IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 3 July 2026, accepted 19 July 2026, date of publication 22 July 2026, date of current version 29 July 2026.

*Digital Object Identifier 10.1109/ACCESS.2026.3715914*

RESEARCH ARTICLE

# Time Series-Based Anomaly Detection in Gas Turbines Using TimeWGAN-GP

**MARIA DEL CARMEN RUBIALES MENA[1], ANTONIO MUÑOZ SAN ROQUE[1], MIGUEL Á. SANZ-BOBI[1], (Senior Member, IEEE), DANIEL GONZALEZ-CALVO[2], AND TOMÁS ÁLVAREZ-TEJEDOR[2]**

[1]Institute for Research in Technology, ICAI School of Engineering, Universidad Pontificia Comillas, 28015 Madrid, Spain
[2]Enel Green Power and Thermal Generation, Endesa Iberia, 28042 Madrid, Spain

Corresponding author: Maria del Carmen Rubiales Mena (mcrubiales@comillas.edu)

**ABSTRACT** Reliable early detection of degradation in gas turbines is essential for preventing failures and optimizing maintenance strategies. However, traditional fault-detection methods struggle to capture the complex temporal dynamics that precede abnormal conditions. This study investigates the use of Generative Adversarial Networks (GANs) for time-series modeling and anomaly datasets augmentation, aiming to learn the evolving behavior of turbine systems directly from sensor data and to address the insufficient amount of failure data for anomaly detection. In the first stage, GANs are trained on historical time-series recordings of turbines operating under normal conditions to address data scarcity and enhance model robustness. Validation results show that the generated sequences maintain strong statistical consistency with the original data, accurately reproducing characteristic temporal patterns of healthy operation. The second stage focuses on anomaly detection by integrating real-time sensor inputs with the learned temporal representations from both real and synthetic datasets. The discriminator detects deviations from expected behavior, enabling early identification of abnormal trends and providing insight into degradation trajectories. Experimental findings demonstrate that the proposed framework captures dynamic transitions more effectively than threshold-based methods, predicts the progression of faults, and supports the construction of an expanded anomaly catalog by generating synthetic profiles that resemble documented failure modes. Overall, the approach improves early-warning capabilities and strengthens data-driven maintenance planning.

**INDEX TERMS** Anomaly detection, fault detection, gas turbine, generative adversarial network, machine learning.

## I. INTRODUCTION

While large volumes of reference data exist for turbines operating under normal conditions, the limited availability of fault data makes it difficult for traditional machine learning algorithms to learn the full spectrum of system behavior. GANs offer a solution by generating synthetic time series that emulate both normal and anomalous conditions, effectively augmenting the dataset and compensating for the lack of failure examples. This enriched dataset enables the model to learn more comprehensive behavioral patterns, improving its ability to detect and forecast anomalies. In the second stage of the methodology, these synthetic sequences are used to model the temporal evolution of faults, allowing for early detection and the identification of degradation trends before actual failures occur.

A Generative Adversarial Network (GAN) is a recent paradigm in Machine Learning (ML) algorithms. The basic idea of GANs deployment is for unsupervised ML techniques but also proved to be better solutions for semi-supervised and reinforcement learning. These factors all together enable GAN networks as comprehensive solutions in plenty of fields such as healthcare, mechanics, banking, face detection, traffic control, etc. [1]

These networks are used for generating new data, such as images or videos, made to look real. Generative networks

The associate editor coordinating the review of this manuscript and approving it for publication was Ahmadreza Montazerolghaem.

existed before GANs, but the Adversarial part adds value and new perspectives. This study proposes the use of GANs to address a critical limitation in gas turbine monitoring: the imbalance between abundant operational data and the scarcity of documented failure instances.

In this framework two models are trained at the same time: the Generative model and the Discriminative model. The Generative model tries to capture the data distribution, while the Discriminative one estimates the probability that some content was generated by the Generative model and did not come from the actual data. The Generative model tries to confuse the Discriminative model, by maximizing the probability that it makes a mistake [2].

These two models are typically implemented by neural networks, but they can be implemented with any form of differentiable system that maps data from one space to the other. The generator tries to capture the distribution of true examples for new data example generation. The discriminator is usually a binary classifier, discriminating generated examples from the true examples as accurately as possible. The optimization of GANs is a minimax optimization problem. The optimization terminates at a saddle point that is a minimum with respect to the generator and a maximum with respect to the discriminator. That is, the optimization goal is to reach Nash equilibrium. Then, the generator can be thought to have captured the real distribution of true examples [3].

Despite the increasing availability of high-frequency sensor data in industrial gas turbines, a persistent limitation for data-driven reliability methods is the severe imbalance between abundant recordings of normal operation and the scarce, often incomplete, documentation of failure events. This imbalance restricts the ability of conventional machine learning models to learn degradation trajectories and generalize early-stage anomalies, which are precisely the conditions in which anomaly detection is most valuable. Motivated by the need for earlier and more reliable identification of incipient faults, this work investigates GANs as a principled mechanism to learn realistic time-series representations of turbine behavior and to alleviate fault-data scarcity through controlled synthetic generation.

The main contributions of this study are threefold. First, a GAN-based time-series generation framework trained on operational sensor sequences was developed to augment limited failure data while preserving the statistical and temporal characteristics of turbine measurements. Second, the generated sequences, together with real observations, were leveraged to support anomaly detection by modeling the temporal evolution of degradation, enabling the identification of abnormal trends before failures fully manifest. Third, the generated data and the resulting anomaly detection capability were validated through quantitative and comparative analyses against reference datasets, demonstrating that the synthetic profiles are both statistically consistent and operationally meaningful for characterizing fault progression. These contributions provide a foundation for constructing a richer anomaly catalog and for enhancing proactive maintenance strategies in data-constrained industrial settings.

## II. STATE OF THE ART

GANs have emerged as a powerful tool in the field of machine learning, with significant applications in various domains, including industrial equipment. GANs were introduced by Ian Goodfellow et al. [5]. In the context of industrial equipment, GANs have been employed to enhance fault detection and diagnosis. For instance, Conditional GANs (CGANs) have been used to generate synthetic data that replicates real-world data, improving the accuracy of fault detection models in digital twins of industrial IIoT systems [6].

One of the significant challenges in industrial applications for failure diagnosis is the lack of sufficient data for training machine learning models. GANs address this issue by generating synthetic data that can augment existing datasets, thereby improving model performance. This approach has been particularly useful in scenarios involving the Industrial Internet of Things (IIoT), where data scarcity can hinder model accuracy [6].

This current application helps in training models to predict impending failures, thereby preventing costly downtime, and improving equipment reliability. In manufacturing, GANs have been applied to improve quality control processes. By generating high-fidelity synthetic data, GANs help in identifying defects and ensuring that products meet quality standards [7].

Despite their potential, the application of GANs in industrial equipment faces several challenges. The effectiveness of GANs depends on the quality of the data used for training. Poor-quality data can lead to inaccurate models and unreliable predictions [8]. Training GANs requires significant computational resources, which can be a limitation in industrial settings with constrained resources [9].

The future of GANs in industrial equipment looks promising, with ongoing research focused on addressing current challenges and exploring new applications. Key areas of interest include developing more stable and efficient training algorithms to enhance GAN performance and reliability [10], combining GANs with other machine learning techniques, such as reinforcement learning and transfer learning, to improve model robustness and applicability [9], enhancing the real-time capabilities of GANs for applications in predictive maintenance and fault detection, and implementing robust security measures to mitigate risks associated with synthetic data and ensure the integrity of GAN-based models.

GANs offer significant potential for advancing the field of industrial equipment, providing innovative solutions for fault detection, data augmentation, predictive maintenance, and quality control. So far, most of the applications have focused on augmenting a fault dataset [11], [12] but this article aims to explore the prognostication of dataset behavior using GANs.

While GANs have been widely utilized to generate synthetic data that enhances the size and diversity of training

datasets, their potential to model and predict future behavior based on learned temporal distributions remains significantly underexplored. Most existing applications focus on static data augmentation, aiming to balance datasets or simulate fault conditions for classification tasks. However, in industrial systems such as gas turbines, the real value lies in understanding how anomalies evolve over time, capturing the subtle transitions from normal operation to early signs of degradation and eventual failure. GANs, particularly those adapted for time series data, offer a unique opportunity to learn these dynamics and generate realistic sequences that reflect the progression of faults. By doing so, they can support not only detection but also augmentation of the failure datasets, enabling predictive maintenance strategies that anticipate failures before they occur. Unlocking this capability requires a shift from snapshot-based modeling to sequence-aware architectures, and from augmentation to simulation of future states grounded in operational history.

By leveraging the ability of GANs to model complex data distributions, this research proposes a novel approach to forecast the behavior of industrial equipment datasets. GANs can capture the underlying patterns and temporal dependencies within the data, enabling the generation of new data that reflects potential future states. This prognostic capability is particularly valuable for predictive maintenance and anomaly detection, where anticipating equipment behavior can lead to timely interventions and improved reliability. The application of GANs for dataset prognostication represents a significant advancement in the field, offering a robust framework for enhancing the predictive accuracy of industrial equipment models [13].

This rationale is consistent with prior GAN-based research on time-series analysis, which has shown that adversarial learning can capture temporal structure for both reconstruction and anomaly assessment. In particular, TadGAN [14]: Time Series Anomaly Detection Using Generative Adversarial Networks models temporal correlations through LSTM-based generators and critics and uses cycle consistency to improve time-series reconstruction, while MAD-GAN [15]: Multivariate Anomaly Detection for Time Series Data with Generative Adversarial Networks extends this perspective to multivariate settings by considering the full set of variables jointly in order to capture latent interactions and by combining discrimination and reconstruction in its anomaly score.

More broadly, a comprehensive review on GANs for time-series [16] signals highlights that preserving time dependence and evaluating generated sequences comprehensively remain central challenges in GAN-based time-series modeling. Taken together, these studies support the use of GANs not only as tools for synthetic data generation, but also as sequence-aware models capable of learning evolving operational patterns, which provides a strong basis for the iterative synthetic-data generation and retraining strategy adopted in this work

In the proposed GAN framework, the methodology begins with the generation of synthetic time series data based on historical observations of gas turbine operation.

This synthetic data is then incorporated back into the original dataset, creating an expanded and enriched input set. The GAN is retrained iteratively using this updated dataset, allowing the model to refine its understanding of normal and abnormal behavior. To structure this process, a moving window approach is applied, at each iteration, the GAN is trained on a segment of the modified dataset that includes the most recent generated observations. This means that the reference data evolves dynamically, with each loop adding new synthetic sequences that reflect the latest learned patterns. As a result, the GAN continuously adapts to the changing data landscape, improving its ability to simulate future behavior and detect early signs of degradation. This iterative learning mechanism is key to enabling the model not just to detect anomalies, but to prognosticate how those anomalies may develop over time, offering a powerful tool for predictive maintenance in gas turbine systems.

By incorporating the generated data into the training process, the GAN can better capture the temporal dependencies and distributional characteristics of the dataset, leading to more accurate forecasts of future behavior. As new data is included into the modified input, the generated output also considers previously generated data, not only the reference period. After a number of iterations, the input vector of the GAN will consist solely of generated data that has captured the nature of the original dataset, and the output will be the evolution of how the temperatures will continue to behave in future observations.

This approach not only enhances the ability of the model to predict potential faults but also provides valuable insights for predictive maintenance, anomaly detection of industrial equipment, ultimately contributing to the reliability and efficiency of industrial operations.

Anomaly detection, predictive maintenance, and fault datasets are integral elements of condition-based maintenance, where machinery behavior is continuously monitored to identify deviations that may indicate emerging failures. These processes rely on systematic data acquisition and processing to detect abnormal patterns and assess equipment health, forming the basis for timely maintenance decisions [17]. Predictive maintenance expands on these capabilities by incorporating prognostic models that estimate the remaining useful life of assets and anticipate future failures, thereby reducing unplanned downtime and optimizing operational reliability [18].

Within the broader field of prognostics and health management, data-driven and knowledge-based methods enable the evaluation of system degradation and facilitate early fault detection, which is essential for effective predictive maintenance strategies [17]. These methods support decision-making by providing insights into equipment status and by highlighting patterns associated with

developing faults, thus enhancing maintenance accuracy and efficiency [18].

The focus of the present research is on data augmentation of scarce fault datasets to improve the robustness and performance of models used for predictive maintenance and fault detection. By enriching limited training data, augmented datasets help strengthen anomaly-detection and prognostic algorithms, ultimately supporting more reliable maintenance decision-making in scenarios where fault data are insufficient.

## III. PREVIOUS ANALYSIS

To understand the operation of a gas turbine, it is crucial to assess the condition of its combustion chambers. Previous analyses [4] have demonstrated that the variability in the exhaust gas temperatures of the turbine accurately reflects the state of the combustion chambers and the turbine overall. By examining how these temperatures fluctuate and the symmetry in their measurements, it is possible to determine whether the turbine is experiencing a malfunction or operating correctly. As part of evaluating the effectiveness of the developed GAN model, it has been verified whether the same operating condition can be detected applying analysis methods from previous research [4] to the generated dataset.

In this previous analysis, the objective was to continuously monitor the combustion process in gas turbines and detect anomalies in the combustion chambers and measurement thermocouples. The study involved four gas turbines, each equipped with twenty-one thermocouples to measure exhaust gas temperatures. Later on, an analysis method that combines several machine learning models, including neural networks and principal component analysis (PCA), to characterize normal behavior and analyze actual temperature is developed. By applying PCA, a neural network and the comparison of the measurement of each thermocouple in each observation with the average of that observation and combining the results of the three analysis methods it was proven that the behavior of the gas turbine could be characterized [4].

The summary of this previous research is shown in FIGURE 1. In this current research, the Principal Component Analysis and the comparison with the observed average will be used to prove the effectiveness of the developed GAN model [4].

This article is divided in five sections: introduction, state of the art of GANs, anomaly dataset augmentation method based on generative adversarial networks, where the developed system is explained with examples, results and conclusions.

## IV. ANOMALY DATASET AUGMENTATION METHOD BASED ON GENERATIVE ADVERSARIAL NETWORKS

This section presents an overview of Conditional Generative Adversarial Networks (cGANs) and their application in developing a forecasting model for gas turbine exhaust gas temperatures. The proposed approach utilizes a categorized failure case as a reference to condition the generation process,

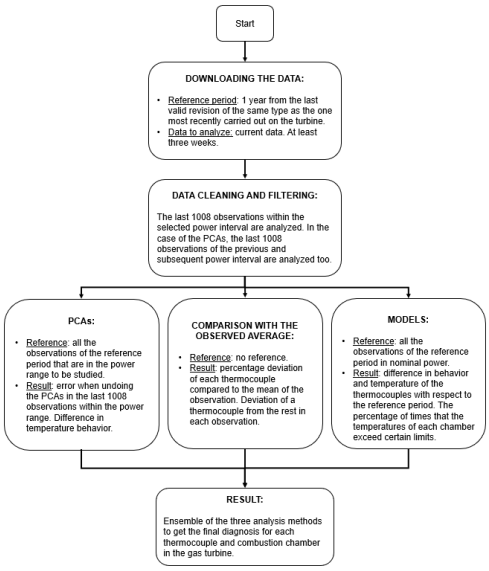



**FIGURE 1. Previously developed analysis of exhaust temperatures method [4].**

enabling the model to simulate the temperature evolution associated with specific failure scenarios.

The combustion chamber is a critical component of a gas turbine, as it is responsible for mixing fuel with air and igniting the mixture to produce the high-temperature, high-pressure gases that drive the turbine blades. Proper combustion is essential for the efficient operation of the turbine. Poor combustion can lead to several serious issues, including significant economic losses due to increased fuel consumption and maintenance costs, disruptions in power supply to customers, and potential damage to the turbine components, which can result in costly repairs and downtime. Ensuring optimal combustion is therefore vital for the reliability and efficiency of gas turbine operations.

Additionally, a few fault signatures are available and will be utilized as seeds for training the proposed GAN. These fault signatures represent specific examples of anomalous behaviors in industrial equipment, and their inclusion in the training process will enable the GAN to learn the distinctive characteristics of faults.

By using these signatures as starting points, the GAN can generate synthetic data that reflects a variety of fault scenarios, thereby enhancing the ability of the model to detect and predict anomalies in the equipment. This approach not only expands the dataset available for training but also strengthens the accuracy and robustness of the model in identifying potential faults.

This continuous enrichment of the training data not only improves the model's exposure to diverse operational patterns but also enhances its ability to recognize subtle deviations that may indicate early-stage faults. By learning from both real and synthetic sequences, the GAN becomes more resilient to noise and variability, ultimately leading to more reliable and timely fault data augmentation.

### A. GENERATIVE ADVERSARIAL NETWORKS

As discussed before, a GAN comprises two neural networks: a generator and a discriminator. The generator network receives a random distribution as input and generates a sample that closely approximates the original data distribution. The random distribution typically originates from a Gaussian distribution, which is used to ensure diverse and realistic data generation.

The discriminator network assesses the generated sample in comparison to the original data, producing an output value between 0 and 1. A value near zero signifies that the generated sample is fake, whereas a value near one signifies that it is real. Both networks undergo independent training. The GAN framework is particularly straightforward when both models are implemented as multilayer perceptron.

Conditional GANs (cGANs) enhance the performance of GANs by incorporating additional information, such as class labels or other data such as the past observations of the variables, into the model. In the generator, the input noise p(z) is combined with the label y or the conditioned data to create a joint hidden representation. Similarly, in the discriminator, both the input x and the label y are used as inputs to the discriminative function. This approach allows cGANs to generate higher quality results by leveraging the extra information provided [4].

cGANs can be trained on labeled datasets, allowing them to assign specific labels to each generated instance. For example, while an unconditional GAN trained on the MNIST dataset, which contains images of handwritten digits from 0 to 9, generates random digits, a conditional MNIST GAN can be directed to generate a specific digit. In progressive GANs, the generator starts by producing a low-resolution image, with subsequent layers progressively adding more detail. This method accelerates GAN training compared to non-progressive GANs and enables the production of high-resolution images [12], [20].

cGANs serve as a key inspiration for the developed system, as they offer the ability to accelerate GAN training while targeting specific conditions and variables. By leveraging the principles of cGANs, the developed system aims to enhance the efficiency of the training process and ensure that the generated outputs meet various predefined criteria. This approach not only speeds up the generation of high-quality data but also allows for greater control over the characteristics of the produced instances, making the system more versatile and effective in addressing diverse requirements.

### B. COMPUTATIONAL RESOURCES

The computational requirements and execution characteristics of the proposed model indicate that the system is lightweight and suitable for deployment in resource-constrained environments. All experiments were conducted on a consumer-grade laptop equipped with a 2.3 GHz Intel Core i5 CPU (4 cores), Intel Iris Plus Graphics 655 (1536 MB), and 8 GB of RAM. Despite the modest hardware, the model, which comprises 42.773 parameters (42.005 trainable), successfully completed training and data generation without the need for specialized accelerators such as GPUs or TPUs. This demonstrates that the approach is not dependent on high-performance computing infrastructure.

The dataset was processed using mini-batches of 64 samples with 21 input features each, and the model required 250 epochs to generate each output row. Therefore, a window of 64 past timesteps was used to generate the next 1 step. Since the selected moving window (64 observations) represents only a small fraction of the total dataset length (1080 observations), most generated samples are effectively conditioned on real data segments rather than previously generated outputs. This design choice limits the propagation of synthetic information across iterations and reduces the impact of recursive self-feeding. Consequently, the generation process remains strongly anchored to real observations, mitigating cumulative drift and ensuring that the conditioning primarily reflects genuine data patterns rather than compounded synthetic deviations.

Producing 1.080 rows (representing one operational week in the original dataset) required approximately two hours of wall-clock time. Given that this full generation cycle corresponds to a multi-day period of real-world operation, the computational overhead remains manageable. The time-to-output ratio suggests that, in practice, the model could be deployed in offline or near-real-time settings without imposing significant operational delays.

Moreover, the relatively small number of parameters ensures a reduced memory footprint, facilitating integration into edge devices, local servers, or embedded systems with limited computing capabilities. The absence of dependence on high-power GPUs also implies lower energy consumption and reduced deployment cost, which are relevant considerations for industrial environments where scalability, availability, and operational efficiency are priorities.

The results suggest that the proposed model architecture is feasible for deployment in industrial scenarios. Its low computational demand, modest training time, and ability to operate on standard hardware support practical implementation in both development and production contexts, including installations where computational resources are constrained or cost-sensitive.

At this stage, the primary objective of the method is to enrich existing datasets for downstream tasks such as model training, validation, and robustness analysis, where strict real-time constraints are not required. The suitability of the approach for online or near-real-time applications would depend on substantial optimization, including parallelized generation strategies, hardware acceleration (e.g., GPUs), or architectural modifications.

Future work will specifically investigate these aspects by leveraging more advanced computational resources and optimized inference pipelines, with the aim of assessing the feasibility of integrating the method into early online

warning systems that operate within typical turbine sampling intervals.

### C. USED DATASET

The dataset employed in this study is confidential and therefore cannot be publicly released. It originates from industrial gas turbine operations and has been made available exclusively for research under non-disclosure constraints. Despite this limitation, the characteristics, structure, and relevant properties of the data can be fully described to ensure methodological transparency. The dataset encompasses operational measurements from four independent gas turbines of type 6FA, all located at geographically close industrial sites. These turbines typically operate within a power range of 55 to 75 MW, representing standard loading conditions for this turbine class.

The recorded variables correspond to the exhaust temperature profile of each turbine, captured through 21 thermocouples installed uniformly around the exhaust section. These sensors provide a detailed thermal representation of the exhaust region, enabling precise monitoring of combustion uniformity, thermal imbalance, and early fault signatures associated with degradation phenomena. Temperature readings were collected at 10-minute intervals, providing a high-resolution temporal dataset capable of supporting anomaly detection, predictive maintenance models, and data-augmentation processes.

The historical record spans a continuous four-year period, covering January 2020 to October 2024. Throughout this interval, no null or missing sensor observations were detected in the available operational segments. However, some discontinuities are present in the data due to scheduled maintenance activities and turbine downtime periods. These gaps are inherent to regular industrial operation and vary between turbines, as each unit follows a different maintenance schedule. Notably, the total downtime per turbine represents less than 10% of the full period, ensuring the availability of a sufficiently large and representative dataset for learning operational behavior and historical fault patterns.

In terms of preprocessing, the dataset required only minimal manipulation prior to its use in training the generative model. All temperature variables were normalized to the [0, 1] range to ensure stable training of the GAN architecture and consistent scaling across turbines. After the generation of synthetic sequences, the data were rescaled back to their original units using the inverse normalization transformation. To distinguish healthy from faulty operating periods, the dataset was segmented according to maintenance logs and operational annotations provided by on-site personnel, enabling a clear separation between periods of correct turbine behavior and periods affected by known faults. Beyond this segmentation and the normalization step, no additional preprocessing, filtering, or outlier removal was applied, preserving the natural statistical characteristics of the original operational data.

The overall dataset therefore provides a robust basis for analyzing exhaust thermal behavior, generating synthetic fault sequences, and evaluating data-augmentation techniques designed to support predictive-maintenance and fault-detection applications.

### D. DEVELOPED MODEL

This research focuses on prognosticating the performance of the turbine based on previous observations of exhaust gas temperatures. This represents a significant advancement in the field of GANs as it emphasizes their application to industrial equipment, specifically through time series analysis, and subsequently in the diagnosis of this fault scenarios using previously proven effective methods.

In the developed system, the generator model is designed to create synthetic data samples from random noise. The input layer accepts a noise vector of 21 variables, corresponding to the number of thermocouples placed to measure the temperatures of the exhaust gases of the combustion chambers of the analyzed turbines. This choice ensures that the stochastic input space of the generator is structurally aligned with the multivariate nature of the target time series, allowing the model to learn joint temporal dependencies across all measurement channels. By matching the dimensionality of the noise vector to the number of thermocouples, the generator is encouraged to produce coherent and physically plausible temperature patterns that reflect the underlying spatial relationships of the combustion system. This input is processed through a series of dense layers, each followed by a LeakyReLU activation function and batch normalization to stabilize the training. The final layer uses a 'tanh' activation function to produce the output, which matches the input shape.

The discriminator, on the other hand, is tasked with distinguishing between real and synthetic data samples. It takes two inputs: the reference dataset for discrimination and the noise vector for training. These inputs are concatenated and passed through dense layers with LeakyReLU activations. The final layer outputs a single value, indicating whether the input is real or fake.

Before the training loop begins, hyperparameters for both the generator and discriminator models are selected using a hyperparameter optimization algorithm. This algorithm, a simplified version of the final GAN architecture, explores various hyperparameter ranges and selects the optimal configuration to ensure effective GAN performance during training [21], [22].

As shown in FIGURE 2, the training process alternates between training the discriminator and the generator based on the same moving period of time from the input vector. Random noise samples are generated and fed into the generator to produce fake samples. The discriminator is trained on both real and fake samples, with appropriate labels. Then, the generator is trained via the GAN model to fool the discriminator.

The input vector consists of a moving period of time of the input vector. In each iteration, the new input vector starts on the second observation of the previous one, treating the input vector and the generated output as a temporal series. In the first iteration the input vector is formed by a dataset of real observations of a gas turbine, which are used as reference, in both regular and faulty operating conditions.

Conditioning is implemented in both the generator and the discriminator. Specifically, the model is a conditional GAN in which the conditioning variable corresponds to the past trajectory of the target variable. That is, the generator receives as input not only a noise vector but also a sequence of past observations, which are concatenated and fed through the first layer of the network. The discriminator is analogously conditioned by receiving the same historical window jointly with either real or generated samples. This design ensures that the synthetic samples produced by the GAN are explicitly dependent on the preceding behavior of the process, rather than on categorical labels.

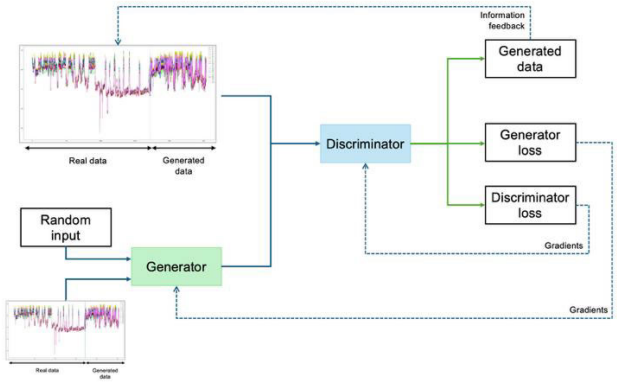



**FIGURE 2.** GAN based developed system.

A gradient penalty is applied to enhance the stability and performance of the GAN. This penalty is particularly useful in Wasserstein GANs (WGANs), where it helps to address issues related to vanishing or exploding gradients. The gradient penalty works by penalizing the gradients of the discriminator when their norm deviates significantly from a specified value, typically one. Conditional GANs (cGANs) provide a foundation for the system by enabling targeted generation based on specific labels. Building on this concept, Wasserstein GANs (WGANs) were chosen to further improve the stability and quality of the training process, ensuring more reliable and high-quality data generation.

This regularization technique ensures that the gradients of the discriminator remain within a reasonable range, promoting smoother and more stable training dynamics. By incorporating the gradient penalty, the GAN can achieve better convergence and generate higher quality synthetic data [23].

The model is trained following the Wasserstein GAN formulation, using the critic as a learned approximation of the Wasserstein-1 distance between the real and generated data distributions. Stability was ensured through Batch Normalization in the generator and tuned optimization hyperparameters. Let $D(\cdot)$ denote the critic output and $G(z)$ the generator output for a noise vector $z \sim N(0, I)$, and let $x$ denote a real sample. The critic receives paired inputs $(x \mid z)$ or $(G(z),z)$, and the training uses labels $-1$ for real samples and $+1$ for generated samples. The implemented Wasserstein loss takes the form $\mathcal{L}(y, \hat{y}) = \mathbb{E}[y \cdot \hat{y}]$, where $\hat{y} = D(\cdot)$.

The critic is updated by minimizing the objective

$$\mathcal{L}_D = \frac{1}{2}(\mathbb{E}_{x,z}[+1 \cdot D(G(z), z)] \underbrace{\quad}_{\text{fake term}} + \mathbb{E}_{x,z}[-1 \cdot D(x, z)]) \underbrace{\quad}_{\text{real term}} = \frac{1}{2}(\mathbb{E}[D(G(z), z)] - \mathbb{E}[D(x, z)]).$$

which corresponds to maximizing the difference between critic scores for real and generated samples, consistent with the Wasserstein-1 distance. The generator is trained with the critic's parameters frozen, and its objective is

$$\mathcal{L}_G = -\mathbb{E}_z[D(G(z), z)],$$

which encourages the generator to produce data that receives higher critic scores and thus reduce the estimated Wasserstein distance. This alternating optimization between critic and generator updates constitutes the core training loop and governs how both networks interact to align the generated and real data distributions.

In the WGAN-GP framework, the discriminator functions as a critic rather than a probabilistic classifier. The critic outputs a real-valued, unbounded score that reflects how closely the input resembles the true data distribution. Higher scores indicate that the sample is more consistent with real observations, whereas lower scores suggest the opposite. This continuous scoring mechanism is fundamental to the estimation of the Wasserstein distance and governs how the generator is optimized during training.

Following adversarial training, the discriminator is no longer used in any part of the generation process. All synthetic data produced for the augmentation task are generated exclusively by the trained generator, without further evaluation, refinement, or feedback from the discriminator. This ensures a complete separation between the training phase, where both networks interact, and the inference phase, which relies solely on the lightweight generator model. During iterative sequence generation, the generator outputs one new instance per loop, and only this network is executed. The discriminator remains inactive throughout inference, making the procedure computationally efficient and suitable for deployment settings where low-overhead generator-only execution is required.

Once training is complete, the generator can produce new synthetic data samples by taking random noise vectors as input. These generated variables are added to the entry vector used for training both in the discriminator and in the generator. This approach allows the system to generate new samples based on the evolution of real data and the newly generated samples, representing a possible evolution of the studied variables.

Finally, all generated samples are then rescaled to match the original data range using the MinMaxScaler. The final synthetic data is stored in a vector for further use. This approach ensures that the GAN model can generate high-quality synthetic data that closely resembles the real data.

The final output of the model is a newly generated time series sample that represents the predicted evolution of the turbine's operational behavior over time. The next step in validating the model involves demonstrating that these generated sequences closely resemble real sensor data, ensuring that the synthetic outputs are both statistically consistent and operationally meaningful before they are used for anomaly detection.

To mitigate the risk of error accumulation inherent in iterative generation pipelines, the method was explicitly designed to produce only one new instance per loop. This incremental strategy limits the amount of generated data reintroduced into the model at any given time, thereby reducing the likelihood of compounding deviations. Additionally, only the later stages of training involved a higher proportion of synthetic data compared to real data.

An analysis of these later iterations showed that the generated instances remained consistent with those produced during the initial loops. In particular, their empirical distributions, as well as their mean and standard deviation across key variables, closely matched the statistics observed in the earliest generated samples. This stability across iterations suggests that the model does not exhibit meaningful drift and maintains robustness even as synthetic data progressively becomes a larger share of the training input.

Regarding the technical characteristics of the proposed model it is based on a Wasserstein GAN architecture tailored to the 21-dimensional operational variables used in this study. The generator receives a 21-element input vector sampled from a standard normal distribution and processes it through a fully connected network composed of two hidden layers. Specifically, the architecture follows the sequence: a dense layer with 128 units, a LeakyReLU activation with $\alpha = 0.01$, Batch Normalization with momentum 0.8, followed by a second dense layer of 256 units with the same activation and normalization settings. The output layer is a fully connected projection with 21 units and a tanh activation, matching the $[-1, 1]$ scaling applied to all input features. The discriminator (critic) takes two inputs, the candidate 21-dimensional data sample and the same 21-dimensional noise vector, and concatenates them before feeding them to two hidden layers with 128 and 64 neurons, respectively. Each layer is followed by a LeakyReLU activation ($\alpha = 0.01$), and the final output is a single linear neuron, consistent with the Wasserstein formulation.

The structure of the generator is detailed in Table 1 which outlines each layer, its output dimensions, and the number of parameters involved in the model. This architecture combines dense transformations, activation functions, and batch normalization steps to progressively map the input space into the desired output domain. The overall configuration comprises 42,773 parameters, of which 42,005 parameters are trainable and 768 parameters are non-trainable, reflecting the presence of normalization statistics that remain fixed during optimization.

**TABLE 1. Structure of generator.**

| Layer (type) | Output Shape | Param # |
|---|---|---|
| dense (Dense) | (None, 128) | 2,816 |
| leaky_re_lu (Le | (None, 128) | 0 |
| batch_normalization | (None, 128) | 512 |
| dense_1 (Dense) | (None, 256) | 33,024 |
| leaky_re_lu_1 | (None, 256) | 0 |
| batch_normalization_1 | (None, 256) | 1,024 |
| dense_2 | (None, 21) | 5,397 |

The structure of the discriminator is summarized in Table 2, which details each layer, its output shape and number of parameters. This architecture comprises a sequence of input, concatenation, dense, and activation layers that collectively define the model's computational flow.

The configuration results in a total of 13,825 parameters, all of which are trainable, indicating that every component contributes to the learning process during optimization.

**TABLE 2. Structure of discriminator.**

| Layer (type) | Output Shape | Param # |
|---|---|---|
| input_layer_1 (InputLayer) | (None, 21) | 0 |
| input_layer_2 (InputLayer) | (None, 21) | 0 |
| concatenate (Concatenate) | (None, 42) | 0 |
| dense_3 (Dense) | (None, 128) | 5,504 |
| leaky_re_lu_2 (LeakyReLU | (None, 128) | 0 |
| dense_4 (Dense | (None, 64) | 8,256 |
| leaky_re_lu_3 (LeakyReLU) | (None, 64) | 0 |
| dense_5 (Dense) | (None, 1) | 65 |

Training employs the Wasserstein loss function $E[y \cdot \hat{y}]$ and the Adam optimizer with a learning rate of $2.0374 \times 10^{-4}$ and $\beta_1 = 0.4$ for both generator and critic, while all other optimizer parameters remain at TensorFlow defaults. No gradient penalty was applied, corresponding to a gradient-penalty coefficient of zero; stability was instead supported through Batch Normalization in the generator and careful tuning of the optimization parameters. The batch size was set to 64, and all feature variables were normalized to the $[-1, 1]$ range to ensure compatibility with the generator's tanh output activation.

A structured hyperparameter-search procedure was conducted prior to full-scale training. To ensure efficient exploration, candidate configurations were evaluated using a reduced subset of the dataset and shorter training intervals.

The search covered ranges for learning rate, $\beta_1$, hidden-layer widths, Batch Normalization momentum, LeakyReLU negative slope, and batch size. Candidate models were assessed according to (i) training stability of the critic and generator losses, and (ii) similarity between generated and real data distributions, evaluated through comparison of variable-wise empirical distributions, means and standard deviations. The selected configuration provided stable critic–generator dynamics and produced synthetic samples whose statistical properties consistently matched those of the real inputs.

The training procedure described in CODE FRAGMENT 1 outlines the adversarial learning process adopted to generate synthetic time-series data. The method operates by iteratively training a generator and a discriminator within a sliding window framework, enabling the model to capture local temporal dependencies present in the data. At each step, the discriminator is trained to distinguish between real samples extracted from the current window and synthetic samples produced by the generator from random noise, while the generator is simultaneously optimized to produce outputs that are indistinguishable from real data according to the discriminator.

The use of Wasserstein loss promotes stable convergence during this adversarial process. By advancing the window across the dataset and retaining a representative generated sample from each iteration, the approach constructs a continuous synthetic sequence that preserves the statistical structure of the original data. Finally, the generated samples are rescaled to the original domain, yielding an augmented dataset suitable for downstream applications.

```
INPUT:
   Time-series dataset X
   Batch size B
   Number of epochs E
  Seed = 42

OUTPUT:
   Synthetic dataset S

BEGIN

1. Normalize dataset X to range [-1, 1]
2. Initialize Generator G
3. Initialize Discriminator D
4. Define Wasserstein loss
5. Initialize empty list S
6. FOR each window i in X of size B DO

    real_samples ← X[i : i + B]

    FOR epoch = 1 TO E DO

      # --- Train Discriminator ---

      Sample random noise Z (size B)
      fake_samples ← G(Z)
      real_labels ← -1
      fake_labels ← +1
      Train D on (real_samples, Z) with real_labels
      Train D on (fake_samples, Z) with fake_labels

      # --- Train Generator ---
      Sample new noise Z
      target_labels ← -1   # generator tries to fool
discriminator
      Freeze D
      Train G via GAN using Z and target_labels
      Unfreeze D

    END FOR

    Append last generated sample to S

  END FOR

7. Trim S to desired length if needed
8. Apply inverse normalization to S
9. RETURN S

END
```

**CODE FRAGMENT 1. Pseudo-code of training loop.**

## V. ABLATION STUDY

Ablation studies are commonly employed to assess the individual contribution of the main components of a machine learning architecture and to determine whether the observed performance improvements can be attributed to specific design choices rather than to the model complexity alone. In the context of GAN-based time-series generation, such analyses are particularly relevant because several interacting elements, including conditioning mechanisms, temporal context length, and adversarial learning strategies, jointly influence the quality of the generated sequences. Therefore, a systematic ablation analysis was conducted to evaluate the impact of the principal components of the proposed framework and to quantify their contribution to the overall generation performance.

The objective of this study is to verify whether the selected architectural and training configurations effectively contribute to preserving the statistical and temporal characteristics of the original turbine datasets.

The ablation study evaluates the contribution of the main architectural components of the proposed framework. Unlike conventional conditional GANs, the proposed framework does not rely on explicit conditioning variables or class labels. Instead, conditioning is implicitly introduced through the sliding-window formulation, where the generator is trained using temporally ordered observations that provide contextual information about the recent evolution of the process.

Consequently, the ablation study focuses on the influence of the temporal conditioning window and on the adversarial training strategy. Specifically, the complete WGAN-GP model is compared against three simplified variants: a reduced temporal context (Window = 32), a standard GAN trained with binary cross-entropy loss (Vanilla GAN), and a WGAN model without gradient penalty.

**TABLE 3. Results of ablation study.**

| Configuration | KLD | JSD | WD | MAE relative | RMSE relative |
|---|---|---|---|---|---|
| **Full Model** | 1.5542 | 0.31648 | 9.72593 | 0.05579 | 0.07388 |
| **Window=32** | 1.6198 | 0.33029 | 9.83470 | 0.04968 | 0.067482 |
| **Vanilla GAN** | 1.6182 | 0.33986 | 12.7993 | 0.05902 | 0.078818 |
| **No Gradient Penalty** | 1.8398 | 0.42397 | 18.3028 | 0.07127 | 0.093021 |

To ensure a comprehensive evaluation, the ablation analysis was performed using the same set of quantitative metrics employed throughout this work. Distribution-based measures, including Kullback-Leibler Divergence (KLD), Jensen-Shannon Divergence (JSD), and Wasserstein Distance (WD), were used to assess the similarity between the statistical distributions of the generated and original datasets. In parallel, relative Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) were considered to quantify the average deviation between generated and reference observations. Together, these metrics provide complementary perspectives on synthetic data quality, evaluating both distributional consistency and pointwise agreement and thereby enabling a balanced assessment of the contribution of each architectural component to the overall performance of the proposed framework.

The results of the ablation study are presented in Table 3. Overall, the complete WGAN-GP configuration achieved the best balance between distributional fidelity and reconstruction accuracy. In particular, it obtained the lowest KLD (1.554), JSD (0.316), and Wasserstein Distance (9.726), indicating superior preservation of the statistical properties of the original dataset. These results suggest that the combination of the Wasserstein objective, gradient penalty regularization, and the selected temporal conditioning window provides the most effective representation of the underlying data distribution.

Reducing the temporal context from 64 to 32 observations resulted in a slight increase in KLD, JSD, and Wasserstein Distance, indicating a small loss of distributional consistency. Although this configuration achieved marginally lower MAE and RMSE values, the higher divergence measures suggest that the shorter window captures local patterns effectively but provides a less accurate representation of the overall data distribution. Since the primary objective of the proposed framework is generative modelling rather than pointwise prediction, the distribution-based metrics were considered more representative of the model quality.

The Vanilla GAN configuration exhibited consistently worse performance than the proposed WGAN-GP model in all distributional metrics, particularly in terms of Wasserstein Distance, which increased from 9.726 to 12.799. These results indicate that replacing the Wasserstein objective with the conventional binary cross-entropy loss reduces the model's ability to accurately reproduce the statistical characteristics of the original turbine behaviour. This observation is consistent with the known advantages of Wasserstein-based training in improving convergence stability and reducing mode collapse in generative models.

The largest degradation was observed when the gradient penalty was removed. In this case, KLD increased to 1.840, JSD to 0.424, and Wasserstein Distance to 18.303, while the relative MAE and RMSE increased to 7.13% and 9.30%, respectively. The substantial deterioration across all evaluation metrics highlights the critical role of the gradient penalty in stabilizing the adversarial training process and improving the fidelity of the generated sequences. Overall, the ablation study confirms that both the Wasserstein formulation and the gradient penalty contribute significantly to the quality of the generated data, while the selected temporal window size provides an appropriate compromise between temporal context and generative performance.

**TABLE 4. Comparison between TimeWGAN-GP and an LSTM baseline.**

| Model | Relative MAE | Relative RMSE |
|---|---|---|
| TimeWGAN-GP (Full Model) | 0.0558 | 0.0739 |
| LSTM Baseline | 0.0640 | 0.0730 |

To provide an additional benchmark against a commonly used time-series forecasting approach, a multivariate LSTM model was trained using the same dataset and temporal window length employed by the proposed framework. The LSTM baseline achieved a relative MAE of 6.40% and a relative RMSE of 7.30%, while the proposed TimeWGAN-GP obtained a relative MAE of 5.58% and a relative RMSE of 7.39%.

These results indicate that both approaches achieve comparable predictive accuracy, with the proposed framework

providing a lower average prediction error. Beyond forecasting performance, however, the proposed method offers the additional capability of learning the underlying data distribution and generating statistically consistent synthetic trajectories, making it suitable not only for prediction tasks but also for dataset augmentation and anomaly analysis. Consequently, the results support the use of TimeWGAN-GP as a competitive alternative to conventional LSTM-based approaches while providing broader functionality for industrial time-series applications.

## VI. RESULTS

This model has been applied to generate normal conditions behavior and failure data from an already categorized fault case so that it can be prove that the generated sample resembles the structure of the real ones and that the augmentation of the time series represents accurately the evolution of the original time series. The reference failure behavior was due to a failure in an injector, which caused air to enter one of the combustion chambers of a gas turbine [4].

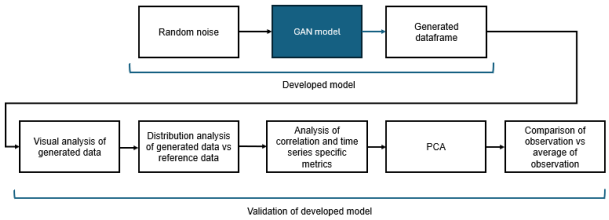



**FIGURE 3. Workflow diagram for GAN model development and validation.**

In this section, the GAN model development and validation process are illustrated, using two real-world examples: one depicting faulty behavior and the other demonstrating regular behavior. The steps outlined in the workflow diagram shown in FIGURE 3, are applied to both examples to highlight the differences and validate the effectiveness of the model.

Upon visual inspection, the deviations and patterns of generated data and reference data is analyzed. This is proven later by using two quantitative methods.

As the GAN generates a temporal series, it is imperative to validate it as such. The metrics used for this validation include Kullback-Leibler Divergence (KLD), Jensen-Shannon Divergence (JSD), Wasserstein Distance (WD), Maximum Mean Discrepancy (MMD), Euclidean Distance (ED) and the correlation between the different temperature variables. These metrics provide a comprehensive evaluation of the system's performance in generating synthetic data that closely resembles real data distributions, ensuring the reliability and accuracy of the temporal series generated by the GAN [24].

Finally, two analysis methods from a previous research are used to compare the results of the GAN model with the reference dataset.

By applying the steps from the workflow diagram to these real-world examples, the capability of the GAN model to distinguish between faulty and regular behavior and its ability to generate time series datasets that are consistent with the temporal evolution of the original reference data is demonstrated, ensuring that each synthetic output reflects realistic operational behavior and supports robust, reliable performance across different input scenarios.

### A. DESCRIPTION OF THE DATASET

Both datasets consist of twenty-one columns each representing a temperature measured by independent thermocouples, placed at the exit of a gas turbine, that measure the temperature of the exhaust gases of a 6Fa gas turbine from General Electric [4]. In FIGURE 4, the main components of a gas turbine can be seen, as well as the placement of the thermocouples which are placed equidistantly.

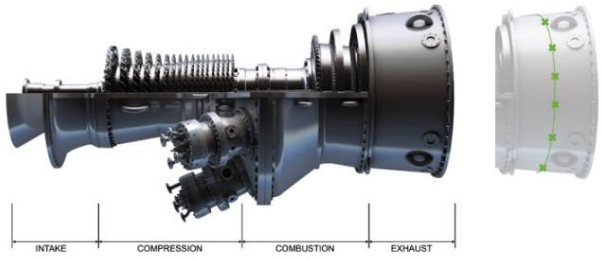



**FIGURE 4. Main Components of a gas turbine [26].**

The proposed methodology has been evaluated across four different 6Fa gas turbines, ensuring a level of variability in operating conditions and system behavior. While only a single fault type is explicitly presented, this is primarily due to the scarcity of labeled failure data, which is a common limitation in industrial applications.

Nevertheless, the selected fault scenario is representative of a broader class of degradation mechanisms and can be reasonably extrapolated to similar underlying failure causes. In this sense, the study establishes a foundational framework that can be extended to additional fault types, datasets, and machinery configurations. It therefore serves as a starting point for further investigations into cross-turbine generalization and the applicability of generative approaches across diverse industrial contexts.

### B. OPERATION OF THE AUGMENTATION METHOD BASED WITH REGULAR OPERATING BEHAVIOR

A dataset formed by six weeks of real observations is used as a reference and as the input for both the generator and the discriminator of the developed GAN model.

The six-week data segment analyzed in this work corresponds to a period in which the turbine was operating under normal and stable conditions, with no observable signs of degradation or fault development. At that time, all operational indicators aligned with expected behavior, and no maintenance alerts or performance deviations had been reported by on-site personnel.

Importantly, this data window is situated six months prior to an unplanned downtime event caused by a failure that will be detailed in the following section. The selection of this period ensures that the baseline behavior used for model training and comparison truly reflects healthy operational dynamics, thereby providing a reliable reference against which the later fault evolution can be contextualized.

The generation of regular operating conditions was evaluated across all four turbines included in the dataset, using several different reference periods for each of the four units. The results were consistently satisfactory and exhibited similar performance across turbines, demonstrating that the method generalizes well under normal operating conditions. To maintain clarity and avoid unnecessary extension of the manuscript, only one representative turbine case is presented in detail.

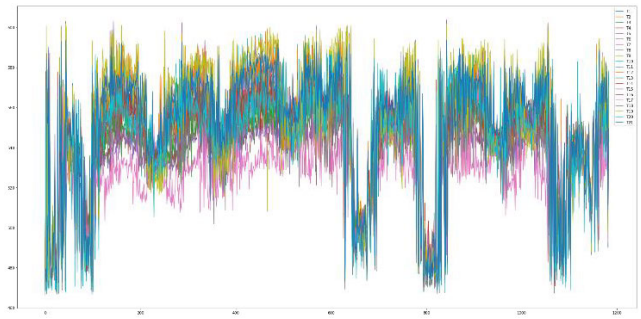


**FIGURE 5.** Generated data by the GAN model in regular conditions.

During each iteration of the training loop, a moving window of reference data is employed on the input vector. In FIGURE 5 the output of the model can be seen, representing a week of generated data.

A comparative analysis of the original input data and the generated dataset is shown in FIGURE 6.

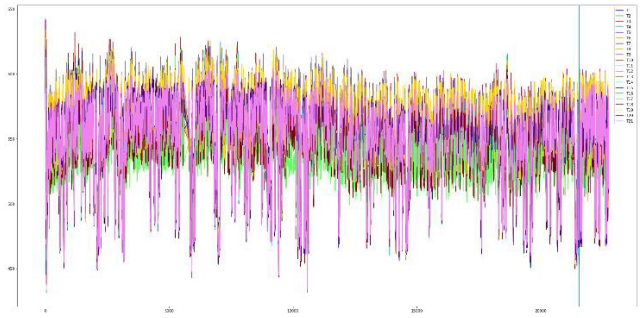


**FIGURE 6.** Original data versus generated data in regular conditions.

FIGURE 7 illustrate the evolution of the adversarial training dynamics and the convergence behaviour of the Wasserstein GAN. In the this plot, the discriminator loss and generator loss are shown over the training steps. During the initial phase, both losses exhibit significant oscillations, reflecting the typical instability of adversarial training as the generator and discriminator compete to improve simultaneously.

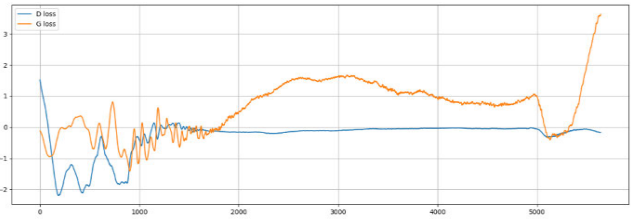



**FIGURE 7.** Generator and discriminator loss during training.

As training progresses, the discriminator loss gradually stabilizes around values close to zero, indicating that it reaches a balance between distinguishing real and generated samples. The generator loss shows a smoother trend after the initial fluctuations, with moderate variations that reflect ongoing adaptation to the discriminator's feedback, although a sharp increase is observed toward the end, suggesting a temporary imbalance in the training dynamics.

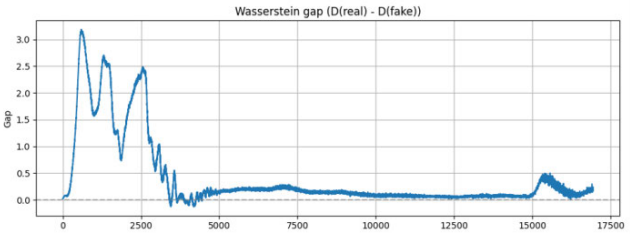



**FIGURE 8.** Wasserstein gap during training.

FIGURE 8 presents the Wasserstein gap, defined as the difference between the discriminator outputs for real and generated samples. At early training stages, the gap reaches relatively high values (above 3), indicating a clear separation between real and synthetic data distributions.

Over time, this gap decreases and stabilizes near zero, which is consistent with the objective of Wasserstein GANs, where convergence is achieved when the generated distribution closely approximates the real one. A slight increase in the gap toward the end of training suggests minor fluctuations and therefore the training stops, but overall, the trend demonstrates that the model successfully reduces the discrepancy between real and generated samples, indicating effective convergence.

In FIGURE 9 the generated values and the original data are highly similar, as shown by their nearly identical probability density functions.

The overlapping regions of these distributions indicate that the variables have similar central tendencies, variances, and overall distributional characteristics.

This visual similarity suggests that all the temperatures are statistically comparable, with minimal differences in their distributions. This will be analyzed using quantitative indicators as a next step of the analysis.

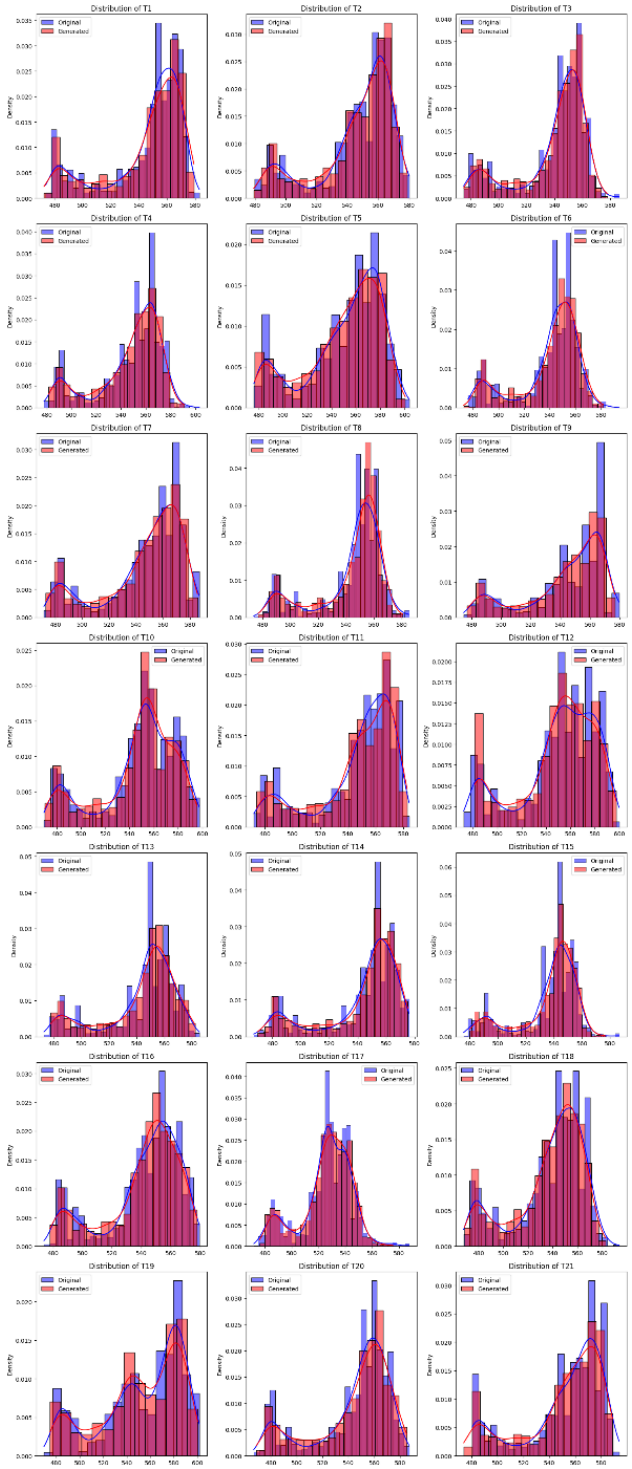



**FIGURE 9. Comparison between the distributions of the generated and original data for each of the variables in regular conditions.**

Moreover, a comparative analysis of the correlation matrices from both datasets is conducted to evaluate how the inter-variable relationships differ under the respective conditions represented by each dataset. By focusing on the correlation, it can be assessed whether the model accurately replicates the inherent randomness and variability of the original correlation structure.

This method isolates the stochastic properties of the data, offering a clearer picture of the model's ability to reproduce the statistical characteristics of the original data without being influenced by specific patterns or structures present in the outputs.

As shown in FIGURE 10, the results indicate similar correlations between variables, confirming that the generated data is statistically consistent with the original data.

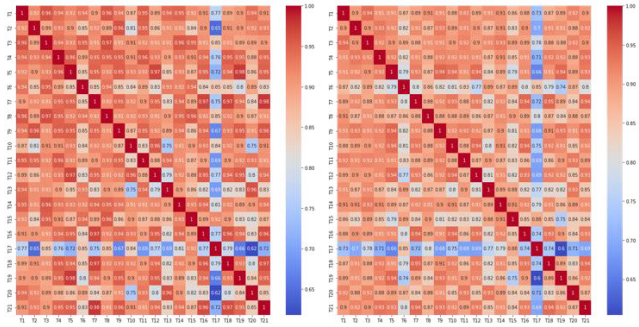


**FIGURE 10. Correlation in regular conditions.**

The next analysis methods employ several key metrics to evaluate the quality of the synthetic data generated by the GAN as a temporal series. The results of these analyses are shown in Table 5.

Kullback-Leibler Divergence (KLD) values are relatively low across all tests, indicating that the synthetic data generated by the system closely approximates the distribution of the real data.

Lower KLD values suggest better performance in terms of matching the real data distribution. Similarly, Jensen-Shannon Divergence (JSD) values are also low, reinforcing the observation that the synthetic data is well-aligned with the real data distribution. JSD, being a symmetric measure, provides a more balanced view of the divergence between distributions.

Wasserstein Distance (WD) values vary significantly across tests, with some tests showing higher values. This metric measures the distance between distributions, and higher values may indicate greater differences between the synthetic and real data distributions.

However, the overall trend suggests that the system performs well in most tests. Maximum Mean Discrepancy (MMD) values are relatively consistent across tests, with slight variations. This metric evaluates the difference between distributions based on their mean embeddings. Consistent MMD values indicate stable performance of the system in generating synthetic data.

Euclidean Distance (ED) values show noticeable variation across tests, with some tests having higher values. Euclidean Distance measures the statistical distance between distributions, and higher values may indicate greater discrepancies. Despite the variations, the system demonstrates reliable performance in most tests. This could be due to the samples having multiple attributes.

**TABLE 5.** Results of metrics for temporal series analysis in regular conditions.

| | KLD | JSD | WD | MMD | ED |
|---|---|---|---|---|---|
| **T1** | 0.0021 | 0.0005 | 27.497 | 0.0127 | 10.38093 |
| **T2** | 0.0016 | 0.0004 | 25.2221 | 0.0092 | 10.2245382 |
| **T3** | 0.0017 | 0.0004 | 24.0348 | 0.0133 | 9.32490647 |
| **T4** | 0.0018 | 0.0005 | 27.1334 | 0.0104 | 8.88450794 |
| **T5** | 0.0026 | 0.0006 | 33.616 | 0.0086 | 9.97032647 |
| **T6** | 0.0018 | 0.0004 | 24.0372 | 0.0117 | 8.85605916 |
| **T7** | 0.0024 | 0.0006 | 30.3573 | 0.0092 | 10.7986929 |
| **T8** | 0.0016 | 0.0004 | 23.1803 | 0.0078 | 9.30813295 |
| **T9** | 0.0018 | 0.0005 | 27.6699 | 0.008 | 10.3894486 |
| **T10** | 0.0027 | 0.0007 | 33.0102 | 0.0065 | 10.2256571 |
| **T11** | 0.0022 | 0.0006 | 29.1953 | 0.0099 | 10.6675301 |
| **T12** | 0.0025 | 0.0006 | 32.8927 | 0.007 | 10.3388236 |
| **T13** | 0.002 | 0.0005 | 26.3263 | 0.0152 | 10.0782315 |
| **T14** | 0.002 | 0.0005 | 26.3769 | 0.008 | 10.6411695 |
| **T15** | 0.0013 | 0.0003 | 20.3473 | 0.0102 | 8.91090513 |
| **T16** | 0.0018 | 0.0005 | 26.3704 | 0.007 | 10.4540341 |
| **T17** | 0.0011 | 0.0003 | 19.7169 | 0.0092 | 8.55128194 |
| **T18** | 0.0022 | 0.0006 | 29.0729 | 0.0117 | 9.41744323 |
| **T19** | 0.0031 | 0.0008 | 37.4341 | 0.0086 | 10.9578485 |
| **T20** | 0.0025 | 0.0006 | 30.0419 | 0.0137 | 10.6308162 |
| **T21** | 0.0025 | 0.0006 | 31.5812 | 0.0107 | 11.0392856 |

Overall, the developed system exhibits robust performance and reliability in generating synthetic data that closely resembles real data.

The low KLD and JSD values, along with consistent MMD, highlight the system's capability to produce high-quality synthetic data. While WD and ED values show some variations, the overall results affirm the effectiveness of the system in replicating real data distributions.

Dynamic Time Warping (DTW) is widely recognized as a robust temporal alignment technique used to quantify the similarity or dissimilarity between time-series sequences, particularly in contexts where temporal distortions, speed variations, or phase shifts are present.

The method aligns sequences by minimizing cumulative pointwise distances along an optimal warping path, producing a scalar distance value whose magnitude depends directly on the characteristics of the underlying data.

As mentioned in [24], DTW distance values are "highly sensitive to variability in amplitude and temporal structure," and therefore must be interpreted only relative to other distances computed under the same preprocessing conditions rather than against any fixed threshold.

**TABLE 6.** Per-column DTW distances.

| | |
|---|---|
| **T1** | 908.451878 |
| **T2** | 951.564693 |
| **T3** | 923.092449 |
| **T4** | 959.189235 |
| **T5** | 973.19719 |
| **T6** | 1001.39389 |
| **T7** | 971.567098 |
| **T8** | 908.239611 |
| **T9** | 949.247238 |
| **T10** | 1034.68874 |
| **T11** | 964.732105 |
| **T12** | 989.678833 |
| **T13** | 988.675329 |
| **T14** | 918.880332 |
| **T15** | 924.352868 |
| **T16** | 953.575127 |
| **T17** | 1079.6068 |
| **T18** | 1014.27558 |
| **T19** | 1024.79844 |
| **T20** | 943.944406 |
| **T21** | 935.503417 |

The Dynamic Time Warping (DTW) values obtained for the 21 normalized variables range from 908.24 to 1079.61, with an overall average distance of 967.56. These values quantify the temporal alignment cost between the real and generated sequences on a per-feature basis. Because the data were z-normalized prior to DTW computation, the resulting distances primarily reflect differences in temporal structure, rather than disparities in magnitude or scale.

Within this dataset-specific context, variables such as T1, T8, T14, T3, and T15 exhibit the lowest DTW values (approximately 908–925) as showed in Table 6, indicating the highest temporal coherence between real and generated sequences. Conversely, features including T17, T10, T19, T18, and T6 display the highest distances (approximately 1000–1080), suggesting comparatively larger deviations in temporal evolution. These intra-dataset variations mirror the heterogeneous behavior often observed in multivariate industrial or sensor-based time series, where different processes or sensors exhibit distinct dynamics over time.

Moreover, the difference between the lowest DTW values and the highest ones represents only a modest variation relative to the overall magnitude of the metric, indicating that

the generated sequences remain consistently aligned with the real data and that the system demonstrates strong temporal fidelity across all variables.

Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) are widely used metrics to quantify the discrepancy between real and generated data, providing complementary perspectives on model performance. MAE measures the average magnitude of the absolute differences between corresponding values, offering a direct and interpretable estimate of the typical prediction error. RMSE, by squaring the differences before averaging and then taking the square root, gives greater weight to large deviations, making it particularly sensitive to outliers. In this study, both metrics are computed by aligning the original and synthetic datasets and restricting the comparison to the minimum common number of observations, after which errors are calculated elementwise across all variables and normalized relative to the mean value of each feature to ensure scale invariance.

**TABLE 7. Relative RMSE and MAE per thermocouple.**

| Thermocouple | MAE_relative | RMSE_relative |
|---|---|---|
| 1 | 0.048287 | 0.066416 |
| 2 | 0.043745 | 0.059236 |
| 3 | 0.042621 | 0.058863 |
| 4 | 0.046736 | 0.062624 |
| 5 | 0.059077 | 0.075981 |
| 6 | 0.043653 | 0.05876 |
| 7 | 0.053373 | 0.070695 |
| 8 | 0.040646 | 0.05685 |
| 9 | 0.048582 | 0.064559 |
| 10 | 0.060941 | 0.078594 |
| 11 | 0.049908 | 0.068239 |
| 12 | 0.05991 | 0.077067 |
| 13 | 0.046047 | 0.062493 |
| 14 | 0.048965 | 0.067857 |
| 15 | 0.038213 | 0.052305 |
| 16 | 0.047809 | 0.062788 |
| 17 | 0.035792 | 0.045887 |
| 18 | 0.052107 | 0.068193 |
| 19 | 0.067609 | 0.084906 |
| 20 | 0.05228 | 0.070929 |
| 21 | 0.055264 | 0.07367 |

The results shown in Table 7 indicate a high degree of agreement between the generated and original datasets, as reflected by the low global relative errors (MAE $\approx$ 0.0497 and RMSE $\approx$ 0.0668). These values demonstrate that, on average, deviations remain below 5% and 7%, respectively, which is generally considered indicative of strong predictive accuracy in continuous-valued variables. The performance is consistent across all thermocouples, with relative MAE values ranging approximately from 3.6% to 6.8% and RMSE values from 4.6% to 8.5%.

This limited dispersion suggests a homogeneous model behavior without significant degradation in any specific variable. Although slightly higher errors are observed in a small subset of thermocouples, these remain within acceptable margins and do not indicate critical deficiencies. Overall, the results support the conclusion that the generative model successfully captures the underlying structure and magnitude of the original data, achieving reliable and stable performance across all measured variables.

The results of the PCA for the generated dataset can be seen in FIGURE 11. This analysis uses a reference period from the same gas turbine, which is categorized as a normal operating condition, to obtain the first five principal components of the twenty-one temperatures of the exhaust gases. This first five principal component explained more than 90% of the data variance [4].

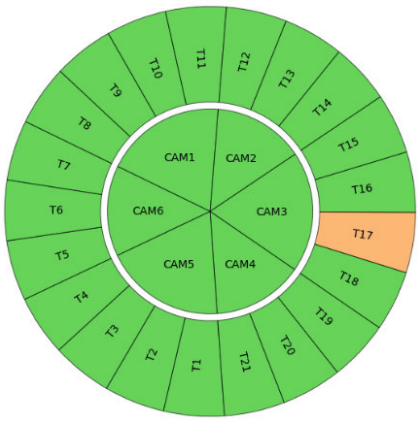



**FIGURE 11. Representation of affected thermocouples in regular conditions.**

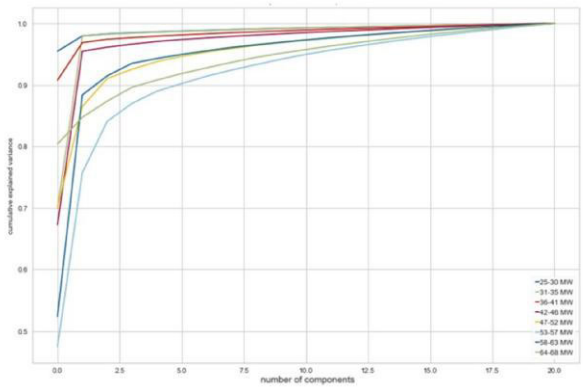
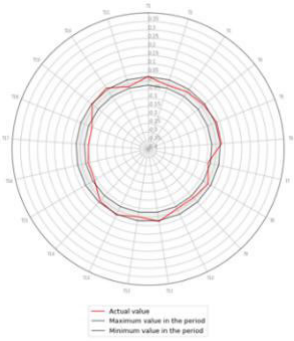



**FIGURE 12. Analysis of explained variance by each principal component [4].**

FIGURE 12 shows that, when the historical observations are segmented by power-range intervals, the first five principal components account for more than 90% of the variance in each interval. Since the analysis in this work is conducted without dividing the data into power-range subsets, it is reasonable to retain the same dimensionality reduction choice. Accordingly, five principal components are used throughout, ensuring consistency with the variance-explained patterns observed across the power-range-specific analyses.

Then, it reverts the transformation to obtain the original dataset and calculates the error between the recalculated original dataset and the original one. Later on, this same transformation is applied to the data that is being studied, which in this case is the generated dataset, but using the same transformation matrix that was calculated for the normal behavior dataset. As a result, this analysis shows no major difference between the variables of the generated dataset and the reference period, which was considered normal conditions behavior [4].

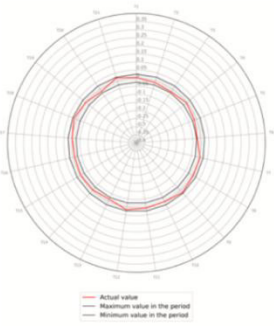


| Chamber | Thermocouple | +dev | -dev |
|---|---|---|---|
| CAM_5 | T1 | 5.07 | 1.44 |
| CAM_5 | T2 | 4.56 | 2.28 |
| CAM_5 | T3 | 0.76 | 6.26 |
| CAM_5 | T4 | 4.14 | 1.78 |
| CAM_6 | T5 | 24.01 | 2.28 |
| CAM_6 | T6 | 2.7 | 22.65 |
| CAM_6 | T7 | 5.33 | 3.38 |
| CAM_1 | T8 | 4.23 | 3.38 |
| CAM_1 | T9 | 2.11 | 2.54 |
| CAM_1 | T10 | 18.01 | 3.72 |
| CAM_1 | T11 | 9.64 | 2.79 |
| CAM_2 | T12 | 26.71 | 2.03 |
| CAM_2 | T13 | 8.62 | 8.54 |
| CAM_2 | T14 | 2.37 | 2.7 |
| CAM_3 | T15 | 0.59 | 33.98 |
| CAM_3 | T16 | 1.35 | 6.09 |
| CAM_3 | T17 | 0.68 | 54.44 |
| CAM_3 | T18 | 0.51 | 14.71 |
| CAM_4 | T19 | 40.32 | 3.3 |
| CAM_4 | T20 | 12.85 | 5.16 |
| CAM_4 | T21 | 24.34 | 1.78 |

**FIGURE 13.** Results of one versus all analysis on generated data in regular conditions.

Lastly, in FIGURE 13, the result of the one versus all analysis is illustrated. In this analysis, for each observation of the generated dataset, the temperature of each thermocouple is compared to the mean temperature of all thermocouples in that observation. When the difference is higher or lower than 2.5% of the standard deviation of the observation, the thermocouple is flagged as erratic. This criterion follows the formulation presented in [4], as the objective here is to assess whether the generated data activates the same previously validated failure-detection mechanism.

Then, the percentage of times that each thermocouple is flagged as erratic for being outside of the accepted interval is calculated and displayed in a table, as well as the difference between the mean and the temperature in the last observation of the interval [4]. This analysis suggests that thermocouples six, fifteen, seventeen and eighteen are in some intervals under the average of all the temperatures for each observation. Only this type of deviation is considered as failure as all the historic failures available for this type of turbines were related to air inlets and decreasing of temperatures in the air chambers, and not to increases.



| Chamber | Thermocouple | +dev | -dev |
|---|---|---|---|
| CAM_5 | T1 | 18.26 | 0.28 |
| CAM_5 | T2 | 11.68 | 0.94 |
| CAM_5 | T3 | 4.74 | 1.18 |
| CAM_6 | T4 | 10.44 | 0.05 |
| CAM_6 | T5 | 2.48 | 0.08 |
| CAM_6 | T6 | 7.29 | 5.26 |
| CAM_6 | T7 | 0.2 | 5.67 |
| CAM_1 | T8 | 7.78 | 2.77 |
| CAM_1 | T9 | 2.44 | 1.01 |
| CAM_1 | T10 | 6.25 | 16.93 |
| CAM_2 | T11 | 5.76 | 6.02 |
| CAM_2 | T12 | 5.25 | 0.67 |
| CAM_2 | T13 | 0.07 | 0.53 |
| CAM_2 | T14 | 1.49 | 0.09 |
| CAM_3 | T15 | 0.14 | 1.27 |
| CAM_3 | T16 | 0.0 | 2.13 |
| CAM_3 | T17 | 0.26 | 23.19 |
| CAM_4 | T18 | 0.06 | 7.51 |
| CAM_4 | T19 | 3.26 | 6.27 |
| CAM_4 | T20 | 1.32 | 7.34 |
| CAM_4 | T21 | 18.4 | 7.2 |

**FIGURE 14.** Results of one versus all analysis in reference dataset.

When comparing this result to the result of the one versus all analysis for the input dataset used for training the GAN, shown in FIGURE 14, it can be noted that some of the thermocouples were also considered to be under the mean of each observation for some intervals, so the result is as expected.

As the model iterates, the most recent synthetic samples are generated using previously generated data rather than the original reference dataset. This recursive process allows the GAN to simulate how the system might behave in future time steps, based on its own learned representations. When these newly generated sequences are compared with the original dataset, the evolution of the variables shows a coherent and consistent pattern that mirrors the historical behavior of the turbine. This indicates that the model not only preserves the structure and dynamics of the original data but also successfully captures the progression of faults over time, making it a reliable tool for temporal augmentation.

### C. OPERATION OF THE AUGMENTATION METHOD BASED ON A FAILURE OF ONE OF THE INJECTORS OF A COMBUSTION CHAMBER

During this period there was a failure in one of the injectors of the third combustion chamber and therefore the temperature of this chamber was lower than expected. FIGURE 15 depicts the evolution of the real temperatures of each of the twenty-one thermocouples, which forms the training dataset for this scenario.

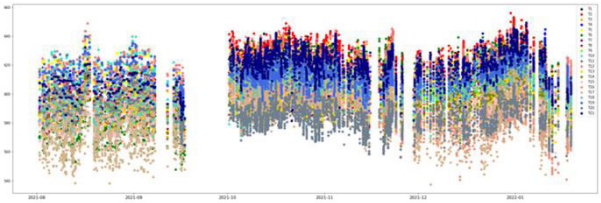


**FIGURE 15.** Evolution of temperatures during the injector failure.

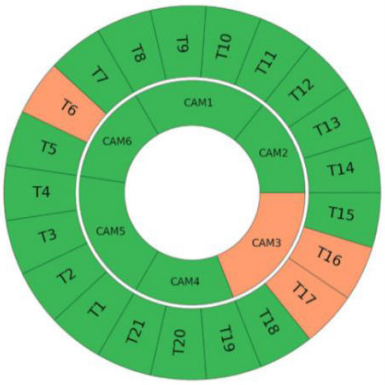



**FIGURE 16.** Representation of affected thermocouples during the injector failure.

After categorizing this historical failure, it was seen that the thermocouples which were more affected by the failure were the 16th and the 17th, both placed in the 3rd combustion chamber at the power that the turbine was operating, as it is shown in FIGURE 17.

The dataset previously described serves as the input for both the generator and the discriminator of the developed GAN model. During each iteration of the training loop, a moving window of reference data is employed on the input vector.

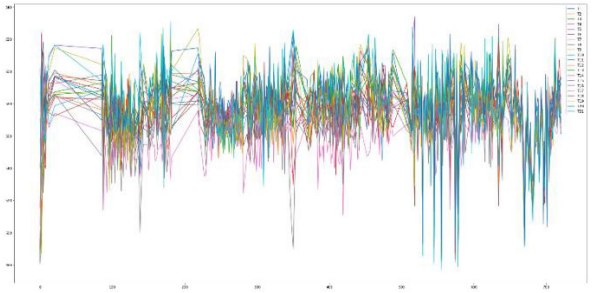


**FIGURE 17.** Generated data by the GAN model in failure.

This technique is designed to maintain the correlation structures of the series, ensuring that the generated data accurately reflects the evolution of the original time series. In FIGURE 18 the output of the model can be seen, it represents a week of generated data, maintaining the same data structure as the original input.

A comparative analysis of the original input data and the generated dataset in FIGURE 18 reveals that both exhibit a similar temporal evolution. This observation indicates that the generated dataset effectively mirrors the dynamic patterns and trends present in the original data.

When this generated data is grouped by combustion chamber to see its evolution, as it can be seen in FIGURE 19, it can be noted that the temperature of the third chamber is consistently lower than the temperature of the rest of the combustion chambers.

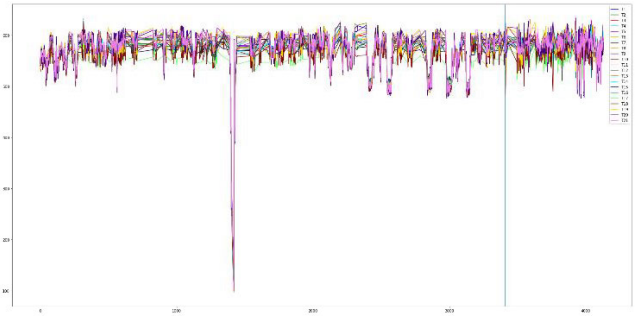


**FIGURE 18.** Original data versus generated data in failure.

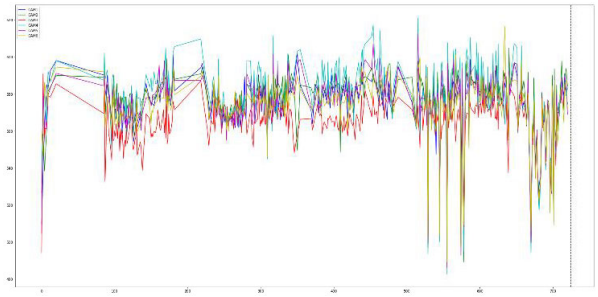


**FIGURE 19.** Average temperature of each combustion chamber represented by generated data in failure.

In the last period of the generated time series, chamber six is also affected, which would make sense since in previous analyses it was demonstrated that when there is a temperature variation due to external factors in one of the combustion chambers, it not only affects it but also the opposite combustion chambers and to a lesser extent the adjacent ones.

In FIGURE 20 the distribution of each of the generated values and the original data exhibits a high degree of similarity, as evidenced by the nearly identical shapes of their probability density functions.

The overlapping regions of the distributions indicate that the variables share similar central tendencies, variances, and overall distributional characteristics. This visual congruence suggests that all of the temperatures are statistically comparable, with minimal divergence in their respective distributions' histograms.

In addition, the correlation of each of the variables in both the generated and the reference dataset shows remarkable similarities, as it can be seen in FIGURE 21.

This indicates that the model has successfully replicated the inherent randomness and variability of the original dataset, ensuring that the statistical properties of the generated data closely match those of the reference data. Such consistency validates the reliability and robustness of the data generation process, confirming that the model performs well in reproducing the stochastic characteristics of the original data.

Table 8 provides a detailed evaluation of the developed system across various tests (T1 to T21) using the metrics KLD, JSD, WD, MMD, and ED.

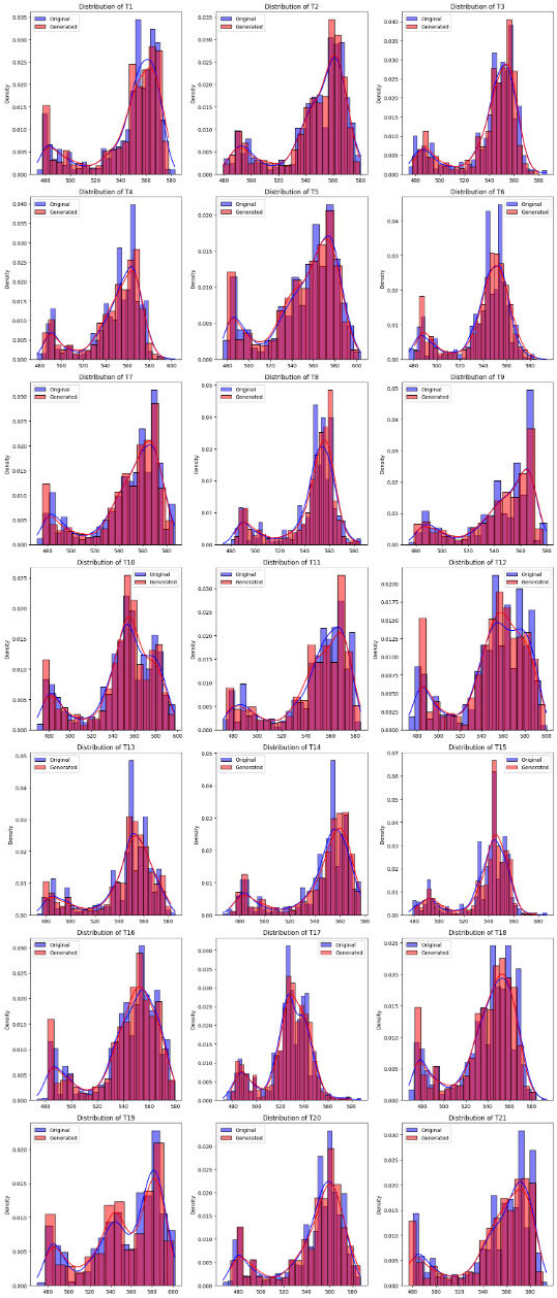


**FIGURE 20.** Comparison between the distributions of the generated and original data for each of the variables in failure.

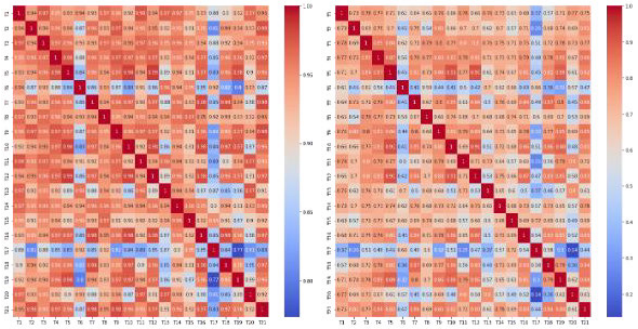


**FIGURE 21.** Correlation for original and generated dataset in failure.

**TABLE 8.** Results of metrics for temporal series analysis in failure conditions.

| | KLD | JSD | WD | MMD | ED |
|---|---|---|---|---|---|
| **T1** | 0.0020 | 0.0005 | 24.7910 | 0.0272 | 7.3861 |
| **T2** | 0.0017 | 0.0004 | 23.3351 | 0.0243 | 7.3429 |
| **T3** | 0.0021 | 0.0005 | 26.7062 | 0.0285 | 7.6001 |
| **T4** | 0.0023 | 0.0006 | 28.2818 | 0.0273 | 7.9375 |
| **T5** | 0.0033 | 0.0008 | 36.2921 | 0.0274 | 8.5982 |
| **T6** | 0.0016 | 0.0004 | 23.2521 | 0.0267 | 6.5682 |
| **T7** | 0.0023 | 0.0006 | 28.2002 | 0.0231 | 8.2114 |
| **T8** | 0.0016 | 0.0004 | 23.1422 | 0.0306 | 7.2643 |
| **T9** | 0.0021 | 0.0005 | 26.3548 | 0.0240 | 7.8794 |
| **T10** | 0.0035 | 0.0009 | 36.7828 | 0.0231 | 8.5069 |
| **T11** | 0.0023 | 0.0006 | 27.9208 | 0.0255 | 7.7047 |
| **T12** | 0.0032 | 0.0008 | 35.4119 | 0.0253 | 8.4103 |
| **T13** | 0.0021 | 0.0005 | 25.5619 | 0.0301 | 7.0948 |
| **T14** | 0.0020 | 0.0005 | 24.2943 | 0.0229 | 7.5918 |
| **T15** | 0.0015 | 0.0004 | 21.7317 | 0.0273 | 6.5414 |
| **T16** | 0.0021 | 0.0005 | 26.8989 | 0.0264 | 7.7381 |
| **T17** | 0.0014 | 0.0004 | 23.1388 | 0.0168 | 6.3247 |
| **T18** | 0.0026 | 0.0006 | 30.7628 | 0.0254 | 8.1859 |
| **T19** | 0.0036 | 0.0009 | 37.5343 | 0.0240 | 8.9396 |
| **T20** | 0.0027 | 0.0007 | 28.7740 | 0.0265 | 7.8564 |
| **T21** | 0.0027 | 0.0007 | 30.1548 | 0.0258 | 8.4555 |

Kullback-Leibler Divergence (KLD) values are consistently low, indicating that the synthetic data closely matches the distribution of the real data. Jensen-Shannon Divergence (JSD) values are also low, confirming the alignment between synthetic and real data distributions.

Wasserstein Distance (WD) values show some variability, with higher values in certain tests, suggesting differences between synthetic and real data distributions in those cases. Maximum Mean Discrepancy (MMD) values are relatively stable, indicating consistent performance in generating synthetic data.

Euclidean Distance (ED) values vary across tests, with some higher values indicating greater discrepancies. Despite these variations, the system generally performs reliably in most tests. This could be due to the samples having multiple attributes.

**TABLE 9.** Per-column DTW distances.

| | |
|---|---|
| **T1** | 865.790614 |
| **T2** | 912.823644 |
| **T3** | 937.385998 |
| **T4** | 958.915789 |
| **T5** | 1005.92716 |
| **T6** | 958.464555 |
| **T7** | 951.426771 |
| **T8** | 891.122148 |
| **T9** | 908.921645 |
| **T10** | 1119.17591 |
| **T11** | 868.967892 |
| **T12** | 985.810271 |
| **T13** | 911.243665 |
| **T14** | 840.025625 |
| **T15** | 898.339264 |
| **T16** | 959.008191 |
| **T17** | 1063.41518 |
| **T18** | 968.5676 |
| **T19** | 1049.25995 |
| **T20** | 923.266669 |
| **T21** | 929.093106 |

**TABLE 10.** Relative MAE and RMSE per thermocouple.

| Thermocouple | MAE_relative | RMSE_relative |
|---|---|---|
| **1** | 0.084075 | 0.098654 |
| **2** | 0.073214 | 0.087082 |
| **3** | 0.084925 | 0.100063 |
| **4** | 0.078997 | 0.096738 |
| **5** | 0.086984 | 0.110297 |
| **6** | 0.077494 | 0.090076 |
| **7** | 0.074341 | 0.093686 |
| **8** | 0.078048 | 0.091369 |
| **9** | 0.0754 | 0.092381 |
| **10** | 0.088989 | 0.112958 |
| **11** | 0.082035 | 0.099673 |
| **12** | 0.08821 | 0.110743 |
| **13** | 0.089445 | 0.102368 |
| **14** | 0.079512 | 0.095212 |
| **15** | 0.084816 | 0.096049 |
| **16** | 0.078502 | 0.094559 |
| **17** | 0.080061 | 0.091988 |
| **18** | 0.080328 | 0.100109 |
| **19** | 0.087185 | 0.110675 |
| **20** | 0.086903 | 0.102796 |
| **21** | 0.083347 | 0.10263 |

These analyses underscore the robustness and precision of the developed system, showcasing its potential to generate high-fidelity synthetic data that closely resembles real data distributions.

The DTW distance obtained for the multivariate comparison in the present case is 8869.94, while per-feature distances range between 840.03 and 1119.18, with an overall average of 947.95 across the 21 variables. All these metrics were obtained after normalizing the datasets.

The per-column DTW results shown in Table 9 allow feature-specific assessment of similarity. The distances span approximately 840–1120, with an average of 947.95, indicating moderate variation across variables.

These variables exhibit relatively low warping costs, indicating that their temporal patterns between the two datasets are highly consistent after normalization.

The multivariate DTW distance represents the cumulative alignment cost across all 21 normalized variables. Since the data were z-normalized prior to analysis, the resulting distance reflects differences primarily in shape and temporal structure, rather than amplitude or scale. The meaning of this value is determined relative to the per-column DTW results, and the variability patterns present across the dataset.

The magnitude of 8869.94 is consistent with multivariate DTW applied to long, multi-dimensional sequences, where accumulated alignment costs naturally grow with dimensionality and sequence length.

The resulting relative errors shown in Table 10 with a global MAE of approximately 8.2% and RMSE of 9.9%, indicate a strong agreement between the generated and original data, suggesting that the generative model is capable of preserving the underlying structure and magnitude of the variables with a satisfactory level of accuracy for practical applications.

To evaluate the effectiveness of the developed GAN model, it has been verified whether the same failure can be detected by applying analysis methods from previous research to the generated dataset [4]. The results of the PCA for the generated dataset can be seen in FIGURE 22.

This analysis suggests that thermocouples T2, T5, T6, T14, and T17 to T20 are affected. When comparing this result with the PCA of the original failure dataset, it can be seen that they coincide. In this case the original result from PCA in FIGURE 23 is shown divided by power range, but this does not apply to the generated dataset, as it only contains exhaust gas temperatures and no other data.

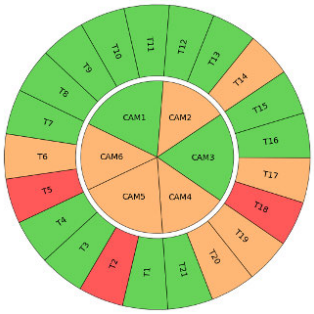



**FIGURE 22.** Results from PCA analysis on generated data in failure.

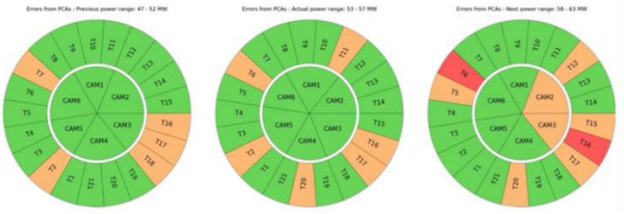


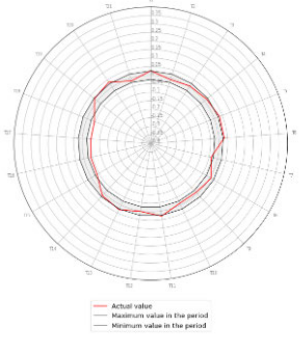
**FIGURE 23.** Results from PCA analysis on original data in failure.

Lastly, in FIGURE 24, the result of the one versus all analysis is illustrated.

This analysis suggests that thermocouples two, six, seven, fifteen, seventeen, eighteen and twenty are continuously under the average of all the temperatures for each observation. In this case this result is also relevant as it highlights that the rest of thermocouples, from other combustion chambers, were also above the average of the observations.

This means that the rest of the combustion chambers that were not affected by the failure were compensating the decrease of temperature on the affected chamber, so that the turbine was able to produce the expected power.

This phenomenon is to be expected during a failure related to the decrease of temperatures in one of the combustion chambers and was not seen in FIGURE 11 or FIGURE 13 when studying regular operating conditions.




| Chamber | Thermocouple | +dev | -dev |
|---|---|---|---|
| CAM_5 | T1 | 5.07 | 1.44 |
| CAM_5 | T2 | 4.56 | 2.28 |
| CAM_5 | T3 | 0.76 | 6.26 |
| CAM_5 | T4 | 4.14 | 1.78 |
| CAM_6 | T5 | 24.01 | 2.28 |
| CAM_6 | T6 | 2.7 | 22.65 |
| CAM_6 | T7 | 5.33 | 3.38 |
| CAM_1 | T8 | 4.23 | 3.38 |
| CAM_1 | T9 | 2.11 | 2.54 |
| CAM_1 | T10 | 18.01 | 3.72 |
| CAM_1 | T11 | 9.64 | 2.79 |
| CAM_2 | T12 | 26.71 | 2.03 |
| CAM_2 | T13 | 8.62 | 8.54 |
| CAM_2 | T14 | 2.37 | 2.7 |
| CAM_3 | T15 | 0.59 | 33.98 |
| CAM_3 | T16 | 1.35 | 6.09 |
| CAM_3 | T17 | 0.68 | 54.44 |
| CAM_3 | T18 | 0.51 | 14.71 |
| CAM_4 | T19 | 40.32 | 3.3 |
| CAM_4 | T20 | 12.85 | 5.16 |
| CAM_4 | T21 | 24.34 | 1.78 |

**FIGURE 24.** Results of one versus all analysis on generated in failure.

## VII. DISCUSSION

The results obtained from this study provide several insights into the applicability of Generative Adversarial Networks for anomaly augmentation in gas turbines. The ability of the GAN-based model to learn the distribution profiles of both normal and faulty operating conditions demonstrates its potential as a tool for enriching datasets where documented anomalies are sparse. The generated sequences exhibited strong consistency with historical reference data, confirming that the model can capture not only pointwise statistical characteristics but also the temporal dependencies that define equipment behavior over time.

A key outcome of the work is the model's capacity to replicate degradation pathways and produce synthetic faults that reflect realistic progression patterns. Comparative analyses with real failure cases revealed that the synthetic samples preserved the structure, temporal evolution, and characteristic transitions observed in the historical data. This fidelity reinforces the suitability of GANs for simulating the early stages of faults, which are often underrepresented in industrial datasets but critical for developing robust augmentation strategies.

The system's discriminative component played a central role in detecting deviations from expected behavior when exposed to real-time or simulated time-series input. Unlike threshold-based approaches that rely on static limits and may fail to capture gradual deterioration, the GAN framework models dynamic evolution, enabling earlier and more nuanced detection of abnormal trends. This capability suggests that the method could significantly improve early-warning systems in industrial environments, where anticipating failures provides substantial operational and economic benefits.

Furthermore, the expanded anomaly catalog created with synthetic data enhances opportunities for training, validation, and benchmarking of other diagnostic models. By generating additional instances of fault profiles that resemble real-world events, the dataset becomes more balanced and representative, mitigating the limitations imposed by scarce or incomplete failure records. This improvement is particularly relevant for data-driven techniques that require large quantities of labeled samples for optimal performance.

The methodological choices made in this study, such as exploring Conditional and Wasserstein GAN variants and applying gradient-penalty strategies for improved training stability, contributed to the quality and reliability of the generated sequences. These architectural considerations highlight the importance of addressing common GAN challenges, such as mode collapse and unstable convergence, when adapting the technique to industrial time series.

However, several aspects require further exploration. The model's performance under varying operational regimes, environmental conditions, or previously unseen fault types remains to be fully evaluated. Additionally, while synthetic data contribute valuable supplementary information, they

must always be validated against domain expertise and real measurements to ensure operational relevance. The integration of hybrid approaches that combine physical modeling and data-driven learning could further reinforce the robustness and interpretability of augmentation systems.

An important aspect to consider in iterative generative frameworks for time series is the potential degradation of temporal fidelity over extended generation horizons. As the model recursively incorporates its own outputs, small discrepancies may accumulate and progressively lead to distributional drift and loss of temporal coherence, particularly when extrapolating far beyond the temporal span observed during training. This behavior arises as long-horizon generation inherently amplifies minor modeling inaccuracies.

In the proposed approach, this effect is mitigated by the one-step-per-iteration generation strategy and the use of a bounded conditioning window, which constrain error propagation and contribute to maintaining stability across short and intermediate horizons. Nevertheless, the reliability of the generated sequences is expected to decrease as the generation length increases, highlighting the importance of restricting usage to controlled temporal ranges aligned with the training distribution. From a practical perspective, the method is therefore best suited for data augmentation scenarios within bounded horizons.

A systematic sensitivity analysis is required to quantitatively assess temporal degradation using distributional similarity metrics, such as Kullback-Leibler Divergence (KLD) and Dynamic Time Warping (DTW), as a function of generation length, thereby highlighting that unbounded or indefinite sequence generation lies beyond the reliable operational scope of the framework.

Overall, the discussion of findings indicates that GAN-based approaches offer a promising direction for advancing industrial failure augmentation, while also underscoring the need for continued refinement and validation before full-scale deployment in operational settings.

Recent advances in deep-learning–based anomaly detection encompass a wide spectrum of paradigms that extend beyond the generative modelling strategy adopted in this work. Numerous studies [27] have demonstrated the effectiveness of GAN-based architectures for industrial time-series anomaly detection, particularly in settings characterized by class imbalance and scarcity of fault data. Other generative models, such as transformer-based GANs, have been shown to effectively capture long-range contextual dependencies in multivariate industrial time-series streams, achieving state-of-the-art detection results across several public datasets.

Beyond pure GAN formulations, recent studies [28] highlight alternative generative neural architectures, such as adjusted-LSTM GAN models, that further enhance reconstruction-based anomaly scoring in both univariate and multivariate temporal domains. These contributions illustrate the breadth and continuous evolution of generative approaches across industrial monitoring, cybersecurity, smart-grid operation, and IoT-based sensing environments.

Another relevant aspect concerns the adaptability of the proposed approach across different turbine configurations and fault typologies. The current implementation is based on a homogeneous setting, involving four turbines of the same model and a specific failure mode characterized by temperature decreases. Extending the framework to other turbine classes would require adjustments in both the input representation and the model architecture, particularly to accommodate variations in the number of sensors, signal dimensionality, and sampling frequency.

For instance, architectures may need to be reconfigured to handle multivariate inputs with differing temporal resolutions, potentially incorporating resampling strategies or temporal alignment mechanisms. Additionally, the conditioning scheme and loss formulation may require adaptation to ensure stability when capturing distinct thermodynamic behaviors. In the case of fault types leading to temperature increases or more complex nonlinear patterns, the generator must be capable of learning alternative temporal dynamics, which could involve rebalancing the training dataset or incorporating domain-specific constraints to guide the generation process.

These considerations highlight that, while the underlying methodology is generalizable, its effective application to heterogeneous industrial scenarios requires careful reparameterization and validation to ensure consistent performance across diverse operating conditions and failure mechanisms.

The proposed framework can be naturally integrated within modern industrial Internet of Things (IoT) architectures, where sensor data are continuously acquired and stored through edge–cloud infrastructures. While the validation presented in this work is conducted in an offline data augmentation setting, the methodology is intended to operate in conjunction with real-time data acquisition systems already deployed in industrial environments. In such scenarios, sensor streams are directly ingested into centralized or distributed databases, enabling seamless incorporation of the generator-only inference without additional data collection stages.

The lightweight nature of the model, characterized by a low parameter count and CPU-based execution, facilitates flexible deployment across heterogeneous infrastructures, including both edge devices and cloud environments. This enables inference to be performed locally, reducing latency and communication overhead, or centrally, leveraging higher computational resources depending on system constraints. In Software-Defined IoT (SD-IoT) frameworks, where software-defined control dynamically orchestrates tasks across distributed nodes, the model can be incorporated into load-balanced pipelines that adapt to resource availability and operational demands.

Furthermore, as the generation process relies on locally available temporal conditioning windows, its execution is inherently resilient to variations in network quality of service

(QoS), making it suitable for environments with fluctuating latency, bandwidth, or connectivity conditions. These characteristics position the proposed approach as a viable component within scalable, real-time predictive maintenance ecosystems based on IoT and SD-IoT paradigms.

## VIII. CONCLUSION

This study demonstrates that Generative Adversarial Networks can effectively model both normal and faulty operating patterns in gas turbine sensor data, enabling the generation of synthetic time series that closely align with real operational behavior. The proposed framework supports failure dataset augmentation by capturing the temporal evolution of degradation, thereby extending beyond traditional detection-oriented approaches. The synthetic datasets produced by the model contribute to a richer and more representative repository of failure modes, strengthening the foundation for data-driven reliability analysis.

Future work will incorporate more diverse operating conditions, expand the anomaly catalog with additional fault categories, and explore hybrid architectures that integrate physical modeling with data-driven learning. Further research will also assess the implementation of the proposed system in real-time monitoring environments to validate its practical deployment for predictive maintenance.

Additionally, future research may include comparative analyses with alternative generative approaches to further position the proposed framework within the broader literature. Such efforts would require a dedicated and rigorous benchmarking study, extending beyond the scope of the present work, demonstrating the relevance, robustness, and applicability of the proposed approach for data augmentation tasks.

## ACKNOWLEDGMENT

The authors gratefully acknowledge the support provided by ENDESA Iberia and by the Chair of Artificial Intelligence Applications for Data-Driven Maintenance (ENDESA-ICAI), as well as its team, in the development of this research.

## REFERENCES

[1] M. Liu and O. Tuzel, "Coupled generative adversarial networks," in *Proc. NIPS*, 2016, pp. 1–2.

[2] S. Tripathi, A. I. Augustin, A. Dunlop, R. Sukumaran, S. Dheer, A. Zavalny, O. Haslam, T. Austin, J. Donchez, P. K. Tripathi, and E. Kim, "Recent advances and application of generative adversarial networks in drug discovery, development, and targeting," *Artif. Intell. Life Sci.*, vol. 2, Dec. 2022, Art. no. 100045, doi: 10.1016/j.ailsci.2022.100045.

[3] J. Gui, Z. Sun, Y. Wen, D. Tao, and J. Ye, "A review on generative adversarial networks: Algorithms, theory, and applications," 2020, *arXiv:2001.06937*.

[4] M. D. C. R. Mena, A. Muñoz, M. Á. Sanz-Bobi, D. Gonzalez-Calvo, and T. Álvarez-Tejedor, "Application of ensemble machine learning techniques to the diagnosis of the combustion in a gas turbine," *Appl. Thermal Eng.*, vol. 249, Jul. 2024, Art. no. 123447, doi: 10.1016/j.applthermaleng.2024.123447.

[5] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Commun. ACM*, vol. 63, no. 11, pp. 139–144, Oct. 2020, doi: 10.1145/3422622.

[6] C. Qian, W. Yu, C. Lu, D. Griffith, and N. Golmie, "Toward generative adversarial networks for the industrial Internet of Things," *IEEE Internet Things J.*, vol. 9, no. 19, pp. 19147–19159, Oct. 2022, doi: 10.1109/JIOT.2022.3163894.

[7] S. Zheng, A. Farahat, and C. Gupta, "Generative adversarial networks for failure prediction," in *Machine Learning and Knowledge Discovery in Databases*. Cham, Switzerland: Springer, 2019, pp. 621–637.

[8] H. Chen, J. Wei, H. Huang, Y. Yuan, and J. Wang, "Review of imbalanced fault diagnosis technology based on generative adversarial networks," *J. Comput. Design Eng.*, vol. 11, no. 5, pp. 99–124, Aug. 2024, doi: 10.1093/jcde/qwae075.

[9] I. Farady, J. Islam, S. Tuarob, H. Ng, and C. Lin, "Gans in industrial surface defect detection: Insights and challenges," SSNR, New York, NY, USA, Tech. Rep., 2023, doi: 10.2139/ssrn.4516131.

[10] A. D. Paroha, "Advancing predictive maintenance in the oil and gas industry: A generative AI approach with GANs and LLMs for sustainable development," in *Communications in Computer and Information Science*. Cham, Switzerland: Springer, 2024, pp. 3–15, doi: 10.1007/978-3-031-71729-1_1.

[11] H. Alqahtani, M. Kavakli-Thorne, and G. Kumar, "Applications of generative adversarial networks (GANs): An updated review," *Arch. Comput. Methods Eng.*, vol. 28, no. 2, pp. 525–552, Mar. 2021, doi: 10.1007/s11831-019-09388-y.

[12] J. Liu, F. Qu, X. Hong, and H. Zhang, "A small-sample wind turbine fault detection method with synthetic fault data using generative adversarial nets," *IEEE Trans. Ind. Informat.*, vol. 15, no. 7, pp. 3877–3888, Jul. 2019, doi: 10.1109/TII.2018.2885365.

[13] L. Ma, X. Zhang, K. Wang, X. Hou, L. Chen, W. Han, and X. Du, "Generative adversarial message passing-based anomaly detection," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 37, nos. 1–2, pp. 1–5 and 20, Mar. 2025, doi: 10.1007/s44443-025-00021-6.

[14] A. Geiger, D. Liu, S. Alnegheimish, A. Cuesta-Infante, and K. Veeramachaneni, "TadGAN: Time series anomaly detection using generative adversarial networks," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2020, pp. 33–43, doi: 10.1109/BIGDATA50022.2020.9378139.

[15] D. Li, D. Chen, B. Jin, L. Shi, J. Goh, and S. K. Ng, "MAD-GAN: Multivariate anomaly detection for time series data with generative adversarial networks," in *Proc. Artif. Neural Netw. Mach. Learn.* Cham, Switzerland: Springer, 2019, pp. 703–714, doi: 10.1007/978-3-030-30493-5_64.

[16] C. Zhang, S. Li, H. Zhang, and Y. Chen, "A comprehensive review on GANs for time-series signals," *Neural Comput. Appl.*, vol. 34, no. 5, pp. 3551–3571, Mar. 2022, doi: 10.1007/s00521-022-06888-0.

[17] A. K. S. Jardine, D. Lin, and D. Banjevic, "A review on machinery diagnostics and prognostics implementing condition-based maintenance," *Mech. Syst. Signal Process.*, vol. 20, no. 7, pp. 1483–1510, Oct. 2006, doi: 10.1016/j.ymssp.2005.09.012.

[18] C. Huang, S. Bu, H. H. Lee, C. H. Chan, S. W. Kong, and W. K. C. Yung, "Prognostics and health management for predictive maintenance: A review," *J. Manuf. Syst.*, vol. 75, pp. 78–101, Aug. 2024, doi: 10.1016/j.jmsy.2024.05.021.

[19] M. Noroozi, "Self-labeled conditional GANs," 2020, *arXiv:2012.02162*.

[20] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of GANs for improved quality, stability, and variation," 2017, *arXiv:1710.10196*.

[21] B. Yilmaz and R. Korn, "A comprehensive guide to generative adversarial networks (GANs) and application to individual electricity demand," *Expert Syst. Appl.*, vol. 250, Sep. 2024, Art. no. 123851, doi: 10.1016/j.eswa.2024.123851.

[22] B. Sabiri, B. El Asri, and M. Rhanoui, "Impact of hyperparameters on the generative adversarial networks behavior," in *Proc. 24th Int. Conf. Enterprise Inf. Syst.*, 2022, pp. 428–438, doi: 10.5220/0011115100003179.

[23] *Train Wasserstein GAN With Gradient Penalty (WGAN-GP)*. Accessed: Jul. 15, 2026. [Online]. Available: https://www.mathworks.com/help//releases/R2021a/deeplearning/ug/trainwasserstein-gan-with-gradient-penalty-wgan-gp.html

[24] W. Choi, J. Cho, S. Lee, and Y. Jung, "Fast constrained dynamic time warping for similarity measure of time series data," *IEEE Access*, vol. 8, pp. 222841–222858, 2020, doi: 10.1109/ACCESS.2020.3043839.

[25] D. Zhang, M. Ma, and L. Xia, "A comprehensive review on GANs for time-series signals," *Neural Comput. Appl.*, vol. 34, no. 5, pp. 3551–3571, 2022.

[26] (2022). *GED7513 6B Product Poster—2021–2022*. [Online]. Available: https://www.gevernova.com/content/dam/gepower-new/global/en_US/downloads/gas-new-site/products/gas-turbines/ged7513-6b-product-poster-2021-2022.pdf
[27] W. Jiang, Y. Hong, B. Zhou, X. He, and C. Cheng, "A GAN-based anomaly detection approach for imbalanced industrial time series," *IEEE Access*, vol. 7, pp. 143608–143619, 2019, doi: 10.1109/ACCESS.2019.2944689.
[28] M. A. Bashar and R. Nayak, "ALGAN: Time series anomaly detection with adjusted-LSTM GAN," *Int. J. Data Sci. Anal.*, vol. 20, no. 6, pp. 5719–5737, Nov. 2025, doi: 10.1007/s41060-025-00810-2.

**MARIA DEL CARMEN RUBIALES MENA** was born in Algeciras, Spain, in 1998. She received the B.S. degree in industrial engineering and the dual M.S. degree in industrial engineering and in smart industry from the ICAI School of Engineering, in 2020 and 2022, respectively. She is currently pursuing the Ph.D. degree in engineering systems modeling, focusing on the applications of artificial intelligence and in particular, generative artificial intelligence, to the maintenance of industrial equipment.

Since 2023, she has been a Business Intelligence Engineer with Endesa Energia, focusing on data manipulation and exploitation of large databases related to the electricity market, based on Oracle, PostgreSQL, and Redshift, among others. Her areas of research interests include the application of artificial intelligence techniques to the monitoring and diagnosis of industrial processes, time series forecasting, and design and manipulation of large databases.

**ANTONIO MUÑOZ SAN ROQUE** received the degree in electrical engineering and the Ph.D. degree from Universidad Pontificia Comillas, Madrid, Spain, in 1991 and 1996, respectively.

He is currently a Research Staff with the Instituto de Investigación Tecnológica, Madrid, and a Full Professor with the ICAI School of Engineering, Comillas Pontifical University. His areas of research interests include the application of artificial intelligence techniques to the operation and maintenance of industrial processes, electricity markets analysis and optimal operation, and time series forecasting.

**MIGUEL Á. SANZ-BOBI** (Senior Member, IEEE) is currently a Professor with the School of Engineering, Department of Computer Science and Artificial Intelligence, Comillas Pontifical University, Madrid, Spain, and also a Researcher with the Institute for Research and Technology (IIT), Comillas Pontifical University. He divides his time between teaching and research in the field of artificial intelligence, applied to the diagnosis and maintenance of industrial processes. He has been a primary researcher in over 40 industrial projects spanning the last 30 years, focusing on the diagnosis of industrial processes, the incipient anomalies, reliability techniques, and data-driven maintenance approaches. All these projects were based on a combination of different artificial intelligence techniques.

**DANIEL GONZALEZ-CALVO** received the Ph.D. degree in industrial engineering from the University of La Laguna, Spain, in 2021. He is responsible for maintenance planning in Enel Green Power and Thermal Generation. He has contributed to multiple projects and events dedicated to predictive maintenance and has focused in advancing the adoption and practical use of artificial intelligence within industrial environments. His specialized in data-driven applications of artificial intelligence, focusing on leveraging advanced analytics and machine learning to optimize predictive maintenance.

**TOMÁS ÁLVAREZ-TEJEDOR** is currently the Head of Iberia Combined Cycle Gas Turbine Oil and Gas Maintenance, Enel Green Power and Thermal Generation. He dedicates his work to the maintenance of combined cycle power plants, with a strong emphasis on predictive maintenance strategies and the dissemination of artificial intelligence models applied to this domain. Over the course of his career, he has led numerous industrial projects focused on improving reliability, detecting incipient anomalies, and implementing data-driven maintenance techniques.