ADVANCED INTELLIGENT SYSTEMS
Open Access
www.advintellsyst.com

Synthetic Electrocardiogram Spectrogram Generation Using Generative Adversarial Network-Based Models: A Comparative Study

Giovanny Barbosa-Casanova, Norelli Schettini, Winston Percybrooks, and Begoña García-Zapirain*

According to the World Health Organization, cardiovascular diseases are the leading cause of death worldwide. The electrocardiogram (ECG) is a widely used noninvasive method for detecting these conditions. However, analyzing long-duration ECG signal recordings can be highly time-consuming for medical professionals. Machine learning and deep learning techniques have emerged as valuable tools to assist in diagnosis. However, class imbalance in medical datasets poses a significant challenge. This work presents a comparative analysis of three generative adversarial network (GAN)-based models—deep convolutional GAN, conditional GAN, and Wasserstein GAN with gradient penalty (WGAN-GP)—to generate synthetic ECG spectrograms. The proposed models are evaluated using the Fréchet inception distance and structural similarity index measure. The results indicate that WGAN-GP models outperform the other two models in terms of intraclass diversity and data quality. These findings suggest that GAN-generated spectrograms can help mitigate data imbalance issues and improve ECG classification models.

1. Introduction

Cardiovascular diseases (CVDs) are a group of disorders affecting the heart and blood vessels. Most are related to atherosclerosis, a condition that develops when plaque accumulates in the arterial walls, narrowing them and restricting blood flow.[1–3] According to data from the World Health Organization (WHO), CVDs are the leading cause of death in the world.[4] In 2019, these diseases caused 32% of all global reported deaths. Of these deaths, 85% were due to heart attack and stroke. Low- and middle-income countries are particularly ill-prepared to face the challenge of CVDs mainly due to the absence of early detection programs and timely access to treatment.[1]

One widespread method used to detect abnormalities in the behavior of the heart is the electrocardiogram (ECG), a noninvasive test that records the heart's electrical activity through electrodes positioned on the body surface.[5,6] The standard ECG is a test that lasts ≈10 min, and during it, the morphology and the waveform of the ECG signal are evaluated. However, due to the short duration of the test, some abnormalities may not be detected. Ambulatory monitoring techniques such as Holter or cardiac event monitors are used in these cases. Nevertheless, for long-term monitoring, ECG signal analysis is tedious and highly time-consuming for specialists.[7–10] Therefore, techniques for analysis, abnormalities detection, and classification of ECG signals based on machine learning (ML) and deep learning (DL) methods are valuable tools for supporting diagnostic tasks. However, one of the leading challenges in developing these tools is the requirement of a large amount of labeled data.[11–13] In the particular case of ECG signals, publicly available databases are highly imbalanced, which affects the performance in detecting or classifying minority classes.[9] Multiple works focused on detecting ECG signal abnormalities have been developed,[6–8,10,14–27] and among them, those that employ convolutional neural networks (CNNs) and spectrograms extracted from the raw ECG signals have shown better classification performance.[6,20–27] However, most of these works mainly use the MIT-BIH Arrhythmia Database,[28] which is highly imbalanced since more than 80% of their data correspond to the majority class.[9,28]

The aforementioned shows the necessity of acquiring a larger quantity of ECG signals to balance the databases used for training of the ML and DL models. Nevertheless, gathering medical data, especially ECG signals, requires specialized staff to acquire,

G. Barbosa-Casanova
Department of Electricity and Electronics
Universidad Francisco de Paula Santander
Av. Gran Colombia No. 12E-96, Colsag, Cúcuta 540003, Norte de Santander, Colombia
E-mail: geinergiovannybc@ufps.edu.co

G. Barbosa-Casanova, B. García-Zapirain
eVIDA Research Group
Universidad de Deusto
Av. de las Universidades 24, Bilbao 48007, Vizcaya, España

N. Schettini, W. Percybrooks
Department of Electrical and Electronics Engineering
Universidad del Norte
Km 5, Vía Puerto Colombia, Puerto Colombia 081007, Atlántico, Colombia

The ORCID identification number(s) for the author(s) of this article can be found under https://doi.org/10.1002/aisy.202500705.

DOI: 10.1002/aisy.202500705

label, and preprocess the data.[9] Furthermore, acquiring data from sudden cardiac events such as heart attacks, or isolated events, such as some arrhythmia types is challenging.[12] On the other hand, when working with medical data regulations related to ethical considerations, privacy, and protection of clinical data, must be considered.[9,12,13,29] Even when data are available, legal requirements for acquisition often involve time-consuming processes that delay research progress.[29] To address the above, many studies suggest the usage of synthetic ECG data, which allows for increasing the diversity and quantity of available data for training ML and DL models.[9,11–13,29,30] Similarly, these studies indicate the existence of various data generation techniques. Among them, those based on the utilization of generative adversarial networks (GANs) have been successfully employed in the generation of medical data.

This work presents a performance comparison of three types of GAN-based models for generating synthetic ECG spectrograms. The models implemented are the deep convolutional GAN (DCGAN), conditional GAN (CGAN), and Wasserstein GAN with gradient penalty (WGAN-GP). These models were trained on spectrograms derived from time-domain ECG signals. The performance of the models was evaluated using the structural similarity index measure (SSIM) and the Fréchet inception distance (FID) score. This work addresses a fundamental gap in medical data augmentation by introducing the first comparative study of GAN-based ECG spectrogram generation. The contributions of this work are as follows: 1) Pioneer application of three GAN architectures (DCGAN, CGAN, and WGAN-GP) specifically adapted for synthetic ECG spectrogram generation—a completely unexplored application domain. 2) First systematic evaluation of GAN models for time–frequency ECG representation synthesis, providing crucial insights for future medical data augmentation strategies. 3) Empirical validation that GAN-generated spectrograms effectively improve classification performance in imbalanced cardiac datasets, opening new avenues for robust diagnostic system development.

While GAN architectures are established, their application to ECG spectrogram synthesis represents a novel paradigm that addresses critical limitations in current medical data augmentation approaches.

The remainder of this article is structured as follows: Section 2 presents related works. Section 3 describes the proposed approach, including the dataset employed, data preprocessing, the generative models developed, and the experimental setup. Section 4 presents the experimental results, and Section 5 provides the conclusions.

2. Related Works

2.1. Spectrogram-Based Methods for ECG Signal Analysis

Farag[6] developed a self-contained CNN that integrates short-time Fourier transform (STFT) computation directly within its architecture for arrhythmia detection from ECG signals. Unlike conventional approaches requiring external preprocessing, the proposed STFT-CNN automatically learns discriminative time–frequency features during training. The model was implemented on a resource-constrained edge device using data from the MIT-BIH Arrhythmia Database, targeting real-time inference. Performance evaluation showed that the STFT-CNN achieved up to 99.1% accuracy and 95% F1-score at the edge with a maximum model size of 90 KB, an average inference time of 9 ms, and a maximum memory usage of 12 MB. These results demonstrate that embedding STFT within CNN layers enables efficient and accurate ECG classification, making the approach particularly suitable for portable and wearable healthcare devices.

Safdar et al.[20] proposed an ECG classification method that combines wavelet-based denoising, frequency filtration, and STFT to generate spectrograms as inputs to a CNN. Using the PTB-XL dataset, the authors focused on distinguishing normal signals from anteroseptal myocardial infarction. Preprocessing involved the discrete wavelet transformation to remove noise and a cutoff frequency of 2 Hz to reduce data size without losing diagnostic information. Spectrograms (64×64 grayscale images) were then created and classified using a CNN with four convolutional layers. Experimental results showed that spectrogram-based inputs achieved higher performance (99.06% accuracy, 100% precision, and 99.83% sensitivity) compared to raw signal inputs while also halving memory usage. The study highlights that spectrogram representations provide more distinguishable features than time-domain signals, enabling simpler CNN architectures to achieve state-of-the-art accuracy in ECG classification.

Alqudah et al.[21] conducted a comparative study evaluating four spectrum-based ECG representations—log-scale STFT, mel-scale STFT, bispectrum, and third-order cumulant—combined with four CNN architectures: AOCT-Net, MobileNet, SqueezeNet, and ShuffleNet. The experiments used 10 502 beats from the MIT-BIH Arrhythmia Database, covering six classes (N, LBBB, RBBB, PVC, APB, and aAP). Results showed that the best-performing CNN was MobileNet with an overall accuracy of 93.8%, while the bispectrum achieved the highest representation performance with an overall accuracy of 93.7%. These findings demonstrate that spectral representations offer strong discriminatory power for arrhythmia classification when combined with lightweight CNNs suitable for mobile and embedded systems.

Eleyan et al.[22] proposed a spectrogram-based arrhythmia classification framework that integrates feature fusion through a three-channel deep learning model. In this approach, ECG signals are transformed into spectrograms using the STFT, while additional visual features are extracted via histograms of oriented gradients and local binary patterns. Each representation is processed by a dedicated 2DCNN, and the resulting feature maps are concatenated and fed into a gated recurrent unit (GRU) for temporal modeling and final classification. The method was evaluated on the MIT-BIH and BIDMC ECG datasets, considering both three-class and five-class classification tasks. Experimental results demonstrated that the proposed model outperformed existing baseline methods, achieving higher accuracy and robustness, thus highlighting the benefits of combining spectrogram and handcrafted visual features within a unified deep learning architecture.

Król-Józaga[23] investigated atrial fibrillation (AF) detection using CNNs applied to three 2D ECG representations: spectrograms, scalograms, and attractor reconstruction (AR). The study employed 5 s raw ECG segments from the MIT-BIH Atrial

Fibrillation Database (AFDB), without preprocessing or handcrafted feature extraction. The CNN architecture was trained and evaluated on both balanced and unbalanced datasets using cross-validation and train-test splits. Results showed that scalograms achieved the best performance with an F1-score of 94%, followed by spectrograms (93%) and AR (89%). These findings demonstrate that short ECG segments transformed into 2D representations can enable effective AF detection with CNNs.

Siekierski and Siwek[24] presented a heartbeat classification method that leverages multisignal ECG spectrograms alongside a CNN enriched with residual blocks. Utilizing data from the MIT-BIH Arrhythmia Database, the authors transformed ECG signals into spectrogram representations and trained a residual CNN architecture to distinguish between different heartbeat types. Results showed that the proposed method achieved an overall accuracy of 98.37%.

Wei et al.[25] introduced a fast and automated DL method for detecting AF using ECG-derived spectrogram representations. In their approach, raw ECG signals undergo preprocessing, are converted into spectrograms, and subsequently fed into a fine-tuned EfficientNet-B0, a pretrained CNN architecture. Evaluated on AF detection tasks, the model achieved impressive performance with an F1-score of 88.2% and an overall accuracy of 97.3%. These results show that combining time–frequency representations with a pretrained CNN architecture enables efficient and highly accurate AF detection.

Singhal and Kumar[26] proposed GSMD-SRST, a multilevel ECG arrhythmia classification framework that integrates Group Sparse Mode Decomposition (GSMD) with the Superlet Transform (SRST). GSMD decomposes ECG signals into constituent sparse modes, improving separation of morphological and rhythmic components, while SRST generates high-resolution time–frequency spectrograms that are subsequently processed by deep CNN architectures, including VGG19, ResNet-18, GoogleNet, and AlexNet. The authors validated their approach using five databases: MIT-BIH Atrial Fibrillation, MIT-BIH Malignant Ventricular Ectopy, Creighton University Ventricular Tachycardia, MIT-BIH Arrhythmia, and Mendeley-II databases. Their proposed method, along with the VGG19 model, achieved an overall accuracy of 99.2%.

Gupta et al.[27] introduced CardioNet, a lightweight DL framework for myocardial infarction (MI) detection using ECG sensor data. Their approach transforms filtered ECG sensor data into time–frequency spectrograms using three methods: the STFT, the Gaussian window-based S-Transform (ST), and the Smoothed Pseudo-Wigner–Ville Distribution. These spectrograms are then processed by several models—benchmark networks such as SqueezeNet and AlexNet, as well as the CardioNet architecture. The study, evaluated on the PTB-ECG dataset, reported an average classification accuracy of 99.82%, a specificity of 99.57%, and a sensitivity of 99.97%. These results demonstrate that the proposed system is suitable for use in wearable devices combined with a cloud-based algorithm for automatic heart disease screening.

Table 1 provides a systematic comparison of spectrogram-based ECG studies, summarizing methodological approaches, datasets utilized, performance metrics, and key findings from the reviewed literature.

2.2. GAN-Based Methods for Synthetic ECG Data Generation

Safdar et al.[31] proposed a data augmentation method in which the new synthetic ECG signals are generated from the original signals. In this method, the original ECG signals, extracted from the PTB-XL ECG dataset, are divided into segments, and then these segments are rearranged. The authors defined two classes of segments: the host and the guest. The first ones are segments that remain in their original place. These segments can be morphologically adapted to integrate with the guest segments. On the other hand, the guest is a segment that is displaced during the rearranged process. The similarity between the generated and the original ECG signals was evaluated by converting the signals into images and computing the SSIM.

Zubair et al.[10] proposed an oversampling strategy to address the problem of imbalanced data. This strategy consists of two steps. First, samples corresponding to the majority class (normal beat) from the MIT-BIH Arrhythmia Database are selected. These samples must be highly similar to samples from the minority class (supraventricular ectopic beat). Then, the selected samples are transformed into samples from the minority class using a translation loss function. To evaluate the effect of the proposed strategy, the authors trained two classification models, the first one with imbalanced data and the other one with data balanced with the proposed method. The metrics employed during the tests were accuracy, sensitivity, specificity, and positive productivity. The results of the tests demonstrated improved classification performance for the model trained with data balanced using the proposed method.

Adib et al.[9] assessed the capacity of five GAN-based models to generate synthetic ECG signals. All these models were trained using signals corresponding to the majority class (normal) from the MIT-BIH Arrhythmia Database. The performance of the models was evaluated using five methods proposed by the authors. The results of the tests show that all models can generate heartbeats with similar morphological features. However, in terms of visual inspection, the models, bidirectional long short-term memory deep convolutional GAN (BiLSTM-DC GAN), GAN, and Wasserstein GAN (WGAN), obtained the best results. On the other hand, among these models, the classical GAN obtained the best productivity rate.

Bechinia et al.[18] proposed using a variant of GANs known as auxiliary classifier GAN as a data augmentation method. The generated data correspond to images representing three types of time-domain ECG signals from the MIT-BIH Arrhythmia Database. The authors trained two different models using both an imbalanced dataset and a balanced dataset created with the proposed method to compare the performance of their approach. The metrics used in this work were accuracy, precision, recall, and F1 score. The results of the tests showed that the models trained with a balanced dataset using the proposed method outperformed the models trained with the imbalanced dataset.

Wulan et al.[30] presented a WaveNet-based model and two GAN-based models for generating synthetic ECG signals corresponding to normal, left bundle branch block, and right bundle branch block beats. The models were trained using data from the MIT-BIH Arrhythmia Database. The GAN-Train and GAN-Test scores were used to evaluate the performance of the models. The

Table 1. Overview of spectrogram-based approaches in ECG studies.

Reference	Method	Database	Metrics	Comments
Farag[6]	STFT-CNN model	The MIT-BIH Arrhythmia Database	Accuracy, recall, precision, and F1-score	In this study, spectrogram-based representations of ECG signals were utilized to train a classification model capable of discriminating among five beat classes. The study does not address the issue of inbalanced classes (It does not explore data augmentation).
Safdar et al.[20]	CNN model	The PTB-XL Dataset	Accuracy, sensitivity, specificity, and precision	In this study, spectrogram-based representations of ECG signals were utilized to train a classification model capable of discriminating among five beat classes. The study addresses the issue of inbalanced classes, but it does so by applying horizontal flipping and image contrast enhancement.
Alqudah et al.[21]	AOCT-Net, MobileNet, SqueezeNet, and ShuffleNet models	The MIT-BIH Arrhythmia Database	Precision, specificity, sensitivity, accuracy, F1-score, and Matthews correlation coefficient (MCC)	In this study, four spectral representations of ECG signals were utilized to train four classification models, per representation, designed to discriminate between six ECG beat classes. The study does not address the issue of inbalanced classes (It does not explore data augmentation).
Eleyan et al.[22]	2DCNN-GRU model	The MIT-BIH Arrhythmia Database and the BIDMC Congestive Heart Failure Database	Accuracy, Recall, Precision, and F1-score	In this study, spectrograms extracted from ECG signals along with visual features were utilized to train classification models designed to discriminate between three or five ECG signal classes. The study does not address the issue of inbalanced classes (It does not explore data augmentation).
Król-Józaga[23]	CNN model	MIT-BIH Atrial Fibrillation Database	Precision, Recall, Accuracy, Area Under the ROC Curve (AUC), and F1-Score	In this study, spectrograms and two other 2D representations extracted from ECG signals were used to train representation-based classification models designed to detect AF from ECG signals. The study does not address the issue of inbalanced classes (It does not explore data augmentation).
Siekierski and Siwek[24]	CNN + Residual Blocks model	The MIT-BIH Arrhythmia Database	Accuracy, Sensitivity, and Specificity	In this study, spectrograms extracted from ECG signals were utilized to train a classification model capable of discriminating among five beat classes. The study addresses the issue of inbalanced classes, but it does so by duplicating four times the samples of the minority class.
Wei et al.[25]	Fine-tuned EfficientNet-B0 model	The PhysioNet Computing in Cardiology Challenge 2017 Database	Precision, recall, specificity, accuracy, F1-score, MCC, and AUC	In this study, spectrograms extracted from ECG signals were used to train a classification model designed to detect AF from ECG signals. The study addresses the issue of inbalanced classes, but it does so by creating and adding another set of spectrograms with a different set of parameters.
Singhal and Kumar[26]	VGG19, RESNET-18, GoogleNet, and Alexnet models	The MIT-BIH Atrial Fibrillation Database, the MIT-BIH Malignant Ventricular Ectopy Database, the Creighton University Ventricular Tachycardia Database, the MIT-BIH Arrhythmia Database, and the Mendeley-II Database	Accuracy, sensitivity, specificity, precision, and F1-score	In this study, spectrograms extracted from ECG signals were utilized to train a classification model designed to discriminate between three ECG signal classes. The study does not address the issue of inbalanced classes (It does not explore data augmentation).
Gupta et al.[27]	Cardionet model	The PTB-ECG dataset	Precision, accuracy, F1-score, sensitivity, and specificity	In this study, spectrograms extracted from ECG signals were used to train a classification model designed to detect MI from ECG signals. The study addresses the issue of inbalanced classes, but it does so by rotating the spectrograms within a range of $-5°$ to $5°$.
Proposed method	GAN-based models	The MIT-BIH Arrhythmia Database	SSIM and FID	In this study, spectrograms extracted from ECG signals were used to train GAN-based models designed to face the imbalanced datasets problem by generating synthetic ECG spectrograms.

tests performed by the authors show that the data distribution of the signals generated by the GAN-based models is more closely approximated to the distribution of actual data.

Hazra and Byun[29] presented SynSigGAN, a GAN-based model that enables the generation of synthetic signals corresponding to encephalograms, electromyographic, photoplethysmography, and 17 ECG types. The training and validation of the model were performed using four biomedical signals databases, including the MIT-BIH Arrhythmia Database. A bidirectional grid long short-term memory (BiGridLSTM) architecture was used for the generator, and a CNN architecture was used for the discriminator. The metrics root mean square error (RMSE), percent root mean square difference (PRD), mean absolute error (MAE), Fréchet distance (FD), and Pearson's correlation coefficient (PCC) were utilized in this work.

Nankani and Dutta Baruah[11] proposed a deep convolutional CGAN for generating three types of cardiac beats following the Association for the Advancement of Medical Instrumentation recommendations. Four standard public databases were employed: MIT-BIH Arrhythmia Database, MIT-BIH Supraventricular Arrhythmia Database, St. Petersburg INCART 12-lead Arrhythmia Database, and China Physiological Signal Challenge 2020. The authors utilized Gaussian noise as latent input to the generator, the binary cross-entropy loss function for the discriminator, and soft labels for faster convergence during training. The model's performance was evaluated using five quantitative metrics: FD, dynamic time warping (DTW), RMSE, maximum mean discrepancy (MMD), and time warp edit distance (TWED). For evaluating the impact on classification performance, the authors implemented a parallel CNN that achieved an accuracy of 0.950, specificity of 0.955, precision of 0.853, and sensitivity of 0.901.

Neifar et al.[12] proposed a GAN-based approach for ECG signal synthesis that incorporates statistical shape priors to better capture the dynamics of ECG signals. In this method, ECG waveforms are represented as 2D shapes modeling both temporal and amplitude variations, and statistical shape models are constructed through clustering, alignment, and principal component analysis of ECG signals. The generator then leverages these priors to produce synthetic signals that preserve realistic morphology. The MIT-BIH Arrhythmia Database was used for evaluation, considering normal, premature ventricular contraction, and fusion beats. The performance of the approach was assessed through qualitative analysis, including cardiologist evaluation of real versus synthetic beats, and quantitative experiments measuring classification improvements and metrics such as RMSE, MAE, earth mover's distance (EMD), and DTW. Results demonstrated that the proposed method generates ECG signals with morphology closer to real signals compared to state-of-the-art GAN baselines.

Rai et al.[13] proposed GAN-SkipNet, a framework designed to mitigate class imbalance in ECG-based arrhythmia detection. Using the MIT-BIH Arrhythmia Database, the authors employed a Bi-LSTM-based GAN to generate synthetic signals for the minority supraventricular (S) and fusion (F) classes. Signal quality was assessed through MSE, SSIM, and PCC with real ECGs. To evaluate the impact on classification performance, the authors introduced three models: SkipCNN, SkipCNN + LSTM, and SkipCNN + LSTM + Attention, which were further enhanced via an ensemble approach. Furthermore, they employed the evaluation metrics: precision, recall, and F1-score. The results confirmed that GAN-SkipNet not only balances imbalanced datasets but also significantly improves arrhythmia classification performance, showing strong potential for clinical application in real-time ECG monitoring.

Yang et al.[32] introduced CECG-GAN, a CGAN enhanced with transformer modules, to address class imbalance and waveform artifacts in ECG-based diagnostics. Using the MIT-BIH Arrhythmia and CSPC2020 datasets, the model employs an encoder–decoder structure to map ECG signals into low-dimensional latent spaces, enabling controlled generation of scarce abnormal beats. Transformer-based parallel processing reduces the waveform jitter and accelerates training compared to recurrent neural network- and CNN-based GANs. Evaluation included quantitative metrics such as PRD, FD, RMSE, and MAE, as well as classification experiments with ResNet50. Results showed that CECG-GAN significantly improved the recognition of minority classes, achieving F1-scores above 98% for arrhythmia categories after dataset augmentation, while maintaining high fidelity between synthetic and real ECGs. These findings highlight the model's capacity to mitigate dataset imbalance and enhance diagnostic performance in cardiac health monitoring.

Despite the significant advances in GAN-based synthetic ECG generation demonstrated by the aforementioned works, a critical limitation emerges that constrains the full potential of these approaches: all existing methods focus exclusively on generating time-domain ECG signals, completely overlooking time–frequency representations such as spectrograms. This represents a substantial gap in the current literature, particularly considering that recent studies have consistently shown that CNN models trained on ECG spectrograms achieve superior classification performance compared to those operating directly on raw time-domain signals.[6,20–27]

The absence of spectrogram generation capabilities in existing ECG-related GAN frameworks is especially problematic given that: 1) spectrograms capture both temporal and frequency-domain characteristics often overlooked in time-domain representations, 2) they provide complementary and potentially more robust features for arrhythmia classification, and 3) they enable better visualization and interpretation of cardiac abnormalities. Furthermore, while data augmentation through synthetic time-domain ECG generation has shown promise, the potential for ECG spectrogram-based data augmentation remains unexplored. This is particularly noteworthy since GAN-based spectrogram generation has already been successfully employed in other fields, such as speech synthesis,[33,34] speech emotion recognition,[35] animal sound classification,[36,37] seismic waveform synthesis,[38] heart sound classification,[39] and human activity classification,[40] where it has proven effective for improving data diversity and model performance. Hence, extending these advances to ECG represents a significant missed opportunity to address dataset imbalance issues more effectively. It is also important to note that conventional image augmentation techniques (e.g., rotation, flipping, cropping) cannot be directly applied to spectrograms, since such transformations distort the underlying temporal and frequency axes, thereby altering the physiological meaning of the signal and introducing artifacts that compromise diagnostic validity.[39]

Table 2. Performance metrics.

Metric	Description
SSIM	This metric quantifies the structural similarity between two images[31]
Accuracy	Ratio between the number of correctly classified samples and the total of samples[45]
Sensitivity	Proportion of actual positive samples that are correctly classified. This metric is also known as Recall or the True Positive Rate (TPR)[63]
Specificity	Proportion of actual negative samples that are correctly classified. This metric is also known as the True Negative Rate (TNR)[63]
Positive productivity	Proportion of positive predictions that are correctly classified. This metric is also known as Precision[63]
F1-score	It is a metric that combines Precision and Recall into a single value by calculating their harmonic mean[63]
GAN-Train	This metric is computed by training a classification model on data generated by a GAN and testing its performance on real data[30]
GAN-Test	This metric is computed by training a classification model on real data and testing its performance on data generated by a GAN[30]
RMSE	This metric measures the square root of the average squared differences between the predicted values and the actual values[63]
PRD	This metric quantifies the distortion or difference between two signals by measuring the relative error[64]
MAE	This metric measures the average of the absolute differences between the predicted values and the actual values[63]
FD	Geometric measurement used to quantify the similarity between continuous curves or trajectories[65]
PCC	It measures the strength and direction of the linear relationship between two variables[63]
DTW	It is a method employed to compare two-time series[66]
MMD	It quantifies the mean discrepancy of two data distributions in kernel space[67]
TWED	It measures the similarity between two given time series, considering both the alignment of their elements and the temporal differences between them[68]
EMD	Also known as Wasserstein Distance, it is a metric that quantifies the minimum cost to transform one probability distribution into another[50]

Table 2 provides an overview of the performance metrics mentioned earlier in this subsection.

3. Experimental Section

3.1. Proposed Approach

This section introduces the proposed approach for comparing the performance in the generation of ECG spectrograms of three GAN-based models. **Figure 1** presents an overview of the proposed approach. In the initial phase of the methodology, 10 s samples of ECG signals corresponding to five types of beats are extracted from the MIT-BIH Arrhythmia Database. Then, the samples are grouped into the "normal" and "abnormal" classes. Next, each time-domain ECG signal is transformed into a 2D time–frequency spectrogram through the STFT. The spectrograms are saved as image files and are used to train the networks: DCGAN, CGAN, and WGAN-GP. The models' performance in generating synthetic spectrograms is measured using the SSIM and the FID score. Furthermore, the models that generated the synthetic spectrograms with the highest quality per class were used as a data augmentation method to balance the datasets employed for training 2D-CNN classifier models.

3.2. The Dataset

Several publicly available ECG databases have been widely used for training ML and DL models to detect ECG signal abnormalities.[41] Among the most prominent datasets are the MIT-BIH Arrhythmia Database,[28] the PTB-XL Database,[42] the St. Petersburg INCART Database,[43] and the AFDB.[44] **Table 3** presents relevant information about these databases.

In this work, the MIT-BIH Arrhythmia Database was selected to train the generative models due to its status as the gold standard for arrhythmia classification, its well-established ground truth annotations validated by cardiologists, and its extensive use in benchmark studies that facilitate performance comparisons,[41] ensuring that our synthetic-spectrogram-based ECG sample generation results can be contextualized within existing performance benchmarks that contribute to the cumulative knowledge base in this field. This database contains 48 half-hour excerpts of two-channel ambulatory ECG recordings digitized at a sampling rate of 360 Hz, with each beat having a respective validated label.[28]

Five types of beats were extracted from this database: normal beat (N), left bundle branch block beat (L), right bundle branch block beat (R), premature ventricular contraction beat (V), and atrial premature beat (A). Ten second ECG samples from each type were extracted following the methodology described in ref. [45]. These five beat types were subsequently grouped into two classes: "normal" and "abnormal." **Table 4** presents a detailed description of the information extracted from the MIT-BIH Arrhythmia Database.

3.3. ECG Data Preprocessing

The function "specgram" from the Matplotlib Python Library[46] was used to transform the time-domain ECG signals into 2D time–frequency spectrograms. This function takes the input signal, computes its STFT, and generates a spectrogram. The parameters used in this work were NFFT set to 512, Fs set to 360, Window set to Hanning, and Noverlap set to 128. The generated spectrograms were stored as 256×256 color images in PNG format. An example of a spectrogram for each class is presented in **Figure 2**.

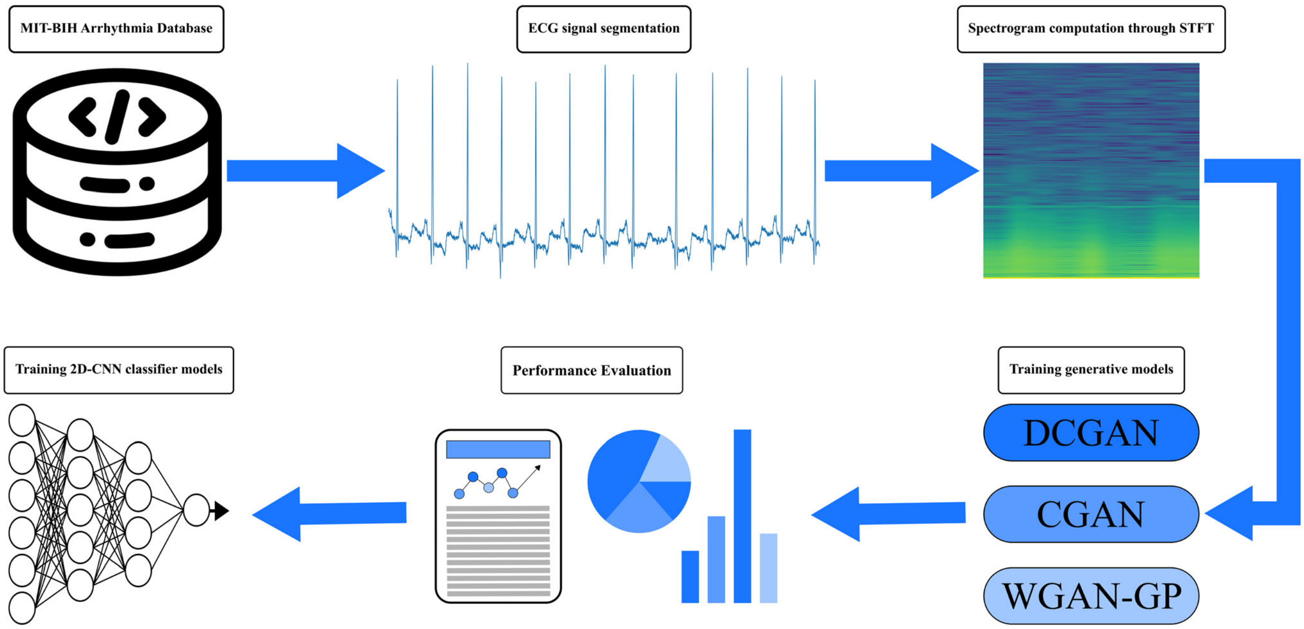

Figure 1. Proposed approach.

Table 3. Characteristics of prominent ECG databases for cardiovascular signal analysis.

Database	Release year	Number of channels	Sampling rate [Hz]	Number of records	Recording duration
MIT-BIH Arrhythmia Database	2005	2	360	48	30 min
PTB-XL Database	2020	12	500	21 837	10 s
St. Petersburg INCART Database	2008	12	257	75	30 min
Atrial Fibrillation Database (AFDB)	2000	2	250	23	10 h

Table 4. Data extracted from the MIT-BIH Arrhythmia Database.

MIT-BIH Arrhythmia Database Code	Record Number	Samples	Class
N	100, 101, 105, 115, 122, 215	1980	Normal
L	109, 111, 214	540	Abnormal
R	118, 124, 212	540	Abnormal
V	106, 223	360	Abnormal
A	207, 209, 232	540	Abnormal

3.4. Generative Models

GANs are a framework proposed by Goodfellow et al.[47] for generating synthetic data. GAN architecture consists of two networks: the generator (G) and the discriminator (D). The functions of G are to capture the real data distribution and generate synthetic (fake) data. In turn, the function of D is to determine if the input data correspond to real or synthetic data. In this framework, both models compete against each other during the training process. G tries to fool D, and D tries to assign the correct label to input data. The loss function used in this framework is defined as follows:

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))] \tag{1}$$

where z is a random noise vector, x is the real data, $p_{\text{data}}(x)$ is the real data distribution, $D(x)$ is the output of the D for a real data sample x, $p_z(z)$ is the prior distribution used by the G, $G(z)$ is the output of the G for an input z, and $D(G(z))$ is the output of the D for a generated data sample $G(z)$.

In this work, we implemented three models—DCGAN, CGAN, and WGAN with gradient penalty (WGAN-GP)—to generate synthetic ECG spectrograms. The DCGANs, originally proposed by Radford et al.,[48] are variations of the original GANs that use a convolutional architecture for the G and the D. Likewise, the CGANs, proposed by Mirza and Osindero,[49] are an extension of the original GANs that introduce the concept of additional information y, which conditions both the G and the D. The loss function of this type of network is defined as follows:

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log D(x|y)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z|y)))] \tag{2}$$

In ref. [50], Arjovsky et al. presented the WGANS, which are variations of the original GANs, that use the EMD (also called Wasserstein-1) as the loss function to improve learning stability and address mode collapse. Furthermore, the D is replaced by a network called the critic that estimates the EMD between real data distribution and synthetic data distribution.

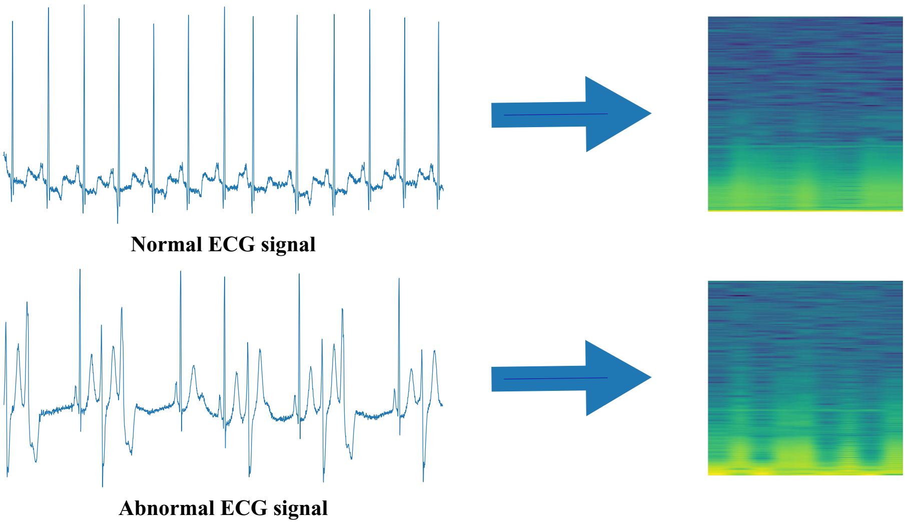

Figure 2. Spectrograms of ECG signals corresponding to normal and abnormal classes.

In this work, the WGAN version presented by Gulrajani et al.[51] called WGAN with gradient penalty (WGAN-GP) was implemented; consequently its loss function takes the form of the EMD plus a penalty that is expressed as follows:

$$L = \mathbb{E}_{\tilde{x} \sim P_g}[D(\tilde{x})] - \mathbb{E}_{x \sim P_r}[D(x)] + \lambda \mathbb{E}_{\hat{x} \sim P_{\hat{x}}} \times [(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2] \tag{3}$$

where $\tilde{x}$ is the generated sample, P_g is the generated data distribution, $D(\tilde{x})$ is the output of the D for a generated sample $\tilde{x}$, x is the real sample, P_r is the real data distribution, $D(x)$ is the output of the D for a real sample x, λ is the gradient penalty coefficient, $\hat{x}$ is interpolated samples between real samples x and generated samples $\tilde{x}$, $P_{\hat{x}}$ is the interpolated samples distribution, and $\|\nabla_{\hat{x}} D(\hat{x})\|_2$ represents the norm of the gradient of the D regarding $\hat{x}$ inputs.

The generative models developed in this work are adaptations of the architecture presented in.[52] This architecture was originally designed to generate 64×64-pixel color images of human faces from a random noise vector of dimensions $100 \times 1 \times 1$. **Table 5** provides comprehensive architectural specifications, including layer-wise parameter distributions and the total trainable parameters. **Figure 3** illustrates the tensor transformation pipeline throughout the models. For the present study, this baseline architecture underwent substantial modifications to accommodate spectrogram synthesis of ECG signals at 256×256-pixel resolution while maintaining the original latent space dimensionality. The DCGAN and WGAN-GP models used spectrograms of the "normal" and "abnormal" classes separately during the training, whereas the CGAN models used both classes simultaneously. All models used a batch size of 128 and were trained for 1000, 2000, and 3000 epochs. The learning rate for DCGAN and CGAN models was 0.0002, and for WGAN-GP models was 0.0001. **Table 6–8** detail the complete architectural specifications for each GAN-based model, including layer-wise parameter counts. **Figure 4–6** provide corresponding tensor flow visualizations demonstrating dimensional transformations across the models' pipeline.

3.5. Experimental Setup

The GAN-based models presented in this work were coded using the Python programming language and the PyTorch framework and were trained on an Ubuntu 22.04.4 LTS operating system running on the g6.12xlarge instance provided by Amazon Elastic Compute Cloud (EC2). This instance is equipped with four NVIDIA L4 Tensor Core GPUs, each with 24 GB of RAM.

The generation of spectrograms and the testing of the developed models were performed on a computer with an Ubuntu 22.04.4 LTS operating system, 32 GB of RAM, Intel Core i9-9900 K CPU,

Table 5. Original architecture of the DCGAN model. Adapted from ref. [52].

Layer	Generator	Discriminator
1	ConvTranspose2d(100, 512) BatchNorm2d(512) ReLU	Conv2d(3, 64) LeakyReLU(0.2)
2	ConvTranspose2d(512, 256) BatchNorm2d(256) ReLU	Conv2d(64, 128) BatchNorm2d(128) LeakyReLU(0.2)
3	ConvTranspose2d(256, 128) BatchNorm2d(128) ReLU	Conv2d(128, 256) BatchNorm2d(256) LeakyReLU(0.2)
4	ConvTranspose2d(128, 64) BatchNorm2d(64) ReLU	Conv2d(256, 512) BatchNorm2d(512) LeakyReLU(0.2)
5	ConvTranspose2d(64, 3) Tanh	Conv2d(512, 1) Sigmoid()
Number of trainable parameters	3 576 604	2 765 568

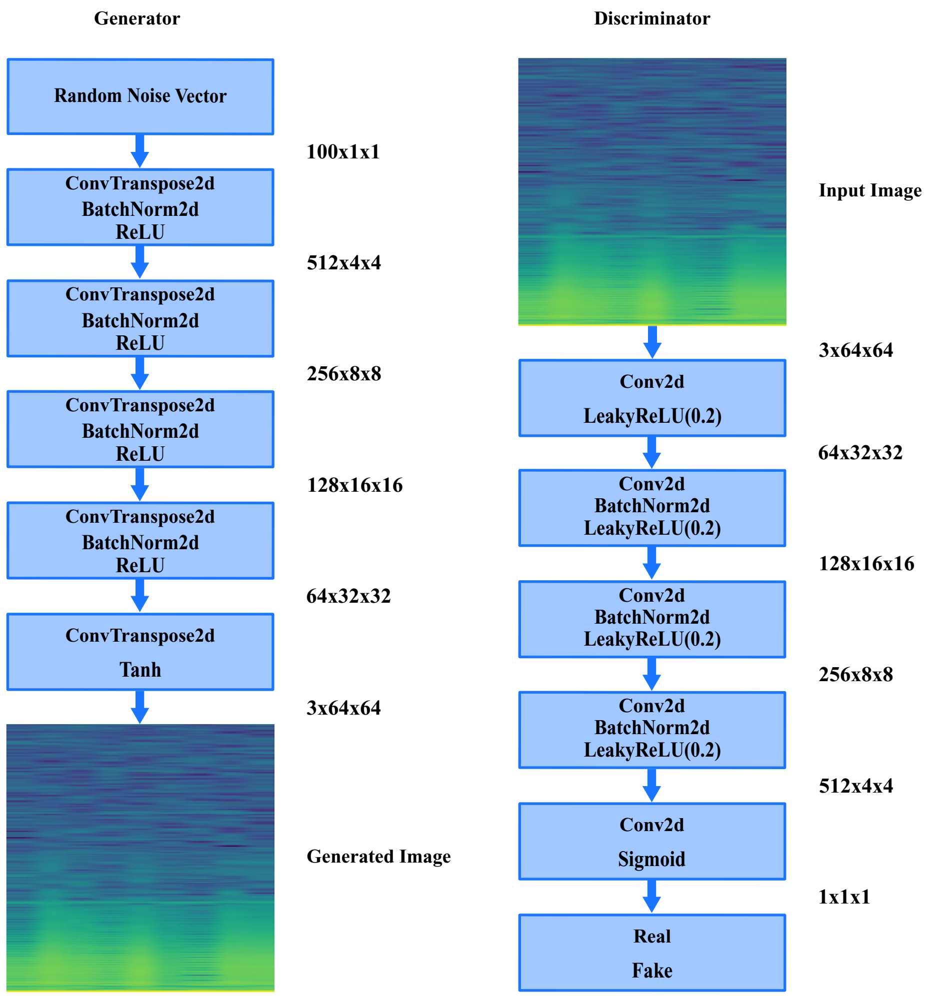

Figure 3. Baseline DCGAN architecture overview with tensor dimension flow.

and two NVIDIA GeForce RTX 2080 Super GPUs, each with 8 GB of RAM. The Python programming language and the PyTorch framework also were used.

4. Experimental Results

4.1. Evaluation Metrics

The SSIM[53] and the FID score[54] were used to assess the performance of the generative models. The SSIM quantifies the structural similarity between two images by comparing their luminance, contrast, and structure.[53,55] This metric has been shown to correlate with human visual perception. The values of this metric range from 0 (images are different) to 1 (images are equal) and are calculated for two given images x and y as follows:

$$\mathrm{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{4}$$

where μ is the mean intensity, σ is the standard deviation, σ_{xy} is the covariance, and C_1 and C_2 are constants to avoid instability.

The FID score is a metric that computes the Fréchet Distance between the feature representations of the real and generated samples extracted using the Inception-V3 model.[54] This metric is widely used to assess the quality of samples generated by GAN-based models, as it has been shown to correlate well with human judgment of visual quality.[56] A lower FID score indicates better performance in sample generation. For two sets of real and generated samples, this metric is calculated as follows:

$$\mathrm{FID}(x, g) = \|m_x - m_g\|_2^2 + (C_x + C_g - 2(C_x C_g)^{1/2}) \tag{5}$$

where m_x and C_x are the mean vector and covariance matrix of feature representations of real samples, respectively; m_g and C_g are the mean vector and covariance matrix of feature representations of generated samples, respectively; and Tr is the matrix trace.

4.2. Generative Model Performance

To evaluate the performance of the models, each of them generated the same quantity of spectrograms that they used during

Table 6. DCGAN model architecture.

Layer	Generator	Discriminator
1	ConvTranspose2d(100, 4096) BatchNorm2d(4096) ReLU	Conv2d(3, 128) LeakyReLU(0.2)
2	ConvTranspose2d(4096, 2048) BatchNorm2d(2048) ReLU	Conv2d(128, 256) BatchNorm2d(256) LeakyReLU(0.2)
3	ConvTranspose2d(2048, 1024) BatchNorm2d(1024) ReLU	Conv2d(256, 512) BatchNorm2d(512) LeakyReLU(0.2)
4	ConvTranspose2d(1024, 512) BatchNorm2d(512) ReLU	Conv2d(512, 1024) BatchNorm2d(1024) LeakyReLU(0.2)
5	ConvTranspose2d(512, 256) BatchNorm2d(256) ReLU	Conv2d(1024, 2048) BatchNorm2d(2048) LeakyReLU(0.2)
6	ConvTranspose2d(256, 128) BatchNorm2d(128) ReLU	Conv2d(2048, 4096) BatchNorm2d(4096) LeakyReLU(0.2)
7	ConvTranspose2d(128, 3) Tanh	Conv2d(4096, 1) Sigmoid()
Number of trainable parameters	185 358 080	179 013 632

Table 7. CGAN model architecture.

Layer	Generator	Discriminator
1	ConvTranspose2d(200, 4096) BatchNorm2d(4096) ReLU	Conv2d(6, 128) LeakyReLU(0.2)
2	ConvTranspose2d(4096, 2048) BatchNorm2d(2048) ReLU	Conv2d(128, 256) BatchNorm2d(256) LeakyReLU(0.2)
3	ConvTranspose2d(2048, 1024) BatchNorm2d(1024) ReLU	Conv2d(256, 512) BatchNorm2d(512) LeakyReLU(0.2)
4	Conv2d(512, 1024) BatchNorm2d(1024) LeakyReLU(0.2) ReLU	ConvTranspose2d(1024, 512) BatchNorm2d(512)
5	ConvTranspose2d(512, 256) BatchNorm2d(256) ReLU	Conv2d(1024, 2048) BatchNorm2d(2048) LeakyReLU(0.2)
6	ConvTranspose2d(256, 128) BatchNorm2d(128) ReLU	Conv2d(2048, 4096) BatchNorm2d(4096) LeakyReLU(0.2)
7	ConvTranspose2d(128, 3) Tanh	Conv2d(4096, 1) Sigmoid()
Number of trainable parameters	191 915 776	180 782 976

Table 8. WGAN-GP model architecture.

Layer	Generator	Critic
1	ConvTranspose2d(100, 4096) BatchNorm2d(4096) ReLU	Conv2d(3, 128) LeakyReLU(0.2)
2	ConvTranspose2d(4096, 2048) BatchNorm2d(2048) ReLU	Conv2d(128, 256) InstanceNorm2d(256) LeakyReLU(0.2)
3	ConvTranspose2d(2048, 1024) BatchNorm2d(1024) ReLU	Conv2d(256, 512) InstanceNorm2d(512) LeakyReLU(0.2)
4	ConvTranspose2d(1024, 512) BatchNorm2d(512) ReLU	Conv2d(512, 1024) InstanceNorm2d(1024) LeakyReLU(0.2)
5	ConvTranspose2d(512, 256) BatchNorm2d(256) ReLU	Conv2d(1024, 2048) InstanceNorm2d(2048) LeakyReLU(0.2)
6	ConvTranspose2d(256, 128) BatchNorm2d(128) ReLU	Conv2d(2048, 4096) InstanceNorm2d(4096) LeakyReLU(0.2)
7	ConvTranspose2d(128, 3) Tanh	Conv2d(4096, 1)
Number of trainable parameters	185 358 080	179 054 464

their training, i.e., 1980 per class. A visual comparison between a spectrogram sample extracted from the training dataset and the spectrograms generated by different GAN architectures for 1000, 2000, and 3000 training epochs for "normal" and "abnormal" classes is presented in **Figure 7** and **8**, respectively. These images show the variation of the generated samples and the models' performance across the training epochs. The intraclass diversity of the data generated by each model was measured using the structural similarity obtained by comparing each spectrogram against the others employing the SSIM. The function "structural_similarity" from the scikit-image library[57] was used to compute this metric per model, and the results are presented in **Table 9**. Values close to 1 in the "Average" column indicate mode collapse. The values in the "Maximum" column show the highest degree of similarity obtained between two spectrograms generated by each model, whereas values in the "Minimum" column show the lowest. Based on these values, it is observed that the WGAN-GP models for 1000 and 2000 epochs and the DCGAN model for 3000 epochs, trained on "abnormal" data, are experiencing mode collapse. Meanwhile, the models trained on "normal" data with the highest SSIM value in the "Average" column are CGAN for 2000 epochs and DCGAN for 1000 epochs. On the other hand, the models that generate the spectrograms with the best intraclass diversity, i.e., with the lowest SSIM value in the "Average" column, are WGAN-GP for 1000 epochs trained on "normal" data and WGAN-GP for 3000 epochs trained on "abnormal" data. Furthermore, the models with the lowest SSIM value in the "Maximum" column are WGAN-GP for

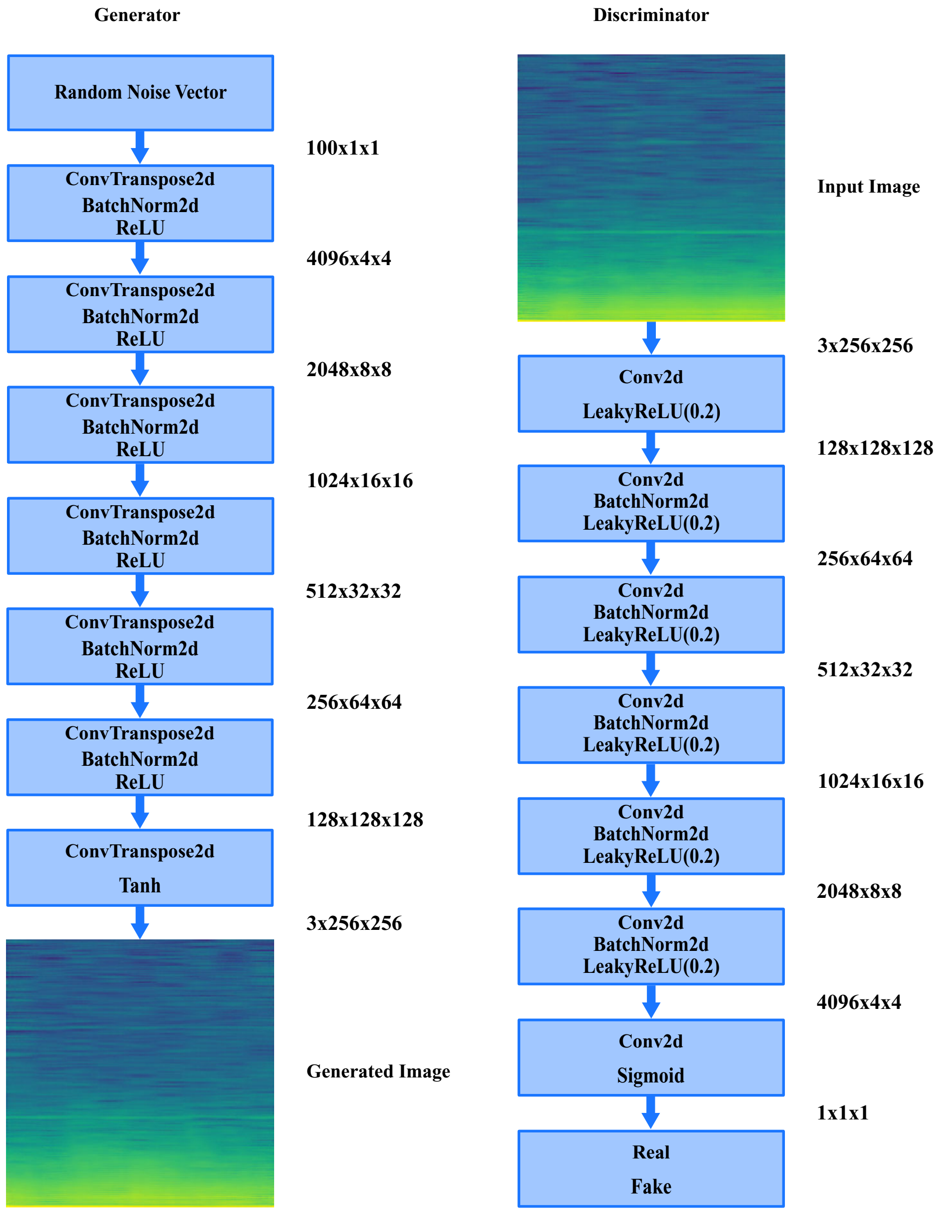

Figure 4. DCGAN architecture overview with tensor dimension flow.

2000 epochs trained on "normal" data and WGAN-GP for 3000 epochs trained on "abnormal" data.

The quality of the data generated by each model was measured using the FID calculated through the "Pytorch-fid" package.[58] Typically, this metric is computed utilizing the "pool_3" layer from the Inception V3 model. However, to employ this layer, the sample size per class, including both real and generated data, should be greater than 2048, corresponding to the dimensionality of the "pool_3" layer. Using a reduced number of samples results in complex numbers and NANs in the covariance calculation.[59] Because the number of training samples was 1980, and following the methodology in refs. [60,61] the "pre-aux" layer, with 768-dimensional features, was used to compute the FID. The results obtained per model are presented in the "FID" column of Table 9. Values close to 0 indicate similarity between the feature representation of the generated samples and the feature representation of the data used during the training of the models, i.e., higher data quality and better model performance. Based on the values presented in Table 9, it is observed that between the models trained on "normal" data, the WGAN-GP model for 2000 epochs has the lowest FID value, i.e., the best performance in quality of the data generated. In turn, between the models trained on "abnormal" data, the WGAN-GP model for 3000 epochs has the best performance. On the other hand, it is important to emphasize that using a layer from the Inception V3 model different from "pool_3" restricts the comparisons between the FID values obtained and

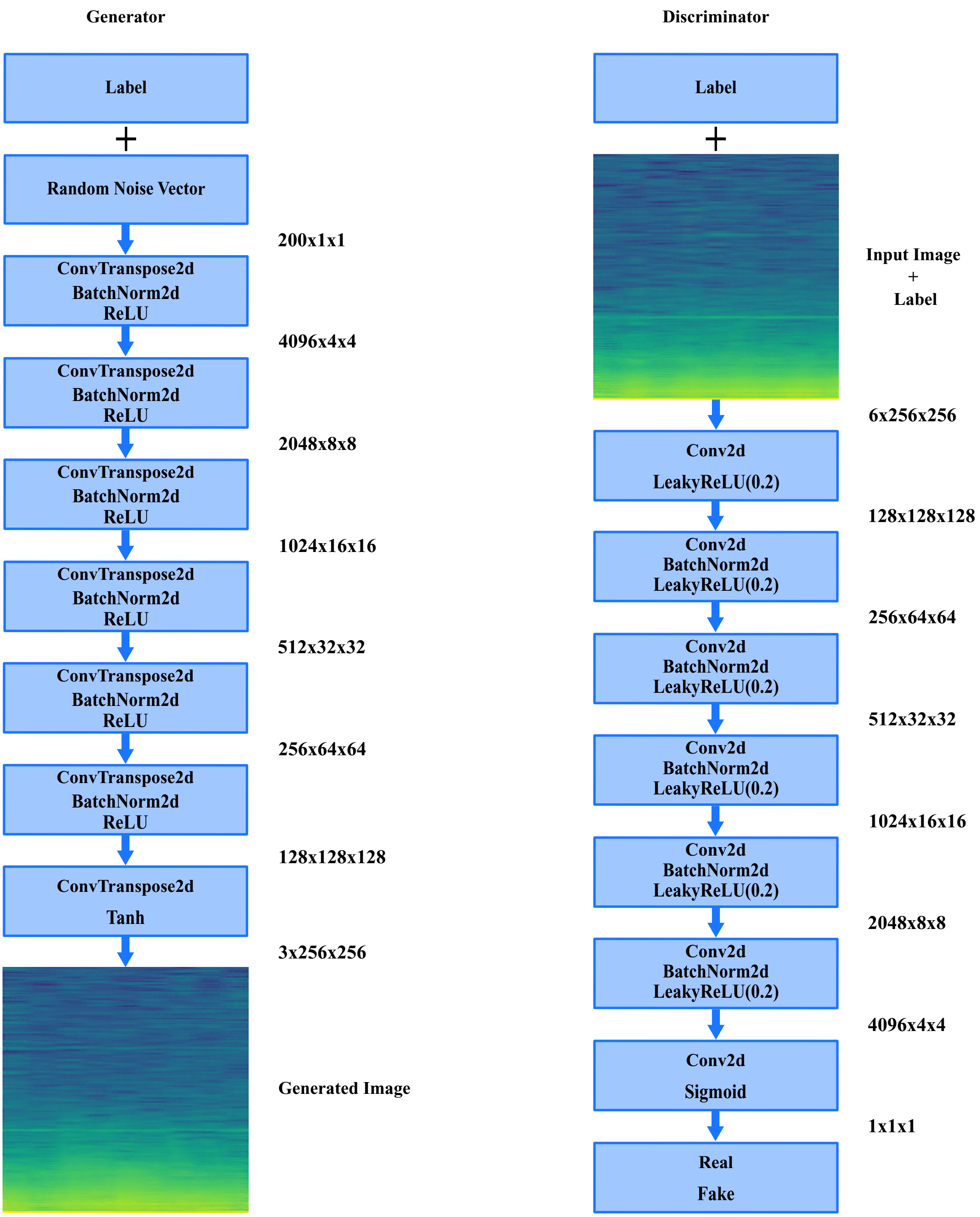

Figure 5. CGAN architecture overview with tensor dimension flow.

the FID values from other works.[58] A correct comparison would require the other works to employ the same "pre-aux" layer.

The computational cost of training each model was measured in terms of runtime and GPU memory usage, employing the Python "time" module and the "nvidia-smi" command, respectively. The results for each model are presented in **Table 10**. These results indicate that the training of WGAN-GP models has the highest GPU memory usage. In turn, the models with the highest runtime are the CGAN. Nevertheless, it is relevant to consider that these models employ both "normal" and "abnormal" data during the training. On the other hand, in Table 10 it can be observed that, for the DCGAN and WGAN-GP models,

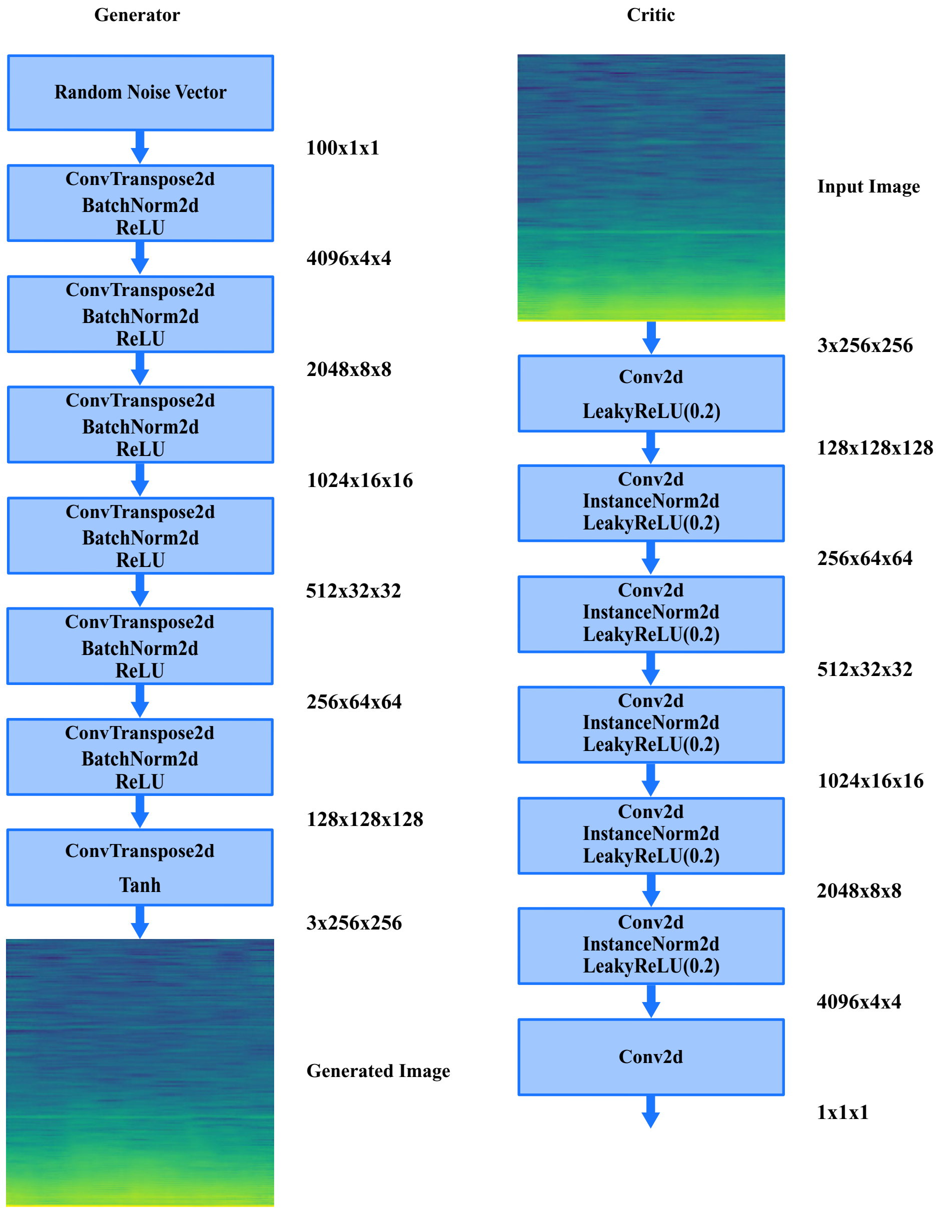

Figure 6. WGAN-GP architecture overview with tensor dimension flow.

there are differences in the runtime according to the class of data used during the training.

4.3. Impact on Classification Performance

The effect of the generated samples on the classification performance of a 2D CNN (2DCNN) model was evaluated by comparing the results obtained when training the model on various datasets using the same hyperparameters (batch size of 256 and 25 epochs). Initially, the model was trained on two imbalanced datasets, one per class, and tested on a balanced test set. Subsequently, two GAN-based models, one per class, were selected based on previous quality assessments to balance the imbalanced datasets. Specifically, the WGAN-GP model trained for 2000 epochs and the WGAN-GP model trained for 3000 epochs were chosen to balance the "normal" and "abnormal" classes, respectively. Each model generated the necessary samples to create ten balanced training datasets from the two original imbalanced datasets. Finally, the 2DCNN model was trained on the GAN-balanced datasets and evaluated on the same balanced test set. The architecture of the 2DCNN model, along with a description of the training and testing datasets, is provided in **Table 11** and **12**.

The metrics precision (Pr), recall (Re), F1-score (F1), and accuracy (Acc) were used to evaluate the classification performance of the 2DCNN model. The "metrics" module from the Sklearn

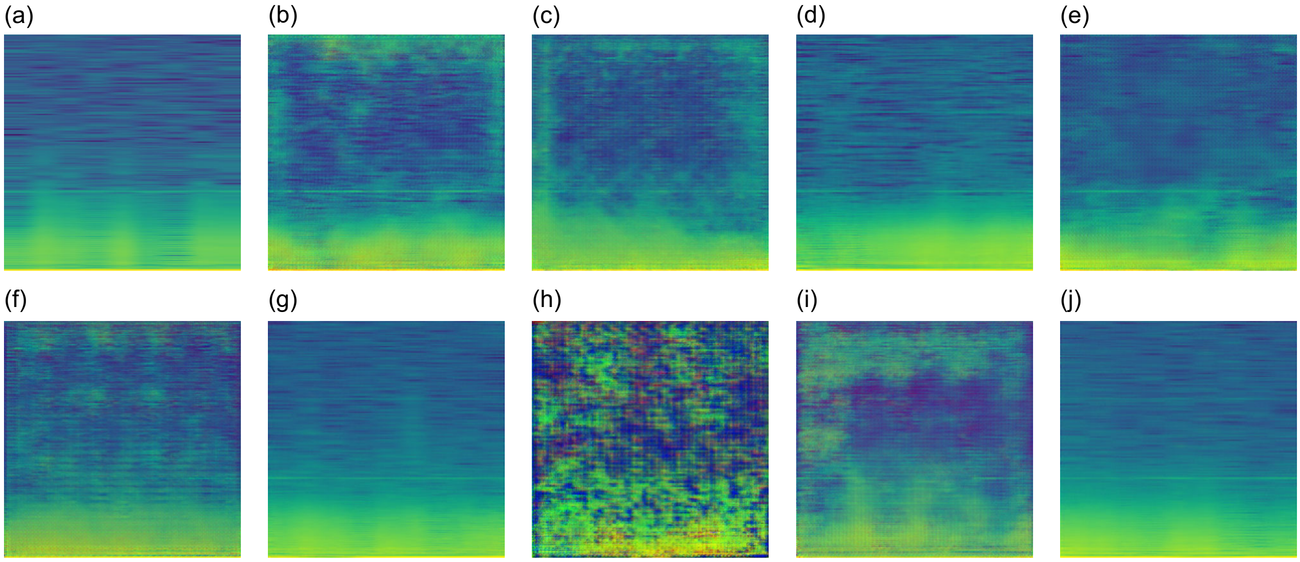

Figure 7. Comparison of original and GAN-generated spectrograms for the "normal" class. a) Spectrogram sample extracted from the training dataset. b) Spectrogram generated by the CGAN model trained for 1000 epochs. c) Spectrogram generated by the DCGAN model trained for 1000 epochs. d) Spectrogram generated by the WGAN-GP model trained for 1000 epochs. e) Spectrogram generated by the CGAN model trained for 2000 epochs. f) Spectrogram generated by the DCGAN model trained for 2000 epochs. g) Spectrogram generated by the WGAN-GP model trained for 2000 epochs. h) Spectrogram generated by the CGAN model trained for 3000 epochs. i) Spectrogram generated by the DCGAN model trained for 3000 epochs. j) Spectrogram generated by the WGAN-GP model trained for 3000 epochs.

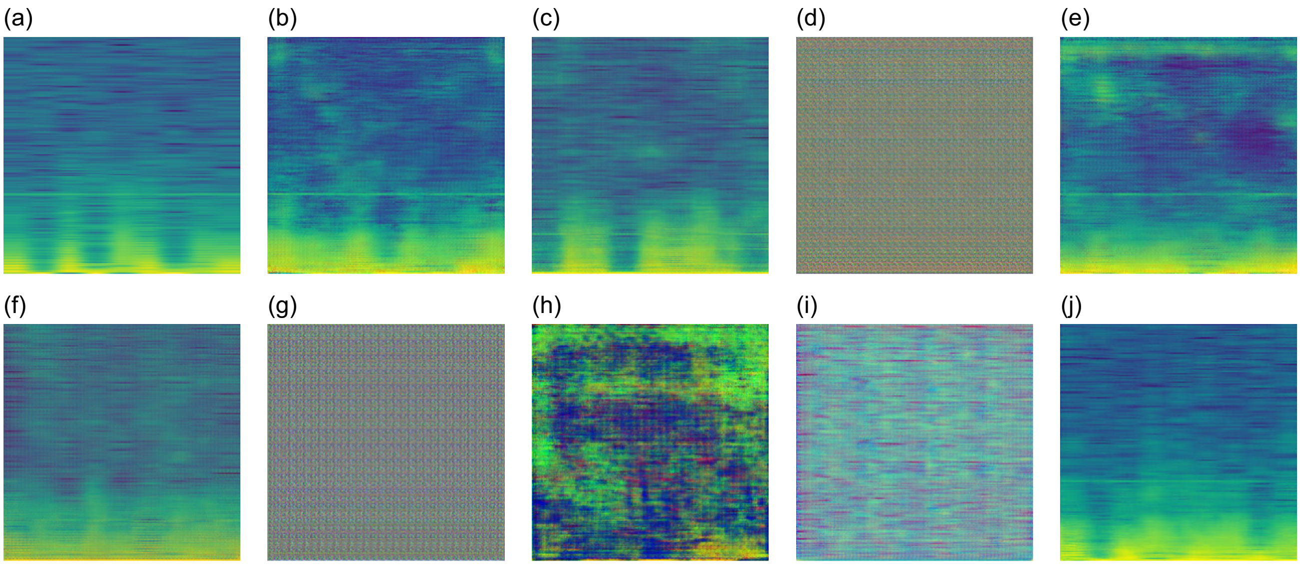

Figure 8. Comparison of original and GAN-generated spectrograms for the "abnormal" class. a) Spectrogram sample extracted from the training dataset. b) Spectrogram generated by the CGAN model trained for 1000 epochs. c) Spectrogram generated by the DCGAN model trained for 1000 epochs. d) Spectrogram generated by the WGAN-GP model trained for 1000 epochs. e) Spectrogram generated by the CGAN model trained for 2000 epochs. f) Spectrogram generated by the DCGAN model trained for 2000 epochs. g) Spectrogram generated by the WGAN-GP model trained for 2000 epochs. h) Spectrogram generated by the CGAN model trained for 3000 epochs. i) Spectrogram generated by the DCGAN model trained for 3000 epochs. j) Spectrogram generated by the WGAN-GP model trained for 3000 epochs.

Python library[62] was employed to compute these metrics. Pr, Re, and F1 were calculated for each class, while Acc was calculated across both classes. **Table 13** presents the results obtained by the models trained on the imbalanced datasets, as well as the arithmetic mean and standard deviation of the results from the models trained on GAN-balanced datasets. These results indicate that the models trained on datasets where the "normal" class was balanced using GANs demonstrate superior

Table 9. Results of performance tests.

Type of network	Epochs	Generated sample	SSIM			FID
			Maximum	Minimum	Average	
CGAN	1000	Abnormal	0.9950	0.4132	0.7328	1.0275
		Normal	0.9958	0.4125	0.7957	1.0648
	2000	Abnormal	0.9943	0.3587	0.7420	0.9900
		Normal	0.9977	0.3992	0.8854	0.9694
	3000	Abnormal	0.9966	0.2886	0.6533	2.4594
		Normal	0.9977	0.2972	0.7817	2.4348
DCGAN	1000	Abnormal	0.9876	0.5691	0.7606	0.4785
		Normal	0.9918	0.6939	0.8321	1.2210
	2000	Abnormal	0.9959	0.7507	0.8632	0.6992
		Normal	0.9902	0.5260	0.7420	1.4592
	3000	Abnormal	0.9974	0.8765	0.9526	1.6102
		Normal	0.9926	0.5307	0.7246	1.5553
WGAN-GP	1000	Abnormal	0.9998	0.9992	0.9996	3.2806
		Normal	0.8949	0.4993	**0.6182**	0.2733
	2000	Abnormal	0.9996	0.9976	0.9991	2.9621
		Normal	**0.8439**	0.6537	0.7260	**0.1745**
	3000	Abnormal	**0.7533**	0.4938	**0.5919**	**0.2907**
		Normal	0.8538	0.7012	0.7621	0.1831

Table 10. Computational cost.

Type of network	Epochs	Class	GPU	Memory usage [GB]	Runtime
CGAN	1000	Abnormal/Normal	NVIDIA L4	16.769	21:00:18.89
	2000	Abnormal/Normal	NVIDIA L4	16.769	41:59:23.68
	3000	Abnormal/Normal	NVIDIA L4	16.769	60:55:52.77
DCGAN	1000	Abnormal	NVIDIA L4	16.622	09:58:13.49
		Normal	NVIDIA L4	16.622	10:03:18.52
	2000	Abnormal	NVIDIA L4	16.622	19:55:25.78
		Normal	NVIDIA L4	16.622	20:11:40.06
	3000	Abnormal	NVIDIA L4	16.622	31:11:40.63
		Normal	NVIDIA L4	16.622	30:21:21.50
WGAN-GP	1000	Abnormal	NVIDIA L4	23.182	15:08:21.10
		Normal	NVIDIA L4	23.182	15:20:08.96
	2000	Abnormal	NVIDIA L4	23.182	30:14:07.93
		Normal	NVIDIA L4	23.182	30:39:35.43
	3000	Abnormal	NVIDIA L4	23.182	45:00:11.62
		Normal	NVIDIA L4	23.182	46:25:24.12

classification performance compared to the model trained on the imbalanced "normal" class. Likewise, the model trained on the imbalanced "abnormal" class achieved classification performance above 85% across all metrics. This may be due to the extracted features from both classes being sufficiently

Table 11. 2DCNN model architecture.

Layer	Classifier
1	Conv2d(3,8) ReLU MaxPool2d(2,2)
2	Conv2d(8,16) ReLU MaxPool2d(2,2)
3	Conv2d(16,32) ReLU MaxPool2d(2,2)
4	Flatten
5	Dropout(0.2)
6	Linear(28 800 128) ReLU
7	Linear(128,1) Sigmoid
Number of trainable parameters	3 689 257

Table 12. Summary of the datasets used for training and testing the 2DCNN model.

Dataset	Number of abnormal samples	Number of normal samples
Training dataset with the "abnormal" class imbalanced	324	1188
Training dataset with the "normal" class imbalanced	1188	324
Training dataset 1 with the "abnormal" class balanced using GAN	1188	1188
Training dataset 2 with the "abnormal" class balanced using GAN	1188	1188
Training dataset 3 with the "abnormal" class balanced using GAN	1188	1188
Training dataset 4 with the "abnormal" class balanced using GAN	1188	1188
Training dataset 5 with the "abnormal" class balanced using GAN	1188	1188
Training dataset 1 with the "normal" class balanced using GAN	1188	1188
Training dataset 2 with the "normal" class balanced using GAN	1188	1188
Training dataset 3 with the "normal" class balanced using GAN	1188	1188
Training dataset 4 with the "normal" class balanced using GAN	1188	1188
Training dataset 5 with the "normal" class balanced using GAN	1188	1188
Testing dataset with both classes equally balanced	396	396

representative to enable classification despite the imbalance. However, more tests are necessary to determine the reasons for these results. Additionally, it is important to note that the "abnormal" class comprises samples corresponding to left bundle branch block beats (L), right bundle branch block beats (R), premature ventricular contraction beats (V), and atrial premature beats (A). On the other hand, the models trained on datasets where the "abnormal" class was balanced using GANs show only a modest improvement in classification performance compared to the model trained on the imbalanced "abnormal" class.

Table 13. Comparison of classification performance between models trained on imbalanced and GAN-balanced datasets.

Model trained on a dataset with	Metrics						
	Abnormal class			Normal class			Acc
	Pr	Re	F1	Pr	Re	F1	
The imbalanced abnormal class	98.8%	85.9%	91.9%	87.5%	99.0%	92.9%	92.4%
The imbalanced normal class.	59.0%	98.0%	73.6%	94.0%	31.8%	47.5%	64.9%
Abnormal class GAN-balanced	99.5% (±0.4%)	86.6% (±1.5%)	92.6% (±0.7%)	88.1% (±1.1%)	99.6% (±0.4%)	93.5% (±0.5%)	93.1% (±0.6%)
Normal class GAN-balanced.	69.8% (±1.4%)	98.4% (±0.4%)	81.6% (±0.8%)	97.3% (±0.6%)	57.3% (±2.9%)	72.1% (±2.2%)	77.9% (±1.3%)

5. Conclusion

The imbalance in available medical datasets is one of the primary challenges researchers face when developing ML models for detecting abnormalities in ECG signals. GAN-based methods offer a promising approach to address this issue. This article presented a performance comparison of three GAN-based models—DCGAN, CGAN, and WGAN-GP—for generating synthetic spectrograms corresponding to "normal" and "abnormal" ECG signals. The intraclass diversity and quality of generated data were measured using the SSIM and the FID score, respectively.

The results demonstrate that WGAN-GP consistently achieves superior performance across both metrics. Specifically, WGAN-GP models trained for 1000 epochs ("normal" data) and 3000 epochs ("abnormal" data) yielded the highest intraclass diversity. Similarly, optimal data quality was achieved by WGAN-GP models trained for 2000 and 3000 epochs on "normal" and "abnormal" data, respectively. However, training these models requires significant computational resources.

Furthermore, data augmentation experiments demonstrate that classification models trained on GAN-balanced datasets improved performance, compared to those trained on imbalanced data, though gains varied by class.

The motivation driving this research—addressing the critical gap in spectrogram-based medical data augmentation—has been successfully validated through our comprehensive experimental framework. Our work demonstrates that the transition from time-domain to time–frequency synthetic data generation is not only a methodological advancement but also a paradigm shift toward more effective cardiac diagnostic systems. The clinical implications are substantial: improved classification performance for minority cardiac abnormalities directly translates to better patient outcomes, particularly in resource-constrained healthcare environments where dataset imbalance severely limits diagnostic accuracy.

This research establishes the foundation for a new direction in medical data augmentation, proving that GAN-based spectrogram generation can effectively bridge the gap between data scarcity and clinical need. The superior performance of WGAN-GP models in generating high-quality synthetic spectrograms provides a concrete pathway for researchers and clinicians to develop more robust, balanced datasets for CVD detection.

Future work will explore several directions: 1) validation using additional databases such as PTB-XL, St. Petersburg INCART, or AFDB, 2) development of GAN-based models for generating spectrograms of specific abnormalities, and 3) investigation of methods to transform time–frequency spectrograms back into time-domain ECG signals.

Acknowledgements

This work was supported by the Bicentenary Grant for doctoral studies from the Ministerio de Ciencia, Tecnología e Innovación de la República de Colombia, BPIN2019000100005.

Conflict of Interest

The authors declare no conflict of interest.

Data Availability Statement

The data that support the findings of this study are openly available in [GAN-Based-ECG-Spectrogram-Generation] at [https://github.com/IngBarbosa69/Synthetic-ECG-Spectrogram-Generation-Using-GAN-Based-Models-AComparative-Study].

Keywords

data augmentation, electrocardiogram, generative adversarial networks, spectrograms

Received: June 27, 2025
Revised: September 9, 2025
Published online: November 3, 2025

[1] World Health Organization, Cardiovascular diseases (CVDs), https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds) (accessed: May 2024).
[2] National Health Service, Cardiovascular disease, https://www.nhs.uk/conditions/cardiovascular-disease/ (accessed: May 2023).
[3] American Heart Association, What is Cardiovascular Disease? https://www.heart.org/en/health-topics/consumer-healthcare/what-is-cardiovascular-disease (accessed: May 2023).
[4] World Health Organization, The top 10 causes of death, https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death (accessed: May 2024).
[5] M. M. Rahman, M. W. Rivolta, F. Badilini, R. Sassi, *Sensors* **2023**, *23*, 1.
[6] M. M. Farag, *IEEE Access* **2022**, *10*, 94469.
[7] G. Bognár, S. Fridli, *Biomed. Signal Process. Control* **2020**, *61*, 102034.

[8] C. K. Jha, M. H. Kolekar, *Biomed. Signal Process. Control* **2020**, *59*, 101875.
[9] E. Adib, F. Afghah, J. J. Prevost, *PLOS ONE* **2025**, *20*, e0271270.
[10] M. Zubair, S. Woo, S. Lim, D. Kim, *IEEE J. Biomed. Health Inform.* **2024**, *28*, 2461.
[11] D. Nankani, R. Dutta Baruah, *Intelligent Systems Reference Library*, 1st ed. (Eds: R. Razavi-Far, A. Ruiz-Garcia, V. Palade, J. Schmidhuber) Springer, Cham **2022**, *217*, pp. 275–304 ch. 12.
[12] N. Neifar, A. Ben-Hamadou, A. Mdhaffar, M. Jmaiel, B. Freisleben, *IEEE Access* **2024**, *12*, 36002.
[13] H. M. Rai, J. Yoo, S. Dashkevych, *Mathematics* **2024**, *12*, 2693.
[14] M. A. Khan, *IEEE Access* **2020**, *8*, 34717.
[15] A. Faraone, R. Delgado-Gonzalo, *2020 2nd IEEE Inter. Conf. on Artificial Intelligence Circuits and Systems (AICAS)*. IEEE, Conference, Genoa, Italy **2020**, pp. 153–157. https://ieeexplore.ieee.org/document/9073950/
[16] A. K. Sangaiah, M. Arumugam, G.-B. Bian, *Artif. Intell. Med.* **2020**, *103*, 101788.
[17] M. S. Islam, M. N. Islam, N. Hashim, M. Rashid, B. S. Bari, F. A. Farid, *IEEE Access* **2022**, *10*, 58081.
[18] H. Bechinia, D. Benmerzoug, N. Khlifa, *IEEE Access* **2024**, *12*, 40827.
[19] T. I. Toma, S. Choi, *Inter. Conf. on Ubiquitous and Future Networks, ICUFN*,IEEE, Conference, Barcelona, Spain **2022**, pp. 483–488.
[20] M. F. Safdar, R. M. Nowak, P. Pałka, *Sensors* **2022**, *22*, 9576.
[21] A. M. Alqudah, S. Qazan, L. Al-Ebbini, H. Alquran, I. A. Qasmieh, *J. Ambient Intell. Humaniz. Comput.* **2022**, *13*, 4877.
[22] A. Eleyan, F. Bayram, G. Eleyan, *Appl. Sci.* **2024**, *14*, 9936.
[23] B. Król-Józaga, *Biomed. Signal Process. Control* **2022**, *74*, 103470.
[24] D. Siekierski, K. Siwek, *Comput. Methods Programs Biomed Update* **2022**, *2*, 100075.
[25] T.-R. Wei, S. Lu, Y. Yan, *Bioengineering* **2022**, *9*, 523.
[26] S. Singhal, M. Kumar, *IEEE Sens. J.* **2024**, *24*, 8160.
[27] K. Gupta, V. Bajaj, I. Ahmad Ansari, *IEEE Sens. J.* **2025**, *25*, 6794.
[28] G. Moody, R. Mark, *IEEE Eng. Med. Biol. Mag.* **2001**, *20*, 45.
[29] D. Hazra, Y.-C. Byun, *Biology* **2020**, *9*, 441.
[30] N. Wulan, W. Wang, P. Sun, K. Wang, Y. Xia, H. Zhang, *Neurocomputing* **2020**, *404*, 122.
[31] M. F. Safdar, P. Pałka, R. M. Nowak, A. A. Faresi, *Biomed. Signal Process. Control* **2023**, *86*, 105114.
[32] Y. Yang, T. Lan, Y. Wang, F. Li, L. Liu, X. Huang, F. Gao, S. Jiang, Z. Zhang, X. Chen, *Sci. Rep.* **2024**, *14*, 14767.
[33] D. Wagner, I. Baumann, T. Bocklet, *Int. J. Speech Technol.* **2024**, *27*, 1093.
[34] Q. Yang, Y. Bai, F. Liu, W. Zhang, *Sci. Rep.* **2024**, *14*, 1.
[35] U. Avci, *IEEE Access* **2025**, *13*, 111647.
[36] E. Kim, J. Moon, J. Shim, E. Hwang, *Sensors* **2023**, *23*, 2024.
[37] P. Li, M. A. Roch, H. Klinck, E. Fleishman, D. Gillespie, E.-M. Nosal, Y. Shiu, X. Liu, *IEEE Trans. Multimed.* **2023**, *25*, 9302.
[38] T. Wang, D. Trugman, Y. Lin, *J. Geophys. Res.: Solid Earth* **2021**, *126*, 1.
[39] G. Zhou, Y. Chen, C. Chien, *BMC Med. Inform. Decis. Mak.* **2022**, *22*, 226.
[40] Y. Yang, Y. Zhang, Y. Lang, B. Li, S. Guo, Q. Tan, *IEEE Trans. Instrum. Meas.* **2023**, *72*, 1.
[41] Q. Xiao, K. Lee, S. A. Mokhtar, I. Ismail, A. L. b. M. Pauzi, Q. Zhang, P. Y. Lim, *Appl. Sci.* **2023**, *13*, 4964.
[42] P. Wagner, N. Strodthoff, R.-D. Bousseljot, D. Kreiseler, F. I. Lunze, W. Samek, T. Schaeffter, *Sci. Data* **2020**, *7*, 154.
[43] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, H. E. Stanley, *Circulation* **2000**, *101*, e215.
[44] G. B. Moody, R. G. Mark, *Comput. Cardiol.* **1983**, *10*, 227.
[45] J. Huang, B. Chen, B. Yao, W. He, *IEEE Access* **2019**, *7*, 92871.
[46] J. Hunter, D. Dale, E. Firing, M. Droettboom, Matplotlib, https://matplotlib.org/stable/ (accessed: March 2024).
[47] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, *Commun. ACM* **2014**, *63*, 139.
[48] A. Radford, L. Metz, S. Chintala, *4th Int. Conf. on Learning Representations, ICLR 2016 - Conf. Track Proc.* Cornell University, Ithaca, NY **2015** pp. 1–16.
[49] M. Mirza, S. Osindero, Conditional Generative Adversarial Nets, http://arxiv.org/abs/1411.1784 (accessed: November 2014).
[50] M. Arjovsky, S. Chintala, L. Bottou, Wasserstein GAN, http://arxiv.org/abs/1701.07875 (accessed: January 2017).
[51] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, A. Courville, *Adv.Neural Inf. Process. Syst.* **2017**, *30*, 5768.
[52] N. Inkawhich, DCGAN TUTORIAL, https://rb.gy/256qfl (accessed: May 2024).
[53] Z. Wang, A. Bovik, H. Sheikh, E. Simoncelli, *IEEE Trans. Image Process.* **2004**, *13*, 600.
[54] N. B. Bynagari, *Asian J. Appl. Sci. Eng.* **2019**, *8*, 25.
[55] S. Maruyama, *Phys. Eng. Sci. Med.* **2023**, *46*, 1131.
[56] Y. Zhou, B. Wang, X. He, S. Cui, L. Shao, *IEEE J. Biomed. Health Inform.* **2022**, *26*, 56.
[57] M. Harfouche, J. Millman, J. Warner, J. Nunez-Iglesias, L. Grüter, M. Corvellec, E. Panfilov, S. van der Walt, scikit-image, https://scikit-image.org/ (accessed: March 2024).
[58] M. Seitzer, pytorch-fid: FID Score for PyTorch, https://pypi.org/project/pytorch-fid/ (accessed: May 2024).
[59] M. Heusel, T. Unterthiner, W. Kan, M. Hamilton, M. Uecker, P. Joshi, Two time-scale update rule for training GANs, https://github.com/bioinf-jku/TTUR (accessed: May 2024).
[60] S. Kora Venu, S. Ravula, *Future Internet* **2020**, *13*, 8.
[61] M. M. Saad, M. H. Rehmani, R. O'Reilly, in *44th Annual Inter. Conf. of the IEEE Engineering in Medicine Biology Society (EMBC)*, IEEE, Conference, Glasgow, Scotland, United Kingdom **2022**, pp. 2049–2052.
[62] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, É. Duchesnay, *J. Mach. Learn. Res.* **2011**, *12*, 2825.
[63] A. Géron, *Hands-On Machine Learning With Scikit-Learn, Keras, And TensorFlow*, 2nd ed., (Ed.: N. Tache), O'Reilly, Sebastopol **2019**.
[64] V. Suterio, P. R. Scalassara, C. M. Agulhari, F. R. Durand, *XXVI Brazilian Congress on Biomedical Engineering, ser. IFMBE Proc.*, (Eds.:R. Costa-Felix, J. C. Machado, A. V. Alvarenga), Springer Singapore, Singapore **2019**, vol. *70/2*. http://link.springer.com/10.1007/978-981-13-2517-5.
[65] S.-W. Cheng, H. Huang, Solving Fréchet Distance Problems by Algebraic Geometric Methods, no. 16208923, http://arxiv.org/abs/2308.14569 (accessed: August 2023).
[66] M. Müller, *Information Retrieval For Music And Motion*. Springer, Berlin, Heidelberg **2007**.
[67] X. Jia, M. Zhao, Y. Di, Q. Yang, J. Lee, *IEEE Trans. Ind. Electron.* **2018**, *65*, 5872.
[68] H. Li, *Inf. Sci.* **2021**, *547*, 592.