

## Article

# The Application of Deep Learning Algorithms for PPG Signal Processing and Classification

Filipa Esgalhado <sup>1,2,3,\*</sup>, Beatriz Fernandes <sup>3</sup>, Valentina Vassilenko <sup>1,2,3</sup> , Arnaldo Batista <sup>1,4</sup>  and Sara Russo <sup>1</sup>

<sup>1</sup> NOVA School of Science and Technology, NOVA University Lisbon, 2829-516 Caparica, Portugal; vv@fct.unl.pt (V.V.); agb@fct.unl.pt (A.B.); s.russo@campus.fct.unl.pt (S.R.)

<sup>2</sup> LIBPhys-Laboratory of Instrumentation, Biomedical Engineering and Radiation Physics, 2829-516 Caparica, Portugal

<sup>3</sup> NMT, S.A., Parque Tecnológico de Cantanhede, Núcleo 04, Lote 3, 3060-197 Cantanhede, Portugal; beatrizserenofernandes00@gmail.com

<sup>4</sup> UNINOVA CTSNOVA, School of Science and Technology, NOVA University Lisbon, 2829-516 Caparica, Portugal

\* Correspondence: feo.cardoso@campus.fct.unl.pt

**Abstract:** Photoplethysmography (PPG) is widely used in wearable devices due to its convenience and cost-effective nature. From this signal, several biomarkers can be collected, such as heart and respiration rate. For the usual acquisition scenarios, PPG is an artefact-ridden signal, which mandates the need for the designated classification algorithms to be able to reduce the noise component effect on the classification. Within the selected classification algorithm, the hyperparameters' adjustment is of utmost importance. This study aimed to develop a deep learning model for robust PPG wave detection, which includes finding each beat's temporal limits, from which the peak can be determined. A study database consisting of 1100 records was created from experimental PPG measurements performed in 47 participants. Different deep learning models were implemented to classify the PPG: Long Short-Term Memory (LSTM), Bidirectional LSTM, and Convolutional Neural Network (CNN). The Bidirectional LSTM and the CNN-LSTM were investigated, using the PPG Synchronously Squeezed Fourier Transform (SSFT) as the models' input. Accuracy, precision, recall, and F1-score were evaluated for all models. The CNN-LSTM algorithm, with an SSFT input, was the best performing model with accuracy, precision, and recall of 0.894, 0.923, and 0.914, respectively. This model has shown to be competent in PPG detection and delineation tasks, under noise-corrupted signals, which justifies the use of this innovative approach.

**Keywords:** PPG; biomedical signal processing; deep learning; neural networks; RNN; CNN; LSTM



**Citation:** Esgalhado, F.; Fernandes, B.; Vassilenko, V.; Batista, A.; Russo, S. The Application of Deep Learning Algorithms for PPG Signal Processing and Classification. *Computers* **2021**, *10*, 158. <https://doi.org/10.3390/computers10120158>

Academic Editor: M. Ali Akber Dewan

Received: 30 October 2021

Accepted: 22 November 2021

Published: 25 November 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Photoplethysmography (PPG) is a non-invasive technique that is used to detect blood volume variations through an infrared light sensor placed on the surface of the skin [1,2]. Correct identification of the PPG waveform and its main features is essential in order to extract several biomarkers, such as heart rate, blood pressure, cardiac output, and blood oxygen saturation, when the red and infrared light are used simultaneously [1,3]. The PPG sensors are usually placed on the distal parts of the human body, such as the arms, fingers, feet, or ears. For this reason, motion artifacts are a main contributor to PPG signal degradation.

Multiple algorithms based on digital filters [4], adaptive thresholds [5], and wavelet transform [6] have been proposed to identify PPG features. However, most studies involve low noise data recordings for which algorithms' robustness to artefacts' contamination cannot be properly evaluated [7]. Since one of the main drawbacks of the PPG signal is its high susceptibility to noise contamination [8], it is crucial to develop an algorithm that is able to overcome this limitation [9].

### Related Work

Deep learning algorithms have been applied in diagnosis, classification, and waveform segmentation on different biomedical signals, such as the electrocardiogram (ECG) and PPG. Ribeiro et al. [10] used a Deep Neural Network for automatic diagnosis with a 12-lead ECG signal as an input. The model achieved an *F1* score above 80% and specificity over 99%. Hannun et al. [11] developed a similar model that had a receiver operating characteristic curve of 0.97, and its average *F1* score was higher than the cardiologist score average. Using the PPG as a model input, Soltane et al. [12] used an Artificial Neural Network to categorize the signal into two classes: healthy and pathologic. A correct classification rate of 94.7% for data set testing was achieved. Liu et al. [13] classified the PPG quality in three categories using a Deep Convolution Neural Network. The best performing algorithm had a 92.5% accuracy. Yen et al. [14] classified hypertension stages, based on PPG signals, using a Deep Residual Network, Convolutional Neural Network, and Bidirectional Long Short-Term Memory model. An accuracy of 76% was achieved in the testing data. Song et al. [15] estimated the heart rate using two different PPG datasets. The optimized Deep Learning model achieved a mean absolute error of 6.02 beats per minute. Alessandrini et al. [16] studied Recurrent Neural Networks to recognize human activity based on PPG and accelerometer data. The developed model achieved a 95% accuracy. Li et al. [17] estimated real-time blood pressure with a Deep Learning model, using the PPG as the input data. A mean error of 4.638 and 3.155 mmHg was achieved for the systolic and diastolic blood pressure, respectively.

Other studies have focused on signal waveform delineation. Laitala et al. [18] proposed a Long Short-Term Memory (LSTM) network to detect R peaks in the ECG. The R peaks are points in the ECG waveform, located in the ventricular depolarization interval, representing in most cases the maximum absolute value in that segment. A double Bidirectional LSTM and Dense layers were chosen, where the best achieved precision was 100%. However, this study used a small subject group ( $n < 15$ ). Kim et al. [19] studied ECG-based biometrics identification and classification. A bidirectional LSTM model was applied to two ECG databases and an overall precision of 100% was achieved in the best performing architecture. Malali et al. [20] segmented the ECG in the P-wave, QRS-complex, and T-wave. The proposed Convolutional-LSTM architecture achieved an accuracy of 94.87%, 96.66%, and 92.73% for the P-wave, QRS-complex, and T-wave, respectively.

The main objective of this work was to develop a robust Artificial Intelligent (AI) detector, in Matlab<sup>®</sup> and Python code, able to accurately detect each PPG beat time limits. This operation, usually referred to as wave delineation or segmentation, allows for PPG peak determination. The PPG peak location along with its time limits are needed for the calculation of the clinical features that this signal provides for diagnosis and research, such as systolic and diastolic points and heart rate variation. The accuracy of these features will impact on the test and validation of machine learning algorithms. In real-life situations, the PPG signal is contaminated with noise, namely movement artifacts, specifically when wearable devices are used, either in clinical or research environments.

Although successful automatic ECG delineation algorithms can be found in the literature, there is a scarcity of similar methodologies applied to the PPG. The herein presented work intends to be a contribution in this respect. Frequency features derived from biomedical signals, such as the ECG or the electroencephalogram (EEG), have been widely used for classification purposes [21,22]. In this work, time-frequency-derived features were obtained using a synchrosqueezed transform, given the non-stationary nature of the PPG signals. Time-frequency representation is established as a reference method for non-stationary signal analysis [23,24], from which features can be derived for classification purposes [21,22,25]. Feeding these time-frequency features to the herein deep learning PPG delineation process can be considered an innovative procedure, along with a sample-by-sample classification method applied to one of the selected models under study.

## 2. Materials and Methods

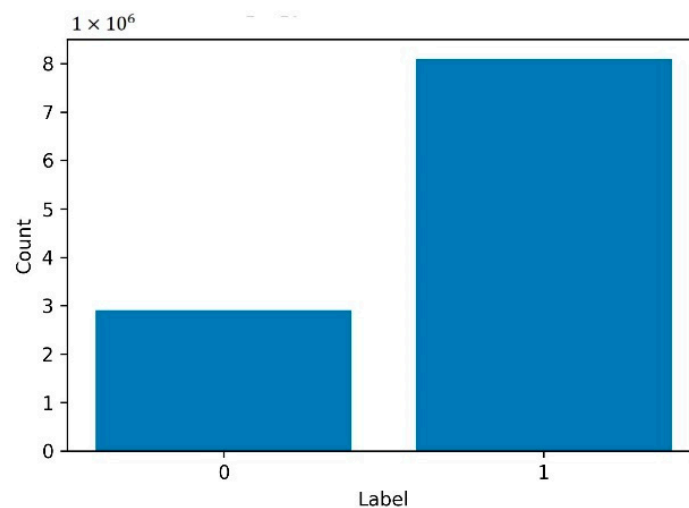
### 2.1. Data Acquisition and Pre-Processing

A total of 47 volunteers of both genders, aged from 18 to 66 years old, participated in the study. All subjects were healthy and had signed an informed consent for study participation. The working database was anonymized. The signals were recorded with a sampling frequency of 2000 Hz from the right index finger by a PPG sensor, model SS4LA, connected to the MP35 equipment of BIOPAC® Systems Inc., Goleta, CA, USA. The recording interval varied between 5 and 7 min.

After recording, pre-processing steps were applied to the signal. A bandpass filter between 0 and 4 Hz [26,27] and a down-sampling to 500 Hz were applied to the PPG data [27]. The signals were then divided into 20 s segments [28].

A total of 1100 signal segments were accounted for in this study. During manual expert labeling, each of the PPG samples was classified as “true” or “false”, which corresponded to areas identified as true PPG or noise, respectively. The noise category included the signal minimums and noisy signal segments. Signal minimums are important for use in projects that include PPG segmentation. For coding purposes, the label class “true” was assigned a value of 1 while the class “false” was replaced by a value of 0. The PPG data and corresponding labels were equally sized matrices (1100 × 10,000). All the previously mentioned pre-processing steps were performed in MATLAB®, version 2020b.

In the initial exploratory analysis of the experimental data, an imbalance of the two labels was detected. A histogram is presented in Figure 1, where label 1, which represents the PPG, corresponds to 74% of the data pool, with label 0 corresponding to noise, 26%. In order to overcome this issue, the parameter *sample\_weights* was included in some models. In this transformation, the samples with the labels 0 and 1 were given a 1.85 and 0.68 weight, respectively. These values were established after an observational study involving weights’ tuning.



**Figure 1.** Histogram of each label, where 0 represents noise and 1 represents the PPG.

### 2.2. Feature Extraction Using Time-Frequency Analysis

Besides using the PPG data as model input, time-frequency features were also extracted for classification. Both methodologies were compared regarding model performance. The non-stationary nature of the PPG was the motivation to use time-frequency-extracted features, given its time-varying frequency content [23,29].

For each PPG segment, a Synchrosqueezed Fourier Transform (SSFT) [29,30] with a Kaiser window of 250 samples was applied. The synchrosqueezing [31] application to the short time Fourier transform resulted in an instantaneous frequency-increased resolution in the time frequency plane [32].

The multicomponent input signal of the SSFT can be defined as [29]:

$$f(t) = \sum_{k=1}^K f_k(t) = \sum_{k=1}^K a_k(t) e^{j2x\phi_k(t)}, \quad (1)$$

where  $k$  is finite,  $a_k(t) > 0$  is a continuously differentiable function,  $\phi_k(t)$  is a two times continuously differentiable function, and  $f_k$  is a mode of  $f$ . The short-time Fourier transform of the  $f$  function, using the spectral window  $g$ , is given by [29]:

$$V_g f(t, \eta) = \int_{-\infty}^{\infty} f(x) g(x-t) e^{-j2x\eta(x-t)} dx, \quad (2)$$

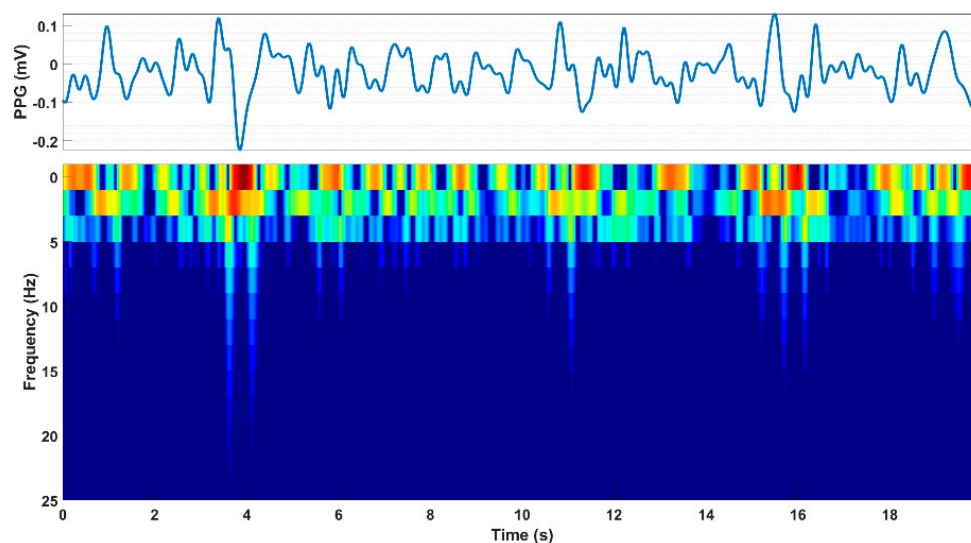
where  $e^{j2x\eta t}$  is a modulation factor. The synchrosqueezed transform is given by [29]:

$$T_g f(t, \omega) = \int_{-\infty}^{\infty} V_g f(t, \eta) \delta(\omega - \Omega_g f(t, \eta)) d\eta, \quad (3)$$

where  $\Omega_g$  is given by:

$$\Omega_g = \frac{1}{j2\pi} \frac{\partial}{\partial t} V_g f(t, \eta) / V_g f(t, \eta), \quad (4)$$

To illustrate the SSFT concept, Figure 2 shows a segment of a noisy PPG (top) and the respective SSFT (bottom), where a color code indicates the energy content along the time axis. Significant noisy components are found around 4, 11, and 16 s, extending to a frequency above 10 Hz. These features are important inputs for the classification process. The real and imaginary parts of the SSFT were divided into two different features for classification.



**Figure 2.** A signal excerpt of a noisy PPG (top) and its time-frequency representation using the SSFT (bottom). Higher frequency energy is present around 4, 11, and 16 s.

### 2.3. Proposed Models

Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs) have been used in different studies [33,34] for time-series processing and classification. Long Short-Time Memory (LSTM) and bidirectional LSTM (BiLSTM) belong to the RNN class, usually applied to time-series data processing and prediction [35,36]. CNNs have also been able to extract deep and time-independent features, while being highly noise-resistant models [37]. Due to these models' established good performance with time-series data, the LSTM, BiLSTM, and CNN were evaluated for PPG waveform detection [33,34].

Dropout layers are widely used as a regularization method [38], where some units of the layers are excluded from activation and weight updates, which will reduce the overfitting effect and improve the model performance. A dropout rate of 0.4 was used in all studied models after each LSTM/BiLSTM layer [39].

Time distributed layers were used in the evaluated models since it allows the application of the dense layer to every timestep of an, at least, three-dimensional array [40]. Instead of the typical whole signal segment processing, a sample-by-sample classification procedure was herein selected. Despite this method being computationally more demanding, it provides the prospective users the possibility of using the validated model parameters on a signal, independently of its length, within the computational capability of the used platform. Consequently, data windowing would not be necessary, thus precluding the segment length selection step, typically a key project stage.

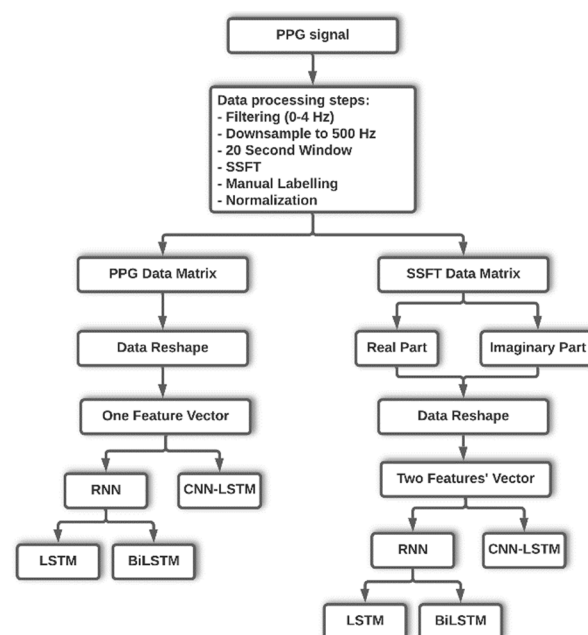
The data was divided into two sets: 70% for training and 30% for testing. A validation set of 20% was selected. Table 1 summarizes the model parameters that were implemented.

**Table 1.** Studied model parameters.

Parameters	Value
Loss Function	Categorical-cross entropy
Optimizer	Adam and SGD <sup>1</sup>
Hidden Activation Function	Sigmoid and Tanh
Dropout Rate	0.4
Learning Rate	$10^{-2}$ , $10^{-3}$ , $10^{-4}$ , $10^{-5}$ and $10^{-6}$

<sup>1</sup> SGD: Stochastic Gradient Descent.

For the RNN models, the data was reshaped. The PPG was flattened into a single column and the SSFT was flattened into two columns, corresponding to the real and imaginary part of the spectrum. A sample-by-sample classification approach was selected. For the RNN model, the input was a signal vector, whereas for the CNN case, the data input was the segments' matrix. Figure 3 represents a summary of the implemented methodology. The Deep Learning algorithms were implemented in Python, version 3.7.



**Figure 3.** Flowchart of the applied methodology.

#### 2.4. Evaluated Metrics

Evaluation of the deep learning algorithm is essential to assess the network learning process evolution and the need for model parameter adjustments. The confusion matrix, represented in Figure 4, describes the complete performance of the model, integrating the number of predictions for each class and their true values.

		Predicted class	
		0	1
Actual class	0	True Negative (TN)	False Positive (FP)
	1	False Negative (FN)	True Positive (TP)

Figure 4. Confusion matrix.

From the confusion matrix, it is possible to extract other metrics, by evaluating different relations between predicted and actual classes. Four different metrics were evaluated in this study:

1. *Accuracy* [41]: measures the correct number of predictions over  $N$ , the total number:

$$Accuracy = \frac{TN + TP}{N}, \quad (5)$$

2. *Precision* [41]: measures the number of true positives over all positive predictions:

$$Precision = TP / (TP + FP), \quad (6)$$

3. *Recall* [41]: measures the proportion of true positives that are correctly predicted as positive:

$$Recall = TP / (TP + FN), \quad (7)$$

4. *F1-Score* [42]: measures the weighted average of the precision and recall:

$$F1\ Score = 2 \cdot \frac{(Precision) \cdot (Recall)}{Precision + Recall}, \quad (8)$$

### 3. Results

For the purpose of this study, several RNN and CNN models were tested. Various parameters were manipulated to evaluate the model performance, such as the number of epochs, batch, number of neurons and layers, activation function, learning rate, and optimizer. The Adam optimizer [43] was used in the output layer. The output of the studied models was activated by the Softmax function [44], where each point corresponded to a set of two values between 0 and 1, representing the probability of a given point being classified as PPG or noise. To get the best overall metrics performance, a new variable threshold was created to adjust the probability of each point being considered in the evaluated class. Accuracy, precision, recall, and the *F1-score* of test data were computed in order to compare the different models.

#### 3.1. LSTM and BiLSTM with PPG Input

The first approach included LSTM models (Table 2). The *MinMaxScaler* function applied in this section scaled the data to be in the range between  $-1$  and  $1$ . The Adam optimizer and the Sigmoid function for the hidden layers were selected. The training accuracy reached a stable value after five iteration epochs. The accuracy and precision maximum values of  $0.749$  for both cases were achieved when the threshold was adjusted. However, despite these apparently good results, the confusion matrix revealed that the

model was biased in the sense that every signal point was classified as PPG. A possible explanation for this behavior is the imbalance of PPG and noise classes. Lowering the threshold value reduced the accuracy and precision but increased the correct number of identified zeros in the classes (Table 2).

**Table 2.** LSTM and BiLSTM models with time distributed, dense, and Softmax layers for PPG input.

Model	Number of Neurons in Each Layer		Epochs	Batch-Size	Accuracy	Precision	Recall	F1-Score
	1st	2nd						
	L-D-S	32						
L-TD(D)-S	50	-	50	40,000	0.709	0.759	0.897	0.822
L-TD(D)-S	200	-	50	40,000	0.719	0.757	0.919	0.830
L-L-TD(D)-S	8	16	50	40,000	0.724	0.755	0.934	0.835
L-L-TD(D)-S	16	32	50	40,000	0.718	0.758	0.917	0.830
L-L-TD(D)-S	32	64	50	40,000	0.720	0.757	0.922	0.831
L-L-TD(D)-S	32	64	50	20,000	0.712	0.766	0.886	0.822
L-L-TD(D)-S	256	128	100	70,000	0.733	0.760	0.940	0.840
B-TD(D)-S	50	-	50	40,000	0.744	0.751	0.982	0.851
B-TD(D)-S	200	-	50	40,000	0.730	0.756	0.965	0.848
B-B-TD(D)-S	8	16	50	40,000	0.744	0.750	0.986	0.852
B-B-TD(D)-S	16	32	50	40,000	0.740	0.751	0.976	0.849
B-B-TD(D)-S	32	64	50	40,000	0.729	0.755	0.945	0.839
B-B-TD(D)-S	32	64	50	20,000	0.739	0.752	0.973	0.848
B-B-TD(D)-S	256	128	100	70,000	0.744	0.756	0.971	0.850

L: LSTM; B: Bidirectional LSTM; TD: Time Distributed; D: Dense; S: Softmax.

The second applied method was the BiLSTM (Table 2), which provided better results than the LSTM. The most complex models under study, with two layers of 256 and 128 neurons, provided an accuracy of 0.733 and 0.744 for the LSTM and BiLSTM, respectively.

### Testing and Improvements

The results presented in Table 2 demonstrate that investing in parameter adjustment to achieve better metric outcomes is viable.

To improve the previous results, multiple approaches were tested. Firstly, a validation split was implemented. In this way, a portion of the training data was separated, called the validation dataset, to evaluate the model performance in each epoch. Due to the PPG data bias, sample weighting was introduced to minimize its effects. For weight balancing, the variable *sample\_weights* was used.

The chosen optimizer on the previously presented models was Adam's, with a 0.001 learning rate. For improvement purposes, a Stochastic Gradient Descent (SGD) optimizer and variable learning rates of  $10^{-3}$ ,  $10^{-4}$ ,  $10^{-5}$ , and  $10^{-6}$  were tested (Table 3).

Additionally, the results in Table 3 reflect a scaling range adjustment using the *MinMaxScaler*, with the parameter values ranging between 0 and 1, instead of  $-1$  and  $1$  (Table 2). In an overall analysis, it was found that this scaling range adjustment did not improve the general scoring results. Sample weighing has proven to balance the classes. However, the model still failed to correctly identify noisy signal segments, as was found upon on a detailed inspection of random signal cases. Changing the optimizer to SGD and decreasing the learning rate slightly changed the model performance. Finally, for the learning rate's tested values, the model's accuracy showed substantial differences. Given that for the same conditions the model's accuracy decreased, this may can be explained by the lower learning rate value applied to the same number of epochs. Clearly, there is an infinite degree of freedom regarding the hyperparameter selection for the models' evaluation. Tables 2 and 3 show just a limited sample of those possibilities. The selected criteria for the hyperparameters' range for these tables were based on the literature and a trial-and-error procedure. From these tables' results, the following model hyperparameters

were selected: the Adam optimizer and a  $10^{-2}$  learning rate. The main deciding factor was the accuracy value.

**Table 3.** Improved LSTM models with validation split (0.2), different optimizers, and learning rates for PPG input.

Model	Optimizer		Number of Neurons in Each Layer			Weights	Epochs	Batch-Size	Accuracy	Precision	Recall	F1-Score
	Opt.	LR	1st	2nd	3rd							
L-L-TD(D)-S	A	$10^{-3}$	16	32	-	-	20	40,000	0.627	0.771	0.715	0.742
L-L-TD(D)-S	A	$10^{-3}$	16	32	-	SW	20	40,000	0.551	0.796	0.539	0.643
L-L-TD(D)-S	A	$10^{-3}$	16	32	-	SW	20	8000	0.574	0.796	0.581	0.672
L-L-TD(D)-S	A	$10^{-3}$	16	32	-	SW	200	1000	0.612	0.788	0.659	0.718
L-L-TD(D)-S	SGD	$10^{-4}$	32	64	-	SW	50	40,000	0.749	0.749	1.000	0.856
L-L-TD(D)-S	SGD	$10^{-5}$	32	64	-	SW	20	40,000	0.749	0.749	1.000	0.856
L-L-TD(D)-S	SGD	$10^{-5}$	32	64	-	SW	50	40,000	0.614	0.741	0.745	0.743
L-L-TD(D)-S	SGD	$10^{-6}$	32	64	-	SW	50	40,000	0.749	0.749	1.000	0.856
L-L-L-TD(D)-S	A	$10^{-3}$	16	32	16	-	20	20,000	0.749	0.749	1.000	0.856
L-L-L-TD(D)-S	A	$10^{-3}$	32	64	32	-	20	20,000	0.749	0.749	1.000	0.856
L-L-L-TD(D)-S	A	$10^{-3}$	64	128	64	-	20	20,000	0.749	0.749	1.000	0.856
L-L-L-TD(D)-S	SGD	$10^{-4}$	32	64	32	SW	50	40,000	0.730	0.754	0.949	0.840
L-L-L-TD(D)-S	SGD	$10^{-5}$	32	64	32	SW	50	20,000	0.563	0.738	0.647	0.690
L-L-L-TD(D)-S	SGD	$10^{-5}$	32	64	32	SW	50	40,000	0.603	0.728	0.750	0.739
L-L-L-TD(D)-S	SGD	$10^{-6}$	32	64	32	SW	50	40,000	0.668	0.741	0.855	0.794

L: LSTM; TD: Time Distributed; D: Dense; S: Softmax; SW: *sample\_weights*; LR: Learning Rate; A: Adam; SGD: Stochastic Gradient Descent.

The next step was to select the activation function of the hidden layers, comparing the models' results. The activation functions under study were the Tanh and Sigmoid. Table 4 shows the results for the BiLSTM models. An improvement of the accuracy (0.744 to 0.745) and precision (0.756 to 0.757) was obtained between Tables 2 and 4, respectively. Therefore, the selected hidden activation function was Tanh for the PPG input models.

**Table 4.** Improved BiLSTM models with a validation split of 0.2 and different hidden activation functions for PPG input.

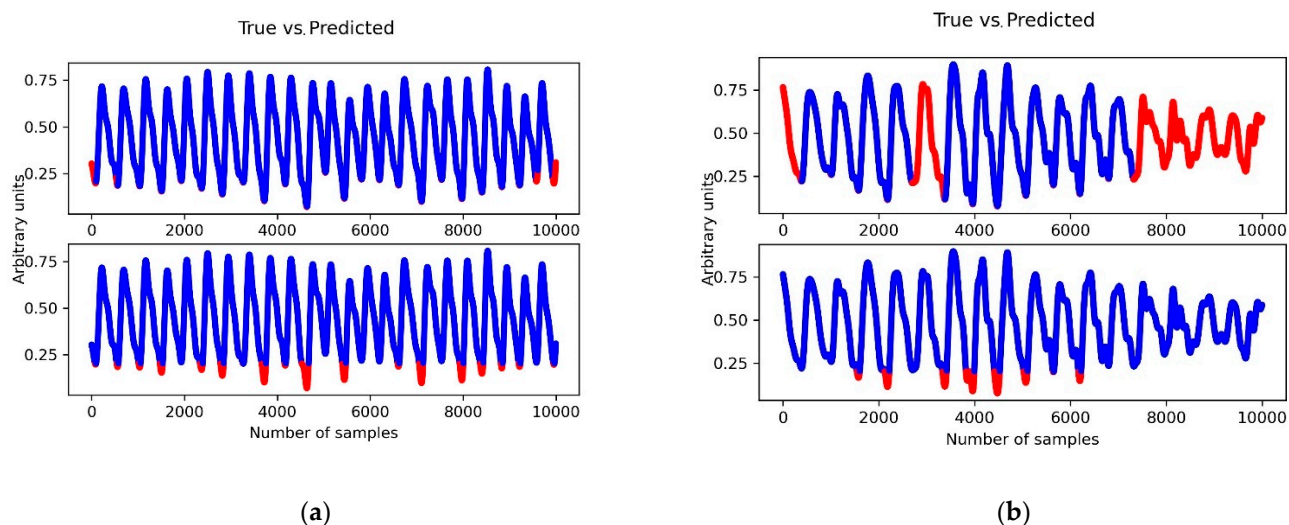
Model	Hidden Activation Function	Number of Neurons in Each Layer			Weights	Epochs	Batch-Size	Accuracy	Precision	Recall	F1-Score
		1st	2nd	3rd							
B-B-TD(D)-S	Sigmoid	50	-	-	SW	100	40,000	0.737	0.753	0.966	0.846
B-B-TD(D)-S	Sigmoid	200	-	-	SW	100	40,000	0.691	0.762	0.854	0.805
B-B-TD(D)-S	Sigmoid	16	32	-	SW	100	40,000	0.717	0.766	0.896	0.826
B-B-TD(D)-S	Sigmoid	16	32	-	-	10	40,000	0.683	0.744	0.881	0.807
B-B-TD(D)-S	Tanh	16	32	-	-	5	40,000	0.691	0.745	0.894	0.813
B-B-TD(D)-S	Tanh	16	32	-	-	10	40,000	0.731	0.749	0.964	0.843
B-B-TD(D)-S	Tanh	16	32	-	SW	20	20,000	0.737	0.752	0.969	0.847
B-B-TD(D)-S	Tanh	16	32	-	SW	10	40,000	0.713	0.759	0.903	0.825
B-B-TD(D)-S	Sigmoid	32	64	-	SW	100	40,000	0.730	0.761	0.931	0.837
B-B-TD(D)-S	Sigmoid	32	64	-	-	10	40,000	0.703	0.746	0.913	0.821
B-B-TD(D)-S	Tanh	32	64	-	-	10	40,000	0.599	0.765	0.671	0.715
B-B-TD(D)-S	Tanh	32	64	-	SW	40	40,000	0.727	0.756	0.939	0.838
B-B-TD(D)-S	Tanh	64	128	-	SW	100	20,000	0.739	0.758	0.957	0.846
B-B-B-TD(D)-S	Tanh	16	32	16	SW	40	20,000	0.708	0.766	0.878	0.818
B-B-B-TD(D)-S	Tanh	16	32	16	-	100	15,000	0.749	0.749	1.000	0.856
B-B-B-TD(D)-S	Tanh	16	32	16	SW	100	15,000	0.729	0.762	0.928	0.837
B-B-B-TD(D)-S	Tanh	64	128	64	SW	100	20,000	0.745	0.757	0.965	0.848
B-B-B-TD(D)-S	Sigmoid	64	128	64	SW	100	20,000	0.733	0.754	0.955	0.843
B-B-B-TD(D)-S	Sigmoid	64	128	64	-	100	20,000	0.743	0.757	0.968	0.850

B: Bidirectional LSTM; TD: Time Distributed; D: Dense; S: Softmax; SW: *sample\_weights*.

A classification example from the best performing BiLSTM model is presented in Figure 5. Two different subject cases are shown on the left Figure 5a and right Figure 5b. On the top plot, the blue and red sample points represent expert classified valid PPG and noise values, respectively. On the bottom plot, blue and red sample points stand for the model-predicted PPG and noise, respectively. The model was able to identify signal minimums in the PPG waveform with and without noise (bottom plots Figure 5a,b). However, for the



portrayed signal, the model could not detect the noise portions, as shown in Figure 5b. It should be noted that red sample points may not be visible due to sample clustering.



**Figure 5.** True labeling versus predicted labeling of PPG signal of BiLSTM models with PPG input: (a) well-defined PPG waves with no noise; (b) PPG signal affected by noise. The red and blue dots represent labels 0 and 1, respectively. Top plots represent expert classified data and bottom plots the result of the model classification. More details to be found in the main text.

### 3.2. BiLSTM with SSFT Input

An SSFT approach was also considered in this work, as mentioned in Section 2.2. Table 5 shows the different BiLSTM models that were tested with an SSFT input. The Adam optimizer and a  $10^{-2}$  learning rate were selected. The evaluated parameters included the hidden activation function and the number of layers, likewise, as shown in Table 4. Sample weighting was applied to all models. The best performance for accuracy and precision were 0.736 and 0.764, respectively, for the BiLSTM with three layers and the Tanh hidden activation function. The PPG input best performance case (Table 4) outperformed the SSFT input accuracy, with 0.745 and 0.736, respectively.

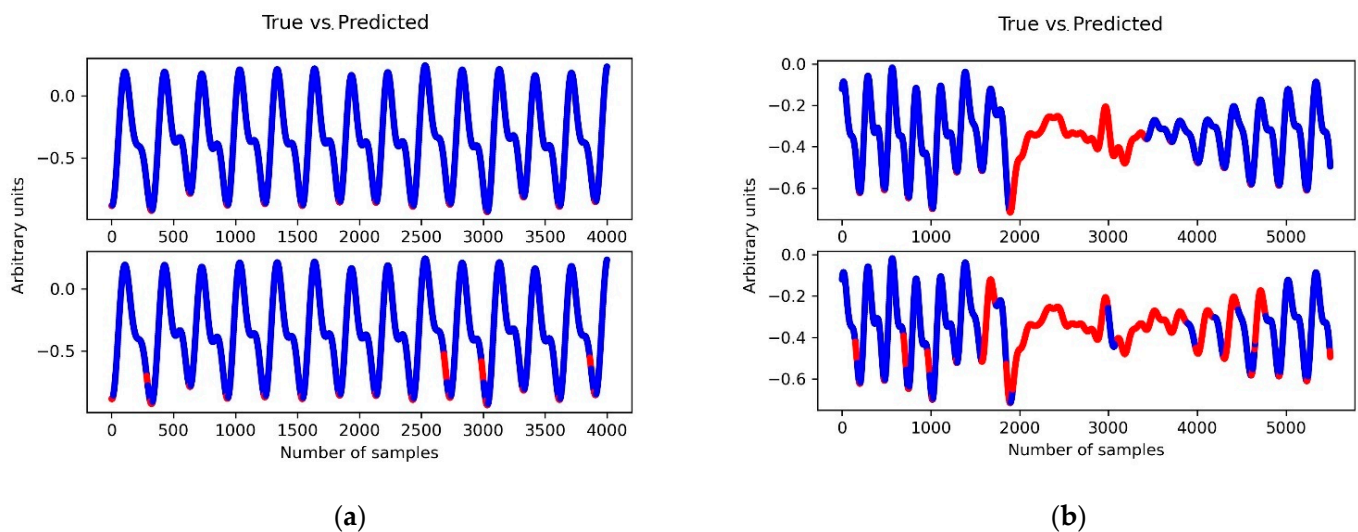
**Table 5.** BiLSTM models with an SSFT input, validation split of 0.2, sample weighting, and different activation functions.

Model	Hidden Activation Function	Number of Neurons in Each Layer			Epochs	Batch-Size	Accuracy	Precision	Recall	F1-Score
		1st	2nd	3rd						
B-B-TD(D)-S	Sigmoid	16	32	-	100	40,000	0.723	0.742	0.961	0.837
B-B-TD(D)-S	Tanh	16	32	-	100	40,000	0.734	0.770	0.847	0.807
B-B-TD(D)-S	Sigmoid	32	64	-	100	20,000	0.713	0.791	0.762	0.776
B-B-TD(D)-S	Tanh	32	64	-	100	20,000	0.713	0.791	0.763	0.777
B-B-TD(D)-S	Tanh	256	512	-	10	20,000	0.735	0.767	0.855	0.809
B-B-B-TD(D)-S	Sigmoid	16	32	16	100	20,000	0.724	0.784	0.798	0.791
B-B-B-TD(D)-S	Tanh	16	32	16	100	20,000	0.735	0.768	0.855	0.809
B-B-B-TD(D)-S	Sigmoid	32	64	32	100	20,000	0.734	0.769	0.849	0.807
B-B-B-TD(D)-S	Tanh	32	64	32	100	20,000	0.736	0.764	0.862	0.810
B-B-B-TD(D)-S	Tanh	256	512	256	10	20,000	0.732	0.772	0.846	0.807

B: Bidirectional LSTM; TD: Time Distributed; D: Dense; S: Softmax.

Figure 6 represents an example of true and predicted labeling with the best performing BiLSTM model with an SSFT input. Regarding the color code and figure organization, please refer to Figure 5. The model was able to identify signal minimums in Figure 6a,b despite red dots not always being visible due to sample clustering. In Figure 6a, the signal was correctly classified as PPG except for four short segments. All the signal minimums were correctly detected (better verifiable in Figure 6). In Figure 6b, the expert labeled noise

(top) was not completely classified as such by the algorithm (bottom), despite the overall acceptable noise classification.



**Figure 6.** True labeling versus predicted labeling of PPG signal of BiLSTM models with SSFT input: (a) well-defined PPG waves with no noise; (b) PPG signal affected by noise. The red and blue dots represent labels 0 and 1, respectively. The top plots represent expert classified data and the bottom plots the result of the model classification. More details are to be found in the main text.

### 3.3. CNN-LSTM with PPG Input

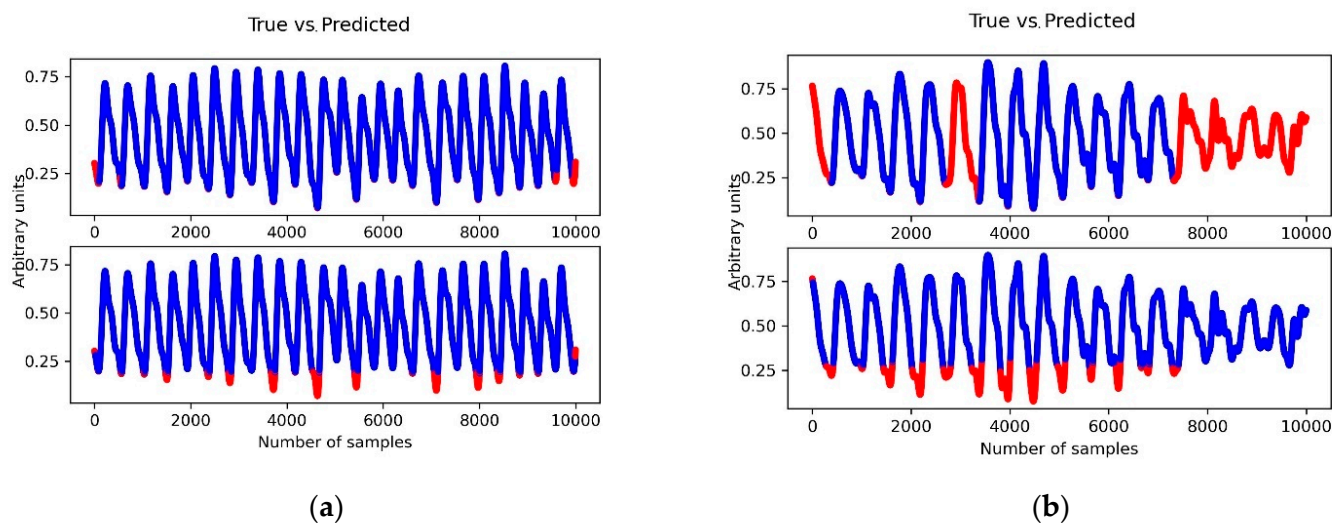
A model with one 1D convolution layer followed by a MaxPool1D [45], a Bidirectional LSTM layer, and a LSTM layer was tested against the previous RNN models (Table 6). The used structure was similar to the one described in Azar et al. [34]. On the convolutional, BiLSTM, and LSTM layers, the Tanh and Softmax activation were used in the hidden and output layers, respectively. The Adam optimizer was used with a learning rate of  $10^{-2}$ . Categorical cross-entropy was also tested as the loss function and compared with the Mean Squared Error (MSE), following the methodology applied in [34]. The MinMaxScaler function applied in this section scaled the data between 0 and 1.

**Table 6.** CNN-LSTM models with a PPG input, validation split of 0.2, Sample Weighting, Tanh activation function, and different loss functions.

Model	Loss Function	Number of Neurons in Each Layer			Epochs	Batch-Size	Accuracy	Precision	Recall	F1-Score
		1st	2nd	3rd						
C-MP-B-L-TD(D)	MSE	16	8	4	10	60	0.669	0.742	0.857	0.795
C-MP-B-L-TD(D)	MSE	32	16	8	10	60	0.646	0.740	0.813	0.775
C-MP-B-L-TD(D)	Categorical	16	8	4	10	60	0.619	0.741	0.756	0.748
C-MP-B-L-TD(D)	Categorical	32	16	8	10	60	0.670	0.742	0.856	0.795
C-MP-B-L-TD(D)	MSE	16	8	4	30	60	0.719	0.749	0.939	0.833
C-MP-B-L-TD(D)	MSE	32	16	8	30	60	0.679	0.745	0.870	0.803
C-MP-B-L-TD(D)	Categorical	16	8	4	30	60	0.675	0.744	0.864	0.800
C-MP-B-L-TD(D)	Categorical	32	16	8	30	60	0.657	0.742	0.831	0.784

C: Conv1D; MP: MaxPool1D; L: LSTM; B: Bidirectional LSTM; TD: Time Distributed; D: Dense; MSE: Mean Squared Error.

The best performing model had a 0.719 accuracy, which is lower than the previously achieved results described in Table 4. Figure 7 shows the model classification in the PPG signal. Regarding the color code, figure organization, and clarifications, please refer to Figure 5.



**Figure 7.** True labeling versus predicted labeling of PPG signal of CNN-LSTM models with PPG input: (a) well-defined PPG waves with no noise; (b) PPG signal affected by noise. The red and blue dots represent labels 0 and 1, respectively. The top plots represent expert classified data and the bottom plots the result of the model classification. More details are found in the main text.

This model amplified the signal minimums, as shown in Figure 7a, but was not able to detect PPG noise portions, as shown in Figure 7b. Note that the sixth PPG beat was expert classified as noise (Figure 7a) because it contained two peaks in the maximum interval region. This is not visible in the figure due to sample clustering. This could be visible by zooming in on the figure.

### 3.4. CNN-LSTM with SSFT Input

Another approach tested a CNN-LSTM network with an SSFT input (Table 7). The structure was similar to the one described in Section 3.2, with different neuron numbers in each layer. The time distributed output layer was followed by a dense layer with two neurons. The hidden layers had a Tanh activation function, and the output activation function was Softmax since the labels were in a categorical format. The loss function used was categorical cross-entropy.

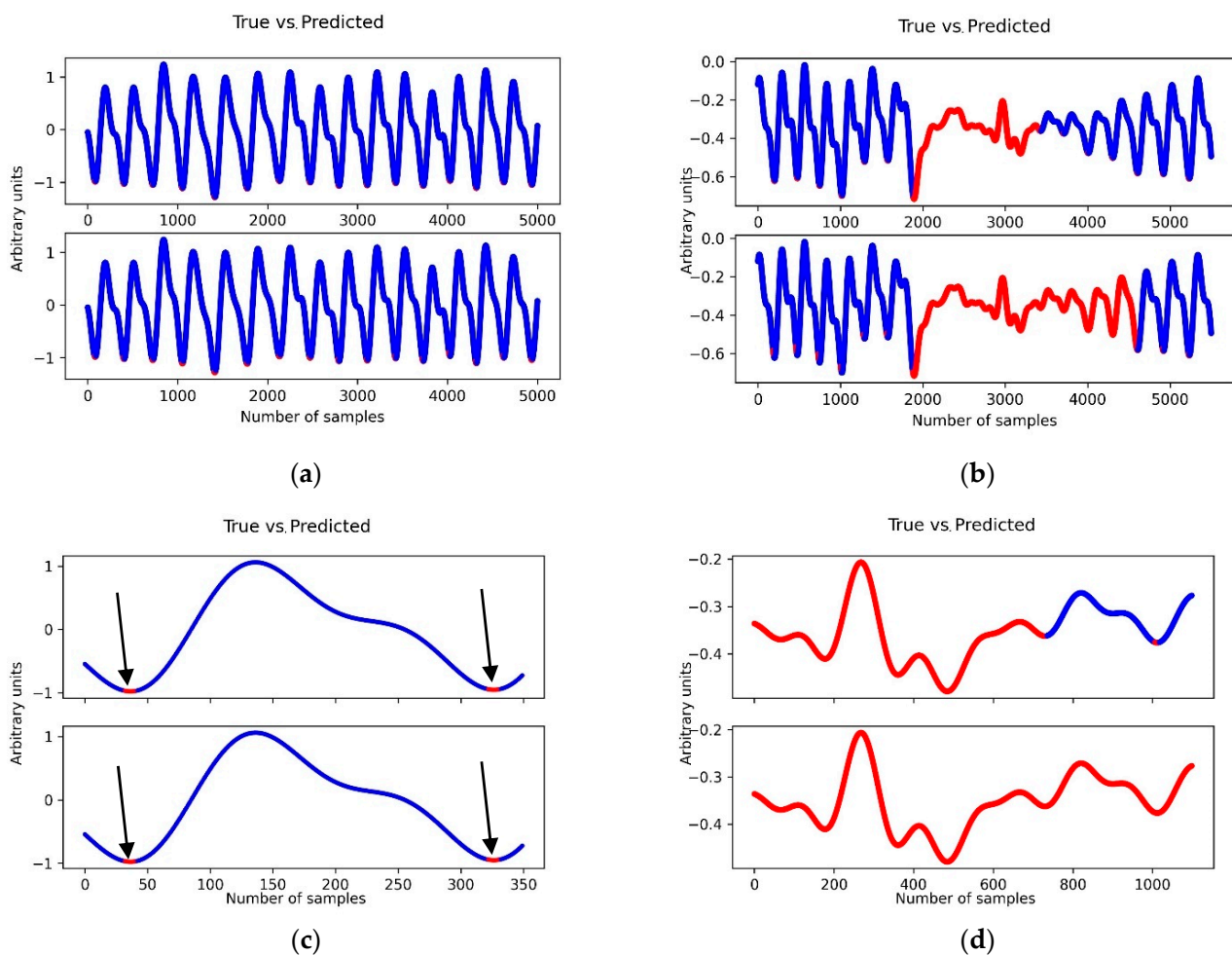
**Table 7.** CNN-LSTM models with an SSFT input, validation split of 0.2, and Tanh activation function.

Model	Number of Neurons in Each Layer			Weights	Epochs	Batch-Size	Accuracy	Precision	Recall	F1-Score
	1st	2nd	3rd							
C-MP-B-L-TD(D)	32	64	32	-	20	30	0.800	0.801	0.907	0.851
C-MP-B-L-TD(D)	32	64	32	SW	20	30	0.752	0.792	0.841	0.816
C-MP-B-L-TD(D)	32	64	32	-	50	100	0.771	0.788	0.889	0.835
C-MP-B-L-TD(D)	32	64	32	-	100	50	0.804	0.805	0.925	0.861
C-MP-B-L-TD(D)	64	128	64	-	200	50	0.894	0.923	0.914	0.918
C-MP-B-L-TD(D)	256	64	48	-	20	30	0.800	0.810	0.907	0.856
C-MP-B-L-TD(D)	256	128	64	SW	20	30	0.787	0.813	0.877	0.844

C: Conv1D; MP: MaxPool1D; L: LSTM; B: Bidirectional LSTM; TD: Time Distributed; D: Dense; SW: *sample\_weights*.

From Table 7, it is deduced that the best accuracy result was achieved for the model with hyperparameters represented in the fifth line. It turns out that this was the best performing model among all the ones presented in this work, with accuracy, precision, and recall of 0.894, 0.923, and 0.914, respectively. A classification example from the best performing CNN-LSTM model is presented in Figure 8. This model was able to segment

each PPG wave correctly, as shown in Figure 8a, and detected the noise with minimum leakage to the PPG waves, as shown in Figure 8b.



**Figure 8.** True labeling versus predicted labeling of PPG signal of CNN-LSTM models with SSFT input: (a) well-defined PPG waves without noise; (b) PPG signal affected by noise. The red and blue dots represent labels 0 and 1, respectively; (c) zoom of a PPG beat in the (a) plot, where the arrows represent the classified local minimums; (d) zoom of the ending noise portion of the plot (b) followed by a PPG beat. The top plots represent expert classified data and the bottom plots the result of the model classification. More details are found in the main text.

#### 4. Discussion and Conclusions

As far as the PPG signal classification is concerned, LSTM networks are one of the most successful architectures in the detection of patterns in time-series data [33]. Different methodologies based on these networks were herein tested. The best results achieved for each architecture are represented in Table 8. The first approach included RNN, where LSTM and Bidirectional LSTM neural networks were explored. The herein selected sample-by-sample classification method was a different approach relative to that currently found in the literature. The advantage of this new method is referred to in Section 2.3. The BiLSTM networks learnt to accurately identify signal minimums, as shown in Figure 5, and achieved an accuracy of 0.745 and recall of 0.965. However, this model did not detect most of the noise regions.

**Table 8.** Best performing models for each studied architecture.

Model	Number of Neurons in Each Layer			Data Input	Epochs	Batch-Size	Accuracy	Precision	Recall	F1-Score
	1st	2nd	3rd							
B-B-B-TD(D)-S	64	128	64	PPG	100	20,000	0.745	0.757	0.965	0.848
B-B-B-TD(D)-S	32	64	32	SSFT	100	20,000	0.736	0.764	0.862	0.810
C-MP-B-L-TD(D)	16	8	4	PPG	30	60	0.719	0.749	0.939	0.833
C-MP-B-L-TD(D)	64	128	64	SSFT	200	50	0.894	0.923	0.914	0.918

C: Conv1D; MP: MaxPool1D; L: LSTM; B: Bidirectional LSTM; TD: Time Distributed; D: Dense; S: Softmax.

For the SSTF transformed PPG signals, the BiLSTM was also tested. Applying a time-frequency transform to the signals before classification provided the model with an increased feature set. This extended data pool also corresponds to a signal projection from the time to the time-frequency domain, where non-stationary components may be better represented. With this approach, the model reached up to an accuracy of 0.736 and recall of 0.862, results that are inferior to the previously mentioned case. However, these models were the only ones with an LSTM-based architecture able to identify noisy regions beyond the signal minimums, as depicted in Figure 6b. These results imply that the RNN-based methods' classification performance could be improved by using a different set of hyperparameters. Further work on this task is expected to be done.

Regarding the CNN-LSTM approach for the PPG data, it correctly identified signal minimums, but it was not able to detect most noise regions, as shown in Figure 7b. However, when the CNN-LSTM was implemented with the SSFT, the best overall results were achieved. The best performing model had an accuracy, precision, and recall of 0.894, 0.923, and 0.914, respectively. With this model, signal minimums were mostly correctly identified, as well as some noisy regions. These results show good agreement with the one presented by Azar et al. [34], where the achieved precision and recall were 0.90 and 0.95, respectively, for a similar CNN-LSTM model with a windowed PPG signal as model input. This comparison has to take into account that different databases were used.

The main goal of this work was to create a Deep Learning Neural Network to detect PPG waveforms with different noise levels. Most of the tested networks were able to detect the signal minimums in order to segment each PPG waveform. However, only models with a time-frequency input could identify with improved accuracy both noise and the signal minimums. The time-frequency transform seems to be a promising tool to be used as a deep learning feature generator, given the herein obtained results. In future work, different model architectures and hyperparameters could be explored. The CNN-LSTM with different time-frequency representations as input, such as the continuous and discrete wavelet transforms, may be tested. Empirical mode decomposition applied to the PPG could also provide a significant data pool for classification.

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

**Funding:** This research was funded by the Fundação para a Ciência e Tecnologia (FCT, Lisbon, Portugal) and NMT, S.A in the scope of the PhD grant PD/BDE/150312/2019 and by FCT within the scope of the CTS Research Unit—Center of Technology and Systems—UNINOVA, under the project UIDB/00066/2020 (FCT).

**Institutional Review Board Statement:** The study was conducted according to the guidelines of the Declaration of Helsinki, and approved by the Ethics Committee of the Hospital da Senhora da Oliveira—Guimarães EPE (protocol code 86/2019 of 6 December 2019).

**Informed Consent Statement:** Informed consent was obtained from all subjects involved in the study.

**Data Availability Statement:** Data is contained within the article.

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

## References

1. Allen, J. Photoplethysmography and its application in clinical physiological measurement. *Physiol. Meas.* **2007**, *28*, R1. [[CrossRef](#)]
2. Elgendi, M. On the Analysis of Fingertip Photoplethysmogram Signals. *Curr. Cardiol. Rev.* **2012**, *8*, 14–25. [[CrossRef](#)]
3. Ram, M.R.; Madhav, K.V.; Krishna, E.H.; Komalla, N.R.; Reddy, K.A. A Novel Approach for Motion Artifact Reduction in PPG Signals Based on AS-LMS Adaptive Filter. *IEEE Trans. Instrum. Meas.* **2012**, *61*, 1445–1457. [[CrossRef](#)]
4. Jang, D.-G.; Park, S.; Hahn, M.; Park, S.-H. A Real-Time Pulse Peak Detection Algorithm for the Photoplethysmogram. *Int. J. Electron. Electr. Eng.* **2014**, 45–49. [[CrossRef](#)]
5. Argüello-Prada, E.J. The mountaineer’s method for peak detection in photoplethysmographic signals. *Rev. Fac. Ing. Univ. Antioquia* **2019**, *90*, 42–50. [[CrossRef](#)]
6. Vadrevu, S.; Manikandan, M.S. A Robust Pulse Onset and Peak Detection Method for Automated PPG Signal Analysis System. *IEEE Trans. Instrum. Meas.* **2019**, *68*, 807–817. [[CrossRef](#)]
7. Siontis, K.C.; Noseworthy, P.A.; Attia, Z.I.; Friedman, P.A. Artificial intelligence-enhanced electrocardiography in cardiovascular disease management. *Nat. Rev. Cardiol.* **2021**, *18*, 465–478. [[CrossRef](#)] [[PubMed](#)]
8. Tamura, T. Current progress of photoplethysmography and SPO<sub>2</sub> for health monitoring. *Biomed. Eng. Lett.* **2019**, *9*, 21–36. [[CrossRef](#)]
9. Cardoso, F.E.; Vassilenko, V.; Batista, A.; Bonifácio, P.; Martin, S.R.; Muñoz-Torrero, J.; Ortigueira, M. Improvements on Signal Processing Algorithm for the VOPITB Equipment. In Proceedings of the DoCEIS: Doctoral Conference on Computing, Electrical and Industrial Systems, Costa de Caparica, Portugal, 7–9 July 2021; pp. 324–330. [[CrossRef](#)]
10. Ribeiro, A.H.; Ribeiro, M.H.; Paixão, G.M.; Oliveira, D.M.; Gomes, P.R.; Canazart, J.A.; Ferreira, M.P.; Andersson, C.R.; Macfarlane, P.W.; Wagner, M., Jr.; et al. Automatic diagnosis of the 12-lead ECG using a deep neural network. *Nat. Commun.* **2020**, *11*, 1760. [[CrossRef](#)] [[PubMed](#)]
11. Hannun, A.Y.; Rajpurkar, P.; 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](#)] [[PubMed](#)]
12. Soltane, M.; Ismail, M.; Rashid, Z.A.A. Artificial Neural Networks (ANN) Approach to PPG Signal Classification. *Int. J. Comput. Inf. Sci.* **2004**, *2*, 58.
13. Liu, S.-H.; Li, R.-X.; Wang, J.-J.; Chen, W.; Su, C.-H. Classification of Photoplethysmographic Signal Quality with Deep Convolution Neural Networks for Accurate Measurement of Cardiac Stroke Volume. *Appl. Sci.* **2020**, *10*, 4612. [[CrossRef](#)]
14. Yen, C.-T.; Chang, S.-N.; Liao, C.-H. Deep learning algorithm evaluation of hypertension classification in less photoplethysmography signals conditions. *Meas. Control* **2021**, *54*, 439–445. [[CrossRef](#)]
15. Song, S.B.; Nam, J.W.; Kim, J.H. NAS-PPG: PPG-Based Heart Rate Estimation Using Neural Architecture Search. *IEEE Sens. J.* **2021**, *21*, 14941–14949. [[CrossRef](#)]
16. Alessandrini, M.; Biagetti, G.; Crippa, P.; Falaschetti, L.; Turchetti, C. Recurrent Neural Network for Human Activity Recognition in Embedded Systems Using PPG and Accelerometer Data. *Electronics* **2021**, *10*, 1715. [[CrossRef](#)]
17. Li, Y.-H.; Harfiya, L.N.; Purwandari, K.; Lin, Y.-D. Real-Time Cuffless Continuous Blood Pressure Estimation Using Deep Learning Model. *Sensors* **2020**, *20*, 5606. [[CrossRef](#)]
18. Laitala, J.; Jiang, M.; Syrjälä, E.; Naeini, E.K.; Airola, A.; Rahmani, A.M.; Dutt, N.D.; Liljeberg, P. Robust ECG R-peak detection using LSTM. In Proceedings of the 35th Annual ACM Symposium on Applied Computing, Brno, Czech Republic, 30 March–3 April 2020; pp. 1104–1111. [[CrossRef](#)]
19. Kim, B.-H.; Pyun, J.-Y. ECG Identification for Personal Authentication Using LSTM-Based Deep Recurrent Neural Networks. *Sensors* **2020**, *20*, 3069. [[CrossRef](#)] [[PubMed](#)]
20. Malali, A.; Hiriyannaiah, S.; Siddesh, G.M.; Srinivasa, K.G.; Sanjay, N.T. Supervised ECG wave segmentation using convolutional LSTM. *ICT Express* **2020**, *6*, 166–169. [[CrossRef](#)]
21. Liang, Y.; Yin, S.; Tang, Q.; Zheng, Z.; Elgendi, M.; Chen, Z. Deep Learning Algorithm Classifies Heartbeat Events Based on Electrocardiogram Signals. *Front. Physiol.* **2020**, *11*, 569050. [[CrossRef](#)]
22. Ruffini, G.; Ibañez, D.; Castellano, M.; Dubreuil-Vall, L.; Soria-Frisch, A.; Postuma, R.; Gagnon, J.-F.; Montplaisir, J. Deep Learning with EEG Spectrograms in Rapid Eye Movement Behavior Disorder. *Front. Neurol.* **2019**, *10*, 806. [[CrossRef](#)]
23. Boashash, B. *Time-Frequency Signal Analysis and Processing*; Academic Press: London, UK, 2016.
24. Alam, M.Z.; Rahman, M.S.; Parvin, N.; Sobhan, M.A. Time-frequency representation of a signal through non-stationary multipath fading channel. In Proceedings of the 2012 International Conference on Informatics, Electronics & Vision (ICIEV), Dhaka, Bangladesh, 18–19 May 2012; pp. 1130–1135. [[CrossRef](#)]
25. Xu, Y.; Zhang, S.; Cao, Z.; Chen, Q.; Xiao, W. Extreme Learning Machine for Heartbeat Classification with Hybrid Time-Domain and Wavelet Time-Frequency Features. *J. Healthc. Eng.* **2021**, *2021*, 6674695. [[CrossRef](#)] [[PubMed](#)]

26. Allen, J.; Murray, A. Effects of filtering on multi-site photoplethysmography pulse waveform characteristics. In Proceedings of the Computers in Cardiology, Chicago, IL, USA, 19–22 September 2004; pp. 485–488. [\[CrossRef\]](#)
27. Béres, S.; Hejmel, L. The minimal sampling frequency of the photoplethysmogram for accurate pulse rate variability parameters in healthy volunteers. *Biomed. Signal Process. Control* **2021**, *68*, 102589. [\[CrossRef\]](#)
28. Gasparini, F.; Grossi, A.; Bandini, S. A Deep Learning Approach to Recognize Cognitive Load using PPG Signals. In Proceedings of the 14th Pervasive Technologies Related to Assistive Environments Conference, Corfu, Greece, 29 June–2 July 2021; pp. 489–495. [\[CrossRef\]](#)
29. Auger, F.; Flandrin, P.; Lin, Y.-T.; McLaughlin, S.; Meignen, S.; Oberlin, T.; Wu, H.-T. Time-Frequency Reassignment and Synchrosqueezing: An Overview. *IEEE Signal Process. Mag.* **2013**, *30*, 32–41. [\[CrossRef\]](#)
30. Salamon, J.; Bello, J.P. Deep Convolutional Neural Networks and Data Augmentation for Environmental Sound Classification. *IEEE Signal Process. Lett.* **2017**, *24*, 279–283. [\[CrossRef\]](#)
31. Daubechies, I.; Lu, J.; Wu, H.-T. Synchrosqueezed wavelet transforms: An empirical mode decomposition-like tool. *Appl. Comput. Harmon. Anal.* **2011**, *30*, 243–261. [\[CrossRef\]](#)
32. Thakur, G.; Wu, H.-T. Synchrosqueezing-Based Recovery of Instantaneous Frequency from Nonuniform Samples. *SIAM J. Math. Anal.* **2011**, *43*, 2078–2095. [\[CrossRef\]](#)
33. Saadatnejad, S.; Oveisi, M.; Hashemi, M. LSTM-Based ECG Classification for Continuous Monitoring on Personal Wearable Devices. *IEEE J. Biomed. Health Inform.* **2020**, *24*, 515–523. [\[CrossRef\]](#) [\[PubMed\]](#)
34. Azar, J.; Makhoul, A.; Couturier, R.; Demerjian, J. Deep recurrent neural network-based autoencoder for photoplethysmogram artifacts filtering. *Comput. Electr. Eng.* **2021**, *92*, 107065. [\[CrossRef\]](#)
35. Hu, J.; Wang, X.; Zhang, Y.; Zhang, D.; Zhang, M.; Xue, J. Time Series Prediction Method Based on Variant LSTM Recurrent Neural Network. *Neural Process. Lett.* **2020**, *52*, 1485–1500. [\[CrossRef\]](#)
36. Schuster, M.; Paliwal, K.K. Bidirectional recurrent neural networks. *IEEE Trans. Signal Process.* **1997**, *45*, 2673–2681. [\[CrossRef\]](#)
37. Zhao, B.; Lu, H.; Chen, S.; Liu, J.; Wu, D. Convolutional neural networks for time series classification. *J. Syst. Eng. Electron.* **2017**, *28*, 162–169. [\[CrossRef\]](#)
38. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
39. Fredriksson, D.; Glandberger, O. Neural Network Regularization for Generalized Heart Arrhythmia Classification. Master's Thesis, Blekinge Institute of Technology, Karlshamn, Sweden, 2020.
40. Singh, A.; Saimbhi, A.S.; Singh, N.; Mittal, M. DeepFake Video Detection: A Time-Distributed Approach. *SN Comput. Sci.* **2020**, *1*, 212. [\[CrossRef\]](#)
41. Powers, D.M.W. Evaluation: From precision, recall and F-measure to ROC, informedness, markedness & correlation. *J. Mach. Learn. Technol.* **2011**, *2*, 37–63. [\[CrossRef\]](#)
42. Lipton, Z.C.; Elkan, C.; Naryanaswamy, B. Optimal Thresholding of Classifiers to Maximize F1 Measure. In Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Nancy, France, 14–18 September 2014; pp. 225–239. [\[CrossRef\]](#)
43. Kingma, D.P.; Ba, J.L. Adam: A method for stochastic optimization. In Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, 7–9 May 2015; pp. 1–15.
44. Nwankpa, C.; Ijomah, W.; Gachagan, A.; Marshall, S. Activation Functions: Comparison of trends in Practice and Research for Deep Learning. *arXiv* **2018**, arXiv:1811.03378.
45. Sun, W.; Zhao, H.; Jin, Z. A facial expression recognition method based on ensemble of 3D convolutional neural networks. *Neural Comput. Appl.* **2019**, *31*, 2795–2812. [\[CrossRef\]](#)