mathematics

Article

Detection of Abnormal Cardiac Response Patterns in Cardiac Tissue Using Deep Learning

Xavier Marimon [1,2,3,*], Sara Traserra [4], Marcel Jiménez [4], Andrés Ospina [5] and Raúl Benítez [1,3]

[1] Automatic Control Department, Universitat Politècnica de Catalunya (UPC-BarcelonaTECH), 08034 Barcelona, Spain
[2] Bioengineering Institute of Technology, Universitat Internacional de Catalunya (UIC), 08195 Barcelona, Spain
[3] Institut de Recerca Sant Joan de Déu (IRSJD), 08950 Barcelona, Spain
[4] Faculty of Veterinary Medicine, Universitat Autònoma de Barcelona, 08193 Barcelona, Spain
[5] Computer Science Department, Universitat Oberta de Catalunya (UOC), 08018 Barcelona, Spain
* Correspondence: xavier.marimon@upc.edu or xmarimon@uic.es

Abstract: This study reports a method for the detection of mechanical signaling anomalies in cardiac tissue through the use of deep learning and the design of two anomaly detectors. In contrast to anomaly classifiers, anomaly detectors allow accurate identification of the time position of the anomaly. The first detector used a recurrent neural network (RNN) of long short-term memory (LSTM) type, while the second used an autoencoder. Mechanical contraction data present several challanges, including high presence of noise due to the biological variability in the contraction response, noise introduced by the data acquisition chain and a wide variety of anomalies. Therefore, we present a robust deep-learning-based anomaly detection framework that addresses these main issues, which are difficult to address with standard unsupervised learning techniques. For the time series recording, an experimental model was designed in which signals of cardiac mechanical contraction (right and left atria) of a CD-1 mouse could be acquired in an automatic organ bath, reproducing the physiological conditions. In order to train the anomaly detection models and validate their performance, a database of synthetic signals was designed (n = 800 signals), including a wide range of anomalous events observed in the experimental recordings. The detector based on the LSTM neural network was the most accurate. The performance of this detector was assessed by means of experimental mechanical recordings of cardiac tissue of the right and left atria.

Keywords: deep learning; autoencoder; cardiac tissue; electrophysiology; electrostimulation; anomaly detection; recurrent neural network; long short-term memory; CD-1 mouse model

MSC: 92; 68T07; 92C55

Citation: Marimon, X.; Traserra, S.; Jiménez, M.; Ospina, A.; Benítez, R. Detection of Abnormal Cardiac Response Patterns in Cardiac Tissue Using Deep Learning. *Mathematics* **2022**, *10*, 2786. https://doi.org/10.3390/math10152786

Academic Editor: Zhaobin Wang

Received: 6 July 2022
Accepted: 3 August 2022
Published: 5 August 2022

1. Introduction

Anomaly detection is a broad field of research focused on the detection of abnormal patterns within a given dataset. Anomaly detectors have been traditionally used in the industrial world for predictive maintenance to identify faults from sensor signals placed on motors and machinery [1]. Anomaly detection is also important in other fields, such as healthcare, structural health monitoring, security, surveillance and fraud detection, where anomalies convey actionable and critical information. Apart from the use of temporal signals, video sequences are also commonly used—for instance, in security surveillance based on the detection of unusual movements in video surveillance footage [2]. Time series anomalies can be classified into two categories: point anomalies and structural anomalies [3]. Point anomalies have historically been the most widely studied and correspond to sample values of the signal statistically different from other signal values [4]. In contrast, structural anomalies are defined as patterns presenting a morphology that differs from the behavior observed in the signal [5].

Currently, machine learning plays an important role in the study of structural anomalies of time series [6]. When correctly labeled signals are available in normal and abnormal classes, the features in the time domain (statistical or morphological features) and in the frequency domain (spectral features) can be examined in order to better identify the features that distinguish between regular and abnormal conditions. These attributes can then be used to develop an anomaly classification algorithm based on labeled data through supervised learning. When labeled data are not available, the standard approach is to use unsupervised learning algorithms such as clustering to identify different behavior in the recordings. The resulting clusters can then be identified as normal and abnormal situations by an expert. This approximation allows for classifying correctly whether it is a normal or abnormal signal but does not give temporal information on the position in which the anomaly responsible for a deviation with respect to the normal signals has taken place.

In the contraction of the heart chambers, the sinus node, which acts as a pacemaker, is responsible for generating an electrical stimulus that activates the two atria. Then, the electrical stimulus propagates to the atrioventricular node through the Purkinje fibers to contract the left and right ventricles [7]. A cardiac anomaly emerges when the conduction pathway generated at the sinus node is interrupted [8]. Therefore, it is necessary to monitor the electrical and mechanical activity of the heart to effectively identify indicative patterns of heart disease to ensure an early diagnosis. For this reason, in the field of translational cardiology, it is relevant to study the behavior of the heart at various scales and under different domains: electrical, mechanical, fluid and chemical (molecular). Different authors have used anomaly detectors in the electrical conduction system from the study of electrocardiogram (ECG) signals [9–24]. However, little attention has been paid to assess anomaly detection in the mechanical contractile signals at the tissue level. Therefore, the present study is focused on the detection of mechanical anomalies in cardiac tissue.

Previous studies have used traditional machine learning methods to detect anomalies in electrical conduction from an ECG signal [9]. Unsupervised learning algorithms used for the detection of anomalies in the ECG signal include the dynamic time warping distance (DTW) [10,11], the Euclidean distance from an adaptive window discord discovery (AWDD) comparing the heartbeats to each other [12,13] and the k-means clustering algorithm [14]. In relation to supervised learning algorithms, the most commonly used are k-nearest neighbor (k-NN) [15], linear discriminant analysis (LDA) [16], quadratic discriminant analysis (QDA) [17], support vector machine (SVM) [18–22] and neural networks (NNs) [23,24].

The rapid development of graphics processing units (GPUs) versus central processing units (CPUs) has shown that deep learning can address artificial intelligence problems more quickly and reliably. Moreover, compared to traditional machine learning algorithms, they do not require cardiology experts to define the most relevant features to be extracted for identifying an anomaly in a given signal as deep learning networks are also able to extract features automatically. Thus, the characteristics extraction phase and the classification phases are integrated into the same deep learning network. The drawback, however, is that deep learning models require large training datasets to achieve high classification accuracy and are difficult to interpret by humans.

Regarding deep learning algorithms, the most frequently used for the detection of ECG signal anomalies is long short-term memory (LSTM) [25]. Convolutional neural networks (CNNs) [25–29] can also be used after the conversion of the signal into a spectrogram image using wavelet transform or short-time Fourier transform [30]. The good results obtained in two-dimensional convolutional neural networks (2D CNNs) in computer vision applications, such as AlexNet, VGG16 or ResNet networks, together with the emergence of the large datasets CIFAR100 and ImageNet have prompted the signal processing community to have a renewed interest in 1D CNNs [31].

In the detection of anomalies in health sciences, there is not always a sufficiently large pathological dataset to design an artificial intelligence algorithm to classify abnormalities based on the features extracted from normal and pathological data. Therefore, the present investigation focuses on studying anomalies by the construction of a model of the behavior

under normal conditions. This approach is well suited for biomedical data as it requires few or no abnormal data. Thus, this article presents a framework that uses the one-class classification approach to construct a deep learning model based on normal data—i.e., a model trained with data under normal conditions. The input of the model is a signal to be analyzed, and the output is the reconstruction of the input signal based on the trained model. The resulting model will present a small reconstruction error when the input is a normal signal and a large reconstruction error when the input is a signal under abnormal conditions. The reconstruction error, defined as the difference between the input and its reconstruction, is indicative of the presence of an anomaly. The structure of this study is described next. First, Section 2 describes the biological preparation of atrial tissue samples (Section 2.1), the experimental setup for recording the mechanical contraction of cardiac tissue (Section 2.2), the creation of the synthetic dataset based on anomalous contractions observed in the experimental recordings (Section 2.3) and the theory and methodology used for the detection of anomalies with deep learning (Section 2.4). Next, in Section 3, the correct operation of the two anomaly detectors (LSTM and autoencoder) is validated with synthetic data (Section 3.1), and then, different application examples with real data are shown using the detector with the highest accuracy (LSTM detector). Finally, Section 4 is reserved for the critical discussion of the results.

2. Materials and Methods

2.1. Preparation of Atrial Tissue Samples

For the present study, tissue samples obtained from the left and right atria of outbred albino CD-1 (Cluster of Differentiation 1) mice bred at the animal facility center of the Universitat Autònoma de Barcelona (UAB) were used. The protocol was approved by the Ethics Committee of the Autonomous University of Barcelona and complied with the Helsinki declaration ethical principles of the World Medical Association (WMA).

2.1.1. Mouse Preparation

CD-1 mice aged between 15 and 18 weeks, weighing between 20 and 40 g, were selected. Housing was under controlled conditions: constant temperature (22 $\pm$ 2 °C) and humidity (55 $\pm$ 10%), a 12-h light/dark cycle and ad libitum access to water and food. The animal was anesthetized by intraperitoneal injections of anesthetics, as shown in Table 1.

Table 1. Composition of anesthetic preparations.

Injection	Composition	Via
First	0.03 mL sodium heparin 50% 0.3 mL sodium chloride heparin 0.9%	Intraperitoneal
Second	1 mg/kg medetomidine 75 mg/kg ketamine	Intraperitoneal

After intraperitoneal injection, the correct effect of anesthesia was confirmed by abolishment of the palpebral and swallowing reflexes. Animals were sacrificed by atlanto-occipital dislocation.

2.1.2. Atria Isolation

Animals were placed in a supine decubitus position, and two thoracic incisions were made to expose the heart and lungs. The heart was extracted from the thoracic cavity and placed on a Sylgard-coated (Dow Corning. Midland, MI, USA) Petri dish containing carboxygenated Krebs solution (Table 2).

Using a microscope, the left and right atria were identified and carefully dissected from the remaining cardiac tissue. Once atria had been isolated and prior to embedding in the bath, they were individually sutured to the force transductor using 3-0 silk sutures.

Table 2. Composition of the carboxygenated Krebs–Ringer solution.

Solution	Composition	Bubbles	pH	Temperature
Carboxygenated Krebs	115.48 mM NaCl 4.61 mM KCl 1.16 mM $MgSO_4$ 21.90 mM $NaHCO_3$ 1.14 mM NaH_2PO_4 2.50 mM $CaCl_2$ 10.10 mM glucose	5% CO_2 95% O_2	7.4	37.5 °C

2.2. Mechanical Recordings

The mechanical activity of the CD-1 atria was assessed using an automatic organ bath system (Letica Scientific Instruments, Barcelona, Spain). This organ bath system is composed of four independent 20 mL chambers filled with carboxygenated Krebs solution (5% CO_2 and 95% O_2), as shown in Figure 1.

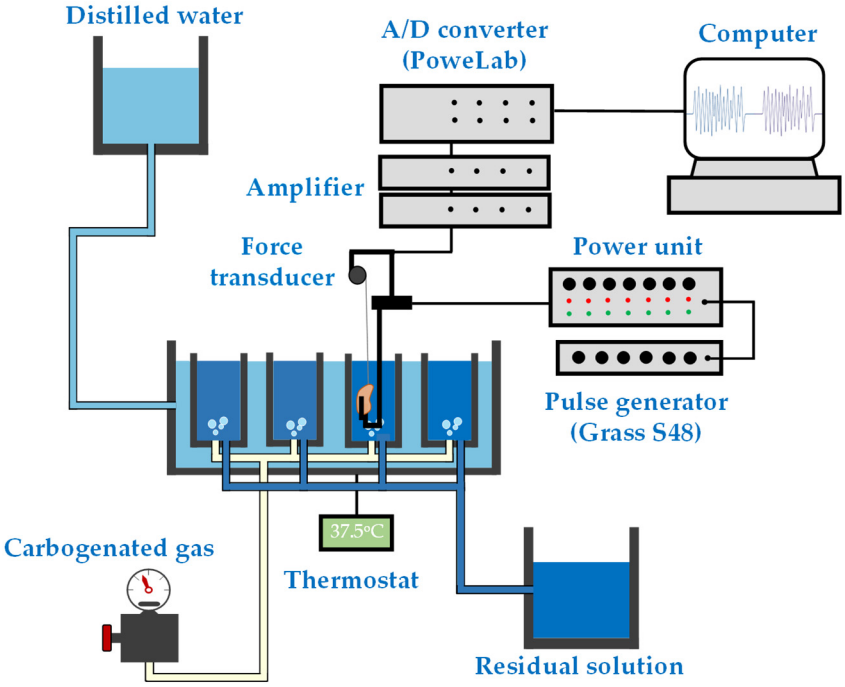

Figure 1. Experimental setup for the recording of mechanical contractions of cardiac tissue.

The temperature was regulated using a Letica 13206 thermostat (Letica Scientific Instruments, Barcelona, Spain), keeping a constant temperature of 37.5 °C. Each atrium, left or right, was placed individually inside a chamber and sutured to the isometric force transducer. The force transducer, R = 360 Ω (Letica Scientific Instruments, Barcelona, Spain), was adjusted to a mass of 0.2 g and allowed to stabilize for 15 min before starting the experiment. Data were registered using an analog-to-digital converter (ADC) (PowerLab 800, AD Instruments, Dunedin, New Zealand) with a sampling frequency of F_s = 1000 Hz.

Electrical Stimulation Protocol

Stimulation was performed using an analog stimulator (Grass S48, Astro-Med, Inc., West Warwick, RI, USA). Electrical stimulation of left atrial samples was applied using a current pulse of 40 mA for 0.4 ms at different stimulation frequencies of 1, 2, 3, 5 and 10 Hz. The frequency of electrical stimulation was related to the frequency with which the cardiac tissue contracted. Right atrial samples were not stimulated as the right atrium has a natural physiological pacemaker.

2.3. Synthetic Dataset

The experiment was based on the creation of a synthetic dataset of mechanical contractions of cardiac tissue. The synthetic dataset includes normal contraction and 7 types of different anomalies and includes a total of 100 signals for each category (i.e., n = 800 signals in total), every one with a signal length of 100,000 samples. For the creation of synthetic anomaly signals, different phenomena of anomalous contractions observed in the experimental recordings have been considered, including the presence of pulses with anomalous amplitude, duration, morphology or timing. The types of anomalies in the synthetic dataset are shown in Table 3. Normal signals were used to train the model and abnormal signals were used to validate both the anomaly classifier and the anomaly detector. In order to make the signals more realistic, noise was added to the signals following a normal distribution with zero mean and a standard deviation σ = 0.005, robustly estimated from experimental signals.

Table 3. Types of signals in the synthetic dataset. The x means not used parameter.

Signal Type	Signal Description	Category	Anomalies N_{anom}	Repetitions N_{rep}
1. Normal signal	Normal contraction	Normal	6	x
2. Lower amplitude	Phenomena of low amplitude contraction	Anomalous	6	x
3. Higher amplitude	Phenomena of large amplitude contractions	Anomalous	6	x
4. Missing pulse	Absence of contraction pulses	Anomalous	1	x
5. Slow pulse decay	Periods of contraction with dynamics of slow decline	Anomalous	1	x
6. Fast pulse decay	Periods of contraction with dynamics of fast decline	Anomalous	1	x
7. Early or anticipated pulse	Periods of anticipated contraction	Anomalous	1	x
8. Pause/block	Periods of absence of contraction pulses	Anomalous	1	10

Different synthetic data generated at a stimulation frequency of 6 Hz, i.e., with the application of a stimulation pulse at approximately every 167 samples, are shown in Figure 2.

2.4. Architecture of Anomaly Detectors

In the framework of mechanical signaling of cardiac tissue contraction, we are interested in being able to localize the exact temporal position with a probability of having an anomaly. For this purpose, two detectors based on deep learning were implemented (Figure 3). As the first detector, a recurrent neural network (RNN) structure of LSTM type was used, whereas an autoencoder neural network was used as the second detector. The advantage of such anomaly detectors is that they can be trained with one-class data obtained from normal behavior without pathological conditions—that is, anomaly data are not necessary.

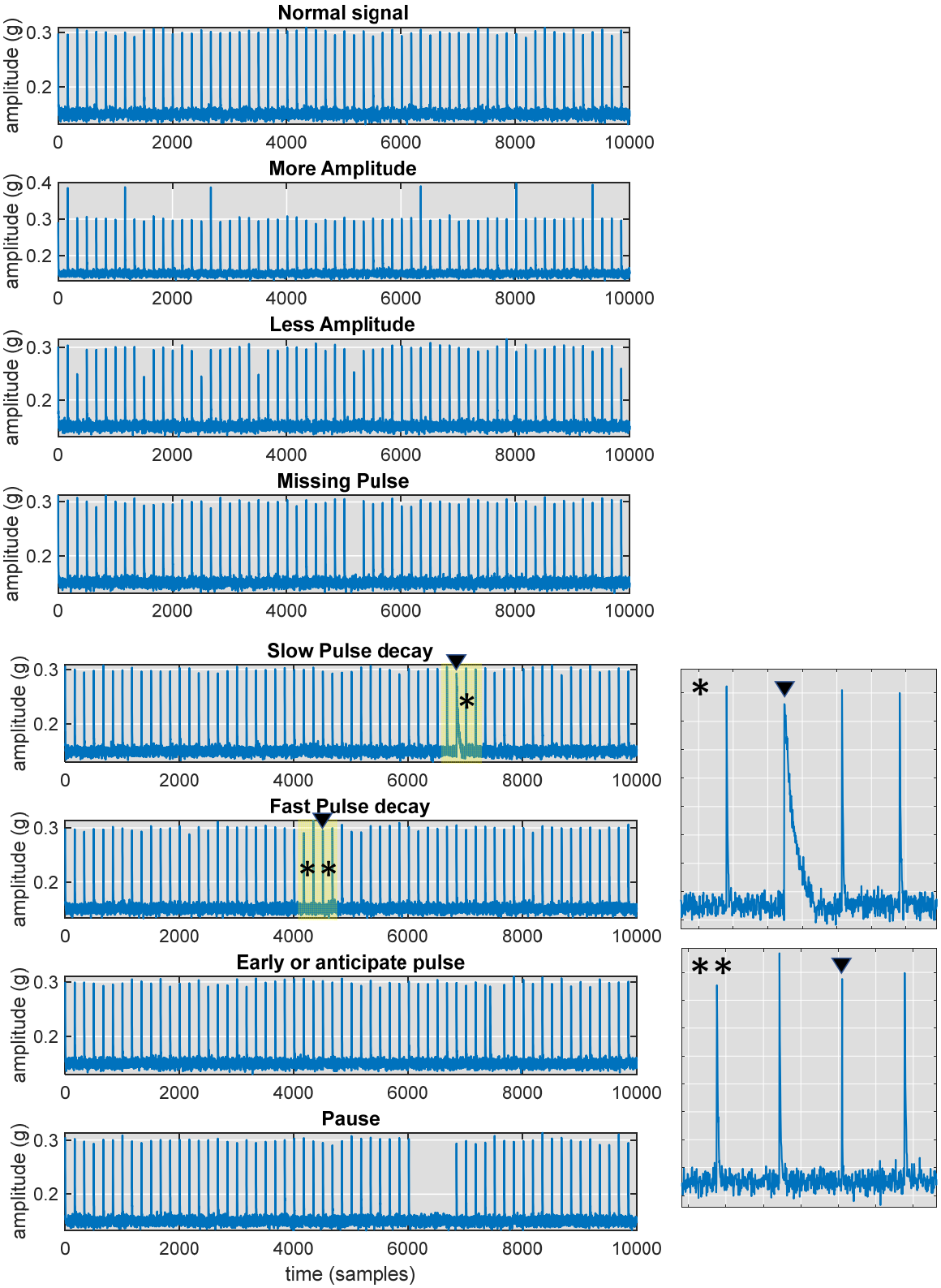

Figure 2. Synthetic signals representing each anomaly at a stimulation frequency of 6 Hz. The asterisk (*) indicates a zoom into figure. The triangle indicates the anomaly location.

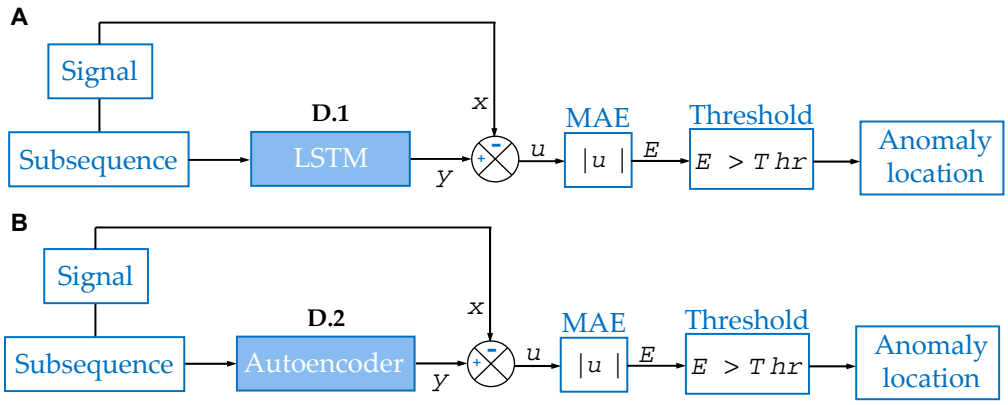

Figure 3. Anomaly detectors based on deep learning. (**A**) Anomaly detector based on an LSTM network. (**B**) Anomaly detector based on an autoencoder.

In the use of autoencoders and LSTM networks as anomaly detectors, when normal data are introduced, the detector can reconstruct the input, and then the error between the input signal and the output signal is small. Conversely, when data containing anomalies are entered into the network, the network cannot reconstruct the input signal and the error becomes larger. The MATLAB r2022a software (Mathworks, Boston, MA, USA) was used to program the code for detecting anomalies, with which custom code was developed and specific libraries were also used, such as Signal Processing Toolbox and Machine Learning Toolbox. A computer with an 11th-generation Intel Core i9 processor (CPU), 32 GB of RAM and an Nvidia GeForce GTX Titan X graphic card (GPU) was used in order to train the deep learning models. The detailed steps necessary to develop the anomaly detector based on an autoencoder (D.1) or an LSTM neural network (D.2) are described as follows (see Supplementary Material S1 and S2):

(a) Organization of data

First of all, mechanical contraction data corresponding to normal and anomalous categories were obtained.

(b) Normalization of data

Signals were then normalized; centered and standardized with the z-score transformation, defined as follows:

$$x_c = \text{z-score}(x) = \frac{x - \mu(x)}{\sigma(x)} \tag{1}$$

where x_c represents centered and standardized data, $\mu(x)$ is the mean of signal x and $\sigma(x)$ is the standard deviation of signal x.

(c) Construction of the subsequences matrix

Signals were arranged as a subsequences matrix, X, an LxK matrix obtained by arranging K signal windows of size L samples, as in Figure 4. The number of windows, K, was defined as follows:

$$K = S - L \tag{2}$$

where K is the number of windows, S is the stride between windows and L is the window size.

Subsequences were organized in an LxK subsequences matrix, X, as follows:

$$X = \begin{bmatrix} x_c(1) & x_c(1+S) & x_c(1+2\cdot S) & \cdots & x_c(1+K\cdot S) \\ x(2) & x_c(2+S) & x_c(2+2\cdot S) & \cdots & x_c(2+K\cdot S) \\ x_c(3) & x_c(3+S) & x_c(3+2\cdot S) & \ldots & x_c(3+K\cdot S) \\ \vdots & \vdots & \vdots & & \vdots \\ x_c(L) & x_c(L+S) & x_c(L+2\cdot S) & \ldots & x_c(L+K\cdot S) \end{bmatrix} \tag{3}$$

where x_c is the normalized signal with N samples; $1 \times N$.

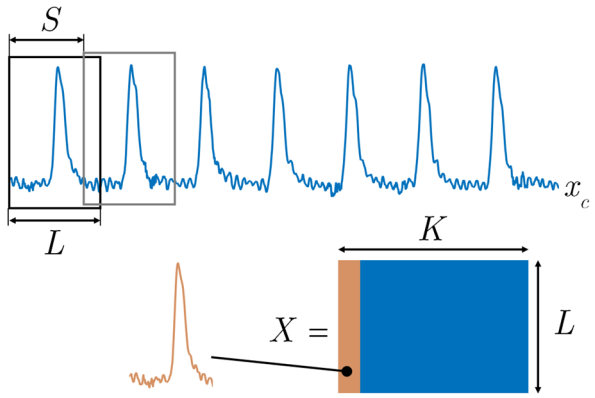

Figure 4. Creation of the subsequence matrix, X, from the contraction time series, x.

(d) Training with normal data

Then, the autoencoder or LSTM network was trained with the sequences of the training data of the normal signal category.

(e) Reconstruction with test data

Then, validation or test data with the corresponding subsequences were taken, and the model trained with normal signals was used to reconstruct the signal.

(f) Reconstruction error

The reconstruction error between the input signal, x, and the reconstructed signal, y (LSTM output), was calculated:

$$E_i = |y_i - x_i| \tag{4}$$

Then, the mean absolute error (MAE) was defined as follows:

$$MAE = \frac{1}{N} \cdot \sum_{i=1}^{N} E_i \tag{5}$$

where x_i is the input signal and y_i is the reconstructed signal produced by the LSTM network or the autoencoder.

(g) Fitting the reconstruction error to a Gaussian distribution

The reconstruction error was adjusted to a univariate normal distribution by means of the maximum likelihood estimation (MLE):

$$E_i \sim N(\hat{\mu}, \hat{\sigma}) \tag{6}$$

where $\hat{\mu}$ is the estimated mean and $\hat{\sigma}$ is the estimated standard deviation.

According to [32], for simplicity, it is recommendable to fit a Gaussian distribution, which is used to assess the likelihood of anomalous behavior. The maximum likelihood estimation (MLE) was calculated as follows [33].

(h) Measure of error distance

When error is found at the tail of a normal distribution, it is probable that an anomaly has occurred, so the rarity of the event can be measured according to how far the location of the error E_i is from the normal distribution, $N(\hat{\mu}, \hat{\sigma})$. For distance calculation, as proposed by different authors [5,32], measurement of the Mahalanobis distance was used since it is a measure of the distance between an error point, E_i, and a statistical distribution. Therefore, in some way, the Mahalanobis distance is being used as an anomaly score. In this study, the univariate Mahalanobis distance, dM, was calculated as follows:

$$dM = \text{dist}(E, N(\hat{\mu}, \hat{\sigma})) = (E - \hat{\mu})^2 \cdot \frac{1}{\hat{\sigma}^2} \tag{7}$$

(i) Envelope extraction

The signal envelope, ζ_n, was calculated by interpolating the Mahalanobis distance, *dM*, with a spline function, where the local maxima separated by at least *np* = 30 samples were selected.

(j) Calculation of the threshold of anomalous samples

Using the Kolmogorov–Smirnov one-sample normality test, it was confirmed that the validation error signals did not follow a normal distribution. Therefore, the threshold, *Thr*, for the detection of anomalous samples was defined as the upper quartile, *Q*3, of the envelope of the validation error plus five-fold the interquartile range (*IQR*):

$$Thr = Q3(\zeta_n) + 5 \cdot IQR(\zeta_n) \tag{8}$$

(k) Temporal localization of the anomalous samples

An anomalous sample is considered when the value of the envelope of the Mahalanobis distance, ζ_n, is greater than the threshold, *Thr*:

$$\zeta_n \geq Thr \tag{9}$$

The entire methodological framework described from sections (a) to (k) is represented graphically in the Supplementary Materials.

2.4.1. Detector Based on an LSTM Network (D.1)

The neural network used in the study was based on simple long short-term memory cells (vanilla LSTM), the output gate of which is described by the following recurrence functions [34]:

$$\begin{aligned} h_t &= o_t \odot \tanh(c_t) \\ o_t &= f_t \odot c_{t-1} + i_t \odot \widetilde{c}_t \end{aligned} \tag{10}$$

where h_t and h_{t-1} are the new state and the previous state of the hidden layer, respectively; c_t is the current state of the cell and $\widetilde{c}_t$ is the cell state candidate.

The symbol $\odot$ in the previous equations refers to element-wise multiplication (Hadamard product). LSTM cells use additional gates for controlling which information of the hidden cell arrives to the input and to the following hidden state. This allows more efficient network learning of the long-term relationship of data. The input gate, i_t, controls the update level of the cell state, c_t. The forget gate controls the extent of forgetfulness of the cell state. Finally, the output gate controls the level of the cell state that has been added to the hidden state, h_t (see Figure 5). Moreover, the cell state candidate adds information to the cell state. Additionally, LSTM cells are able to learn long-term dependencies.

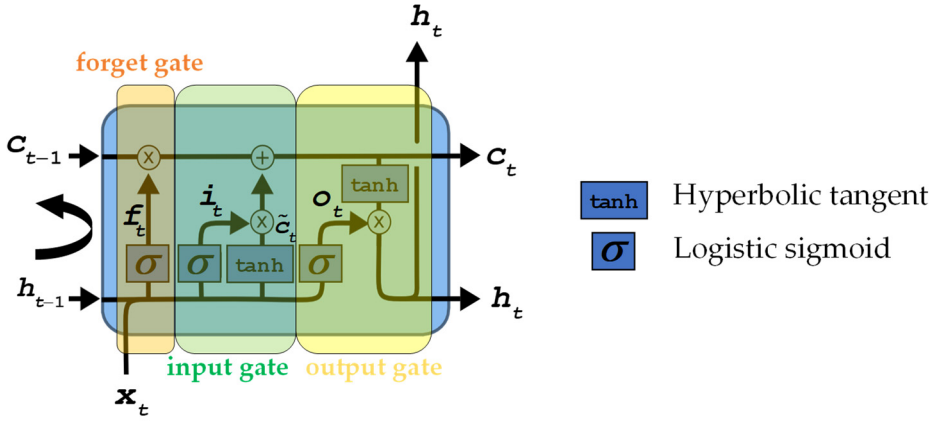

Figure 5. Control gates of a simple LSTM cell (vanilla LSTM).

Table 4 summarizes the governance equations of a cell in the LSTM network [34].

Table 4. Governance equations of an LSTM cell.

Part	Equation
Hidden state	$h_t = o_t \odot \tanh(c_t)$
Cell state (cell output)	$c_t = f_t \odot c_{t-1} + i_t \odot \widetilde{c}_t$
Cell state candidate	$\widetilde{c}_t = \tanh(U_c \cdot x_t + W_c \cdot h_{t-1} + b_c)$
Input gate	$i_t = \sigma(U_i \cdot x_t + W_i \cdot h_{t-1} + b_i)$
Forget gate	$f_t = \sigma(U_f \cdot x_t + W_f \cdot h_{t-1} + b_f)$
Output gate	$o_t = \sigma(U_o \cdot x_t + W_o \cdot h_{t-1} + b_o)$

In the table, x_t is the current input gate; U_i, U_f and U_o are the weights of the hidden layer for the input gate, forget gate and output gate, respectively; W_i, W_f and W_o are the weights of the hidden state for the input gate, forget gate and output gate, respectively; b_i, b_f and b_o represent the bias for the input gate, forget gate and output gate, respectively; h_t is the new state (hidden); h_{t-1} is the previous state (hidden); σ is the logistic sigmoid activation function and tanh is the hyperbolic tangent function.

Sigma, σ, indicates that the gates' activation function is defined as a logistic sigmoid function (logsig) as follows:

$$f(x_t) = \sigma(x_t) = \frac{1}{1 + e^{-x_t}} \tag{11}$$

In a vanilla LSTM, the sigmoid function is used as the main activation function for the input gate, the output gate and the forget gate because it outputs a value between 0 and 1, and it can either let complete information flow or not throughout the gates. Additionally, to overcome the vanishing gradient problem caused by the derivative of the activation function, which is very close to zero in the gradient descent methods, we needed a function whose second derivative can be sustained for a long time before going to zero. Then, in the cell state candidate recurrence formula, the hyperbolic tangent function (tanh) ensures that the gate values are scaled between -1 and 1, thus increasing stability during gradient descent. Bidirectional LSTM (BiLSTM), which is simply the union of two conventional LSTM cells, allows cells to retrieve sequence information in both ways, either backward or forward, to continue the information transmission [35]. Figure 6 compares a conventional LSTM cell with a bidirectional LSTM cell.

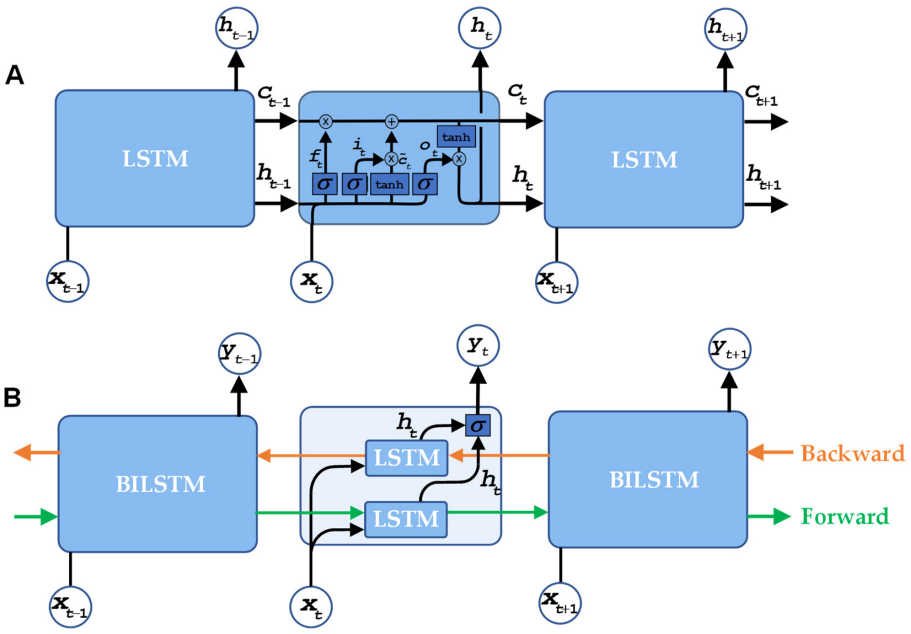

Figure 6. Comparison of a unidirectional LSTM cell with a bidirectional cell. (**A**) Unidirectional LSTM cell (LSTM). (**B**) Bidirectional LSTM cell (BiLSTM).

For the detector based on a LSTM network, a neuronal network with 9 layers and 8 connections was developed (see Figure 7).

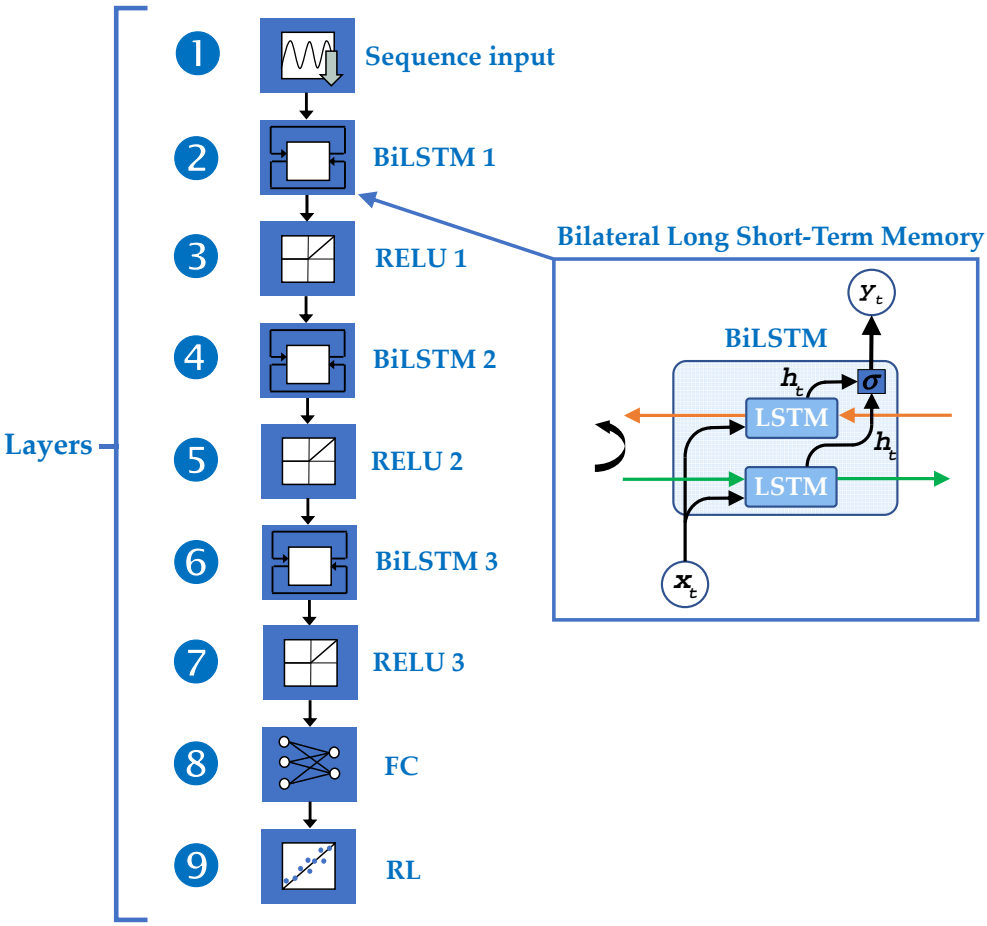

Figure 7. Architecture of long short-term memory (LSTM) network created for anomaly detection.

In addition to LSTM cells, rectified linear units (RELUs), fully connected (FC) layers and one last regression layer (RL) were used. RELU activation functions are responsible for setting the negative values between layer and layer to zero; they are not important for learning as they do not have saturation regions and they avoid stagnation during some phases of the network training, unlike other activation functions. The hyperparameters that define the architecture of the LSTM network used for detecting anomalies in the signals of the mechanical contraction of cardiac tissue are summarized in Table 5. The definition of the best hyperparameters was achieved using an empirical study.

For the training of the LSTM network, as these are real values, the loss function has been defined as a mean squared error (MSE) function, which calculates the error between network predictions and target values for the regression task:

$$J(W) = \frac{1}{N} \cdot \sum_{t=1}^{N} (y_t - \widetilde{y}_t)^2 \tag{12}$$

where y_t represents ground truth data and $\widetilde{y}_t$ is the output value of the neural network.

Table 5. Hyperparameters defining the architecture of the LSTM network.

Layer Name	Type	Activations	Learnables	States
1. Input	Sequence input	1	-	-
2. BiLSTM 1	BiLSTM	520	Input weights: 2080 × 1 Recurrent weights: 2080 × 260 Bias: 2080 × 1	Hidden states: 520 × 1 Cell states: 520 × 1
3. RELU 1	RELU	520	-	-
4. BiLSTM 2	BiLSTM	320	Input weights: 1280 × 520 Recurrent weights: 1280 × 160 Bias: 1280 × 1	Hidden states: 320 × 1 Cell states: 320 × 1
5. RELU 2	RELU	320	-	-
6. BiLSTM 3	BiLSTM	520	Input weights: 2080 × 320 Recurrent weights: 2080 × 260 Bias: 2080 × 1	Hidden states: 520 × 1 Cell states: 520 × 1
7. RELU 3	RELU	520		
8. Fully Connected	FC	1	Weights: 1 × 520 Bias: 1 × 1	-
9. Regression Layer	RL	1	Mean squared error	-

Numerical optimization that minimizes the loss of function by means of successive iterations, l, starting from a random initial value was used. In our case, for training the LSTM-type neural network, the adaptative moment estimation (Adam) algorithm was used; this is a stochastic gradient descent method based on adaptative estimation of the learning rates of moments of first, m_l, and second order, v_l, to further improve convergence time [36]. Computationally, it is a very efficient method as it uses little memory. Classical gradient descent methods use a single learning rate for all parameters, whereas the Adam method aims to improve network training by using adaptative training rates that can be adapted automatically to the loss function that is being optimized. When updating an iteration with the Adam method, a moving average is used such as the following:

$$\begin{aligned} &\text{Update formula}: W_{l+1} = W_l - \frac{\alpha \cdot \hat{m}_l}{\sqrt{\hat{v}_l} + \varepsilon} \\ &\hat{m}_l = \frac{m_l}{(1-\beta_1{}^l)} \quad m_l = \beta_1 \cdot m_{l-1} + (1-\beta_1) \cdot \nabla J(W_l) \\ &\hat{v}_l = \frac{v_l}{(1-\beta_2{}^l)} \quad v_l = \beta_2 \cdot v_{l-1} + (1-\beta_2) \cdot [\nabla J(W_l)]2 \end{aligned} \tag{13}$$

where m_l and v_l are first- and second-order moments of the current state, respectively; W_l, W_{l+1} and W_{l-1} are the weight vectors of the current, previous and next states, respectively; ∇J is the gradient of the loss function and β_1 and β_2 are the exponential decay rates of the first and second moments, respectively.

If gradients contain mostly noise, updating the weights with the moving average allows the moving average of the gradient to become smaller, and therefore the weight updates also become smaller. The moments indicated by the symbol (ˆ) are the moments with correction of the bias that appears at the beginning of training. The use of a regularizer allows for applying penalties on layer parameters during optimization. These penalties add to the loss function that optimizes the network. In the present study, a regularization penalty, L_2, was used. Additionally, adding a regularization term to the loss function weights, $J(W)$, is one way to reduce overfitting [37,38]. Then, the loss function after L_2 regularization was expressed as follows:

$$J_{L2}(W_l) = J(W_l) + L_2 \cdot W^2 \tag{14}$$

where $J_{L2}(W)$ is the loss function with L_2 regularization, W is the weight vector of the current state and L_2 is the penalty of L_2 regularization.

All parameters used in the training options of the LSTM neural networks are summarized in Table 6. Parameters related to the Adam optimization algorithm are based on [36].

Table 6. Training parameters for the LSTM network.

Description	Parameter Name	Value
Optimization algorithm	Solver name	Adam
First moment rate	Gradient decay factor	$\beta_1 = 0.9$
Second moment rate	Squared gradient decay factor	$\beta_2 = 0.999$
Epsilon	Epsilon	$\varepsilon = 10^{-8}$
Maximum number of epochs	Max epochs	30
Mini-batch size	Mini batch size	10
Shuffle the data	Shuffle	Once
Initial learning rate	Initial learnrate	$\alpha_0 = 0.001$
L_2 regularization	L_2 regularization	$L_2 = 0.0001$

2.4.2. Autoencoder-Based Detector (D.2)

Autoencoders are unsupervised learning algorithms with unlabeled data that are based on neural networks and consist of two parts: an encoder and a decoder. The function of the encoder is to compress data. For given input data $x = \{x_1, x_2, x_3, \ldots, x_N\}$, the encoder compresses the N-dimensional input in a code $h\{h_1, h_2, h_3, \ldots, h_M\}$, where $M < N$, named latent representation, which includes most of the input information. The architecture of a basic autoencoder is shown in Figure 8.

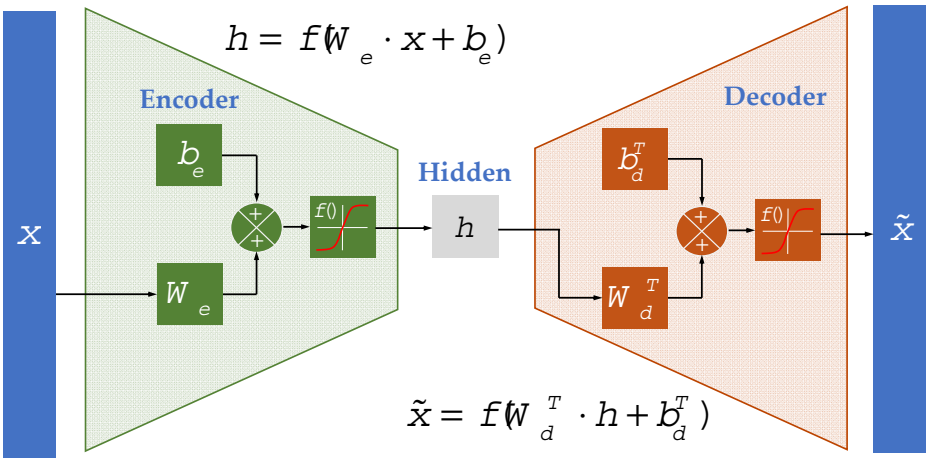

Figure 8. Basic autoencoder structure.

The network of the autoencoder can be mathematically represented by the corresponding encoding and decoding equations (see Figure 8):

$$\begin{aligned} h &= f(W_e \cdot x + b_e) \\ \tilde{x} &= f({W_d}^T \cdot h + b_d^T) \end{aligned} \tag{15}$$

where x and $\tilde{x}$ are the input and output signal vectors of the autoencoder $N \times 1$, respectively; h is the hidden state $M \times 1$; W_e and W_d are the weight vectors of the encoder and decoder $M \times N$, respectively; and b_e and b_d are the bias of the encoder and the decoder $M \times 1$, respectively.

In order to avoid the vanishing gradient problem, the hyperbolic tangent function (tanh) and logistic sigmoid function (logsig) are commonly used. Both are s-shaped curves; the only difference is that logsig lies between 0 and 1 whereas tanh lies between 1 and -1. In this study, the logistic sigmoid activation function (logsig) has been used for both the

encoder and the decoder because it is commonly used for models where probabilities have to be predicted. Since the probability of anything exists only between the range of 0 and 1, the tanh function does not have a clear relationship to modeling probability.

For the definition of the loss function of the autoencoder, since the input, x, has a real value, a mean squared error sparse (msesparse) function was used. This function calculates the error between the input and the output value of the autoencoder as well as adding two terms of regularization:

$$J(W) = \frac{1}{N} \cdot \sum_{i=1}^{N} (x_i - \widetilde{x}_i)^2 + \Omega_{L2} + \Omega_{KL} \tag{16}$$

where N is the number of samples of the input signal vector; x_i and $\widetilde{x}_i$ are the values of the sample at the input and output of the autoencoder, respectively; Ω_{L2} is the regularization term L_2 and Ω_{KL} is the sparsity regularization term.

In Equation (11), it can be observed that two regularization terms are used with the objective of avoiding overfitting. L_2 regularization affects weights, whereas dispersion regularization, W, affects the activity of the hidden layers, h. To minimize loss function, the stochastic gradient descent (SGD) was used. The regularization term L_2 or ridge regularization is expressed as the i-th sum of the squares of the weights, W, multiplied by the weight decay parameter, λ, which tends to decrease the magnitude of weights and helps to avoid overfitting [39,40]:

$$\Omega_{L2} = \lambda \cdot \frac{1}{2} \cdot \sum_{l=1}^{L} \sum_{i=1}^{s_l} \sum_{j=1}^{s_{l+1}} \left(W_{ji}^{l}\right)^2 \tag{17}$$

where λ is the weight decay parameter, L is the number of layers, s_l is the number of units in the layer l and W_{ji} is the weight between unit j of layer l and unit i of layer $l + 1$.

It is interesting to be able to achieve a mean activation, ρ, of each hidden neuron, j, close to the dispersion parameter; that is, $\hat{\rho}_j = \rho$. Typically, the dispersion parameter, ρ, takes a small value close to zero. In the present study, $\rho = 0.1$ was selected. In order to meet this restriction, activations of the hidden unit should be mostly close to 0. To achieve this, the dispersion regularization term, Ω_{KL}, penalizes when $\hat{\rho}_j$ is significantly deviated from ρ on the basis of the Kullback–Leibler divergence (KL) concept [41]:

$$\Omega_{KL} = \beta \cdot \sum_{j=1}^{M} D_{\mathrm{KL}}(\rho \parallel \hat{\rho}_j) \tag{18}$$

where M is the number of hidden neurons; β is the dispersion regularization parameter; $\hat{\rho}$ is the vector of the mean activity of the hidden layer, h; and ρ is the sparsity proportion parameter.

The Kullback–Leibler divergence (KL) is a measure between two statistical distributions—specifically the distance between two distributions of Bernoulli random variables, one with a mean ρ and the other with a mean $\hat{\rho}$. The dispersion regularization parameter, β, regulates the weight of the dispersion term, $\Omega_{KL.}$

The parameters used in the training options of the autoencoder are summarized in Table 7. The values of the regularization parameters and loss function selection are based on [39].

Table 7. Training parameters for the autoencoder detector.

Description	Parameter Name	Value
Number of hidden nodes	Hidden size	200
L_2 regularization	L_2 weight regularization	$\lambda = 0.0001$
Regularization of dispersion	Sparsity regularization	$\beta = 0.0001$
Sparsity proportion	Sparsity proportion	$\rho = 0.1$
Maximum number of epochs	Max epochs	200
Loss function	Loss function	msesparse
Activation function (encoder)	Encoder transfer function	logsig
Activation function (decoder)	Decoder transfer function	logsig

3. Results

3.1. Synthetic Data

The intersection over union (IOU) and Sørensen–Dice index (SDI) criteria are commonly used to calculate the accuracy of an anomaly detector in synthetic data.

$$IOU = \frac{\text{intersection}}{\text{union}} = \frac{\text{ROI}_D \cap \text{ROI}_{GT}}{\text{ROI}_D \cup \text{ROI}_{GT}}. \in [0,1] \tag{19}$$

where ROI_D is the anomalous region detected by the anomaly detector and ROI_{GT} is the ground truth anomaly region.

The region of interest (ROI) corresponds to the cardinalities of the two regions (i.e., the number of samples in each set). Figure 9 shows the practical calculation of the intersection over union (*IOU*) when the region of interest of the ground truth (ROI_{GT}) and the region of interest detected by the anomaly detector (ROI_D) are known.

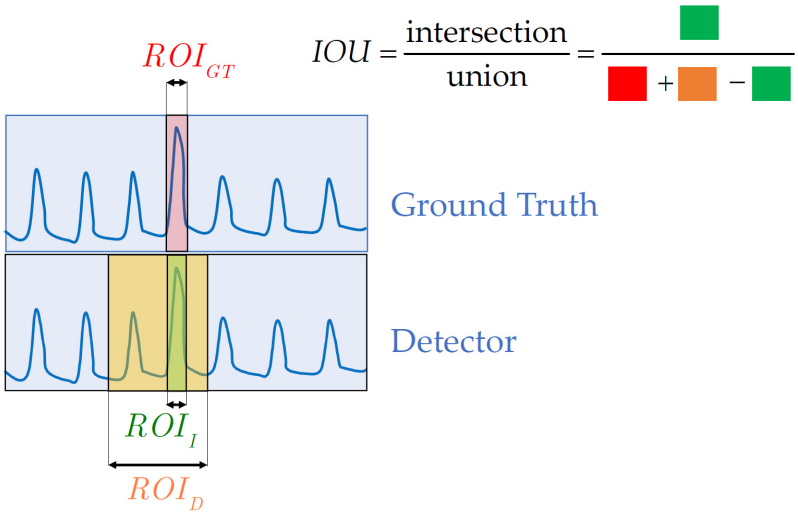

Figure 9. Method of obtaining the intersection over union (IOU) in signals. Colors boxes in the formula indicates the different regions of interest (ROI), according to the figure on the left.

Alternatively, the IOU can be written using the definition of true positive (TP), false positive (FP) and false negative (FN), as follows:

$$IOU = \frac{\text{intersection}}{\text{union}} = \frac{TP}{TP + FP + FN} \in [0,1] \tag{20}$$

This index is known by several other names, especially F1 score. In a similar way, the Sørensen–Dice index (SDI) or F1 score is defined as follows:

$$SDI = \frac{2 \cdot TP}{2 \cdot TP + FP + FN} \in [0,1] \tag{21}$$

It differs from the *IOU* index which only counts true positives once in both the numerator and the denominator. Both indices, *IOU* and Sørensen–Dice, range in the closed interval between 0 and 1. They can be interpreted as a similarity measure over sets.

Figure 10 shows the detection accuracy using the anomaly detectors based on the autoencoder and LSTM for one particular case of a synthetic signal: higher amplitude.

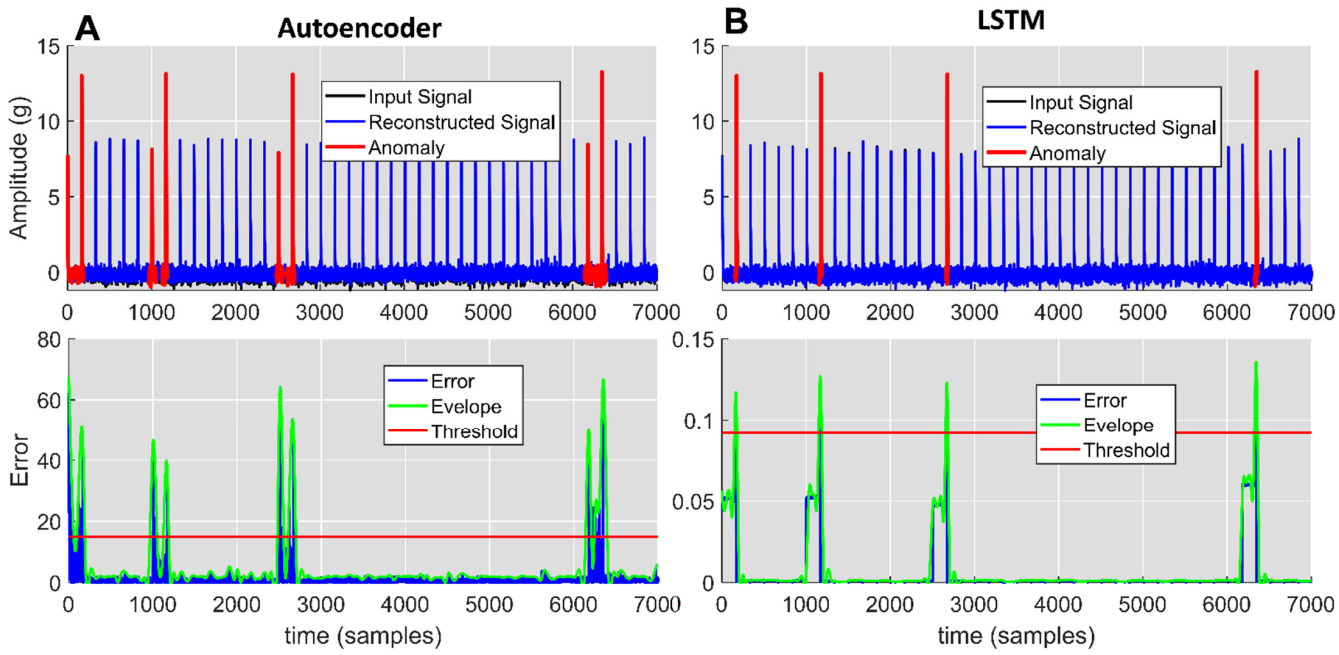

Figure 10. Detection of anomalies in the synthetic signal type: higher amplitude. (**A**) Autoencoder-based detector. (**B**) LSTM-based detector.

Table 8 shows the accuracy obtained by the classifiers based on the autoencoder and the LSTM-type neural network; we decided to use the *SDI* to evaluate the accuracy. Synthetic data which model seven types of anomalies representative of the contractile signals of heart tissue, excluding normal signals, were used for evaluation (see Table 5).

Table 8. Anomaly detectors' accuracy.

Signal Type	Sørensen–Dice Index (SDI)	
	Autoencoder	LSTM
2. Lower amplitude	0.55	0.83
3. Higher amplitude	0.71	0.99
4. Missing pulse	0.63	0.91
5. Slow pulse decay	0.67	0.95
6. Fast pulse decay	0.61	0.88
7. Early or anticipated pulse	0.51	0.77
8. Pause/block	0.61	0.9

The LSTM neural-network-based framework presented the best accuracy. It is also noted that the early pulse signal was the most difficult signal to detect in both approaches.

3.2. Experimental Data

This section describes the application of the anomaly detection framework based on deep learning to experimental data. In the previous section, it was validated with synthetic data that LSTM offers better accuracy in detecting anomalies (see Section 3.1). Therefore, the LSTM-type anomaly detector was tested on experimental signals of mechanical contraction of left atrium cardiac tissue (AE) stimulated at successive frequencies of 1, 2, 4, 6, 8

and 10 Hz and also on experimental data from the unstimulated right atrium, since it is controlled by pacemaker cells. For stimulated left atria recordings, normal experimental recordings were used for network training, while for right atria recordings, normal experimental signals stimulated at a frequency of 5 Hz were used, as the observed contraction in the heart tissue of the left atrium is approximately 5 Hz. Specifically, 16 records were selected from 12 mice where representative anomalies were observed.

Figure 11 shows the experimental signals of the normal category used for the training of the fault detector used in the experimental signals.

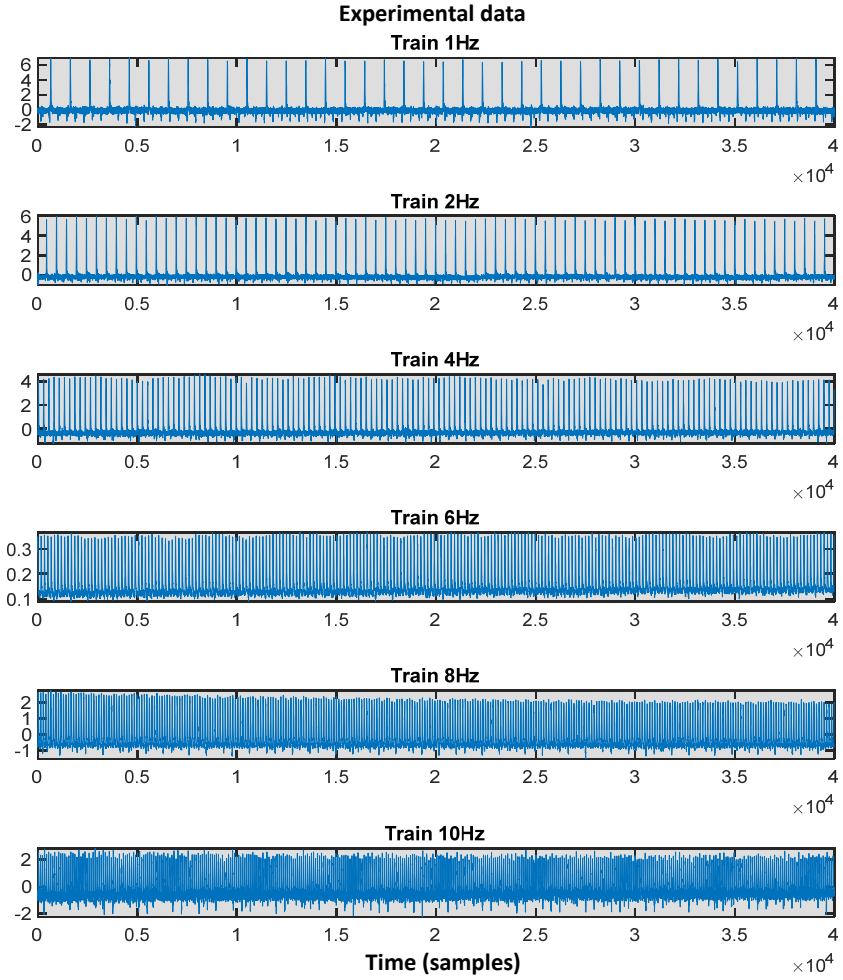

Figure 11. Normal experimental signals with different stimulation frequencies (1 to 10 Hz). These signals were used for training the anomaly detector that was applied to experimental signals.

Figure 12 shows the LSTM-type anomaly detector applied to experimental mechanical signals from the cardiac tissue of the left and right atria of CD-1 mice. It visually confirms that the detector is able to satisfactorily identify the anomalies present in the signal.

Figure 12A shows an anomaly that appeared in a left atrium when stimulated at 5 Hz. Figure 12B shows anomalies detected in a right atrium, which remained unstimulated as it has a natural physiological pacemaker. These atria that did not follow a stimulation pattern were more prone to the genesis of anomalies. Specifically, two cardiac arrest phenomena appeared, which were correctly identified by the anomaly detector.

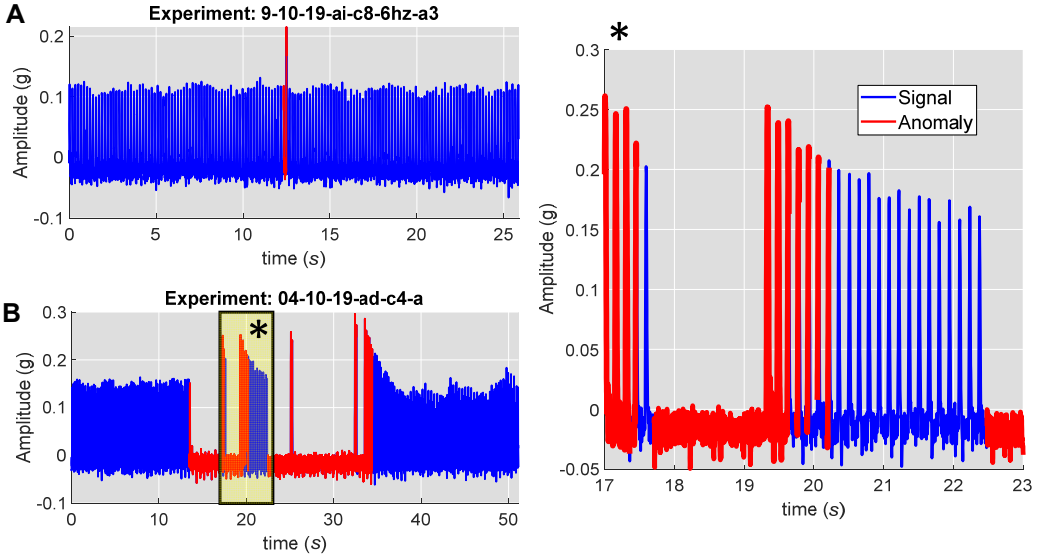

Figure 12. Anomalies detected in experimental signals. (**A**) Left atrium. (**B**) Right atrium. The asterisk (*) indicates a zoom into figure. The triangle indicates the anomaly location.

4. Discussion

We have reported a method for the detection of anomalies in cardiac tissue based on deep learning, in which two anomaly detectors have been evaluated. The first detector used a recurrent neural network (RNN) of long short-term memory (LSTM) type, while the second used an autoencoder. The proposed methodology is "normal-only" because training of the algorithm does not require labeling of the whole dataset, but only of normal data. This is an approach that uses the power of deep learning representation without usual a labeling constraint, just normal data. No previous studies have been found in the literature where the methodology of detecting anomalies is applied to mechanical contractions in cardiac tissue. Nevertheless, we can compare it with the accuracy achieved by deep-learning-based anomaly detectors in cardiac electrical signals, which is in the range between 0.75 and 0.1 (Zhao et al., 2021) [42]. In our study, despite dealing with a signal with a high signal-to-noise ratio and a lot of biological variability, when we used the LSTM-based anomaly detector, we achieved a maximum accuracy of 0.99 and an accuracy of 0.88 in the worst case.

A set of synthetic data was constructed, which allowed for accurately describing the different anomalies observed in the experimental recordings of the mouse cardiac atria. These artificial data were used to validate the correct performance of the anomaly detection methodology that was subsequently applied to real experimental data.

The results of the anomaly detector framework presented in this study have been exported to an *html* file with interactive graphics where the anomalous regions are highlighted. In addition, this file incorporates basic visualization tools that allow the researcher or clinician to perform actions such as zooming in, zooming out, scaling and downloading the results as an image file (png file). This whole set of interactive tools allows to delve deeper into the analysis or explanation of anomalies which are associated with the generation of cardiac arrhythmias and to easily gain actionable insights. Files with an *html* extension can be used in any operating system (OS), do not require computer resources and can be used in offline mode (without Internet connection). All *html* interactive graphics files are freely accessible on a Git repository page (Github) under the MIT Open software license (See supplementary data section).

From a biological point of view, this methodology may be of interest in the fields of heart physiology, pharmacology or pathology. Although the aim of this work was primarily methodological, it is important to note that the high frequency of heartbeats occurring physiologically in a mouse can make quantification difficult. The detection of abnormalities

is a first step in the detection of different cardiac pathologies associated with changes in contraction amplitude (inotropic), frequency (chronotropic) or rhythm (arrhythmias). This type of experimental approach, together with traditional methods of calculating the parameters associated with contractions (amplitude, duration, frequency, etc.), may be useful in future studies where these alterations are studied.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/math10152786/s1, S1: Autoencoder-based anomaly detector framework; S2: LSTM-based anomaly detector framework. Additionally, interactive graphics files are freely accessible on a Git repository page under the MIT Open software license (https://github.com/xavierMarimon/CardiacTissueAnomaly/).

Author Contributions: Conceptualization, X.M., R.B. and M.J.; data curation, X.M., S.T., M.J. and A.O.; formal analysis, X.M. and R.B.; funding acquisition, R.B. and M.J.; investigation, X.M., S.T., R.B., M.J. and A.O.; methodology, X.M., S.T., R.B. and M.J.; project administration, R.B. and M.J.; resources, R.B. and M.J.; software, X.M. and A.O.; supervision, R.B. and M.J.; validation, X.M., S.T., R.B., M.J. and A.O.; visualization, X.M. and A.O.; writing—original draft, X.M. and R.B.; writing—review and editing, X.M., S.T., R.B., M.J. and A.O. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Spanish Ministry of Science and Innovation, grant number PID2020-116927RB-C22.

Institutional Review Board Statement: The animal study protocol was approved by the Institutional Ethics Committee of Universitat de Autónoma de Barcelona.

Informed Consent Statement: Not applicable.

Data Availability Statement: Codes are available from the authors upon request.

Acknowledgments: We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X Pascal GPU used for this research and Marta Pulido for editing the manuscript and editorial assistance.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Lindemann, B.; Fesenmayr, F.; Jazdi, N.; Weyrich, M. Anomaly Detection in Discrete Manufacturing Using Self-Learning Approaches. *Procedia CIRP* **2019**, *79*, 313–318. [CrossRef]
2. Gupta, M.; Gao, J.; Aggarwal, C.C.; Han, J. Outlier Detection for Temporal Data: A Survey. *IEEE Trans. Knowl. Data Eng.* **2014**, *26*, 2250–2267. [CrossRef]
3. Chandola, V.; Cheboli, D.; Kumar, V. *Detecting Anomalies in a Time Series Database*; Technical Report; University of Minnesota: Minneapolis, MN, USA, 2009.
4. Hawkins, D.M. Identification of Outliers. Chapman and Hall: London—New York 1980, 188 S., £ 14, 50. *Biom. J.* **1987**, *29*, 198. [CrossRef]
5. Keogh, E.; Lin, J.; Fu, A. HOT SAX: Efficiently Finding the Most Unusual Time Series Subsequence. In Proceedings of the Fifth IEEE International Conference on Data Mining (ICDM'05), Houston, TX, USA, 27–30 November 2005; pp. 226–233.
6. Lis, A.; Dworakowski, Z.; Czubak, P. An Anomaly Detection Method for Rotating Machinery Monitoring Based on the Most Representative Data. *J. Vibroeng.* **2021**, *23*, 861–876. [CrossRef]
7. Yang, H.; Kan, C.; Liu, G.; Chen, Y. Spatiotemporal Differentiation of Myocardial Infarctions. *IEEE Trans. Autom. Sci. Eng.* **2013**, *10*, 938–947. [CrossRef]
8. Yang, H. Multiscale Recurrence Quantification Analysis of Spatial Cardiac Vectorcardiogram Signals. *IEEE Trans. Biomed. Eng.* **2011**, *58*, 339–347. [CrossRef]
9. Li, H.; Boulanger, P. A Survey of Heart Anomaly Detection Using Ambulatory Electrocardiogram (ECG). *Sensors* **2020**, *20*, 1461. [CrossRef]
10. Berndt, D.J.; Clifford, J. Using Dynamic Time Warping to Find Patterns in Time Series. In Proceedings of the KDD Workshop, New York, NY, USA, 26 April 1994.
11. Sivaraks, H.; Ratanamahatana, C.A. Robust and Accurate Anomaly Detection in ECG Artifacts Using Time Series Motif Discovery. *Comput. Math. Methods Med.* **2015**, *2015*, 1–20. [CrossRef]
12. Chuah, M.C.; Fu, F. ECG Anomaly Detection via Time Series Analysis. In *Frontiers of High Performance Computing and Networking ISPA Workshops*; Parimala, T., Ed.; Springer: Berlin, Germany, 2007; pp. 123–135. [CrossRef]

13. Pham, N.D.; Le, Q.L.; Dang, T.K. HOT ASAX: A Novel Adaptive Symbolic Representation for Time Series Discords Discovery. In *Lecture Notes in Computer Science*; (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); Springer: Berlin, Germany, 2010; Volume 5990, pp. 113–121. ISBN 3642121446.
14. Veeravalli, B.; Deepu, C.J.; Ngo, D. Real-Time, Personalized Anomaly Detection in Streaming Data for Wearable Healthcare Devices. In *Handbook of Large-Scale Distributed Computing in Smart Healthcare*; Khan, S.U., Zomaya, A.Y., Abbas, A., Eds.; Springer: Berlin, Germany, 2017; pp. 403–426. [CrossRef]
15. Christov, I.; Gómez-Herrero, G.; Krasteva, V.; Jekova, I.; Gotchev, A.; Egiazarian, K. Comparative Study of Morphological and Time-Frequency ECG Descriptors for Heartbeat Classification. *Med. Eng. Phys.* **2006**, *28*, 876–887. [CrossRef]
16. de Chazal, P.; Reilly, R.B. A Patient-Adapting Heartbeat Classifier Using ECG Morphology and Heartbeat Interval Features. *IEEE Trans. Biomed. Eng.* **2006**, *53*, 2535–2543. [CrossRef]
17. Llamedo, M.; Martínez, J.P. Heartbeat Classification Using Feature Selection Driven by Database Generalization Criteria. *IEEE Trans. Biomed. Eng.* **2011**, *58*, 616–625. [CrossRef]
18. Gretton, A.; Smola, A.; Huang, J.; Schmittfull, M.; Borgwardt, K.; Schölkopf, B. Covariate Shift by Kernel Mean Matching. In *Dataset Shift in Machine Learning*; The MIT Press: Cambridge, MA, USA, 2008; pp. 131–160.
19. Li, K.; Du, N.; Zhang, A. Detecting ECG Abnormalities via Transductive Transfer Learning. In Proceedings of the ACM International Conference on Bioinformatics, Computational Biology and Biomedicine-BCB'12, Orlando, FL, USA, 8–10 October 2012; pp. 210–217. [CrossRef]
20. Ye, C.; Kumar, B.V.K.V.; Coimbra, M.T. Heartbeat Classification Using Morphological and Dynamic Features of ECG Signals. *IEEE Trans. Biomed. Eng.* **2012**, *59*, 2930–2941. [CrossRef]
21. Zhang, Z.; Dong, J.; Luo, X.; Choi, K.-S.; Wu, X. Heartbeat Classification Using Disease-Specific Feature Selection. *Comput. Biol. Med.* **2014**, *46*, 79–89. [CrossRef] [PubMed]
22. Rajesh, K.N.V.P.S.; Dhuli, R. Classification of ECG Heartbeats Using Nonlinear Decomposition Methods and Support Vector Machine. *Comput. Biol. Med.* **2017**, *87*, 271–284. [CrossRef] [PubMed]
23. Thomas, M.; Das, M.K.; Ari, S. Automatic ECG Arrhythmia Classification Using Dual Tree Complex Wavelet Based Features. *AEU. Int. J. Electron. Commun.* **2015**, *69*, 715–721. [CrossRef]
24. Wess, M.; Manoj, P.D.S.; Jantsch, A. Neural Network Based ECG Anomaly Detection on FPGA and Trade-off Analysis. In Proceedings of the 2017 IEEE International Symposium on Circuits and Systems (ISCAS), Baltimore, MD, USA, 28–31 May 2017; pp. 1–4.
25. Kiranyaz, S.; Ince, T.; Gabbouj, M. Real-Time Patient-Specific ECG Classification by 1-D Convolutional Neural Networks. *IEEE Trans. Biomed. Eng.* **2016**, *63*, 664–675. [CrossRef] [PubMed]
26. Kiranyaz, S.; Ince, T.; Gabbouj, M. Personalized Monitoring and Advance Warning System for Cardiac Arrhythmias. *Sci. Rep.* **2017**, *7*, 9270. [CrossRef]
27. Rajpurkar, P.; Hannun, A.Y.; Haghpanahi, M.; Tison, G.H.; Bourn, C.; Turakhia, M.P.; Ng, A.Y. Cardiologist-Level Arrhythmia Detection and Classification in Ambulatory Electrocardiograms Using a Deep Neural Network. *Nat. Med.* **2019**, *25*, 65–69. [CrossRef]
28. Acharya, U.R.; Fujita, H.; Lih, O.S.; Hagiwara, Y.; Tan, J.H.; Adam, M. Automated Detection of Arrhythmias Using Different Intervals of Tachycardia ECG Segments with Convolutional Neural Network. *Inf. Sci.* **2017**, *405*, 81–90. [CrossRef]
29. Zhou, H.; Kan, C. Tensor-Based ECG Anomaly Detection toward Cardiac Monitoring in the Internet of Health Things. *Sensors* **2021**, *21*, 4173. [CrossRef]
30. Li, D.; Zhang, J.; Zhang, Q.; Wei, X. Classification of ECG Signals Based on 1D Convolution Neural Network. In Proceedings of the 2017 IEEE 19th International Conference on e-Health Networking, Applications and Services (Healthcom), Dalian, China, 12–15 October 2017; pp. 1–6.
31. Bailly, R.; Malfante, M.; Allier, C.; Ghenim, L.; Mars, J. *Deep Anomaly Detection Using Self-Supervised Learning: Application to Time Series of Cellular Data*; IEEE Access: Porto, Portugal, 2021.
32. Malhotra, P.; Vig, L.; Shroff, G.M.; Agarwal, P. *Long Short Term Memory Networks for Anomaly Detection in Time Series*; ESANN: Bruges, Belgium, 2015.
33. McLachlan, G.; Peel, D. *Finite Mixture Models*; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2000; ISBN 9780471721185.
34. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
35. Wang, J.; Zhang, J.; Wang, X. Bilateral LSTM: A Two-Dimensional Long Short-Term Memory Model With Multiply Memory Units for Short-Term Cycle Time Forecasting in Re-Entrant Manufacturing Systems. *IEEE Trans. Ind. Inform.* **2018**, *14*, 748–758. [CrossRef]
36. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. In Proceedings of the 3rd International Conference for Learning Representations, San Diego, CA, USA, 22 December 2015.
37. Bishop, C.M. *Pattern Recognition and Machine Learning*; Springer: New York, NY, USA, 2006.
38. Murphy, K.P. *Machine Learning: A Probabilistic Perspective*; The MIT Press: Cambridge, MA, USA, 2012.
39. Olshausen, B.A.; Field, D.J. Sparse Coding with an Overcomplete Basis Set: A Strategy Employed by V1? *Vis. Res.* **1997**, *37*, 3311–3325. [CrossRef]
40. Nair, V.; Hinton, G.E. 3D Object Recognition with Deep Belief Nets. In *Advances in Neural Information Processing Systems*; Bengio, Y., Schuurmans, D., Lafferty, J., Williams, C., Culotta, A., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2009; Volume 22.

41. Coates, A.; Ng, A.Y. The Importance of Encoding versus Training with Sparse Coding and Vector Quantization. In Proceedings of the 28th International Conference on International Conference on Machine Learning, Washington, DC, USA, 28 June–2 July 2011; Omnipress: Madison, WI, USA, 2011; pp. 921–928.
42. Zhao, X.; Yan, H.; Hu, Z.; Du, D. Deep Spatio-Temporal Sparse Decomposition for Trend Prediction and Anomaly Detection in Cardiac Electrical Conduction. *IISE Trans. Healthc. Syst. Eng.* **2022**, *12*, 150–164. [CrossRef]