A Shallow Domain Knowledge Injection (SDK-Injection) Method for Improving CNN-Based ECG Pattern Classification

Soyeon Oh and Minsoo Lee *

Department of Computer Science and Engineering, Ewha Womans University, Seoul 03760, Korea; soyeon.oh@ewha.ink
* Correspondence: mlee@ewha.ac.kr; Tel.: +82-2-3277-3401

Featured Application: The proposed method can improve the accuracy of an existing parameter-optimized CNN for ECG pattern classification. Especially, by applying the proposed method to single ECG-based analysis, the existing accuracy can be improved to a level similar to, or higher than, that of multi-ECG-based analysis. Since wearable devices, such as smart watches, mainly measure a single ECG stream, the proposed method can be applied to the core technologies of wearables-based smart hospital or healthcare services. In addition to aiding heart disease diagnoses, it can be applied to various IoT services, based on general analyses using ECG.

Abstract: ECG pattern classification for identifying the progress status of various heart diseases is a typical nonlinear problem. Therefore, deep learning-based automatic ECG diagnosis is being widely studied, and for this purpose, the CNN is mainly used to classify ECG patterns. In this case, it is hard to expect any further improvement in accuracy after optimizing the parameters. We propose a shallow domain knowledge injection method that can improve the accuracy of the existing parameter-optimized CNN. The proposed method can improve the accuracy by effectively injecting shallow domain knowledge, that can be acquired by non-medical experts, into the existing parameter-optimized CNN. The experiments show that the proposed method can be applied to both heart disease diagnoses and general ECG classification tasks, while improving the existing accuracy for both types of tasks.

Keywords: convolutional neural network; attention mechanism; ECG; time series data; classification

1. Introduction

ECG (electrocardiogram) patterns reflect the medical characteristics of the progress status for various heart diseases and are used as major criteria for diagnosing heart diseases. Traditionally, medical experts have diagnosed various heart diseases by analyzing ECG patterns, based on deep domain knowledge, which is specialized medical knowledge acquired through internal medicine theory and long-term clinical training. Meanwhile, as the usage of wearable devices, such as smart watches, increases, the need for smart hospital or healthcare services based on the IoT environment is increasing. Therefore, automatic ECG diagnosis based on ECG pattern classification is one of the core technologies necessary in such smart hospital or healthcare services.

ECG pattern classification on the status of various heart diseases is a typical nonlinear problem. Traditional data mining-based automatic ECG diagnosis techniques require ECG segmentation-based feature extraction to classify ECG patterns of various heart disease progress states [1–4], and, based on this, a diagnosis algorithm for each heart disease is developed [5,6]. Recent deep learning-based automatic ECG diagnosis techniques are being studied in various ways with emphasis on important heart diseases that cause emergency situations [7–12], while showing higher performance than traditional data mining-based techniques. Most deep learning-based automatic ECG diagnosis
methods diagnose each heart disease by applying a neural network to the results of ECG segmentation preprocessing [13–15].

Deep learning-based automatic ECG diagnosis techniques use CNN in various ways to classify ECG patterns. Deep learning is unexplainable [16,17], and the most basic approach to reach target accuracy is parameter optimization. However, it is generally hard to expect any further improvement in accuracy after the parameters are optimized. Meanwhile, ECG-based heart disease diagnosis by medical experts with deep domain knowledge has been successfully performed from the past to the present and is the basis of modern clinical medicine. Therefore, it could be expected that the accuracy of existing parameter-optimized CNN for ECG classification could be improved by applying such deep domain knowledge.

However, it is hard to apply deep domain knowledge directly into CNNs. First, using deep domain knowledge, in other words, applying the medical characteristics of ECG waveforms to classify ECG patterns, is a very complex nonlinear problem. Second, due to the fact that deep learning is unexplainable, it is hard to establish a theoretical background that can reflect deep domain knowledge into the model design. Finally, since the subjects performing deep learning-based ECG analysis are non-medical experts, such as data analysts and developers, it is impossible to utilize deep domain knowledge without the help of medical experts—this eventually leads to a cost problem.

In this paper, we propose a shallow domain knowledge injection (SDK-Injection) method that can additionally improve the accuracy of the existing parameter-optimized CNN. The proposed method applies curve fitting-based smoothing and an attention mechanism to inject shallow domain knowledge for a single ECG stream and expands the resulting variants to multivariate input. Through this, it is possible to effectively inject shallow domain knowledge that can be acquired by non-medical experts into the existing parameter-optimized CNN, and further improve the existing accuracy. The experiments show that the proposed method can be applied to both heart disease diagnosis and general ECG classification tasks and improve the accuracy for both types of tasks.

The contributions of this study are as follows:

- We propose a method to further improve the accuracy of the existing parameter-optimized CNN by using accumulated medical domain knowledge, which could not be utilized due to the inexplicability of deep learning.
- By utilizing shallow domain knowledge that can be acquired even by non-medical experts, it is possible to further improve the existing accuracy while minimizing the cost of introducing medical experts.
- By utilizing the existing CNN parameters that have already been optimized, the accuracy can be further improved without the cost of an additional parameter optimization process.
- By applying the proposed method to the single ECG-based analysis, the original accuracy can be improved to a level similar to, or higher than, the accuracy of the multi-ECG-based analysis. As a result, it can be applied to smart hospital or healthcare services using wearable devices, such as smart watches, that measure just a single ECG stream.

2. Related Work
2.1. Deep and Shallow Domain Knowledge for the ECG Pattern Classification

The ECG pattern consists of sub-patterns, such as P, Q, R, S, and T waves. Each of these sub-patterns, in a medical sense, is closely related to the status of various heart diseases. For example, for patients with ischemia, a characteristic T wave, different from the normal state, is observed. Medical experts can accurately recognize the non-linear characteristics of these T waves by using internal medicine theories and expertise. Through long-term clinical training. In this paper, we will refer to deep domain knowledge as the knowledge that can be used at the level of medical experts who have been trained for a long time. Medical experts with such deep domain knowledge have successfully performed ECG-based heart disease diagnosis from the past to the present, and this is the
basic diagnosis method in modern clinical medicine. Therefore, the accuracy of the existing parameter-optimized CNN may be improved by applying the deep domain knowledge about ECG patterns. However, it is not easy to apply such deep domain knowledge to deep learning-based ECG classification tasks. This is because such deep domain knowledge, in other words, the ability to classify the characteristic waveform of ECG from a medical point of view, is a very complex, nonlinear problem. In addition, due to the inexplicability of deep learning, it is very difficult to design a model that sufficiently reflects deep domain knowledge. Moreover, since data scientists or developers who perform deep learning-based ECG pattern classification tasks are non-medical experts, it is almost impossible to acquire or utilize deep domain knowledge without the help of medical experts. Additionally, the introduction of medical experts eventually causes a problem of cost. On the other hand, knowledge such as sub-patterns, and their approximate ranges, which are the criteria for diagnosing a specific heart disease, can be acquired relatively easily even by non-medical experts. That is, non-medical experts cannot know the detailed medical characteristics of the T wave of ischemia, but at least it can be known that the T wave plays a major role in the diagnosis of ischemia. This knowledge can also be acquired through pre-investigation for the data analysis or visualization of data samples. Via a comparison of this knowledge with deep domain knowledge, we refer to it as shallow domain knowledge.

2.2. Automatic ECG Diagnosis and ECG Pattern Classification

Most ECG signals measured from sensors contain noise. Since such noise can distort the characteristics, most automatic ECG diagnoses perform preprocessing to remove the noise. For preprocessing, smoothing is mainly used. A commonly used ECG smoothing technique in recent studies is FIR [18]. However, in this study the Savitzky–Golay filter [19,20], a curve fitting-based smoothing technique that has been traditionally used for ECG preprocessing, was used.

Classification of ECG patterns on the status of various heart diseases is a typical nonlinear problem. Therefore, traditional data mining-based, automatic ECG diagnosis techniques require ECG segmentation-based feature extraction to classify ECG patterns for various heart disease states [1–4]. ECG segmentation is a technique for recognizing major sub-patterns that reflect the medical characteristics of ECG patterns, such as P and Q waves, QRS complex, RR interval, and ST segment. Such ECG segmentation was researched using data mining in [1,2]. Recently, signal processing-based ECG segmentation methods, such as those presented in [3,4], are also being studied. Data mining based automatic ECG analysis techniques first perform ECG segmentation, as in [1–4], and then develop a diagnosis algorithm for each heart disease by characterizing the recognized sub-patterns [5,6]. A system architecture for automatic ECG diagnosis is proposed in [21], which classifies ECG data using the decision tree and multilayer perceptron (MLP) after performing preprocessing and data mining-based feature extraction. Although these data mining-based automatic ECG diagnosis techniques are explainable, they are not suitable for real-time analysis in a wearable-based environment.

Recently, deep learning-based automatic ECG diagnosis techniques are being actively studied. These deep learning-based automatic ECG diagnosis techniques are being studied preferentially for important heart diseases related to emergency situations, such as atrial fibrillation and myocardial infarction [7–12]. In addition, they show high performance compared with traditional data mining-based methods. Most of the deep learning-based automatic ECG diagnosis techniques [13–15] studied so far diagnose each heart disease by applying a neural network to the results of conventional ECG segmentation, such as in [1–4]. Ref. [13] diagnoses arrhythmia by analyzing with CNN applied to sub-patterns that were obtained by applying ECG segmentation to the result of wavelet transform preprocessing. Refs. [14,15] also extract features through existing ECG segmentation and then analyze them based on CNN. Ref. [22] proposes an automatic ECG diagnosis system which analyzes ECG data collected from a mobile environment after performing preprocessing and feature extraction. Especially, [22] uses sliding segmentation instead of the existing
fixed segmentation to extract features and proposes an advanced neural network based on a residual network using a large kernel for the computational efficiency during the model training phase. However, although this traditional ECG segmentation plays a very important role, it may be considered unsuitable for real-time analysis because of its complex steps. Therefore, studies on deep learning-based ECG segmentation techniques, such as [23,24], are being attempted, but additional research is needed due to the inexplicability of deep learning.

Deep learning-based automatic ECG diagnosis techniques use CNN in various ways to classify ECG patterns. As in [7,8], ECG analysis is performed by ensembles of various types of neural networks and CNNs. Here, CNN is introduced to extract features for pattern classification and RNN is introduced to reflect periodicity, such as heart rate. Ref. [9] also extracts features by using the CNN instead of the segmentation-based, feature extraction method, and passes them to the LSTM to analyze atrial fibrillation. It uses the focal loss function to increase the analysis accuracy of imbalanced data. Additionally, an attention mechanism is introduced so that the periodicity of the ECG signal can be used heavily for analysis. In this case, as in [10–12], by applying the attention mechanism based on the context generated from RNNs, such as an LSTM, the periodic characteristics of the ECG signal are reflected in the analysis result. Additionally, as shown in [25,26], the feature extraction performance is improved by applying the attention mechanism to the CNN. The SDK-Injection method proposed in this paper adds variants generated by applying curve fitting-based smoothing and SDK-Injecting attention to a single ECG stream to improve accuracy. Similarly, Ref. [27] tried to improve the accuracy by extending a single ECG stream to 2D spectrogram and applying 2D-CNN instead of 1D-CNN. However, the SDK-Injection method proposed in this paper differs from [27] in that it does not generate an image-type spectrogram, but constructs a multivariate input by adding variants. In addition, unlike [27], since it constructs a multivariate input, rather than an image, the existing 1D-CNN model is used to improve the analysis accuracy. Although deep learning-based automatic ECG diagnosis techniques are being studied in various ways, there is still a limitation in that they are model design techniques, optimized for the purpose of diagnosing a specific heart disease.

3. SDK-Injection Method

The proposed SDK-Injection method is designed, as shown in Figure 1, by combining the following three techniques, and through this, the accuracy of the original parameter-optimized CNN for ECG pattern classification, that is, the Base-CNN, can be further improved.

- Technique 1: Curve fitting-based smoothing;
- Technique 2: SDK-Injecting attention;
- Technique 3: Expansion to multivariate.

The proposed method applies curve fitting-based smoothing and an attention mechanism to inject shallow domain knowledge on the single ECG stream pattern which is the original data input. Additionally, the resulting variants are expanded to multivariate input delivered to the Base-CNN. Through this, it is possible to effectively inject into Base-CNNs the shallow domain knowledge at a level that non-medical experts can acquire, and the accuracy of the original Base-CNN can be further improved. The experimental results in Section 4 show that the SDK-Injection method, designed by combining all three techniques, shows the best performance when compared with the methods designed with other possible combinations of Technique 1, Technique 2, and Technique 3.
Figure 1. SDK-Injection method proposed in this paper. It takes univariate original data, which is a single ECG stream pattern, and applies three techniques. Technique 1 is curve fitting-based smoothing by Savitzky–Golay filter and Technique 2 is SDK-Injecting attention to inject shallow domain knowledge. Technique 3 is expansion to multivariate, which expands the univariate input to multivariate input by adding three variants generated from the original data by applying Technique 1 and Technique 2. Multivariate input, generated by applying the SDK-Injection method, is passed to the existing parameter-optimized Base-CNN.

Technique 1 applies curve fitting-based smoothing to the original data. Through this, the adverse effect of the noise included in the original data could be offset during the feature extraction process. Technique 2 inserts shallow domain knowledge through an attention mechanism. The resulting data from applying Technique 2 has a relatively high weight for the sub-pattern range, in other words, the shallow domain knowledge, which is known as important criteria for diagnosis of target heart disease. Technique 3 composes a
four-variate input by adding three variants to the original data. Variants to be added are smoothed data, attention output, and smoothed attention output, as shown in Figure 1. Smoothed data is data from which noise has been removed by applying Technique 1 to the original data. Attention output is data with high weight for shallow domain knowledge by applying Technique 2 to the original data. Smoothed attention output is data obtained by applying Technique 2 to the smoothed data, obtained by applying Technique 1 to the original data. That is, the smoothed attention output is data with high weights for shallow domain knowledge in the noise-removed ECG pattern. Technique 3 composes the original data, smoothed data, attention output, and smoothed attention output in the form of a two-dimensional array and delivers it as input to the existing parameter-optimized Base-CNN.

In particular, Technique 3 which is the expansion to multivariate technique that composes a four-variate input by adding three variants to the original single ECG stream pattern, contributes greatly to the improvement of accuracy through SDK-Injection. This is shown by the experimental results in Section 4, where the performance of the SDK-Injection method designed including Technique 3 is remarkably better than the methods designed only with the combination of Technique 1 and Technique 2. The three techniques constructing the SDK-Injection method are described in more detail below.

3.1. Technique 1: Curve Fitting-Based Smoothing

Unlike ECG waveforms visualized by professional equipment in hospitals, which are generally used for heart disease diagnosis by medical experts, an ECG stream, directly collected through a sensor attached to a wearable device, contains noise. Such noise may cause distortion of the characteristics of the ECG waveform. This may degrade the classification performance and also reduce the effectiveness of shallow domain knowledge injection. Therefore, the SDK-Injection method introduces curve fitting-based smoothing to deal with this noise.

SDK-Injection method uses the Savitzky–Golay filter for curve fitting-based smoothing. The Savitzky–Golay filter is a curve fitting method that fits points within a given window interval, based on a polynomial function of a given order. Figure 2 shows the result of applying the Savitzky–Golay filter to the sample pattern extracted from the datasets used in the experiment in Section 4. The black line is the data before curve fitting, and all of the data contain noise; although, there may be a difference depending on the dataset. The red line is smoothed data after curve fitting by the Savitzky–Golay filter, and because outliers caused by noise are corrected it is easy to check the characteristics and trends of ECG patterns.

The degree of smoothing by the Savitzky–Golay filter, that is, the degree of noise correction, is controlled by two parameters: window size and polynomial order. However, the window size must always be an odd number, and the polynomial order must be smaller than the window size. At this time, if the polynomial order is set similar to the window size, the curve fitting algorithm cannot work properly. On the other hand, if the polynomial order is too small, important feature points may be lost. Therefore, it is necessary to set a polynomial order of an appropriate size compared to the window size. That is, it is not the window size alone that affects the smoothing degree, but the value of the appropriate polynomial order compared to the window size.

Figure 3 shows the original patterns and smoothed patterns by changing the size of the polynomial order, while fixing the window size to 21 for the 2 patterns. The top of Figure 3 is the smoothing result with the window size set to 21 and polynomial order set to 17. As such, if the polynomial order is set similar to the window size, the curve fitting algorithm does not work properly. On the other hand, when the polynomial order is 7 or less, in Figure 3, the sub-pattern containing the maxima with the maximum value, commonly known as the R wave, is lost. These R waves are very important in ECG analysis. Therefore, in all the experiments in this paper, smoothing was performed with fixed parameters, such as window size 21 and polynomial order 13, which can correct the noise without damaging the R wave, while the curve fitting algorithm works well.
Figure 2. Original (black) and smoothed (red) pattern samples from five datasets used in the experiments. Smoothed patterns were generated by Savitzky–Golay filter with window size 21 and polynomial order 13.

The degree of smoothing by the Savitzky–Golay filter, that is, the degree of noise correction, is controlled by two parameters: window size and polynomial order. However, the window size must always be an odd number, and the polynomial order must be smaller than the window size. At this time, if the polynomial order is set similar to the window size, the curve fitting algorithm cannot work properly. On the other hand, if the polynomial order is too small, important feature points may be lost. Therefore, it is necessary to set a polynomial order of an appropriate size compared to the window size. That is, it is not the window size alone that affects the smoothing degree, but the value of the appropriate polynomial order compared to the window size.

Figure 3 shows the original patterns and smoothed patterns by changing the size of the polynomial order, while fixing the window size to 21 for the 2 patterns. The top of Figure 3 is the smoothing result with the window size set to 21 and polynomial order set to 17. As such, if the polynomial order is set similar to the window size, the curve fitting algorithm does not work properly. On the other hand, when the polynomial order is 7 or less, in Figure 3, the sub-pattern containing the maxima with the maximum value, commonly known as the R wave, is lost. These R waves are very important in ECG analysis. Therefore, in all the experiments in this paper, smoothing was performed with fixed parameters, such as window size 21 and polynomial order 13, which can correct the noise without damaging the R wave, while the curve fitting algorithm works well.
Figure 3. Smoothing effects according to the polynomial orders of Savitzky–Golay filter. The window size was fixed at 21 and polynomial orders were applied to the 2 sample patterns from the ECG200 dataset in the order of 17, 13, 5, and 3, from top to bottom in the figure.

3.2. Technique 2: SDK-Injecting Attention

The two ECG patterns in Figure 4 are those of an ischemia patient and a healthy person, respectively. The sub-pattern that is the criterion for diagnosing ischemia is known as the T wave. Medical experts diagnose ischemia by comprehensively considering the shape of the T wave, that is, the location and curvature, and the relationship between the surrounding sub-patterns. This is deep domain knowledge for ischemia diagnosis, with expertise accumulated through internal medicine theory and long-term clinical training. Data scientists or developers who perform deep learning-based ECG analysis cannot acquire or utilize such deep domain knowledge because they are non-medical experts. Even if medical experts are invited to use deep domain knowledge at a high cost, it is not easy to apply such deep domain knowledge to deep learning. This is because deep learning is not yet sufficiently explainable.

The SDK-Injection method improves classification accuracy by using only the fact that the T wave part is an important sub-pattern for ischemia diagnosis, instead of using such deep domain knowledge. That is, only the range of the T wave part shown in green in Figure 4 is shallow domain knowledge used in ischemia diagnosis based on the SDK-Injection method. This shallow domain knowledge is simple medical fact that non-medical experts, such as data scientists and developers, can find through research. If doing this is not easy, shallow domain knowledge can be grasped in a similar way that a typical data exploration task is performed before a full-scale data analysis. In other words, just by visualizing a few sample patterns in the training dataset, it is possible to roughly grasp the range of sub-patterns that make the significant difference between the two patterns.

The SDK-Injection method injects such shallow domain knowledge through SDK-Injecting Attention, as shown in Figure 4. The SDK-Injection method receives a scope of
sub-patterns that are the diagnostic criteria, or the shallow domain knowledge, through a predefined interface. Then, the shallow domain knowledge part is filled with 1 and the rest of the part is filled with 0, and the metadata with same length as the original data is generated. After concatenating the original data and the metadata, it is delivered to the dense layer, having softmax as an activation function. The output of this dense layer is the attention score, shown in Figure 4, which is a probability vector indicating the weight that each part of the original data has in the pattern classification task, according to the shallow domain knowledge injection. Additionally, the attention output can be obtained as the result of element-wise product for these attention scores and original data. The attention output is the ECG pattern data for the classification task in which each part of the original data is weighed according to the shallow domain knowledge. By passing this attention output to the input of the existing parameter-optimized Base-CNN, pattern classification reflecting shallow domain knowledge can be performed.

3.2. Technique 2: SDK-Injecting Attention

The two ECG patterns in Figure 4 are those of an ischemia patient and a healthy person, respectively. The sub-pattern that is the criterion for diagnosing ischemia is known as the T wave. Medical experts diagnose ischemia by comprehensively considering the shape of the T wave, that is, the location and curvature, and the relationship between the surrounding sub-patterns. This is deep domain knowledge for ischemia diagnosis, with expertise accumulated through internal medicine theory and long-term clinical training. Data scientists or developers who perform deep learning-based ECG analysis cannot acquire or utilize such deep domain knowledge because they are non-medical experts. Even if medical experts are invited to use deep domain knowledge at a high cost, it is not easy to apply such deep domain knowledge to deep learning. This is because deep learning is not yet sufficiently explainable.

![Figure 4. T wave part as shallow domain knowledge (green) and SDK-Injecting attention. The sample patterns are for the diagnosis of ischemia. The fact is known that the shape of the T wave is one of the main criteria for the diagnosis of ischemia. Therefore, the range of the T wave part (green) is used as shallow domain knowledge and injected by SDK-Injecting attention.](image)

3.3. Technique 3: Expansion to Multivariate

The SDK-Injecting method constructs a multivariate input by adding three variants to the original data, which are single ECG pattern data, as shown in Figure 5, in order to effectively reflect the shallow domain knowledge to the classification task. Additionally, it is transferred to the existing parameter-optimized Base-CNN. The three additional variants are smoothed data, attention output and smoothed attention output. Smoothed data can
be created by applying curve fitting-based smoothing of Technique 1 to the original data. Attention output can be generated by applying SDK-Injecting attention of Technique 2 to the original data. Additionally, the smoothed attention output can be obtained by applying Technique 2 to the smoothed data, generated through Technique 1.

Figure 5. Expansion from univariate to multivariate by adding variants. Added variants, such as smoothed data, attention output, and smoothed attention output, are generated from the original data.

Technique 3 composing the SDK-Injection method constructs a 2D-matrix multivariate form of input by adding, as variants, the smoothed data, attention output, and smoothed attention output, generated from the original data. A classification task with original data, which is a single ECG stream, as input is a univariate problem. However, the classification task with the data obtained through Technique 3 as new input is a multivariate problem. In other words, Technique 3 extends the existing single ECG-based analysis to a four-variate analysis. Therefore, features can be extracted by considering original data, smoothed data, attention output, and smoothed attention output, simultaneously. In addition, the existing parameters of the parameter-optimized Base-CNN can be used as they are without additional training. This is because the neural network structure is not changed while expanding to multivariate by constructing a multivariate input in the form of a 2D-matrix. The experiment in Section 4 shows that the classification accuracy is higher for the case when expansion through Technique 3 is performed following Technique 1 and Technique 2, than it is for the case when Technique 2 is applied alone. This means that the core technology
of the SDK-Injection method is Technique 3, which effectively applies the shallow domain knowledge injected through Technique 2 to the existing Base-CNN.

Additionally, in the experiment in Section 4, due to Technique 3, the SDK-Injection method to a single ECG-based analysis achieves higher accuracy than when performing multiple ECG-based analysis. These experimental results show the possibility that the SDK-Injection method can be applied to a smart hospital or healthcare service in a wearable-based environment that measures a single ECG stream. ECG-based diagnosis in hospitals by medical experts is a typical multivariate problem. In order to diagnose a disease based on ECG, a hospital measures up to 12 ECG streams by attaching several electrodes to body positions determined according to medical standards. Since the shape of the ECG stream induced by each electrode is slightly different, it is said that the most accurate diagnosis is possible when these 12 lead ECG streams are considered simultaneously. In fact, some heart diseases can only be accurately diagnosed by measuring multiple ECG streams. Therefore, such multi-ECG stream-based analysis is not suitable for wearable-based smart hospital or healthcare services. On the other hand, Technique 3 of the SDK-Injection method generates variants from a single ECG stream and uses them to expand to multivariate, so it is suitable for a wearable-based environment that measures a single ECG stream.

In Technique 3, when generating variants from a single ECG stream, Technique 1’s curve fitting-based smoothing and Technique 2’s SDK-Injecting attention are utilized. However, as another option for generating variants from a single ECG stream, we can consider generative neural networks. That is, a generative neural network can be trained to generate variants reflecting each characteristic of 12 lead ECG streams, measured by specialized devices in a hospital. However, training of generative neural networks is very tricky because it is based on the min–max problem. In contrast, the variants from Technique 3 of the SDK-Injection were generated by transforming a single ECG stream, based only on shallow domain knowledge. In other words, Technique 3 is efficient in that it is easy to generate variants and expand to multivariate based on this, while showing stable training performance compared with generative neural networks.

4. Performance Evaluation Results and Discussion

4.1. Dataset and Environment

The proposed SDK-Injection method, Base-CNN, and all comparison methods in the experiment were all developed based on Keras 2.2.4 with Tensorflow 1.14.0 as a backend, and were trained using NVIDIA GeForce GTX 965M GPU. A total of five datasets, as shown in Table 1, were used for model training and testing. These data are all from the UEA and UCR Time Series Classification Repository [28] and the datasets created from data of PhysioNet [29]. AF-D1, AF-D2, and TwoLead have very small train datasets. For AF-D1, AF-D2, and ECG200, the ratio of training dataset size to test dataset size was 1:1. On the other hand, in ECG5000 and TwoLead, the ratio of the training dataset size to the test dataset size was 1:9 and 1:50, respectively.

Among them, there are a total of four datasets for diagnosing heart disease—AF-D1, AF-D2, ECG200, and ECG5000—and through these, three heart diseases can be diagnosed: atrial fibrillation, ischemia, and congestive heart failure. AF-D1 and AF-D2 are multivariate datasets for the diagnosis of atrial fibrillation. In this paper, the proposed SDK-Injection method was applied by using AF-D1 and AF-D2 as univariate inputs. For comparison, AF-D1 and AF-D2 were composed as multivariate inputs and were analyzed based on the Base-CNN. For the four datasets related to the diagnosis of heart disease, shallow domain knowledge can be derived from the diagnostic criteria of each heart disease. Meanwhile, TwoLead is a dataset for general ECG classification not related to heart disease diagnosis. Therefore, there are no medical diagnostic criteria that can be used as shallow domain knowledge.
Table 1. Dataset Description.

| No. | Dataset   | Size (Train) | Size (Test) | Pattern Length | No. of Classes | Class Ratio (Train) | Class Ratio (Test) | Standard Deviation (Class Ratio) | Target Disease |
|-----|-----------|--------------|-------------|----------------|----------------|--------------------|-------------------|----------------------------------|----------------|
| 1   | AF-D1     | 15           | 15          | 640            | 3              | 1                  | 1                 | 0                                | Atrial Fibrillation |
| 2   | AF-D2     | 15           | 15          | 640            | 3              | 1                  | 1                 | 0                                | Atrial Fibrillation |
| 3   | ECG200    | 100          | 100         | 96             | 2              | 1                  | 2.23              | 1.78                             | ischemia        |
| 4   | ECG5000   | 500          | 4500        | 140            | 5              | 146                | 119.41            | 72.27                            | Congestive Heart Failure |
| 5   | TwoLead   | 23           | 1139        | 82             | 2              | 1.09               | 1                 | 0.03                             | Non-disease      |

Class Ratio (Train) and Class Ratio (Test) in Table 1 are the distribution ratios between class labels for the training and test datasets of each dataset. For example, in AF-D1, the ratio of the three classes is 1:1:1 for a training dataset consisting of 15 patterns, so each class is uniformly distributed with 5 patterns. In ECG5000, the ratio of 5 classes is 119.41:72.27:3.91:7.95:1 for the test dataset consisting of 4500 patterns, so each class is non-uniformly distributed with 2627, 1590, 86, 175, and 22 patterns, respectively. In this paper, the degree of imbalance in the class distribution for each dataset is quantified by Equation (1), and this result is shown in Table 1 under “Standard Deviation (Class Ratio)”.

\[
\sigma = \sqrt{\frac{\sum_{i=1}^{n} (R_{Traini} - \bar{R}_{Train})^2}{n}} + \sqrt{\frac{\sum_{i=1}^{n} (R_{Testi} - \bar{R}_{Test})^2}{n}}
\]  

(1)

\(R_{Train}\) and \(R_{Test}\) are class distribution ratio vectors of the training and test datasets in each of the datasets. That is, for each dataset, \(R_{Train}\) is composed of the Class Ratio (Train) value in Table 1, and \(R_{Test}\) is composed of the Class Ratio (Test) value in Table 1. \(n\) is the number of classes in each dataset. As shown in Equation (1), \(\sigma\) is the average of the standard deviation of \(R_{Train}\) and the standard deviation of \(R_{Test}\), and quantifies the class imbalance of the dataset. That is, as the Standard Deviation (Class Ratio) value of Table 1 is closer to 0 it is a balanced dataset with an even class distribution, and as this value increases it is an imbalanced dataset in which classes are unevenly distributed. From this, it can be seen that AF-D1, AF-D2, and TwoLead are balanced datasets, and ECG200 and ECG5000 are imbalanced datasets. In addition, it can be seen that the class distribution of ECG5000 is more uneven than ECG200.

4.2. Experiments and Results

In this paper, the performance of the proposed method is verified through a total of five experiments. Experiment 1 verifies the performance of the proposed method by comparing the accuracy of the existing Base-CNN with the case of applying the SDK-Injection method using the 5 datasets. Experiment 2 compares the performance when the SDK-Injection method is applied with AF-D1 and AF-D2 as univariate inputs, respectively, and when multivariate analysis is performed based on the Base-CNN, composed of multivariate inputs. In Experiment 3, the design of the SDK-Injection method, which combines the three techniques of Technique 1, Technique 2, and Technique 3, is experimentally verified.
Experiment 4 verified how the optimized shallow domain knowledge derived from medical facts affects the accuracy improvement. Finally, Experiment 5 experimentally validates one kind of method that can derive shallow domain knowledge for general ECG classification tasks that are not for the purpose of diagnosing of heart disease.

The five experiments evaluate performance of the SDK-Injection and all other compared methods, according to the following six metrics:

- **Accuracy** = Test accuracy of the model (%);
- **Precision** = \( \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \);
- **Sensitivity (Recall)** = \( \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \);
- **Specificity** = \( \frac{\text{True Negative}}{\text{True Negative} + \text{False Positive}} \);
- **F1 − Score** = \( \frac{2 \times \text{Precision} \times \text{Sensitivity}}{\text{Precision} + \text{Sensitivity}} \);
- **Training Time** = Training time of the model (sec).

Accuracy is the test accuracy of the model evaluated based on the test dataset, and training time is the measured training time of the model in seconds. The precision is the ratio of the actual target labels among those classified as target labels by the model. Sensitivity or recall is the ratio of those classified as target labels by the model among those actual target labels. Additionally, since precision or sensitivity values may be biased in an imbalanced dataset, the performance can be evaluated using f1-score, which is the harmonic average of precision and sensitivity. Among the five datasets used in the experiment, there also exists multi-classification datasets with more than two classes. Therefore, the true positive, false positive, true negative, and false negative were calculated for each class in each dataset, and the precision, sensitivity, specificity, and f1-score for each class were calculated with this. Then, the metric values for each class were averaged to obtain the precision, sensitivity, specificity, and f1-score for each dataset. In each of the five experiments, the values of accuracy, precision, sensitivity, specificity, and f1-score were summarized in the form of a table. In particular, the accuracy values in each table and the training time measured during the model training phase were visualized in a graph form.

### 4.2.1. Result of Experiment 1

Table 2 summarizes the accuracy, precision, sensitivity, specificity, and f1-score when the proposed SDK-Injection method is applied to each dataset and when only the Base-CNN is applied. The last column of the table summarizes how much the performance has improved from the Base-CNN by applying the SDK-Injection method. Among the metrics in the table, the accuracy is visualized in Figure 6a. That is, Figure 6a is the result of comparing the accuracy when the proposed SDK-Injection method is applied to each dataset and the accuracy when the existing Base-CNN is applied. When the SDK-Injection method was applied to all five datasets, the accuracy was improved compared with the case when only the Base-CNN was applied. However, the improvement in accuracy differed depending on the dataset. Figure 6b shows the difference between the accuracy when the SDK-Injection method was applied to each dataset and when only the Base-CNN was applied, along with the class imbalance of each dataset. AF-D1, AF-D2, and TwoLead are balanced datasets. On the other hand, ECG200 and ECG5000 are imbalanced datasets. At this time, the accuracy improvement when SDK-Injection is applied is very large, from about 15.36% to 20% in balanced datasets. However, for imbalanced datasets, the accuracy improvement when SDK-Injection is applied is relatively low. In particular, AF-D1 and AF-D2 showed the most uniform class distribution with a Standard Deviation (Class Ratio) value of 0, and the accuracy improvement was the largest at 20% when the proposed technique was applied. TwoLead has a nearly uniform class distribution, but is relatively imbalanced dataset compared with AF-D1 and AF-D2. Additionally, it can be seen that the accuracy improvement when the proposed method is applied is 15.36%, which is slightly lower than AF-D1 and AF-D2. ECG5000, the most imbalanced dataset, shows the lowest accuracy improvement at 0.62%. That is, the accuracy improvement when the
SDK-Injection method is applied is inversely proportional to the degree of class imbalance of the dataset. As shown in Table 2, not only the accuracy but also the precision, sensitivity, specificity, and f1-score improved when SDK-Injection was applied to all datasets. Similar to the accuracy, the improvement is remarkable in the balanced dataset, and relatively small performance improvement is shown in the imbalanced dataset. Therefore, the proposed SDK-Injection method shows higher performance on a balanced dataset.

Table 2. Performance evaluation of Base-CNN and SDK-Injection.

| Metric  | Dataset | Base-CNN | SDK-Injection | Increasement by SDK-Injection |
|---------|---------|----------|---------------|-------------------------------|
| Accuracy (%) | AF-D1   | 26.67    | 46.67         | 20.00                         |
|         | AF-D2   | 40.00    | 60.00         | 20.00                         |
|         | ECG200  | 92.00    | 95.00         | 3.00                          |
|         | ECG5000 | 93.76    | 94.38         | 0.62                          |
|         | TwoLead | 69.89    | 85.25         | 15.36                         |
| Precision | AF-D1   | 0.30     | 0.47          | 0.17                          |
|         | AF-D2   | 0.33     | 0.70          | 0.37                          |
|         | ECG200  | 0.92     | 0.95          | 0.03                          |
|         | ECG5000 | 0.70     | 0.75          | 0.05                          |
|         | TwoLead | 0.72     | 0.86          | 0.14                          |
| Sensitivity | AF-D1   | 0.27     | 0.47          | 0.20                          |
|         | AF-D2   | 0.40     | 0.60          | 0.20                          |
|         | ECG200  | 0.92     | 0.95          | 0.03                          |
|         | ECG5000 | 0.52     | 0.58          | 0.06                          |
|         | TwoLead | 0.70     | 0.86          | 0.16                          |
| Specificity | AF-D1   | 0.63     | 0.73          | 0.10                          |
|         | AF-D2   | 0.70     | 0.80          | 0.10                          |
|         | ECG200  | 0.92     | 0.95          | 0.03                          |
|         | ECG5000 | 0.98     | 0.98          | 0.00                          |
|         | TwoLead | 0.70     | 0.86          | 0.16                          |
| F1-Score | AF-D1   | 0.28     | 0.46          | 0.18                          |
|         | AF-D2   | 0.35     | 0.60          | 0.25                          |
|         | ECG200  | 0.92     | 0.95          | 0.03                          |
|         | ECG5000 | 0.56     | 0.63          | 0.07                          |
|         | TwoLead | 0.70     | 0.86          | 0.16                          |

Table 3 summarizes the precision, sensitivity, specificity, and f1-score for each class in each dataset for the same experiment situation. Class s of AF-D1, class s of AF-D2, and class 3 of ECG5000 showed sensitivity values of 0.20, 0.00, and 0.07, respectively, and precision values of 0.14, 0.00, and 0.43 in the Base-CNN, respectively. That is, these classes have relatively low classification performance compared with other classes in the same dataset. Interestingly, by applying SDK-Injection, the classification performance of classes which previously had very low classification performance was greatly improved. The sensitivity of these three classes were improved to 0.40, 0.40, and 0.41, respectively, after applying the SDK-Injection. The precision was also improved to 0.50, 1.00, and 0.60, respectively. In particular, in the case of class 3 of ECG5000, an imbalanced dataset, the f1-score also improved from 0.12 to 0.49. In conclusion, by applying the SDK-Injection, the performance was improved for not only the accuracy but also the precision, sensitivity, specificity, and f1-score for all datasets. In addition, it can be seen that the SDK-Injection can significantly improve classification performance for those classes that previously showed low classification performance, as well as classes that have been classified well in the past.
SDK-Injection was applied to all datasets. Similar to the accuracy, the improvement is remarkable in the balanced dataset, and relatively small performance improvement is shown in the imbalanced dataset. Therefore, the proposed SDK-Injection method shows higher performance on a balanced dataset.

Figure 6. Performance evaluation when the SDK-Injection method is applied to each dataset: (a) test accuracy when SDK-Injection is applied to each dataset (red) and test accuracy of Base-CNN for each dataset (sky blue); (b) standard deviations of ratio of class labels (gold) for each dataset and difference in accuracy between SDK-Injection method and Base-CNN (dark blue). Ratio of class labels is shown in Table 1. The datasets with standard deviations close to 0 are balanced datasets.

Table 3. Performance evaluation of Base-CNN and SDK-Injection per classes.
4.2.2. Result of Experiment 2

In general, when medical experts diagnose heart disease in hospitals, multi-ECG streams are measured through a professional device and these are integrated into the consideration. In other words, the ECG-based heart disease problem is a multivariate problem. Among the datasets used in the experiment, AF-D1 and AF-D2 were originally multivariate datasets for the diagnosis of atrial fibrillation. When the Base-CNN is used, multivariate analysis for atrial fibrillation diagnosis can be performed by configuring AF-D1 and AF-D2 as multivariate inputs as shown in Figure 7. On the other hand, the proposed method uses each of AF-D1 and AF-D2 as a univariate input in the form of a single ECG stream. That is, the existing multivariate problem can be solved by applying the SDK-Injection method with each of AF-D1 and AF-D2 as univariate inputs.

![Figure 7. Dual-variate ECG analysis with Base-CNN using dataset AF-D1 and AF-D2 for the diagnosis of atrial fibrillation.](image)

Table 4 summarizes the accuracy, precision, sensitivity, specificity, and f1-score when solving the existing multivariate problem by applying the SDK-Injection method to the univariate input. Among them, the accuracy is visualized in Figure 8a. That is, Figure 8 shows the accuracy in (a) and the training time in (b) when solving the existing multivariate problem by applying the SDK-Injection method to the univariate input. When AF-D1 and AF-D2 are configured as multivariate inputs and analyzed using the Base-CNN, the accuracy is 20% higher than that of the Base-CNN-based analysis with AF-D1 as the univariate input. It is 6.67% higher than the accuracy when analyzing based on the Base-CNN with AF-D2 as a univariate input. This is because these AF-D1 and AF-D2 datasets for diagnosing atrial fibrillation are multivariate, and higher performance is shown when using multivariate input than when using univariate input based on a neural network with the same structure.
Table 4. Performance evaluation of univariate Base-CNN, multivariate Base-CNN, and SDK-Injection.

| Metric   | Dataset     | Base-CNN (Univariate) | Base-CNN with Multivariate (AF-D1, AF-D2) | SDK-Injection (Univariate) |
|----------|-------------|-----------------------|------------------------------------------|---------------------------|
| Accuracy (%) | AF-D1       | 26.67                 | 46.67                                    | 46.67                     |
|           | AF-D2       | 40.00                 |                                          | 60.00                     |
| Precision | AF-D1       | 0.30                  | 0.48                                     | 0.47                      |
|           | AF-D2       | 0.33                  |                                          | 0.70                      |
| Sensitivity | AF-D1      | 0.27                  | 0.47                                     | 0.47                      |
|           | AF-D2       | 0.40                  |                                          | 0.60                      |
| Specificity | AF-D1     | 0.63                  | 0.73                                     | 0.73                      |
|           | AF-D2       | 0.70                  |                                          | 0.80                      |
| F1-Score  | AF-D1       | 0.28                  | 0.45                                     | 0.46                      |
|           | AF-D2       | 0.35                  |                                          | 0.60                      |

Figure 7. Dual-variate ECG analysis with Base-CNN using dataset AF-D1 and AF-D2 for the diagnosis of atrial fibrillation.

Figure 8. Performance when applying SDK-Injection on univariate data compared with Base-CNN with multivariate data: (a) accuracy when applying Base-CNN (sky blue) and SDK-Injection (red) to univariate inputs AF-D1 and AF-D2, respectively, compared with the accuracy of the Base-CNN with dual-variate input composed with AF-D1 and AF-D2 (yellow); (b) additionally, training time for the same situation.

If the SDK-Injection method is applied with AF-D1 as a univariate input, it shows the same level of accuracy as that of the Base-CNN with multivariate input applied. In addition, the accuracy when applying the SDK-Injection method with AF-D2 as a univariate input is 13.33% higher than that of the Base-CNN with multivariate input applied. This shows that, by applying the SDK-Injection method, even though only the univariate input is used, equal or higher accuracy than the existing multivariate analysis can be achieved.

However, the SDK-Injection method took the longest training time. Nevertheless, this experimental result shows the possibility of effectively solving the existing multi-ECG stream-based analysis problem by applying SDK-Injection to a single ECG stream, measured through a wearable device. In particular, the precision, sensitivity, specificity, and f1-scores, shown in Table 4, are also higher when the SDK-Injection is applied to
AF-D2 than when the Base-CNN with multivariate input is applied. Additionally, when the SDK-Injection is applied to AF-D1, all metrics except the precision are the same or higher than when the Base-CNN with multivariate input is applied. This shows the possibility that the SDK-Injection method can be applied to smart hospital or healthcare services in a wearable-based environment.

4.2.3. Result of Experiment 3

SDK-Injection method consists of three techniques: Technique 1, Technique 2, and Technique 3. By combining these three techniques, various methods can be derived, as shown in Table 5 in addition to SDK-Injection. Table 5 specifies the types of input data and combinations of techniques to construct Base-CNN, SDK-Injection method, and other possible comparison methods. Additionally, this can be schematically shown in Figures 9 and 10. First, the proposed SDK-Injection method injects shallow domain knowledge into the existing parameter-optimized Base-CNN by applying all three techniques. As comparison methods for this, Compared 1 performs only Technique 1, that is, curve fitting-based smoothing. Compared 2 injects shallow domain knowledge by applying only Technique 2, that is, SDK-Injecting attention. Compared 3 consists of Technique 1 and Technique 2, where the SDK-Injecting attention of Technique 2 is applied to the smoothed data by Technique 1. Compared 4 is composed of Technique 1 and Technique 3, where Technique 3 constructs multivariate input by adding variants obtained via curve fitting-based smoothing by Technique 1. Compared 4 has three options according to the number of variants added. Compared 5 is composed of Technique 2 and Technique 3, where the SDK-Injecting attention injects shallow domain knowledge into the data which is added as variants to construct a dual-variate input.

Table 6 summarizes the accuracy, precision, sensitivity, specificity, and f1-score of each dataset when applying the proposed SDK-Injection, Base-CNN and other compared methods given in Table 5 and Figure 10. Among them, the accuracy is visualized in Figure 11a. That is, Figure 11 compares the performance of the proposed SDK-Injection, Base-CNN, and other comparison methods for each dataset. First, for all datasets, SDK-Injection shows the highest accuracy compared against Base-CNN or other comparison methods. Additionally, in Table 6, the precision, sensitivity, specificity, and f1-score of the SDK-Injection are the highest. However, as an exception, the specificity of ECG5000 has the same value of 0.98 in the SDK-Injection and all compared methods. In other words, it was empirically proved that it was appropriate to design the SDK-Injection method by combining all three techniques. In addition, we can infer which technique affects the improvement in accuracy by comparing the accuracy of other comparison methods.

Table 5. Description of SDK-Injection and other comparison methods derived from the combination of three techniques.

| No. | Name               | Technique 1 | Technique 2 | Technique 3 | Variants                                      |
|-----|--------------------|-------------|-------------|-------------|-----------------------------------------------|
| 1   | Base-CNN           | X           | X           | X           | 1: Original Data                              |
| 2   | SDK-Injection      | O           | O           | O           | 1: Original Data, 2: Smoothed Data (21 window size, 13 order), 3: Attention Output, 4: Smoothed Attention Output |
| 3   | Compared 1         | O           | X           | X           | 1: Original Data                              |
| 4   | Compared 2         | X           | O           | X           | 1: Original Data                              |
| 5   | Compared 3         | O           | O           | X           | 1: Original Data                              |
Table 5. Cont.

| No. | Name          | Technique 1 | Technique 2 | Technique 3 | Variants                                           |
|-----|---------------|-------------|-------------|-------------|---------------------------------------------------|
| 6   | Compared 4    | O           | X           | O           | 1: Original Data                                   |
|     | (add 1 variant) |             |             |             | 2: Smoothed Data (21 window size, 15 order)       |
| 7   | Compared 4    | O           | X           | O           | 1: Original Data                                   |
|     | (add 2 variants) |             |             |             | 2: Smoothed Data (21 window size, 15 order)       |
|     |               |             |             |             | 3: Smoothed Data (21 window size, 11 order)       |
| 8   | Compared 4    | O           | X           | O           | 1: Original Data                                   |
|     | (add 3 variants) |             |             |             | 2: Smoothed Data (21 window size, 15 order)       |
|     |               |             |             |             | 3: Smoothed Data (21 window size, 11 order)       |
|     |               |             |             |             | 4: Smoothed Data (21 window size, 9 order)        |
| 9   | Compared 5    | X           | O           | O           | 1: Original Data                                   |
|     |               |             |             |             | 2: Attention Output                                |

Figure 9. Schemes of SDK-Injection and other comparison method (Base-CNN) derived from the combination of three techniques.

In all datasets except AF-D2, the accuracy was improved when applying Compared 2, which consists of Technique 2 alone. Additionally, Compared 5, which consists of Technique 2 and Technique 3, also helped improve accuracy for all datasets except AF-D1. Compared 3, which consists of Technique 1 and Technique 2, helped to improve the accuracy for AF-D1, ECG5000, and TwoLead, and maintained the same accuracy in ECG200, but decreased the accuracy for AF-D2. The sensitivity and specificity of Compared 2 and Compared 5 also mostly increased than those of the Base-CNN. In summary, injecting shallow domain knowledge through SDK-Injecting attention of Technique 2 can help improve performance.
Table 6 summarizes the accuracy, precision, sensitivity, specificity, and f1-score of each dataset when applying the proposed SDK-Injection, Base-CNN and other compared methods given in Table 5 and Figure 10. Among them, the accuracy is visualized in Figure 11a. That is, Figure 11 compares the performance of the proposed SDK-Injection, Base-CNN, and other comparison methods for each dataset. First, for all datasets, SDK-Injection shows the highest accuracy compared against Base-CNN or other comparison methods. Additionally, in Table 6, the precision, sensitivity, specificity, and f1-score of the SDK-Injection are the highest. However, as an exception, the specificity of ECG5000 has the same value of 0.98 in the SDK-Injection and all compared methods. In other words, it was empirically proved that it was appropriate to design the SDK-Injection method by Figure 10.

Figure 10. Schemes of other comparison methods (from Compared 1 to Compared 5) derived from the combination of three techniques.

On the other hand, Compared 1, consisting of only curve fitting-based smoothing of Technique 1, helped to improve the accuracy for AF-D1, AF-D2, and ECG5000, but decreased the accuracy in TwoLead, and maintained the same accuracy for ECG200. Additionally, Compared 4, which adds variants generated through Technique 1, did not help to improve accuracy for the rest of the datasets except for ECG200 and ECG5000. Even for TwoLead, when Compared 4 and Compared 1 were applied, the accuracy was rather decreased. When we consider the comparison methods based on Technique 2, those without Technique 1, which are Compared 2 or Compared 5, did not reduce the accuracy, but those including Technique 1—Compared 3—decreased the accuracy, with respect to AF-D2. The sensitivity and specificity of Compared 4 in most cases were also the same or decreased when compared with the Base-CNN. That is, the curve fitting-based smoothing
of Technique 1 helps to improve accuracy as a preprocessing method, but it may distort important features depending on the dataset.

Table 6. Performance evaluation of Base-CNN, SDK-Injection, and other comparison methods.

| Metric      | Dataset | Base-CNN | SDK-Injection | Compared 1 | Compared 2 | Compared 3 | Add 1 Variants | Add 2 Variants | Add 3 Variants | Compared 5 |
|-------------|---------|----------|---------------|------------|------------|------------|----------------|----------------|----------------|------------|
| Accuracy (%)| AF-D1   | 26.67    | 46.67         | 40.00      | 33.33      | 33.33      | 26.67          | 26.67          | 26.67          | 26.67      |
|             | AF-D2   | 40.00    | 60.00         | 46.67      | 40.00      | 33.33      | 40.00          | 33.33          | 40.00          | 46.67      |
|             | ECC200  | 92.00    | 95.00         | 92.00      | 93.00      | 92.00      | 93.00          | 93.00          | 93.00          | 93.00      |
|             | ECG5000 | 93.76    | 94.38         | 94.16      | 94.11      | 94.11      | 94.33          | 94.24          | 94.13          | 94.18      |
|             | TwoLead | 69.89    | 85.25         | 69.53      | 72.52      | 70.41      | 65.67          | 66.64          | 67.95          | 77.88      |
| Precision   | AF-D1   | 0.30     | 0.47          | 0.33       | 0.22       | 0.28       | 0.21           | 0.17           | 0.17           | 0.17       |
|             | AF-D2   | 0.33     | 0.70          | 0.31       | 0.44       | 0.25       | 0.30           | 0.22           | 0.37           | 0.48       |
|             | ECC200  | 0.92     | 0.95          | 0.92       | 0.93       | 0.92       | 0.94           | 0.93           | 0.94           | 0.92       |
|             | ECG5000 | 0.70     | 0.75          | 0.70       | 0.68       | 0.64       | 0.73           | 0.72           | 0.68           | 0.69       |
|             | TwoLead | 0.72     | 0.86          | 0.71       | 0.78       | 0.74       | 0.66           | 0.69           | 0.68           | 0.79       |
| Sensitivity | AF-D1   | 0.27     | 0.47          | 0.40       | 0.33       | 0.33       | 0.27           | 0.27           | 0.27           | 0.27       |
|             | AF-D2   | 0.40     | 0.60          | 0.47       | 0.40       | 0.33       | 0.40           | 0.33           | 0.40           | 0.47       |
|             | ECC200  | 0.92     | 0.95          | 0.91       | 0.92       | 0.92       | 0.92           | 0.92           | 0.92           | 0.93       |
|             | ECG5000 | 0.52     | 0.58          | 0.57       | 0.56       | 0.57       | 0.57           | 0.58           | 0.55           | 0.56       |
|             | TwoLead | 0.70     | 0.86          | 0.70       | 0.77       | 0.72       | 0.66           | 0.67           | 0.68           | 0.79       |
| Specificity | AF-D1   | 0.63     | 0.73          | 0.70       | 0.67       | 0.67       | 0.63           | 0.63           | 0.63           | 0.63       |
|             | AF-D2   | 0.70     | 0.80          | 0.73       | 0.70       | 0.67       | 0.70           | 0.67           | 0.70           | 0.73       |
|             | ECC200  | 0.92     | 0.95          | 0.91       | 0.92       | 0.92       | 0.92           | 0.92           | 0.92           | 0.93       |
|             | ECG5000 | 0.98     | 0.98          | 0.98       | 0.98       | 0.98       | 0.98           | 0.98           | 0.98           | 0.98       |
|             | TwoLead | 0.70     | 0.86          | 0.70       | 0.77       | 0.72       | 0.66           | 0.67           | 0.68           | 0.79       |
| F1-Score    | AF-D1   | 0.28     | 0.46          | 0.35       | 0.26       | 0.25       | 0.23           | 0.20           | 0.20           | 0.20       |
|             | AF-D2   | 0.35     | 0.60          | 0.37       | 0.41       | 0.29       | 0.32           | 0.25           | 0.36           | 0.45       |
|             | ECC200  | 0.92     | 0.95          | 0.92       | 0.92       | 0.92       | 0.93           | 0.93           | 0.93           | 0.93       |
|             | ECG5000 | 0.56     | 0.63          | 0.61       | 0.58       | 0.62       | 0.62           | 0.63           | 0.59           | 0.60       |
|             | TwoLead | 0.70     | 0.86          | 0.69       | 0.77       | 0.71       | 0.66           | 0.66           | 0.68           | 0.79       |

Figure 11. Performance when applying SDK-Injection and other comparison methods on each dataset: (a) accuracy when applying SDK-Injection and other comparison methods on each dataset; (b) training time for the same situation.
Interestingly, the accuracy improvement of the SDK-Injection method is the most significant than that of any other comparison methods. In particular, as shown by the results of Compared 2, Compared 3, and Compared 5, the injection of shallow domain knowledge based on Technique 2 helps to improve accuracy; moreover, the SDK-Injection method is significantly high in accuracy compared with the accuracy of Compared 2, Compared 3, and Compared 5. The precision, sensitivity, specificity, and f1-scores, shown in Table 6, are also remarkably high in the case of SDK-Injection. This means that, although the shallow domain knowledge is injected by SDK-Injecting attention of Technique 2, the accuracy improvement can be further increased if the injected shallow domain knowledge is transmitted to the neural network by constructing multivariate input according to Technique 3 of the SDK-Injection method. As a result, the effect of the injected shallow domain knowledge on accuracy improvement can be maximized according to the method of transferring the injected shallow domain knowledge to the neural network. Additionally, it can be said that the SDK-Injection method is designed to maximize the effect of the injected shallow domain knowledge on accuracy improvement, compared with other comparison methods.

4.2.4. Result of Experiment 4

The four datasets, except for TwoLead, are aimed at diagnosing heart disease. Therefore, as shown in Table 7, the shallow domain knowledge can be derived from the sub-pattern that is the criteria for diagnosis of each heart disease. AF-D1 and AF-D2 are datasets for the diagnosis of atrial fibrillation. Atrial fibrillation can be diagnosed based on irregular heart rate. Since the QRS complex is observed at every heartbeat interval, the appearance pattern of the QRS complex within a certain range can be reflected in the shallow domain knowledge. When sample patterns were extracted and visualized for each class, it was confirmed that the appearance frequency of the QRS complex per unit time was different for each class. In order to inject the frequency of appearance of the QRS complex per unit time into shallow domain knowledge, there may be several possible methods, such as introducing an RNN. However, techniques for injecting shallow domain knowledge in a more complex form, based on an ensemble with other types of neural networks, is to be further studied in the future. Instead, in order to reflect this shallow domain knowledge up to the present study, first, the kernel size of the Base-CNN was set as the QRS complex length, and the middle part of the pattern was set as the shallow domain knowledge range. This is because data loss may occur in the process of cutting and patterning stream data in the front and rear portions of the pattern. Additionally, ECG200 is a dataset for the diagnosis of ischemia, and the characteristic T wave is the main diagnostic criterion for ischemia. Therefore, the range of the T wave was set as the shallow domain knowledge.

| No. | Dataset | Pattern Length | Target Disease          | Sub-Pattern as Criteria | Shallow Domain Knowledge |
|-----|---------|----------------|-------------------------|-------------------------|-------------------------|
|     |         |                |                         |                         | Start Index | End Index  |
| 1   | AF-D1   | 640            | Atrial Fibrillation     | QRS                     | 70          | 300        |
| 2   | AF-D2   | 640            | Atrial Fibrillation     | QRS                     | 70          | 300        |
| 3   | ECG200  | 96             | ischemia                | T                       | 10          | 55         |
| 4   | ECG5000 | 140            | Congestive Heart Failure| P                       | 100         | 135        |
| 5   | TwoLead | 82             | Non-disease             | Differences btw. Patterns | 0          | 70         |
ECG5000 is a dataset for diagnosis of congestive heart failure. At this time, the main diagnostic criterion is R-on-T, which is a phenomenon in which the R wave appears too early and rises above the T wave of the previous ECG pattern. That is, when R-on-T occurs, an abnormality also occurs in the P wave. In addition, abnormal findings of the P wave are diagnostic criteria for various heart diseases. Therefore, the shallow domain knowledge for ECG5000 was set to the range of the P wave. Since TwoLead is not a dataset for diagnosing heart disease, shallow domain knowledge could not be derived from specific medical facts. In this paper, Experiment 5 tried a method applying the proposed SDK-Injection method to this general ECG classification task, and based on this, TwoLead’s shallow domain knowledge was derived, as shown in Table 7.

Table 8 compares the performance improvement effect of the injected optimized shallow domain knowledge when the SDK-Injection method is applied to each dataset, as shown in Table 7. Additionally, from the metrics in Table 8, accuracy is visualized in Figure 12a. That is, as seen in Figure 12, the accuracy and training time were compared when the optimized shallow domain knowledge was set for the sub-pattern range, derived based on medical facts, as above, and when the sub-pattern range was set to 0 and the full length of the pattern. As a result, the highest accuracy, precision, sensitivity, specificity, and f1-score were achieved when the optimized shallow domain knowledge in Table 7 was set for all datasets. In other words, shallow domain knowledge derived based on medical facts can be effectively injected into neural networks through the SDK-Injection method, and can help improve accuracy. Meanwhile, TwoLead has the same accuracy when applying the optimized shallow domain knowledge, seen in Table 7, and when using the shallow domain knowledge as the full length of the pattern. This was analyzed in Experiment 5.

4.2.5. Result of Experiment 5

As shown in Table 7, the optimized shallow domain knowledge that can be derived based on each heart disease diagnosis criteria exists. Non-medical experts, such as data scientists and developers, can learn this and inject it into the neural network through the proposed SDK-Injection method to improve the existing accuracy. However, there is no shallow domain knowledge that can be derived from medical facts for general ECG classification tasks that are not related to heart disease diagnosis, such as TwoLead. We hypothesized that, if there is no shallow domain knowledge that can be derived from medical facts, classification accuracy can be improved by setting the shallow domain

| Dataset     | Shallow Domain Knowledge | Accuracy (%) | Precision | Sensitivity | Specificity | F1-Score |
|-------------|--------------------------|--------------|-----------|-------------|-------------|----------|
|             |                          | Length       | Start     | End         |             |          |
|             |                          | Start Index  | End Index |             |             |          |
| AF-D1       | Optimized                | 0            | 0         | 0           | 20.00       | 0.17     | 0.20     | 0.60     | 0.18     |
| AF-D1       | Max Pattern              | 0            | 640       | 40.00       | 0.40        | 0.70     | 0.29     |
| AF-D2       | Optimized                | 0            | 0         | 0           | 40.00       | 0.28     | 0.40     | 0.70     | 0.32     |
| AF-D2       | Max Pattern              | 0            | 640       | 46.67       | 0.52        | 0.47     | 0.73     |
| ECG200      | Optimized                | 0            | 0         | 0           | 91.00       | 0.91     | 0.89     | 0.90     |
| ECG200      | Max Pattern              | 0            | 96        | 92.00       | 0.93        | 0.90     | 0.91     |
| ECG5000     | Optimized                | 0            | 0         | 0           | 94.02       | 0.68     | 0.55     | 0.98     | 0.58     |
| ECG5000     | Max Pattern              | 0            | 140       | 94.11       | 0.66        | 0.55     | 0.98     | 0.58     |
| TwoLead     | Optimized                | 0            | 0         | 0           | 56.19       | 0.72     | 0.56     | 0.56     | 0.47     |
| TwoLead     | Max Pattern              | 0            | 82        | 85.25       | 0.86        | 0.85     | 0.85     | 0.85     | 0.85     |
knowledge to maximize the difference between class patterns. Based on this hypothesis, Table 9 shows the experimental results of one possible method for deriving a sub-pattern range that can replace shallow domain knowledge for general ECG classification tasks that are not related to heart disease. The accuracy of Table 9 is visualized in Figure 13. The red solid line in Figure 13 shows that the accuracy of the SDK-Injection method as the range of the sub-pattern to be set as shallow domain knowledge increased by 10 from index 0 to the pattern length 82.

### Table 9. Performance Evaluation of SDK-Injection Among Various Lengths of Shallow Domain Knowledge on the Non-Disease ECG.

| Dataset          | Shallow Domain Knowledge | Accuracy (%) | Precision | Sensitivity | Specificity | F1-Score |
|------------------|--------------------------|--------------|-----------|-------------|-------------|----------|
|                  | Start Index | End Index    |            |             |             |          |
| TwoLead          | 0           | 0            | 56.19      | 0.72        | 0.56        | 0.56     | 0.47     |
|                  | 0           | 10           | 73.22      | 0.79        | 0.74        | 0.74     | 0.73     |
|                  | 0           | 20           | 69.62      | 0.81        | 0.70        | 0.70     | 0.67     |
|                  | 0           | 30           | 77.35      | 0.78        | 0.78        | 0.78     | 0.78     |
|                  | 0           | 40           | 64.71      | 0.66        | 0.65        | 0.65     | 0.65     |
|                  | 0           | 50           | 78.75      | 0.80        | 0.78        | 0.78     | 0.78     |
|                  | 0           | 60           | 72.52      | 0.73        | 0.73        | 0.73     | 0.73     |
|                  | 0           | 70           | 85.25      | 0.86        | 0.86        | 0.86     | 0.86     |
|                  | 0           | 82           | 85.25      | 0.86        | 0.85        | 0.85     | 0.85     |

The black dotted line is the gradient of the absolute difference between the representative patterns of each class in the TwoLead dataset for the same situation. As a result, the change in accuracy according to the sub-pattern range almost coincided with the change of gradient of the absolute difference between the representative patterns of each class of TwoLead. That is, as the difference between the pattern values representing the two classes of TwoLead increased, the classification accuracy also increased. As a result, the

**Figure 12.** Performance compared when applying optimized shallow domain knowledge and other length of shallow domain knowledge for each dataset: (a) accuracy when applying optimized shallow domain knowledge compared with length with 0 and maximum pattern length; (b) training time for the same situation.
highest accuracy of 85.25% was obtained when the shallow domain knowledge range was set to the index 70, where the difference in pattern values started to accumulate the most, and 82, which was the total length of the pattern. The precision, sensitivity, specificity, and f1-score, shown in Table 9, also showed similar patterns to the change in accuracy in Figure 13. In addition, with index 70, where the difference in values between patterns started to accumulate the most, each showed the highest value. The accuracy and precision had the same values for the index 70 and the total length of the pattern, which was 82, but the sensitivity, specificity, and f1-score were slightly higher at index 70 than at index 82. In other words, considering the sensitivity, specificity, and f1-score, it is most appropriate to set the shallow domain knowledge with index 70, where the difference in pattern values is accumulated the most, rather than the entire length of the pattern, which is 82.

Figure 13. Accuracy (red) and gradient of differences between values of representative patterns of each class (black) according to the length of the subpattern starting from the index 0 for the TwoLead dataset, which is a general ECG classification task not related to heart disease.

5. Conclusions

In this paper, we proposed the SDK-Injection method, that can further improve the existing ECG pattern classification accuracy by injecting shallow domain knowledge, which can be acquired by non-medical experts, into the existing parameter-optimized CNN. The proposed method was designed by combining three techniques, which are curve fitting-based smoothing, shallow domain knowledge injection based on the attention mechanism, and using variants obtained from these for expansion to multivariate input. Additionally, it showed the highest performance among the compared methods composed of different possible combinations of techniques. In addition, the experiments showed that the proposed method can be applied to both heart disease diagnosis and general ECG classification tasks, improving the original accuracy for both types of tasks. Interestingly,
by applying the proposed method to single ECG-based analysis, the accuracy could be improved compared with that of multi-ECG-based analysis. For heart disease diagnosis tasks, the highest performance improvement was achieved when appropriate shallow domain knowledge, corresponding to each heart disease, was injected. On the other hand, there is no corresponding shallow domain knowledge for general ECG pattern classification tasks that are not related to the diagnosis of heart disease. In this case, the highest performance improvement was achieved when the range where the change in the difference value was accumulated the most was injected, instead of the shallow domain knowledge.

The proposed method showed a high improvement in performance for balanced data, but a relatively low improvement in performance for imbalanced data. Therefore, we plan to study a method for significantly improving performance even when the proposed method is applied to imbalanced data. Second, it should be possible to inject more diverse kinds of shallow domain knowledge, for example, the order in which each sub-pattern appears, and not only the current range of sub-patterns which are the criteria for diagnosis of each heart disease. Thirdly, we plan to expand our approach which calculates the gradient after finding the difference between the representative pattern values for each class, and work toward applying the proposed method to general ECG classification tasks, beyond than the diagnosis of heart disease. Additionally, we plan to collect additional datasets other than the current AF-D1 and AF-D2, and develop a method of converting a multi-ECG-based analysis task to a single ECG-based analysis task, by applying the proposed method. Finally, we plan to further research on applying our method to various types of neural networks, for example, using generative neural networks to generate variants for the Technique 3. In particular, the proposed method can currently only be applied on the CNN-based model. However, LSTM is also being used in various ways for ECG analysis, either alone or in combination with CNN. Therefore, we will develop an advanced SDK-Injecting attention, which can apply the proposed method to LSTM-based neural networks. Through these future works, the proposed method can be developed as a core technology for various ECG-based diagnoses in smart hospitals or healthcare services, in an actual wearables-based environment.

Author Contributions: Methodology, software, and writing, S.O.; writing—review and editing, M.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2021R1F1A1062559).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No new data were created or analyzed in this study. The program codes to perform the experiments using five ECG datasets are available online at http://dwlab.ewha.ac.kr/mlee/codes/sdk-injection/source.7z (accessed on 18 December 2021). The programs are written in Python.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References
1. Moses, D. A Survey of Data Mining Algorithms Used in Cardiovascular Disease Diagnosis from Multi-Lead ECG Data. *Kuwait J. Sci.* 2015, 42, 206–235.
2. Glushica, B.; Aleksovski, B.; Kuhar, A. Automatic Detection of Characteristic Segments of a Recorded ECG Signal with Noise Handling Methods. *IOP Conf. Ser. Mater. Sci. Eng.* 2021, 1032, 012047. [CrossRef]
3. Zifan, A.; Saberi, S.; Moradi, M.H.; Towhidkhah, F. Automated ECG Segmentation Using Piecewise Derivative Dynamic Time Warping. *Int. J. Biol. Med. Sci.* 2006, 6, 181–185.
4. Beraza, I.; Romero, I. Comparative Study of Algorithms for ECG Segmentation. *Biomed. Signal Process. Control* 2017, 34, 166–173. [CrossRef]
5. Luz, E.J.; Da, S.; Schwartz, W.R.; Cámara-Chávez, G.; Menotti, D. ECG-Based Heartbeat Classification for Arrhythmia Detection: A Survey. *Comput. Methods Programs Biomed.* 2016, 127, 144–164. [CrossRef] [PubMed]

6. Barman, T.; Ghongade, R.; Ratnaparkhi, A. Rough Set Based Segmentation and Classification Model for ECG. In *Proceedings of the 2016 Conference on Advances in Signal Processing (CASIP)*, Pune, India, 9–11 June 2016; pp. 18–23.

7. Rai, H.M.; Chatterjee, K. Hybrid CNN-LSTM Deep Learning Model and Ensemble Technique for Automatic Detection of Myocardial Infarction Using Big ECG Data. *Appl. Intell.* 2021, 1–19. [CrossRef]

8. Essa, E.; Xie, X. An Ensemble of Deep Learning-Based Multi-Model for ECG Heartbeats Arrhythmia Classification. *IEEE Access* 2021, 9, 103452–103464. [CrossRef]

9. Petmezas, G.; Haris, K.; Stefanopoulos, L.; Kilintzis, V.; Tzavelis, A.; Rogers, J.A.; Katsaggelos, A.K.; Maglaveras, N. Automated Atrial Fibrillation Detection Using a Hybrid CNN-LSTM Network on Imbalanced ECG Datasets. *Biomed. Signal Process. Control* 2021, 63, 102194. [CrossRef]

10. Zhang, Y.; Li, J. Application of Heartbeat-Attention Mechanism for Detection of Myocardial Infarction Using 12-Lead ECG Records. *Appl. Sci.* 2019, 9, 3328. [CrossRef]

11. Han, C.; Huang, R.; Yu, F.; Huang, X.; Cui, L. EasiECG: A Novel Inter-Patient Arrhythmia Classification Method Using ECG Waves. In *Proceedings of the 2020 25th International Conference on Pattern Recognition (ICPR)*, Milan, Italy, 10–15 January 2021; pp. 1–8.

12. Jiang, M.; Gu, J.; Li, Y.; Wei, B.; Zhang, J.; Wang, Z.; Xia, L. HADLN: Hybrid Attention-Based Deep Learning Network for Automated Arrhythmia Classification. *Front. Physiol.* 2021, 12, 683025. [CrossRef] [PubMed]

13. Wang, T.; Lu, C.; Sun, Y.; Yang, M.; Liu, C.; Ou, C. Automatic ECG Classification Using Continuous Wavelet Transform and Convolutional Neural Network. *Entropy* 2021, 23, 119. [CrossRef] [PubMed]

14. Couceiro, R.; Carvalho, P.; Henriques, J.; Antunes, M.; Harris, M.; Habetha, J. Detection of Atrial Fibrillation Using Model-Based ECG Analysis. In *Proceedings of the 2008 19th International Conference on Pattern Recognition*, Tampa, FL, USA, 8–11 December 2008; pp. 1–5.

15. Chen, X.; Cheng, Z.; Wang, S.; Lu, G.; Xu, G.; Liu, Q.; Zhu, X. Atrial Fibrillation Detection Based on Multi-Feature Extraction and Convolutional Neural Network for Processing ECG Signals. *Comput. Methods Programs Biomed.* 2021, 202, 106009. [CrossRef] [PubMed]

16. Pouyanfar, S.; Sadiq, S.; Yan, Y.; Tian, H.; Tao, Y.; Reyes, M.P.; Shyu, M.-L.; Chen, S.-C.; Iyengar, S.S. A Survey on Deep Learning: Algorithms, Techniques, and Applications. *ACM Comput. Surv.* 2019, 51, 1–36. [CrossRef]

17. Holzinger, A. From Machine Learning to Explainable AI. In *Proceedings of the 2018 World Symposium on Digital Intelligence for Systems and Machines (DISA)*, Košice, Slovakia, 23–25 August 2018; pp. 55–66.

18. Lastre-Domínguez, C.; Shmaliy, Y.S.; Ibarra-Manzano, O.; Munoz-Minjares, J.; Morales-Mendoza, L.J. ECG Signal Denoising and Features Extraction Using Unbiased FIR Smoothing. *BioMed. Res. Int.* 2019, 2019, 1–16. [CrossRef] [PubMed]

19. Krishnan, S.R.; Seelamantula, C.S. On the Selection of Optimum Savitzky-Golay Filters. *IEEE Trans. Signal Process.* 2013, 61, 380–391. [CrossRef]

20. Awal, M.A.; Mostafa, S.S.; Ahmad, M. Performance Analysis of Savitzky-Golay Smoothing Filter Using ECG Signal. *Int. J. Comput. Inf. Technol.* 2011, 1, 24.

21. Hussain, I.; Park, S.J. Big-ECG: Cardiographic Predictive Cyber-Physical System for Stroke Management. *IEEE Access* 2021, 9, 123146–123164. [CrossRef]

22. Tseng, K.-K.; Wang, C.; Xiao, T.; Chen, C.-M.; Hassan, M.M.; de Albuquerque, V.H.C. Sliding Large Kernel of Deep Learning Algorithm for Mobile Electrocardiogram Diagnosis. *Comput. Electr. Eng.* 2021, 96, 107521. [CrossRef]

23. Sereda, I.; Alekseev, S.; Koneva, A.; Kataev, R.; Osipov, G. ECG Segmentation by Neural Networks: Errors and Correction. In *Proceedings of the 2019 International Joint Conference on Neural Networks (IJCNN)*, Budapest, Hungary, 14–19 July 2019; pp. 1–7.

24. Sodmann, P.; Vollmer, M. ECG Segmentation Using a Neural Network as the Basis for Detection of Cardiac Pathologies. In *Proceedings of the 2020 Computing in Cardiology*, Rimini, Italy, 13–16 September 2020; pp. 1–4.

25. Wang, S.; Li, R.; Wang, X.; Shen, S.; Zhou, B.; Wang, Z. Multiscale Residual Network Based on Channel Spatial Attention Mechanism for Multilabel ECG Classification. *J. Healthc. Eng.* 2021, 2021, e6630643. [CrossRef] [PubMed]

26. Gao, F.; Li, Z. Recognition of ECG Signals by Convolutional Neural Network Based on Attentional Mechanism. In *Proceedings of the 2021 4th International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE)*, Changsha, China, 26–28 March 2021; pp. 124–127.

27. Ullah, A.; Tu, S.; Mehmood, R.M.; Ehatisham-ul-haq, M. A Hybrid Deep CNN Model for Abnormal Arrhythmia Detection Based on Cardiac ECG Signal. *Sensors* 2021, 21, 951. [CrossRef] [PubMed]

28. Time Series Classification Website. Available online: http://www.timeseriesclassification.com/dataset.php (accessed on 6 September 2021).

29. PhysioNet Databases. Available online: https://www.physionet.org/about/database/ (accessed on 6 September 2021).