





Article

Classifying Cardiac Arrhythmia from ECG Signal Using 1D CNN Deep Learning Model

Adel A. Ahmed ^{1,*} , Waleed Ali ¹ , Talal A. A. Abdullah ²  and Sharaf J. Malebary ¹ ¹ Information Technology Department, Faculty of Computing and Information Technology-Rabigh, King Abdulaziz University, Jeddah 25729, Saudi Arabia² Computer & Information Sciences Department, Universiti Teknologi PETRONAS, Seri Iskandar 32610, Malaysia

* Correspondence: aaaabdullah1@kau.edu.sa; Tel.: +966-563884738

Abstract: Blood circulation depends critically on electrical activation, where any disturbance in the orderly pattern of the heart's propagating wave of excitation can lead to arrhythmias. Diagnosis of arrhythmias using electrocardiograms (ECG) is widely used because they are a fast, inexpensive, and non-invasive tool. However, the randomness of arrhythmic events and the susceptibility of ECGs to noise leads to misdiagnosis of arrhythmias. In addition, manually diagnosing cardiac arrhythmias using ECG data is time-intensive and error-prone. With better training, deep learning (DL) could be a better alternative for fast and automatic classification. The present study introduces a novel deep learning architecture, specifically a one-dimensional convolutional neural network (1D-CNN), for the classification of cardiac arrhythmias. The model was trained and validated with real and noise-attenuated ECG signals from the MIT-BIH dataset. The main aim is to address the limitations of traditional electrocardiograms (ECG) in the diagnosis of arrhythmias, which can be affected by noise and randomness of events, leading to misdiagnosis and errors. To evaluate the model performance, the confusion matrix is used to calculate the model accuracy, precision, recall, f1 score, average and AUC-ROC. The experiment results demonstrate that the proposed model achieved outstanding performance, with 1.00 and 0.99 accuracies in the training and testing datasets, respectively, and can be a fast and automatic alternative for the diagnosis of arrhythmias.

Keywords: cardiac arrhythmia; deep learning; electrocardiogram; classification; CNN**MSC:** 68T01; 68M10

Citation: Ahmed, A.A.; Ali, W.; Abdullah, T.A.A.; Malebary, S.J. Classifying Cardiac Arrhythmia from ECG Signal Using 1D CNN Deep Learning Model. *Mathematics* **2023**, *11*, 562. <https://doi.org/10.3390/math11030562>

Academic Editors: Yazhou Yao and Xiaoshui Huang

Received: 30 December 2022

Revised: 15 January 2023

Accepted: 18 January 2023

Published: 20 January 2023



Copyright: © 2023 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

Globally, cardiovascular diseases (CVDs) have the highest mortality rate [1]. The World Health Organization (WHO) identifies cardiovascular diseases (CVDs) as the leading cause of death worldwide, accounting for an estimated 31% of all fatalities annually [2]. The Middle East, Asia, and Russia have much higher rates than the rest of the globe [3,4]. The Cardiovascular electrical system has three major categories, including electrical (cardiac arrhythmia), circulatory (blood vessel abnormality), and structural [5]. Cardiac arrhythmia refers to a collection of irregular heartbeats caused by a malfunctioning heart's electrical system.

Electrocardiogram (ECG) is a well-known diagnostic technique for cardiac arrhythmias that documents physiological heart activity throughout time [6]. According to a recent estimate, the annual global number of ECG recordings exceeds 300 million and is expected to increase in the future [7], and the number continues to rise. The main reason for the popularity of the, compared to CT and MRI, ECG is that it is a simpler and less expensive test. ECG is a non-invasive test, it only requires the placement of electrodes on the skin, and it can provide information about the electrical activity of the heart, heart rate, and the presence of certain conditions, such as arrhythmias or heart attacks [8]. ECG can be performed easily, quickly and in most cases it is painless [9]. It can also be conducted repeatedly to monitor the progress of some conditions.

Each type of cardiac arrhythmia has a unique effect and requires a unique type of treatment [10]. Therefore, it is vital for cardiologists to precisely diagnose the kind of arrhythmia before administering medication. However, it is not easy to manually detect ECG components due to the significant differences in morphology. Moreover, visual identification, the current standard, can lead to subjective biases between and among observers [5]. For example, Figure 1 shows that different ECG signal classes have distinctive heartbeat characteristics and patterns, such as fusion (F), normal (N), supraventricular-ectopic (S), and ventricular-ectopic (V) beat. Therefore, researchers are working on alternative methods [11–13] that do not require visual and manual interpretations, such as deep learning.

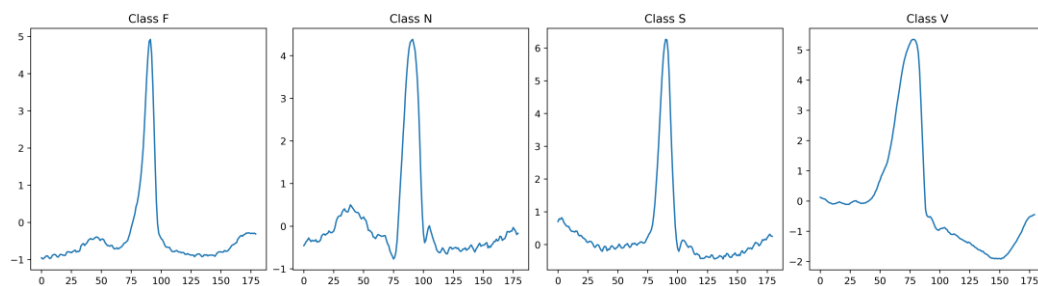


Figure 1. Example ECG patterns for different heartbeat types.

The ability of deep learning (DL) to automatically extract important features and self-learn from them to distinguish between classes makes it a promising alternative for classifying cardiac arrhythmia. Moreover, DL can reduce features automatically and handle large and noisy datasets [14,15]. Therefore, deep learning can be seen in various applications, including malware detection [16,17], image processing [18], and healthcare [19,20]. Furthermore, several recent studies have been proposed to classify cardiac arrhythmias using deep learning techniques such as convolutional neural networks (CNNs) [5,21,22].

Convolutional neural networks (CNNs) are well-known deep learning techniques for 2D data, such as image processing [23] and segmentation [24], due to the ability to learn complex features and maintain spatial relationships between features.

One-dimensional convolutional neural networks (1D CNNs) are a modified variant of 2D convolutional neural networks (CNNs) designed for applications involving 1D signals, such as bio-medical signal processing [25–27], and music genre classification [28].

This work presents a classification model based on DL by fine-tuning the hyperparameters of the 1D-CNN to address the limitations of traditional electrocardiograms (ECG) in the diagnosis of arrhythmias, which can be affected by noise and randomness of events, leading to misdiagnosis and errors. The proposed DL model incorporates three blocks, and each block comprises two 1D-CNN layers, a max-pooling layer, a dropout layer, and a batch-normalization layer. The main goal is to develop a fast and automatic alternative for the diagnosis of arrhythmias. The model correctly detects four types of arrhythmias, including fusion (F), normal (N), supraventricular-ectopic (S), and ventricular-ectopic (V) beat from ECG lead II signal. The confusion matrix and AUC-ROC curve were utilized to evaluate the model's performance. To support the model's validity, the CNN model's accuracy and recall (sensitivity) performance are compared to current research. The main contributions of this article are summarized as follows:

- Proposing a simple yet effective method for extracting the heartbeat from the signals by segmenting the signal centered on the R-peak point to ensure that all the critical features, such as QRS Complex, P-wave and T-wave, are correctly extracted.
- Developing a novel CNN architecture that shows outstanding performance in identifying four types of cardiac arrhythmias compared to existing work in the literature.
- Evaluating the optimal CNN hyperparameters in terms of filter, activation function, kernel size, and the number of layers.

The rest of the article is arranged in the following manner.

Section 2 reviews related work in arrhythmia classification using ECG signals. The proposed 1D-CNN model for classifying cardiac arrhythmia is explained in Section 3. Section 4 discusses the results of the experiment in comparison to state-of-the-art works. Section 5 concludes the work presented in this article and provides some suggestions and future work.

2. Related Works

Various research works on classifying cardiac arrhythmia from ECG signals can be divided into two approaches: the non-deep learning approach (traditional machine learning) and the deep learning approach.

The traditional ML approach uses machine learning algorithms such as support vector machine (SVM), decision trees (DTs), and random forest (RF) to classify cardiac arrhythmia. For instance, [29] proposed a computational system for diagnosing cardiac arrhythmia using k-nearest neighbor (KNN) and DTs. The model was trained based on 14 features extracted from the MIT-BIH dataset. DTs outperformed KNN with 0.96, 0.99, and 0.84 for accuracy, sensitivity, and specificity, respectively. The authors of [30] detected myocardial infarctions from 10 s ECG signals using SVM. The model was trained using 14 features extracted by the principal component (PCA) technique. The model achieved an overall accuracy of 0.96. The authors of [31] developed a model to detect the narrowing of three types of coronary arteries (CAD). The model is trained with SVM, uses 25 features, and achieves an overall accuracy of 0.96, a sensitivity of 1.00, and a specificity of 0.88. The authors of [32] proposed the Naïve Bayes model to detect five types of cardiac arrhythmias from ECG signals. The best model performance was based on four features extracted using higher-order statistics (HOS). The model obtained an overall accuracy of 0.94, a specificity of 0.57, and a recall of 0.99. In [13], various tree-based ML algorithms, such as Logistic Model Trees, Naïve Bayes Tree, and RF, were trained to classify arrhythmias from 23 recordings and trained to classify 11 classes. The RF scored the best results with an accuracy of 0.97, a specificity of 0.95, and a recall of 0.97. In [33], a genetic algorithm-based backpropagation neural network (GA-BPNN) technique for ECG identification was developed to categorize six distinct types of arrhythmias with an accuracy of 0.97.

Although these traditional machine learning models have performed well in classifying arrhythmias, they have some limitations, such as feature selection and poor classification performance on large datasets [34]. In other words, most machine learning algorithms require feature selection to reduce complexity and enhance performance. However, selecting important features requires additional work and might differ from one method to another.

On the other hand, deep learning (DL) overcomes these limitations and improves performance due to its ability to extract features automatically and handle large datasets. The authors of [6] trained a CNN-BiLSTM to classify five types of cardiac arrhythmia from the MIT-BIH dataset; the model obtained 0.98 accuracy, 0.91 sensitivity and 0.91 specificity. The authors of [35] proposed a CNN algorithm to classify heartbeats into five classes and achieved an overall accuracy of 0.93. Two-and five-second ECG signals from the St. Petersburg and Fantasia datasets are used in [36] to build a CAD prediction model using CNN. The proposed model can discriminate between arrhythmias with an accuracy of 0.94 for the two second model and 0.95 for the five-second model. A study [37] proposed a network of CNNs and BiLSTM to categorize five ECG arrhythmias with a recognition accuracy of 0.96.

Although several works have been conducted on identifying cardiac arrhythmias using deep learning techniques, there is still a lack of an effective CNN method for cardiac arrhythmia classification. In this study, we develop a novel CNN architecture to efficiently classify four types of cardiac arrhythmia from ECG lead II signal.

3. Proposed Methodology

The proposed 1D-CNN model is suggested for classifying cardiac arrhythmia into four phases: data acquisition, data preprocessing, training of 1D-CNN for classifying

cardiac arrhythmias and performance evaluation. The proposed classification of cardiac arrhythmias based on the 1d-CNN algorithm is shown in Figure 2.

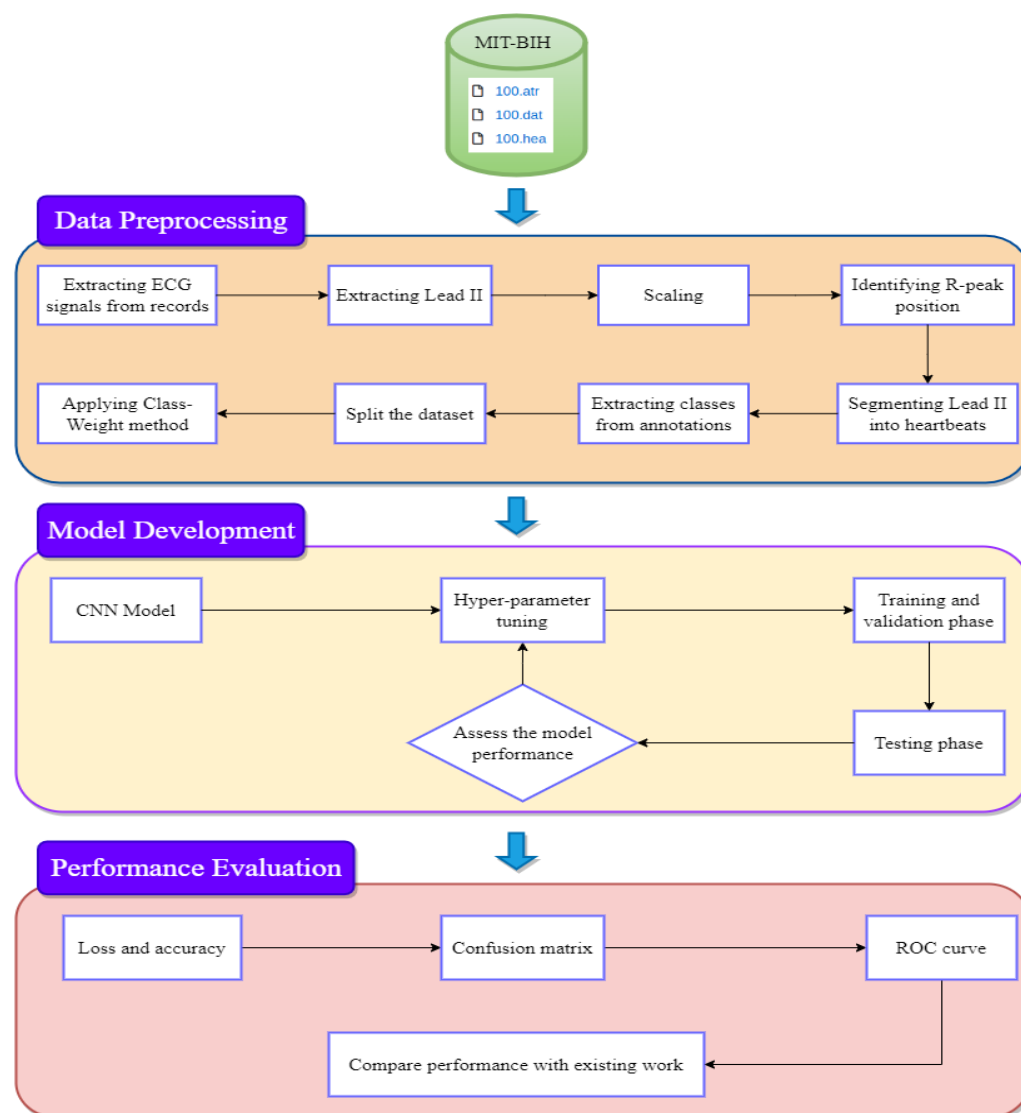


Figure 2. The proposed classification of cardiac arrhythmias based on 1D-CNN model.

3.1. Dataset Description

In this work, the original and noise-attenuated ECG signals obtained from the MIT-BIH [38] dataset are used as a data source to classify four arrhythmias according to the Association for the Advancement of Medical Instrumentation (AAMI) standard EC57 [39]. The MIT-BIH dataset contains 48 ECG recordings, thirty minutes long, obtained from 47 patients. Each record has two types of ECG signals: Lead II and Lead V5. The recordings were digitized across a 10 mV range at 360 Hz per channel in 11-bit resolution. In this experiment, ECGs Lead II have been extracted, scaled, and segmented into four types of arrhythmias to train and test the proposed CNN model.

3.2. Dataset Preprocessing

The 48 ECG records were converted to NumPy arrays, and the signals from the leads II were extracted to train the model. NumPy arrays are multi-dimensional arrays used for scientific computing in Python [40]. They are homogeneous, meaning each element in the array must be of the same type. They are fixed-size at creation, unlike Python lists. It is a requirement that all elements in a NumPy array are of the same data type, and they

are indexed by a tuple of non-negative integers. NumPy arrays are written mostly in C language and are stored in contiguous memory locations, which makes them faster and more powerful than Python lists.

Each extracted signal was scaled and segmented into heartbeats with a window length of 180 features for each heartbeat centered around the R-peak. The steps of extracting the heartbeats from the ECG signal II are described in Figure 3.

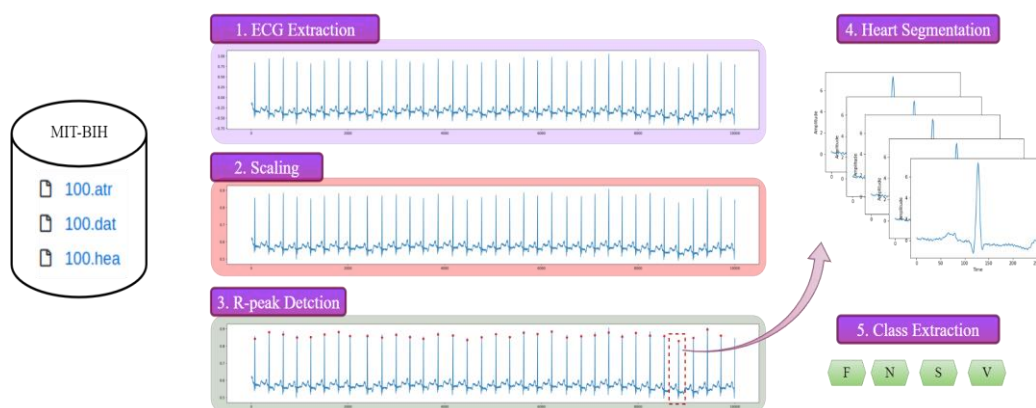


Figure 3. Heartbeat extraction pipeline.

3.2.1. Heartbeat Extraction

To extract the heartbeats from the MIT-BIT lead II, we propose a simple yet effective method (a visual representation is shown in Table 1) to extract heartbeats from signals. The main steps of the extraction method are described as follows:

- Extracting ECG lead II signals from the patient records and converting them into NumPy array.
- Scaling the extracted signals to ensure that all signals have the same mean and standard deviation, which helps classify several arrhythmias correctly.
- Detecting the R-peak position using XQRS from the WFDB library to segment the heartbeats.
- Segmenting the signal into heartbeat windows with a length of 180 features in each window centered around the R-peak position.
- Extract each heartbeat class from the annotations given by cardiologists in the dataset.

Table 1. Arrhythmia classes, related annotation and sample size of each class in the dataset.

Class	Annotation	Sample Size
N	Normal	89,694
	Left/Right bundle branch block	
	Atrial escape	
	Nodal escape	
S	Atrial premature	2814
	Aberrant atrial premature	
	Nodal premature	
	Supra-ventricular premature	
V	Premature ventricular contraction	6487
	Ventricular escape	
F	Fusion of ventricular and normal	779
Total		99,774

Table 1 shows the categorization of the four types of ECG heartbeats, which are divided into four classes according to AAMI [39].

It is essential to highlight that the proposed technique extracts all the key regions, such as QRS Complex, P-wave and T-wave, that are generally used by cardiologists to identify arrhythmia. Moreover, all extracted beats contain exact window sizes, which is crucial to train the model.

3.2.2. Data Preparation

After segmenting the signals into heartbeats, the total number of heartbeats is 99,774 for the 48 records. Each class contains a different number of heartbeat samples. The heartbeats are categorized into 89,694 for normal beats, 6487 for ventricular ectopic beats, 2814 for supraventricular ectopic beats, 779 for fusion beats, and 24 for unknown beats. The unknown beats have dropped because they contain unclassified beats and have a significantly lower number of instances [41]. The heartbeats were split into 75% (74,830 samples) for training and validation and 25% (24,944 samples) for testing. We used the stratify parameter to ensure that all the classes had been split equally into the training and testing data. The utilization of the stratify parameter guarantees that the proportion of samples is maintained consistently across each class, for both the training and testing data. The training and validation dataset splits into 80% (59,864 samples) for training and 20% (14,966 samples) for validation.

Table 1 shows a significant imbalance among the four classes (i.e., class ‘N’ has 89,694, while class ‘F’ has only 779). An unbalanced dataset leads to a classification bias toward classes with more samples, which leads to deficient performance in classifying categories with fewer samples [42]. Therefore, estimating class-weight technique [43] is applied to balance the datasets. The class-weight technique adjusts the model’s cost function such that misidentifying an instance belonging to a minority class incurs a more significant penalty than misidentifying an example belonging to a majority class [14]. This strategy can improve the model’s accuracy by rebalancing the class distribution.

3.3. Training of One-Dimensional Convolutional Neural Network (1D-CNN) for Classifying Cardiac Arrhythmia

Convolutional neural network (CNN) is a deep learning network that can infer high-level features from input features [44]. A CNN is conceptually comparable to a multilayer perceptron (MLP), where each neuron possesses an activation function that translates the weighted inputs into the outputs. CNNs consist of three main layers: the convolutional layer, the pooling layer, and the fully connected layer. With proper training, CNNs can be utilized in many applications, including speech recognition [45], structure engineering [46,47], and image processing [48].

1D-CNN is a modified version of CNN designed for 1D signals, especially for sparse data unsuitable for traditional CNN [49]. 1D-CNNs are similar to 2D-CNNs but are used for processing one-dimensional data such as audio or text. Moreover, 1D-CNNs use 1D convolutional filters to extract features from the data, while 2D-CNNs use 2D convolutional filters. 1D-CNNs also have fewer parameters than 2D-CNNs, which makes them more computationally efficient [49]. It is particularly well suited for signal data that has temporal component, including time series data, because it is able to extract local features from the signal that are robust to small shifts in time. Here are a few reasons why 1D CNNs are particularly well suited for signal data:

- **Time-invariant feature learning:** A 1D CNN is able to learn time-invariant features from a signal, meaning that it is able to extract features that are robust to small shifts in time. This is important for signal data because signals are often affected by noise, variations in the measurement scale, and non-stationarity. With a 1D CNN, a network can learn to extract relevant features that are robust to these variations, resulting in improved performance.
- **Local feature extraction:** In 1D CNNs, the convolutional layers are able to extract local features from the signal, which are important for signal data, as signals are often

composed of local patterns. These patterns could be variations, such as frequency, amplitude, or shape, which the CNN is able to learn and extract.

- Translation invariance: 1D CNNs are also translation invariant, which means that they are able to detect the same pattern, regardless of its location in the signal. This is particularly useful for signals where the location of the pattern is not known.

Multiscale feature learning: The combination of convolution and pooling operations in a 1D CNN allows the network to learn features at different scales and resolutions, which is useful for signal data because signals often have patterns at different scales.

These properties, along with the ability of CNNs to learn complex and abstract feature representations, makes them a good fit for signal data. Algorithm 1 shows an example of the basic structure of a 1D CNN algorithm in pseudocode.

Algorithm 1. The 1D-CNN pseudocode.

1D-CNN model architecture

```

Input: input_data, test_data, filter, kernel, blocks, cls_weight
Output: Predictions
Start Algorithm (1D-CNN)
1 | model = Sequential ()
2 | For (block in blocks) do
3 |   model. Add (Conv1D (filter, (kernel,)), activation = 'relu')
4 |   model. Add (Conv1D (filter, (kernel,)), activation = 'relu')
5 |   model. Add (MaxPooling1D (2))
6 |   model. Add (Dropout (0.2))
7 |   model. Add (BatchNormalization ())
8 | End;//For loop
9 | model. Add (Flatten ())
10 | model. Add (Dense (512), activation = 'relu')
11 | model. Add (Dropout (0.2))
12 | model. Add (Dense (4), activation = 'softmax')
Phase 1, Compile the model
   | model. Compile (loss = 'sparse_categorical_crossentropy',
13 | optimizer = Adam (learning_rate = 0.001, decay =  $1e^{-6}$ ),
   | metrics = ['accuracy'])
Phase 2, Fit the model
   | model. Fit (input_data,
   | batch_size = 512,
14 | epochs = 500,
   | validation_split = 0.20,
   | class_weight = cls_weight)
Phase 3, Predict arrhythmias
   | Predictions = model. Predict (test_data)
   | Return Predictions
15 End;//Algorithm

```

This pseudocode assumes that the CNN class has already been implemented with functions for applying filters and combining output and that the convolutional and fully connected layers are stored in lists within the CNN object. The input data are passed through the convolutional layers, with each layer applying a set of filters to the input using 1D convolution and producing output data. The output data from the final convolutional layer is then passed through the fully connected layers, with each layer combining the output from the previous layer into a single set of predictions. The final set of predictions is then returned as the output of the 1D CNN.

Due to its compact and simple configuration performance, 1D-CNN is also suitable for real-time applications and low-cost hardware implementations. Equation (1) represents a single convolution of a signal, $x_1^n = [x_1, x_2, \dots, x_n]$, in which n denotes the total number of points [50], h denotes the activation function, l denotes layer index, b denotes the bias of

the j th feature map, M denotes the kernel size, W_m^j denotes the feature map's weight and m th denotes the filter index.

$$C_i^{lj} = h(b_j + \sum_{m=1}^M W_m^j x_{i+m-1}^j) \quad (1)$$

This work developed a 1D CNN model to classify cardiac arrhythmias based on signals from ECG lead II. An intensive experiment was conducted to select the minimal model architecture with optimal parameters to improve the model performance. The architecture of the proposed model is represented in Figure 4.

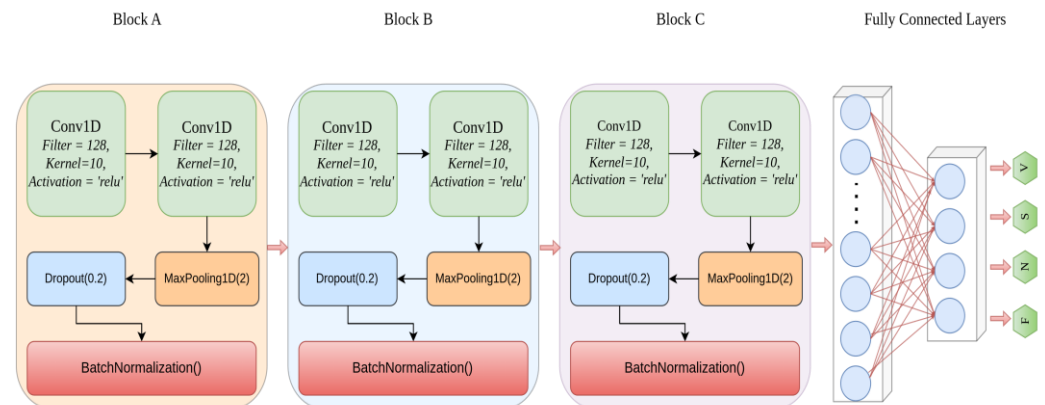


Figure 4. The proposed 1D-CNN model architecture.

Figure 4 shows that the model consists of three convolution blocks (A, B, C). Each block contains two 1D-CNN layers, a max-pooling layer, a dropout layer, and a batch-normalization layer. Each 1D-CNN layer contains 128 filters with a kernel-window size of 10 for each filter. Each layer of a 1D-CNN is activated using the rectified linear unit (ReLU) function. Activation functions are crucial to increase the expressiveness of neural networks and enhance the approximation capability between the network's different layers [34].

The max-pooling layer is applied after the 1D-CNN with a pooling size of two to highlight the most present feature by calculating the largest value in each patch. To accelerate the learning speed of the model structure, max-pooling down-samples the input representation by selecting the largest value inside a spatial region [34]. The setting of the network hyperparameters was determined through an empirical approach, involving experimentation with different values to find the optimal configuration.

As shown in Table 2, to reduce the overfitting, a dropout layer of 0.20 and the normalization layer were applied [49]. During the training period, the dropout layer randomly sets the inputs at each step with zero frequency. Dropout regularization reduces interdependence between layers by probabilistically dropping some of the nodes in the same layer. The dropped neuron weights are ignored, significantly improving the model's generalization capacity [51].

Table 2. Parameter tuning of the proposed model.

Parameter	Value
Filter	128
Kernel size	10
Dropout	0.2
Learning rate	0.001
Decay	1×10^{-6}
Batch size	512
Epoch	500

The normalization of CNN layers accelerates model convergence during training and prevents gradient growth [52]. In addition, the batch-normalization layer guarantees that the transformation of the various batches remains within a specified range, stabilizing the learning process and accelerating the parameters' convergence [34].

The fully connected layer combines the information from the preceding layers to create the final output. In the flatten layer, the preceding layer's output is transformed into a single vector to be implemented as input for the dense layer. Each neuron in the dense layer gets the outputs of all neurons in the layer underneath it and conducts matrix-vector multiplication. Table 2 illustrates the proposed structure of the CNN model. The first dense layer has 512 nodes and an activation function of ReLU, followed by a 0.20 dropout layer. The second dense layer consists of four nodes, each representing a class, and a SoftMax activation function to classify the output into four arrhythmia types. The model was fitted with sparse cross-entropy as the loss function and the Adam optimizer with a learning rate of 0.001 and a decay factor of 1×10^{-6} as the optimization function. The model fits the training and validation datasets with 512-epoch batches and 500 iterations.

4. Results and Discussion

4.1. Performance Matrices

The confusion matrix and AUC-ROC curve, frequently used to assess machine learning models, were used to assess the model performance. Specifically, the model is evaluated using accuracy, precision, recall (sensitivity), f1 score, specificity, and ROC curve, which are described below:

- Accuracy: how often the model is correct.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (2)$$

- Precision: how often the model predicts a class as positive relative to the total number of positives in all classes.

$$Precision = \frac{TP}{TP + FP} \quad (3)$$

- Recall: how many times the model correctly predicts the class to be positive.

$$Recall = \frac{TP}{TP + FN} \quad (4)$$

- Specificity: Specificity is defined as the proportion of true negatives correctly identified by the model. It is also referred to as the true negative rate (TNR) or selectivity. It measures the ability of the model to accurately identify negative examples.

$$Specificity = \frac{TN}{TN + FP} \quad (5)$$

- F1-score describes the weighted average of recall and precision.

$$F1 - score = \frac{2 * recall * precision}{recall + precision} \quad (6)$$

- AUC-ROC Curve: The Area Under Curve (AUC) is a measure of the thresholds between true and false-positive rates. The Receiver Operating Characteristic curve (ROC) visually illustrates the trade-off between sensitivity and specificity, where the x -axis represents the false-positive rate, and the y -axis represents the true-positive rate. The AUC evaluates the capability of the ROC curve to distinguish between classes. The larger the AUC number, the better the performance.

$$AUC = \frac{1}{2} \left(\frac{TP}{TP + FN} + \frac{TN}{TN + FP} \right) \quad (7)$$

4.2. Performance of the Proposed Method

The experimental dataset used in this work is from the MIT-BIH, which is currently used in a lot of ECG research and contains accurate and thorough expert annotation [38,53]. As discussed in Section 3, the ECG lead II signals of the 47 patients have been extracted, scaled, and segmented into heartbeats. The working environment for training the model consisted of one NVIDIA GeForce GTX 1070 GPU with 16 GB of RAM.

The initial phase of the experiments involved dividing the dataset into 75% (74,830 samples) for training and validation and 25% (24,944 samples) for testing. The model was trained for 500 epochs with a batch size of 512 per epoch. The accuracy and loss curves of the conducted experiment are illustrated in Figure 5. The loss function in 1D-CNN quantifies the discrepancy between the expected outcome and the outcome produced by the 1D-CNN algorithm. It is used to measure how far an estimated value is from its true value [54]. In this experiment, the sparse categorical cross entropy loss function is used for our multiclass classification task. This loss function computes the logarithm of the output index, which is indicated by the ground truth. This means that the loss is computed only once per instance and the summation is omitted, leading to better performance [54]. The formula for the sparse categorical cross entropy loss is as follows:

$$J(w) = -\log(\hat{y}_y) \quad (8)$$

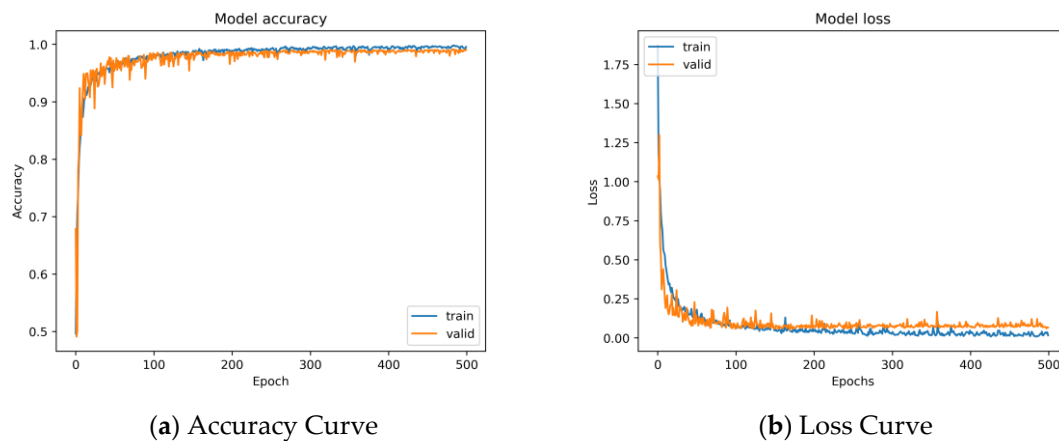


Figure 5. Accuracy and loss curves of the 1D-CNN model.

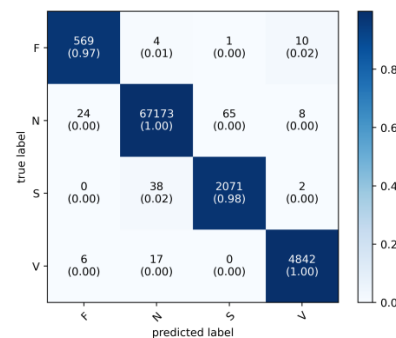
Figure 5 shows that both the training and validation curves increased in a stable manner.

Furthermore, the proposed model achieved remarkable precision, recall, f1-score, AUC, average accuracy, and loss in the training and testing datasets. Table 3 shows the performance matrices used to evaluate the model in the training and testing datasets. It is worth mentioning that all the numbers in the manuscript have been rounded to 2 decimal numbers.

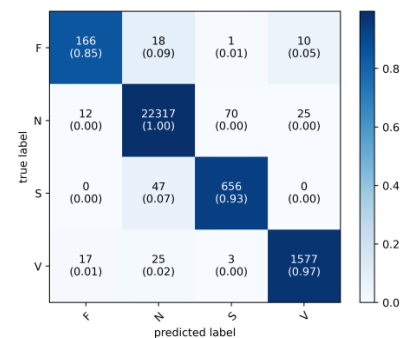
The average in the table refers to the mathematical mean of all classes in a particular measurement without considering the proportion of each class in the dataset. For example, in the training dataset, the model scores an average of 0.99 per cent in recall, and 0.98 per cent in precision and f1-score, respectively. Amongst the four classes, class N and class V secure perfect results of 1.00 in all matrices, whereas class F scores the worst, with 0.95, 0.97, 0.96 in precision, recall and f1-score, respectively. Figure 6 presents the number of correctly and incorrectly classified samples in the training and testing dataset, along with their percentages.

Table 3. Confusion matrix report of 1D-CNN model.

Matrix	Training Dataset					Testing Dataset				
	F	N	S	V	Average	F	N	S	V	Average
Precision	0.95	1.00	0.97	1.00	0.98	0.85	1.00	0.90	0.98	0.93
Recall	0.97	1.00	0.98	1.00	0.99	0.85	1.00	0.93	0.97	0.94
F1-score	0.96	1.00	0.98	1.00	0.98	0.85	1.00	0.92	0.98	0.93
AUC	1.00	1.00	1.00	1.00	1.00	0.99	1.00	1.00	1.00	1.00
Accuracy			1.00					0.99		
Loss			0.02					0.06		



(a) Training-based confusion matrix.

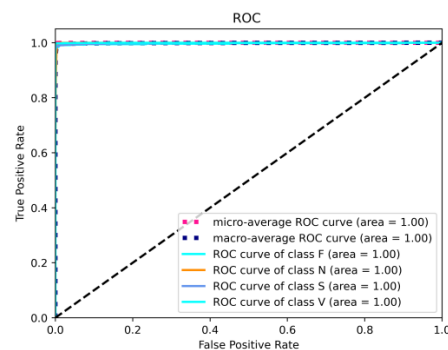


(b) Testing-based confusion matrix.

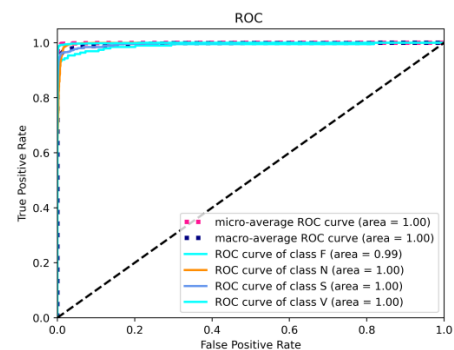
Figure 6. Confusion matrix of the 1D-CNN model for arrhythmia categorization.

In the test dataset, the average of all classes decreased by 0.5 per cent in all classes compared to the training dataset, with overall 0.93, 0.94, and 0.93 precision, recall and f1-score, respectively. Class N is still in the lead with a perfect score in all performance matrices, while class V dropped 0.02 per cent in precision and f1-score and 0.03 per cent in recall. Class S dropped 0.07 per cent in precision, 0.05 per cent in recall and 0.06 per cent in F1-score. Even though the class weight technique has been applied to adjust the cost function of the model, class F is still the worst class, with 0.85 per cent in all classes with a 0.10 per cent drop. We can relate the low score of class F to the sample size, which only has 584 and 195 samples; among them, only 15 and 29 samples were misclassified in the training and testing datasets, respectively.

AUC is another measurement score to evaluate the model's performance in discriminating between classes. It measures the trade-off between the true-positive and false-positive rates, which can graphically represent using the ROC curve. Figure 7 shows the ROC curve on the training and testing datasets. The figure shows a perfect score in all classes in the training and testing datasets except class F in the testing set with 0.99 per cent.



(a) Training-based ROC curve.



(b) Testing-based ROC curve.

Figure 7. ROC curve of the 1D-CNN model for arrhythmia categorization.

Micro-average and macro-average are two ways of summarizing the information of the multiclass ROC curves. Micro-averaging aggregates the contributions from all the classes to compute the average metrics as follows:

$$TPR = \frac{\sum(TP_c)}{\sum(TP_c + FN_c)}; FPR = \frac{\sum(FP_c)}{\sum(FP_c + TN_c)} \quad (9)$$

Macro-averaging requires calculating the metric individually for each class and then averaging the results, hence treating all classes equally a priori.

4.3. Comparison of the Proposed Method to Other Previous Works

Further comparison with existing work in the literature showed that our proposed network scores superior performance in distinguishing different classes exceeding traditional and deep learning models. Furthermore, the experiment in this paper achieved excellent accuracy compared to existing works in the literature, with total training and testing dataset accuracy of 1.00 and 0.99 per cent, respectively.

Compared to traditional machine learning models such as [29,30,32], and [13], the proposed 1D-CNN model exceeds them in terms of accuracy with an estimate of 0.05 to 0.02 percent. Moreover, machine learning models tend to apply feature engineering to select high-weight features to reduce complexity. For instance, [29] used Fourteen features from ECG signals to train SVM and DTs to diagnose cardiac arrhythmia. The authors of [32] trained a Naïve Bayes model using only four features. Using such an approach in ECG signals might cause losing important features and break up the temporal relationships between features [34]. Moreover, feature engineering requires additional processes to choose the feature selection method and the best number of features to train the model.

On the other hand, the deep learning approach automatically reduces feature complexity while training the model. CNN, for instance, can automatically learn the unique relationships between features. Therefore, CNN is commonly used with spatially related data due to its ability to effectively model spatial localities using shared weights for the filters [55]. Table 4 summarizes selected state-of-the-art studies and compares them to our work. It is worth mentioning that all the existing work in Table 4 used the MIT-BIH dataset.

Table 4. Existing work for arrhythmia classification from ECG signals.

Author	Classifier	Classes	Accuracy	Recall (Sensitivity)	Specificity
Guo et al. [56]	DenseNet -GRU	5	0.92	0.82	0.96
Singh et al. [57]	Ensemble SVM	4	0.93	0.47	NA
Singh et al. [58]	RNN-LSTM	2	0.88	0.92	0.83
Kachuee et al. [59]	Deep residual CNN	5	0.93	0.93	N/A
Essa et al. [60]	CNN-LSTM	4	0.96	0.69	0.95
Shadmand et al. [61]	BBNN	2	0.98	0.73	0.99
Hannun et al. [62]	1D-CNN	12	0.95	0.77	0.97
Hassan et al. [6]	CNN-BiLSTM	5	0.98	0.91	0.91
Zhou et al. [22]	CNN-ELM	4	0.99	0.94	0.98
Xu et al. [37]	CNN + BiLSTM	5	0.96	0.96	0.96
Our work	1D CNN	4	0.99	0.94	0.99

BBNN = Block-based neural network.

Compared to CNN models such as [62] and [5], the proposed model exceeds them in terms of accuracy with an estimate of 0.04 and 0.06 per cent, respectively. Moreover, our model scores higher specificity with an estimate of 0.02 and 0.07 per cent. Similarly, [59] developed a model with thirteen layers of 1D CNN to predict five types of arrhythmias and achieved less than our model with an estimate of 0.04 and 0.01 in accuracy and sensitivity.

Compared to hybrid models that combine two architectures such as [56,58], our CNN model surpasses them by at least 0.07 per cent in accuracy and sensitivity and 0.03 per

cent in specificity. For instance, [56] developed a model incorporating a densely connected convolutional neural network (DenseNet) and gated recurrent unit network (GRU) and scores less than our model with an estimate of 0.07, 0.17, and 0.02 in accuracy, sensitivity, and specificity, respectively. In [58], a deep learning model combining RNN and LSTM is developed to classify the normal and abnormal beats from ECG signals. Our model scores better in accuracy sensitivity and specificity with 0.11, 0.02, and 0.16, respectively. In [22], a convolutional neural network and extreme learning machine (CNN-ELM) is trained to classify four classes of arrhythmias and scored 0.01 less than our model in specificity. In [37], a CNN-BiLSTM model is developed to classify five types of arrhythmias and scored 0.03 less than our model in accuracy and specificity.

5. Conclusions

Classification of cardiac arrhythmias is essential to help physicians diagnose cardiovascular diseases. This work proposed a classification model containing three blocks of 1D CNN to classify cardiac arrhythmia from ECG lead II signal. The proposed 1D-CNN model has demonstrated its efficiency in predicting four arrhythmia classes with an outstanding performance. The 1D CNN model can help physicians diagnose cardiovascular disease while reducing physician workload. Although the proposed architecture has an excellent performance, certain limitations still had to be considered when interpreting the results. The distribution of categories in the MIT-BIH dataset we used for training and testing was quite unbalanced. Although the proposed architecture has an excellent performance, certain limitations still had to be considered when interpreting the results. The distribution of categories in the MIT-BIH dataset we used for training and testing was quite unbalanced. Even though we have addressed the imbalance with the class weight approach, the imbalance of the data still has some impact on the model generalization. Furthermore, cardiac arrhythmias can vary greatly between patients; therefore, a larger dataset is needed to train deep learning models to handle such variability and to better generalize new cases, which will be our aim in future work.

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

Funding: King Abdulaziz University-Institutional Funding Program for Research and Development-Ministry of Education: IFPIP:536-830-1443.

Data Availability Statement: Not applicable.

Acknowledgments: This research work was funded by Institutional Fund Projects under grant no. (IFPIP:536-830-1443). The authors gratefully acknowledge technical and financial support provided by the Ministry of Education and King Abdulaziz University, DSR, Jeddah, Saudi Arabia.

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

References

1. Al'Aref, S.J.; Maliakal, G.; Singh, G.; van Rosendael, A.R.; Ma, X.; Xu, Z.; Alawamlh, O.A.; Lee, B.; Pandey, M.; Achenbach, S.; et al. Machine learning of clinical variables and coronary artery calcium scoring for the prediction of obstructive coronary artery disease on coronary computed tomography angiography: Analysis from the CONFIRM registry. *Eur. Heart J.* **2020**, *41*, 359–367. [\[CrossRef\]](#) [\[PubMed\]](#)
2. Abdolmanafi, A.; Duong, L.; Dahdah, N.; Adib, I.R.; Cheriet, F. Characterization of coronary artery pathological formations from OCT imaging using deep learning. *Biomed. Opt. Express* **2018**, *9*, 4936. [\[CrossRef\]](#) [\[PubMed\]](#)
3. Acharya, U.R.; Fujita, H.; Lih, O.S.; Adam, M.; Tan, J.H.; Chua, C.K. Automated detection of coronary artery disease using different durations of ECG segments with convolutional neural network. *Knowl.-Based Syst.* **2017**, *132*, 62–71. [\[CrossRef\]](#)
4. Husain, K.; Zahid, M.S.M.; Hassan, S.U.; Hasbullah, S.; Mandala, S. Advances of ECG Sensors from Hardware, Software and Format Interoperability Perspectives. *Electronics* **2021**, *10*, 105. [\[CrossRef\]](#)

5. Acharya, U.R.; Oh, S.L.; Hagiwara, Y.; Tan, J.H.; Adam, M.; Gertych, A.; San Tan, R. A deep convolutional neural network model to classify heartbeats. *Comput. Biol. Med.* **2017**, *89*, 389–396. [CrossRef]
6. Hassan, S.U.; Zahid, M.S.M.; Abdullah, T.A.A.; Husain, K. Classification of cardiac arrhythmia using a convolutional neural network and bi-directional long short-term memory. *Digit. Health* **2022**, *8*, 1–13. [CrossRef]
7. Schläpfer, J.; Wellens, H.J. Computer-Interpreted Electrocardiograms: Benefits and Limitations. *J. Am. Coll. Cardiol.* **2017**, *70*, 1183–1192. [CrossRef]
8. Müller, N.L. Computed tomography and magnetic resonance imaging: Past, present and future. *Eur. Respir. J.* **2002**, *19*, 3–12. [CrossRef]
9. Bizopoulos, P.; Koutsouris, D. Deep Learning in Cardiology. *IEEE Rev. Biomed. Eng.* **2019**, *12*, 168–193. [CrossRef]
10. Martis, R.J.; Acharya, U.R.; Adeli, H. Current methods in electrocardiogram characterization. *Computers in Biology and Medicine. Comput. Biol. Med.* **2014**, *48*, 133–149. Available online: <https://www.sciencedirect.com/science/article/pii/S0010482514000432> (accessed on 19 December 2022). [CrossRef]
11. Kumar, M.; Pachori, R.B.; Acharya, U.R.; Bilas, R.; Acharya, U.R. Characterization of coronary artery disease using flexible analytic wavelet transform applied on ECG signals. *Biomed. Signal Process. Control* **2017**, *31*, 301–308. [CrossRef]
12. Zhu, H.; Cheng, C.; Yin, H.; Li, X.; Zuo, P.; Ding, J.; Lin, F.; Wang, J.; Zhou, B.; Li, Y.; et al. Automatic multilabel electrocardiogram diagnosis of heart rhythm or conduction abnormalities with deep learning: A cohort study. *Lancet Digit. Health* **2020**, *2*, e348–e357. Available online: <https://www.sciencedirect.com/science/article/pii/S2589750020301072> (accessed on 19 December 2022). [CrossRef] [PubMed]
13. Latif, G.; al Anezi, F.Y.; Zikria, M.; Alghazo, J. EEG-ECG Signals Classification for Arrhythmia Detection using Decision Trees. In Proceedings of the 4th International Conference on Inventive Systems and Control, ICISC 2020, TamilNadu, India, 8–10 January 2020; pp. 192–196. [CrossRef]
14. Chou, Y.H.; Hong, S.; Zhou, Y.; Shang, J.; Song, M.; Li, H. Knowledge-shot learning: An interpretable deep model for classifying imbalanced electrocardiography data. *Neurocomputing* **2020**, *417*, 64–73. [CrossRef]
15. Abdullah, T.A.A.; Zahid, M.S.M.; Ali, W. A review of interpretable ml in healthcare: Taxonomy, applications, challenges, and future directions. *Symmetry* **2021**, *13*, 2439. [CrossRef]
16. Abdullah, T.A.A.; Ali, W.; Abdulghafor, R. Empirical study on intelligent android malware detection based on supervised machine learning. *Int. J. Adv. Comput. Sci. Appl.* **2020**, *11*, 215–224. [CrossRef]
17. Abdullah, T.A.A.; Ali, W.; Malebary, S.; Ahmed, A.A. A Review of Cyber Security Challenges, Attacks and Solutions for Internet of Things Based Smart Home. *IJCSNS Int. J. Comput. Sci. Netw. Secur.* **2019**, *19*, 139–146.
18. Al-Hiyali, M.I.; Yahya, N.; Faye, I.; Hussein, A.F. Identification of autism subtypes based on wavelet coherence of BOLD FMRI signals using convolutional neural network. *Sensors* **2021**, *21*, 5256. Available online: <https://www.mdpi.com/1216014> (accessed on 19 December 2022). [CrossRef]
19. Alizadehsani, R.; Roshanzamir, M.; Abdar, M.; Beykikhoshk, A.; Khosravi, A.; Panahiazar, M.; Koohestani, A.; Khozeimeh, F.; Nahavandi, S.; Sarrafzadegan, N. A database for using machine learning and data mining techniques for coronary artery disease diagnosis. *Sci. Data* **2019**, *6*, 227. [CrossRef]
20. Al-Hiyali, M.I.; Yahya, N.; Faye, I.; Khan, Z.; Alsaih, K. Classification of BOLD FMRI signals using wavelet transform and transfer learning for detection of autism spectrum disorder. In Proceedings of the 2020 IEEE-EMBS Conference on Biomedical Engineering and Sciences (IECBES), Langkawi Island, Malaysia, 1–3 March 2021; pp. 94–98. Available online: <https://ieeexplore.ieee.org/abstract/document/9398803/> (accessed on 19 December 2022).
21. Hassan, S.U.; Zahid, M.S.M.; Husain, K. Performance comparison of CNN and LSTM algorithms for arrhythmia classification. In Proceedings of the 2020 International Conference on Computational Intelligence (ICCI), Bandar Seri Iskandar, Malaysia, 8–9 October 2020; pp. 223–228. Available online: <https://ieeexplore.ieee.org/abstract/document/9247636/> (accessed on 19 December 2022).
22. Zhou, S.; Tian, B. Electrocardiogram soft computing using hybrid deep learning CNN-ELM. *Appl. Soft Comput.* **2020**, *86*, 105778. Available online: <https://www.sciencedirect.com/science/article/pii/S1568494619305599> (accessed on 19 December 2022). [CrossRef]
23. Hu, H.; Zhang, Z.; Xie, Z.; Lin, S. Local relation networks for image recognition. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019; pp. 3464–3473. Available online: http://openaccess.thecvf.com/content_ICCV_2019/html/Hu_Local_Relation_Networks_for_Image_Recognition_ICCV_2019_paper.html (accessed on 19 December 2022).
24. Gite, S.; Mishra, A.; Kotecha, K. Enhanced lung image segmentation using deep learning. *Neural Comput. Appl.* **2022**, 1–15. [CrossRef]
25. Liu, X.; Wang, H.; Li, Z.; Qin, L. Deep learning in ECG diagnosis: A review. *Knowl.-Based Syst.* **2021**, *227*, 107187. [CrossRef]
26. 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]
27. Kiranyaz, S.; Ince, T.; Gabbouj, M. Personalized monitoring and advance warning system for cardiac arrhythmias. *Sci. Rep.* **2017**, *7*, 9270. Available online: <https://www.nature.com/articles/s41598-017-09544-z> (accessed on 19 December 2022). [CrossRef] [PubMed]

28. Allamy, S.; Koerich, A.L. 1D CNN Architectures for Music Genre Classification. In Proceedings of the 2021 IEEE Symposium Series on Computational Intelligence, SSCI 2021, Orlando, FL, USA, 5–7 December 2021. [\[CrossRef\]](#)
29. Acharya, U.R.; Fujita, H.; Adam, M.; Lih, O.S.; Hong, T.J.; Sudarshan, V.K.; Koh, J.E. Automated characterization of arrhythmias using nonlinear features from tachycardia ECG beats. In Proceedings of the 2016 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2016, Budapest, Hungary, 9–12 October 2016; pp. 533–538. [\[CrossRef\]](#)
30. Dohare, A.K.; Kumar, V.; Kumar, R. Detection of myocardial infarction in 12 lead ECG using support vector machine. *Appl. Soft Comput. J.* **2018**, *64*, 138–147. [\[CrossRef\]](#)
31. Alizadehsani, R.; Hosseini, M.J.; Khosravi, A.; Khozeimeh, F.; Roshanzamir, M.; Sarrafzadegan, N.; Nahavandi, S. Non-invasive detection of coronary artery disease in high-risk patients based on the stenosis prediction of separate coronary arteries. *Comput. Methods Programs Biomed.* **2018**, *162*, 119–127. [\[CrossRef\]](#) [\[PubMed\]](#)
32. Marinho, L.B.; Nascimento, N.D.M.; Souza, J.W.M.; Gurgel, M.V.; Filho, P.P.R.; de Albuquerque, V.H.C. A novel electrocardiogram feature extraction approach for cardiac arrhythmia classification. *Future Gener. Comput. Syst.* **2019**, *97*, 564–577. [\[CrossRef\]](#)
33. Li, H.; Yuan, D.; Ma, X.; Cui, D.; Cao, L. Genetic algorithm for the optimization of features and neural networks in ECG signals classification. *Sci. Rep.* **2017**, *7*, 41011. Available online: <https://www.nature.com/articles/srep41011> (accessed on 19 December 2022). [\[CrossRef\]](#)
34. Zheng, Z.; Chen, Z.; Hu, F.; Zhu, J.; Tang, Q.; Liang, Y. An automatic diagnosis of arrhythmias using a combination of CNN and LSTM technology. *Electronics* **2020**, *9*, 121. [\[CrossRef\]](#)
35. Boursalie, O.; Samavi, R.; Doyle, T.E. M4CVD: Mobile machine learning model for monitoring cardiovascular disease. *Procedia Comput. Sci.* **2015**, *63*, 384–391. [\[CrossRef\]](#)
36. Butun, E.; Yildirim, O.; Talo, M.; Tan, R.S.; Acharya, U.R. 1D-CADCapsNet: One dimensional deep capsule networks for coronary artery disease detection using ECG signals. *Phys. Med.* **2019**, *70*, 39–48. [\[CrossRef\]](#)
37. Xu, X.; Jeong, S.; Li, J. Interpretation of Electrocardiogram (ECG) Rhythm by Combined CNN and BiLSTM. *IEEE Access* **2020**, *8*, 125380–125388. [\[CrossRef\]](#)
38. Moody, G.B.; Mark, R.G. The impact of the MIT-BIH arrhythmia database. *IEEE Eng. Med. Biol. Mag.* **2001**, *20*, 45–50. Available online: <https://ieeexplore.ieee.org/abstract/document/932724/> (accessed on 19 December 2022). [\[CrossRef\]](#)
39. ANSI/AAMI EC57:2012; d Testing and Reporting Performance Results of Cardiac Rhythm and ST Segment Measurement Algorithms. American National Standards Institute: Washington, DC, USA, 2017.
40. NumPy. NumPy Array Objects. *NumPy* 18 December 2022. Available online: <https://numpy.org/doc/stable/reference/arrays.html> (accessed on 11 January 2023).
41. Gai, N.D. ECG beat classification using machine learning and pre-trained convolutional neural networks. *arXiv* **2022**, arXiv:2207.06408. Available online: <http://arxiv.org/abs/2207.06408> (accessed on 11 January 2023).
42. Hosgunor, E. “How to Handle Imbalance Data and Small Training Sets in ML.” Towards Data Science, Medium, 28 October 2020. Available online: <https://towardsdatascience.com/how-to-handle-imbalance-data-and-small-training-sets-in-ml-989f8053531d> (accessed on 11 January 2023).
43. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830. Available online: <https://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf?ref=https://githubhelp.com> (accessed on 19 December 2022).
44. Feng, S.H.; Xu, J.Y.; Shen, H.B. Artificial intelligence in bioinformatics: Automated methodology development for protein residue contact map prediction. In *Biomedical Information Technology*; Academic Press: Cambridge, MA, USA, 2020; pp. 217–237. Available online: <https://www.sciencedirect.com/science/article/pii/B9780128160343000079> (accessed on 19 December 2022).
45. Chan, W.; Park, D.; Lee, C.; Zhang, Y.; Le, Q.; Norouzi, M. SpeechStew: Simply Mix All Available Speech Recognition Data to Train One Large Neural Network. *arXiv* **2021**, arXiv:2104.02133. Available online: <http://arxiv.org/abs/2104.02133> (accessed on 19 December 2022).
46. Yu, Y.; Liang, S.; Samali, B.; Nguyen, T.N.; Zhai, C.; Li, J.; Xie, X. Torsional capacity evaluation of RC beams using an improved bird swarm algorithm optimised 2D convolutional neural network. *Eng. Struct.* **2022**, *273*, 115066. [\[CrossRef\]](#)
47. Yu, Y.; Samali, B.; Rashidi, M.; Mohammadi, M.; Nguyen, T.N.; Zhang, G. Vision-based concrete crack detection using a hybrid framework considering noise effect. *J. Build. Eng.* **2022**, *61*, 105246. [\[CrossRef\]](#)
48. Ding, K.; Ma, K.; Wang, S.; Simoncelli, E.P. Comparison of Full-Reference Image Quality Models for Optimization of Image Processing Systems. *Int. J. Comput. Vis.* **2021**, *129*, 1258–1281. [\[CrossRef\]](#)
49. Kiranyaz, S.; Avci, O.; Abdeljaber, O.; Ince, T.; Gabbouj, M.; Inman, D.J. 1D convolutional neural networks and applications: A survey. *Mech. Syst. Signal Process.* **2021**, *151*, 107398. [\[CrossRef\]](#)
50. MAlkhodari, M.; Fraiwan, L. Convolutional and recurrent neural networks for the detection of valvular heart diseases in phonocardiogram recordings. *Comput. Methods Programs Biomed.* **2021**, *200*, 105940. [\[CrossRef\]](#)
51. 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. Available online: http://www.jmlr.org/papers/volume15/srivastava14a/srivastava14a.pdf?utm_content=buffer79b43&utm_medium=social&utm_source=twitter.com&utm_campaign=buffer (accessed on 20 December 2023).

52. Jason, B. A Gentle Introduction to Batch Normalization for Deep Neural Networks—Machine Learning Master. Available online: <https://machinelearningmastery.com/batch-normalization-for-training-of-deep-neural-networks/> (accessed on 20 December 2022).
53. Zhang, B.; Liu, J. Discriminative Convolutional Sparse Coding of ECG Signals for Automated Recognition of Cardiac Arrhythmias. *Mathematics* **2022**, *10*, 2874. Available online: <https://www.mdpi.com/1771970> (accessed on 20 December 2022). [[CrossRef](#)]
54. Heaton, J. Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep learning. *Genet. Program. Evolvable Mach.* **2018**, *19*, 305–307. [[CrossRef](#)]
55. Jozefowicz, R.; Vinyals, O.; Schuster, M.; Shazeer, N.; Wu, Y. Exploring the Limits of Language Modeling. *arXiv* **2016**, arXiv:1602.02410. [[CrossRef](#)]
56. Guo, L.; Sim, G.; Matuszewski, B. Inter-patient ECG classification with convolutional and recurrent neural networks. *Biocybern. Biomed. Eng.* **2019**, *39*, 868–879. [[CrossRef](#)]
57. Singh, V.; Reddy, U.S.; Bhargava, G.M. A Generic and Robust System for Automated Detection of Different Classes of Arrhythmia. *Procedia Comput. Sci.* **2020**, *167*, 1801–1810. [[CrossRef](#)]
58. Singh, S.; Pandey, S.K.; Pawar, U.; Janghel, R.R. Classification of ECG arrhythmia using recurrent neural networks. *Procedia Comput. Sci.* **2018**, *132*, 1290–1297. Available online: <https://www.sciencedirect.com/science/article/pii/S1877050918307774> (accessed on 20 December 2022). [[CrossRef](#)]
59. Kachuee, M.; Fazeli, S.; Sarrafzadeh, M. ECG heartbeat classification: A deep transferable representation. In Proceedings of the 2018 IEEE International Conference on Healthcare Informatics, ICHI 2018, New York, NY, USA, 4–7 June 2018; pp. 443–444. [[CrossRef](#)]
60. Essa, E.; Xie, X. An ensemble of deep learning-based multi-model for ECG heartbeats arrhythmia classification. *IEEE Access* **2021**, *9*, 103452–103464. [[CrossRef](#)]
61. Shadmand, S.; Mashoufi, B. A new personalized ECG signal classification algorithm using block-based neural network and particle swarm optimization. *Biomed. Signal Process. Control.* **2016**, *25*, 12–23. [[CrossRef](#)]
62. 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](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.