranhão -Brazil. His current interests include medical image processing, geographical information systems and scientific visualization. He is the coordinator of the NCA-UFMA Applied Computing Center. Has experience in Computer Science, with emphasis on Graphics Processing, working mainly on the following topics: Virtual and Augmented Reality, Computer Graphics, GIS, Medical Image Processing and Volumetric Visualization. He is a member of SBC (Brazilian Computer Society) and ACM (Association for Computing Machinery).