

Review

# Applications of Machine Learning in Ambulatory ECG

Joel Xue <sup>1,\*</sup> and Long Yu <sup>2</sup><sup>1</sup> Department of Biomedical Informatics, Emory University, Atlanta, GA 30322, USA<sup>2</sup> Diagnostic Cardiology, GE Healthcare, Wauwatosa, WI 53226, USA; long.yu@ge.com

\* Correspondence: joel.q.xue@emory.edu

**Simple Summary:** The ambulatory ECG (AECG) is an important diagnostic tool for many heart electrophysiology-related cases. This review covers some key Ambulatory ECG applications of Machine Learning algorithms, which include both statistical learning and neural network-based deep learning algorithms.

**Abstract:** The ambulatory ECG (AECG) is an important diagnostic tool for many heart electrophysiology-related cases. AECG covers a wide spectrum of devices and applications. At the core of these devices and applications are the algorithms responsible for signal conditioning, ECG beat detection and classification, and event detections. Over the years, there has been huge progress for algorithm development and implementation thanks to great efforts by researchers, engineers, and physicians, alongside the rapid development of electronics and signal processing, especially machine learning (ML). The current efforts and progress in machine learning fields are unprecedented, and many of these ML algorithms have also been successfully applied to AECG applications. This review covers some key AECG applications of ML algorithms. However, instead of doing a general review of ML algorithms, we are focusing on the central tasks of AECG and discussing what ML can bring to solve the key challenges AECG is facing. The center tasks of AECG signal processing listed in the review include signal preprocessing, beat detection and classification, event detection, and event prediction. Each AECG device/system might have different portions and forms of those signal components depending on its application and the target, but these are the topics most relevant and of greatest concern to the people working in this area.

**Keywords:** ambulatory ECG; machine learning; deep learning; pattern recognition; noise reduction; Holter ECG



**Citation:** Xue, J.; Yu, L. Applications of Machine Learning in Ambulatory ECG. *Hearts* **2021**, *2*, 472–494.

<https://doi.org/10.3390/hearts2040037>

Academic Editor: Peter Macfarlane

Received: 2 August 2021

Accepted: 8 October 2021

Published: 13 October 2021

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



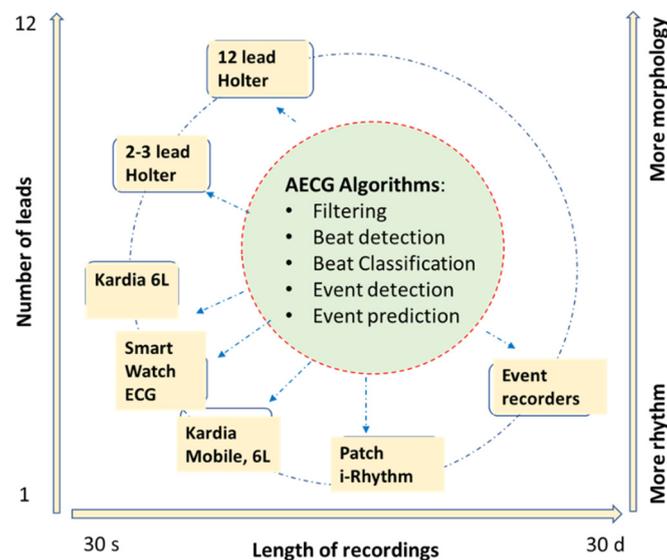
**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Ambulatory electrocardiograms (AECG) have evolved greatly from the traditional 24–48 h of Holter monitoring devices. Now, AECGs can last as short as 30 s to as long as 30 days. There is a much wider range for lead selection, with AECGs coming in the form of a small patch with a small vector to a full-scale 12-lead electrocardiogram (ECG) with a wide coverage of heart electrical activity. The clinical applications for AECGs have also expanded from limited arrhythmia analysis to morphology analysis of ST level and QT interval and risk stratification/prediction [1]. Figure 1 is a diagram of the scope for AECG devices and algorithms. In the figure, the most noticeable development of AECG in recent years is innovation in miniaturized devices and applications such as patch devices, wearable devices such as smart watches (Apple Watch), and convenient home use devices such as AliveCor's Kardia Mobile™. These recent innovations are not only convenient to use by patients, they are also FDA-approved medical devices and medical applications [2–4].

Although there have been many new and exciting developments for AECGs involving device type, data capacity, and physical size, the core of AECGs—the basic algorithms, as shown in Figure 1's core circle, which include signal conditioning, beat detection, event

detection, and interpretation—have not changed very much. However, key challenges for each of the four processing tasks listed above remain, particularly in noise handling, how to reduce noise during preprocessing, and how to differentiate noise and signals during beat detection and classification. The reason noise continues to be a challenge for AECGs is mainly because most AECGs are recorded outside of hospitals. Therefore, there is no trained medical personnel to monitor the signal quality, patients move around and engage in various activities while carrying the recording devices, and the recording time can also be long. It also needs to be indicated that most recent AECG recordings based on home-use devices such as Kardia Mobile/6L and Apple Watch are short-segment/noncontinuous vs. traditional continuous recordings from Holter recording of 24–72 h. For long Holter recordings, some learning algorithms can be applied for a period to accumulate initial templates, and the learning process can be updated with the longitudinal data to refine the template matching and beat detection. On the other hand, short recordings of AECG cannot afford too long of a learning segment, and thus they are more dependent on pre-learned model performance.



**Figure 1.** There are a wide range of AECG devices and applications. The recording length can be from 30 s to 30 days, and the number of leads of ECG can be from 1 to 12. At the center of all these lies the AECG algorithms including filtering, beat detection and classifications, and event detection and prediction. For devices with 1 or 2 leads, the events are mainly in rhythm abnormalities, such as sinus, AFIB, or tachycardia/bradycardia. For devices with more leads, some morphology analysis can be added such as ST, QT, LVH, or BBB.

Thirty-five years ago, a limited capacity microprocessor-based AECG algorithm could already achieve AECG beat detection accuracy to around 99% for normal beats and 96% for ectopic beats [5]. It is this remaining 2–3% of AECG signals that remain uncharacterized due to the challenges discussed above. Now the question or challenge is whether modern techniques offer significantly better performances than the ‘old’ ones.

In recent years, with the rapid development of machine learning (ML), deep learning (DL) in particular, many researchers have applied ML/DL methods to AECG algorithms [6–11]. Unlike the previous wave of interest in ML and neural networks around the 1980s to the 1990s, the recent development in the field has shown more promise, primarily due to the availability of larger training data sets and the maturity of the ML algorithms, especially convolution neural networks (CNN) and recurrent neural networks (RNN) [12,13]. One major advancement of the CNN-based AECG algorithms is that they can directly process ECG waveforms after initial preprocessing, without the need for ECG feature extraction, which the last generation of neural network algorithms relied on. Theoretically, since feature extraction of ECG signals from noisy data is usually difficult and time-consuming,

the advantage of not relying on predefined feature extraction is that it can increase the efficiency of building new AECG algorithms from scratch if a large training data set is available.

The traditional 24–48 h Holter AECG might be described as the first attempt at big data analysis, but with less impressive results and with far greater effort compared to some current methods. An average of 100,000 to 200,000 plus heartbeat cycles of ECG data needed to be processed for a single data recording. Either automatic or semi-automatic learning models, which include template matching and clustering, were used during the analysis [14]. These types of learning algorithms were mainly limited to current patient data, instead of using a wide group of patients' data for training sets as the most current DL algorithms do. Most recent applications of DL methods to AECG have used large data sets consisting of multiple patients' data recordings as the training set, followed by the application of automatic pattern matching during analysis. A combination of large data sets at the pre-learning stage and the individual data's continuous learning can be a key to improve the analysis accuracy for such a relatively long AECG analysis [15].

After the initial excitement of experiencing ML/DL's performance of processing AECG, it is important to understand how ML algorithms work and perform in comparison to more traditional algorithms. Instead of a general overview of ML/DL techniques, which have been recently discussed in other reviews, this review focuses on how these new ML/DL algorithms can perform better in recognizing differences between physiologically meaningful signals and noise, and ways in which these new algorithms can be used together with traditional models to achieve even better performance and interpretability. Interpretability is very important for most medical applications, not only because it can help us better understand how the algorithms work, but also because it is more relevant for causality analysis—fundamental to helping physicians find causes and better treatments. We here cover how these points are relevant to each key step of AECG processing.

## 2. A Summary of Machine Learning Algorithms Used for AECG

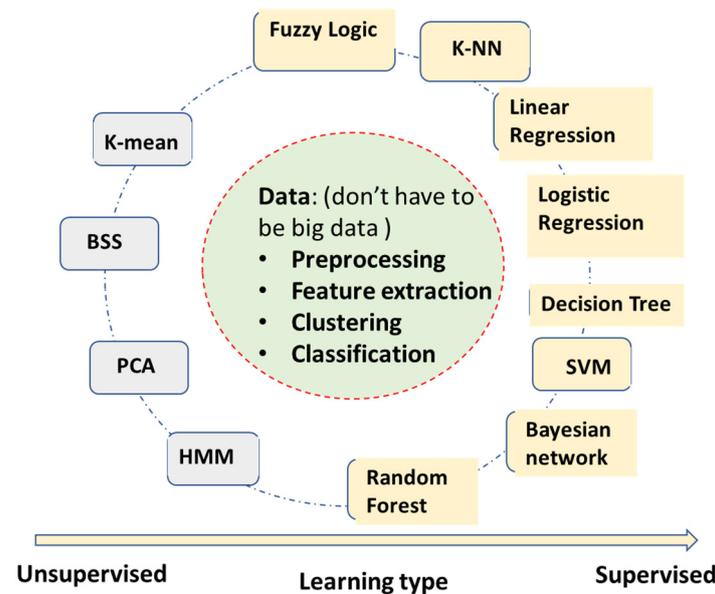
Various machine learning algorithms have been used for AECG signal processing and detection for several decades, although DL algorithms have only become more widely used in recent years [13–16]. Therefore, we can list DL vs. non-DL algorithms separately. The purpose of this section is not to introduce each algorithm, rather for the convenience of the discussions in a later section.

### 2.1. Machine Learning Algorithms without Deep Learning

As shown in Figure 2, there are many ML algorithms in this category. We can divide them into supervised and unsupervised learning [17]. Supervised learning requires an input-label/reference pair for training the algorithm, while unsupervised learning does not need reference/labeling (note: this might not be a complete list of all the algorithms).

- **Fuzzy logic algorithm**—Using fuzzy logic for rule-based classifications. The fuzzy logic uses a 'soft edge' decision boundary to replace the 'crispy' decision boundary on the basis of multiple criteria. It maintains its interpretable nature while being able to adapt to the complicated decision boundaries. However, the algorithm complexity can increase very fast with the increase of the number of input features.
- **Linear regression**—It builds a linear correlation model between inputs and outputs, mostly used for continuous value output.
- **Logistic regression**—It also builds the correlation model between inputs and outputs, converging to binary level output by logistic functions such as sigmoid. It can be used for categorical classifications.
- **Decision tree**—Automatically generates a rule-based classification by a tree-like model. It usually has very intuitive flowchart symbols and rules, and therefore it is simple to understand and interpret. However, it can be relatively inaccurate compared to other predictors with the similar data.

- **SVM (support vector machine)**—A binary classification to maximize the distance between two classes. It has become one of the most robust binary classifiers. It also can perform nonlinear classification with its kernel functions.
- **Bayesian network (naïve Bayesian network)**—Applies simplified Bayesian theorem for classifications. It builds a probabilistic relationship between symptoms and measurements with the causes of diseases, and therefore is usually a better causality model than neural networks' 'black box' models. However, it requires more known knowledge such as conditional/joint probabilities of input variables.
- **k-NN (k-nearest neighbors)**—A very intuitive classification algorithm wherein a sample is classified on the basis of a common majority rule of its k closest neighbors. It is a very effective classification algorithm with limited training samples. It works better with a small number of input features.
- **Random forest**—An ensemble learning method by constructing multiple decision trees. A test sample is classified on the basis of the selections made by most trees. It has some relationship with the decision tree algorithm but is better for avoiding overfitting. It has become one of the most widely used classification algorithms outside of neural networks models.



**Figure 2.** A diagram for the most widely used statistical learning algorithms. Most of the non-deep learning algorithms listed here can work on moderate-sized data since their independent parameters are limited. These algorithms mostly work on previously extracted features, or they can help to identify the best features such as PCA. On the left is unsupervised learning mainly used for clustering and feature optimization; on the right are supervised learning algorithms, requiring pairs of inputs and labels. All these methods have been used widely in AECG applications in the last several decades.

Here is a list for the most widely used **unsupervised algorithms**:

- **K-mean**—A clustering algorithm based on vector quantization that partitions  $n$  observations into  $K$  clusters, e.g., clustering ECG beats into templates in Holter analysis (not to be confused with k-NN, which is a supervised classification algorithm).
- **BSS (blind signal separation)**—Separates a set of source signals with little information about source signals, e.g., separate signal and noise of AECG.
- **PCA (principal component analysis)**—One of the most popular unsupervised algorithms for data reduction and feature optimization, e.g., determining which ECG features are more important for AFIB detection.

- **HMM (hidden Markov model)**—Assume signals to be a Markov process, current state,  $X(n)$  only depend on immediate previous state  $X(n-1)$ . e.g., to describe R-R interval sequence of AFIB ECGs.

These groups of algorithms mostly belong to statistical learning and modeling with mathematical equations, instead of distributed weights and layers in neural networks and DL algorithms. Most statistical algorithms listed here can work on moderate-sized data since their independent parameters are limited. These algorithms mostly work on previously extracted features, or they can help to identify the best features.

## 2.2. Neural Network Deep Learning Algorithms for AECG

Different from the statistical learning algorithms above, neural network (NN) learning algorithms are based on the parallel and layered structure. The optimization is usually done through minimizing cost functions by some sort of backpropagation of gradients [18]. Previous NN learning started in the 1980s with many applications in ECG processing and pattern recognition [19,20]. The NN models were mostly 1–2 hidden layers, so-called ‘shallow learning’, whose inputs were mostly pre-extracted features or localized waveforms [21]. The current wave of DL models of NN started in around 2012, although a CNN-based DL algorithm for digits was published earlier [22,23]. A list of the most widely used DL algorithms is found below (Figure 3 is a diagram for a summary of DL algorithms).

### Supervised Learning:

- **Convolutional neural network deep learning (CNN)**—The most popular DL algorithm with many applications in AECG. The multi-layer structures of CNN act as a filter bank, and the nonlinear activation functions of CNN act as feature extraction. Therefore, the CNN models are capable of handling original ECG waveforms directly.
- **Recursive neural network deep learning (RNN)**—RNN models add links among their hidden units (cells) to add memories for time series signals, similar to the concept of IIR filters used in signal processing. AECG signals are time-series signals whose features such as P-R and R-R intervals can be well fitted to certain time-series models.
- **Transfer learning**—This algorithm is very useful for DL models without enough training data for the current application. A pre-trained DL model for another domain is transferred to the current application. It only re-trains fewer layers of the transferred model or adds fewer new layers for retraining.
- **Ensemble learning**—This algorithm is not only useful for DL models. Random forest learning mentioned above is a type of ensemble learning. General ensemble learning can be extended to a wide range of groups of model collections and voting, suitable for complicated detection and prediction applications. Statistically, ensemble learning can also be viewed as an application of the central limit theorem. However, in practical use, it needs to know if computation power is enough to handle multiple models running.
- **Self (semi)-supervised learning**—This learning method deals with limited labeled data and large amounts of unlabeled data, which almost all applications are facing. It can be very useful in applications of AECG, although there are not too many mentioned thus far.
- **Online learning**—It allows for a pre-trained large model to be updated with the new coming samples, while still maintaining good performance for previously trained samples. This is a particularly good concept but lacks matured algorithms thus far. Some of the transfer learning examples can be viewed as special cases for online learning with only new training samples updated from a pre-trained model, but ‘old samples’ are most likely forgotten.

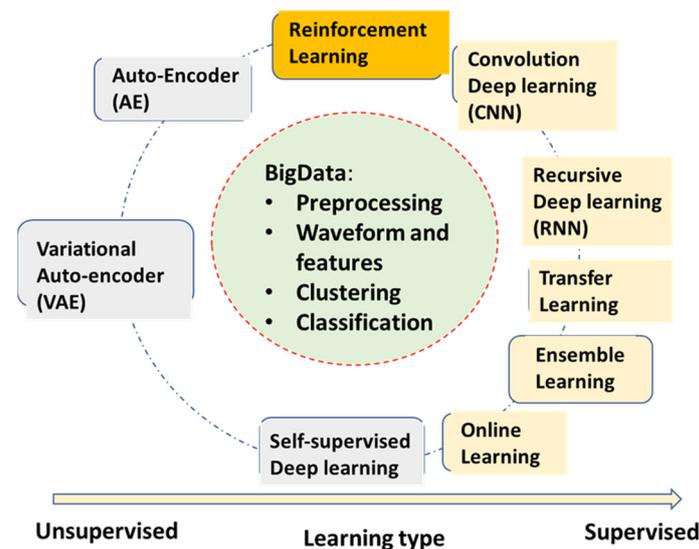
### Unsupervised learning:

- **Auto-encoder (AE)**—This algorithm is a DL model for encoding the original input, such as ECG waveforms. After training, it can represent ECG waveform with a much-condensed latent variable vector.
- **Variational auto-encoder (VAE)**—It has the same structure and training as AE. However, during the applications, one can adjust its latent vector to form different variation patterns of the original waveforms, for example, to synthesize different noise patterns.

**Reinforcement learning (RL):**

Reinforcement learning is the third large type of learning algorithm that recently achieved very promising results in many fields, especially in game playing [24,25]. RLs use agents to fit into an environment in order to maximize reward. There are also some attempts of RL for AECG processing but with limited successes [26].

Figure 3 is a diagram of a summary of DL algorithms. In summary, DL models need a large database to train. The rapid development of DL models goes side-by-side with the availability of big data and big computation power. For AECG, a large ECG waveform database with corresponding labels is needed for training purposes.

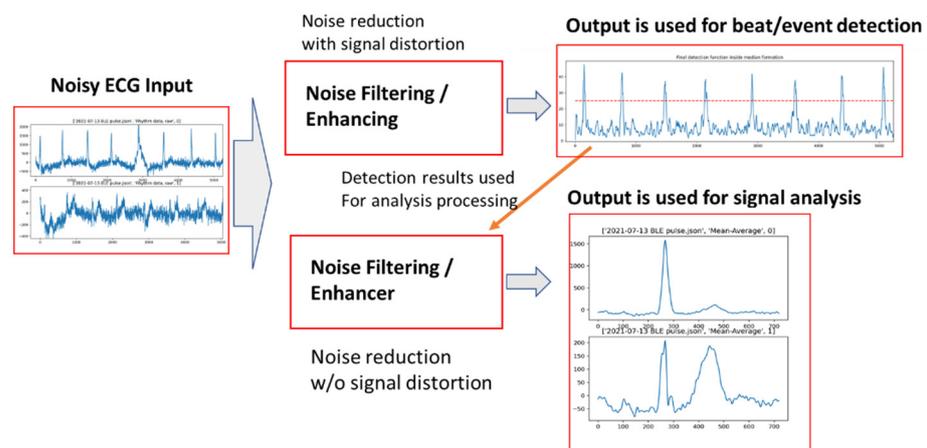


**Figure 3.** A diagram of a summary of DL algorithms. DL models rely on big data. For AECG, a large ECG waveform database with corresponding labels is needed for training purposes. In the list, CNN and RNN are the most popular DL models. Transfer learning and ensemble learning also become practical for AECG. However, there has been limited use of reinforcement learning and self-supervised learning thus far. AE and VAE are very useful for noise detection and feature extraction.

**3. AECG Signal Preprocessing—Noise Filtering**

The main purpose of the preprocessing of AECG signals is to reduce noise while bringing minimum distortion to the original signals. Any type of filtering can distort the signals, and therefore we need to make sure that the distortion is tolerable and meets the standards defined for each application.

A typical noise handling task in AECG can be described by a diagram shown in Figure 4, where there are two parallel paths for preprocessing: one (the upper path) is mainly for noise reduction and signal-to-noise ratio (SNR) improvement, but with some signal distortion; the other (the bottom path) is also for noise reduction but with minimal signal distortion. The results from the first path can be used to assist the second path, e.g., signal averaging; as shown in the figure, an average beat is formed with the trigger points detected from the first path.



**Figure 4.** Signal preprocessing can have 2 parallel paths, one (the upper path) is mainly for noise reduction and SNR improvement, but with some signal distortion, while the other (the bottom path) is also for noise reduction but with minimal signal distortion. The results from the first path can be used to assist the second path, e.g., signal averaging; as shown in the figure, an average beat is formed with the trigger points detected from the first path.

### 3.1. AECG Signal Processing-Noise Reduction

The conventional bandwidth for AECG is 0.5–40 Hz, compared to a diagnostic resting ECGs 0.05–150 Hz [1,27,28]. Since most AECG applications are focused on rhythm and arrhythmia analysis, the main task for preprocessing is to enhance the QRS complex as shown in the upper path of Figure 4. With the applications expanding to other ECG morphology analyses such as ST and QT, preprocessing also needs to consider reducing the signal distortion to other signal segments while enhancing the QRS complex segment, but we here focus on the mainstream application of AECGs.

If the noise’s frequency contents are higher than 40 Hz, it is called ‘out-band’ noise, which can usually be removed or reduced by a bandpass filter. If the noise’s frequency contents are within 40 Hz, then it is called ‘in-band’ noise, which can be caused by motion and muscle contraction; this is almost unavoidable for AECG. It is this in-band noise that needs to be dealt with using more specialized methods. Compared to the preprocessing of diagnostic ECGs with very a small tolerance for signal distortion, AECG signals are allowed to have a larger tolerance for signal distortion. Therefore, there are more choices for filtering methods.

It should be noted that if preprocessing is only to form some type of detection function for the purpose of QRS or P wave detection—not for analyzing ECG morphology—then processing methods can be more flexible as long as signal-to-be-detected can be enhanced while uninterested portions can be attenuated. For example, the QRS detection function applied in this classic paper [5] enhanced QRS complex and attenuated noise, and even P and T waves, such as the output of the upper path of Figure 4.

Usually, there are two parallel paths for ECG signal preprocessing, as shown in Figure 4, one for beat detection (which maximizes the SNR of QRS) and one for more detailed ECG analysis with a minimized distortion to the original waveform shape. If needed, signal averaging or median beats can be formed from the second path signal output.

### 3.2. Early Stage of ML Filtering of AECG

As most recent studies using ML for AECGs are focused on beat-classification and event detection, it needs to be pointed out that many ML technologies are evolved from the various adaptive filtering algorithms that include the earliest attempts to AECG’s powerline interference reduction [29]. In those ECG adaptive filtering applications, the models were learned and converged in real time, and most of the time the model is linear and very compact and therefore it can converge fast. There are some similarities of those

adaptive filtering models to the current ML learning model, such as minimizing the error of prediction and target with the steepest descent of gradient of the errors.

The first nonlinear adaptive filtering was proposed and applied to AECG in the early 1990s [19]. This research created a neural network model for real-time filtering, and it achieved better performance for beat detection for noisy data. The reason is that the nonlinear model can adapt to color noise better than linear models so as to improve the accuracy of the match filters.

The wavelet methods are also among the advanced noise filtering methods for AECG [30]. It is also a nonlinear signal processing method, in which different wavelet transformed scales were filtered differently and then inverse transformed back to time-domain signals.

### 3.3. Using Deep Learning Models for ECG Denoising

The ML algorithms can be used for both the detection function and waveform analysis. The first major task of any ML algorithm is to read and format the ECG data from originally sampled formats, which can be an involved task since data can come from different devices with different formats. For AECG, large data sets are mostly shared with the following three formats: (1) Physionet-MIT data format [31], (2) International Society for Holter and Noninvasive Electrocardiology (ISHNE) format [1], and (3) devices' outputs in Json/XML format with readable ECG data.

The study of [32] applied several DL models to ECG signal denoising, in which a multilayer CNN model and an RNN model were used separately for a supervised regression. For the training set, inputs are noisy ECG signals, and the targets are denoised ECG. The model is more like an auto-encoder where input and target pair have the same signals, with the exception of the target signal, which is to be a leaned version of the input signal [33].

The auto-encoder is designed to extract the independent latent representation, and is also employed in the conditioning of ECG signals [33–36]. The auto-encoder usually has an inversely symmetric architect where the input signal is transformed into a latent representation vector with significantly fewer dimensions. The signal is then reconstructed solely on the basis of the latent representation vector. Therefore, with a clean signal as a target for reconstruction from noisy inputs, an auto-encoder can perform as an ECG signal conditioner. The input of an auto-encoder is usually ECG signal without the need for labels, making it much more achievable than supervised DL methods. Some even applied generative adversarial networks (GAN) in feeding auto-encoders to further reduce the need for large datasets [37]. Due to the dimension reduction nature of the method, auto-encoders can also be applied in other applications such as compression [38–40].

The study of [32] combined conventional filtering with a CNN DL model to obtain a higher SNR improvement. The initial filtering of ECG signals included a bandpass filter (0.1–30 Hz), an IIR notch filter, and a wavelet filter for removing baseline wander. The filtered signals were fed into a 15-layer CNN model.

Unlike arrhythmia analysis or beat classification where ground truth requires medical expertise, for ECG denoising analysis, a training data set can be obtained by adding synthetic noise to a presumed clean ECG signal [41,42].

These types of DL model-based denoising filters have much more computational burden than other types of filters. Therefore, usage is limited for applications with computational and/or battery life limitations.

## 4. AECG Beat Detection and Classification

Beat detection and classification are the essential tasks of any AECG application. The good news is that the accuracy of beat detection (QRS complex) had already reached more than 99% before any advanced ML algorithms were invented or applied years ago [5]. However, the performances of those beat detections are very much dependent on the test databases. In addition, it needs to be indicated that the performance of beat classifiers is

usually not as good as the beat detection only algorithms. The challenges for current ML enthusiasts are not just to prove that ML, especially DL algorithms, can work on AECG beat detection and classification, but also to show improved performance for the remaining 1% of the widely used test databases, and also for improved performance on more challenge databases.

#### 4.1. Conventional Algorithms for Beat Detection and Classification

In a conventional AECG analysis program, after the preprocessing and the detection function is formed, a combination of thresholding and pattern matching algorithms are used for both beat detection and classification. Here, the task of beat detection is to detect any QRS complex, regardless of its source of origin, which can start from sinus, atrial, junctional, ventricle, fusion, etc. The beat classification separates beats into different categories or templates. There are two main challenges for both beat detection and classifications: noise handling and the time-series nature of the beats.

##### 4.1.1. Use Both Thresholding and Template Pattern Matching

The biggest challenge is still noise—electrode motion noise, muscle noise, powerline interference, etc. Noise detection is a major part of any AECG algorithm. After various filtering and techniques are used, as reviewed in the previous section, AECG's SNR is improved so much that the beat detection accuracy can be as good as 99% or even higher. For a 24-h Holter recording, there are about 100,000 beats on average. Thus, a 99% accuracy means there are still about 1000 beats missed. We would hope a Holter ECG reviewer does not have to correct each miss beat-by-beat, otherwise, it could be a time-consuming task. A good thresholding method is to find the optimal threshold between QRS complex and the noise floor. Due to the nature of the AECG's relatively high noise level, sometimes the separation of beat and noise is difficult.

The pattern matching method can work better than using a threshold only, especially under low SNR situations in which a bank of collected templates is matched with an underlying signal beat. If the correlation coefficient is high, then a beat can be identified. If the correlation coefficient is low, then a new template might be added to the template collection. The matching can be feature-based or waveform-based. The feature-based template matching can be more computationally efficient, but it requires a feature extraction process.

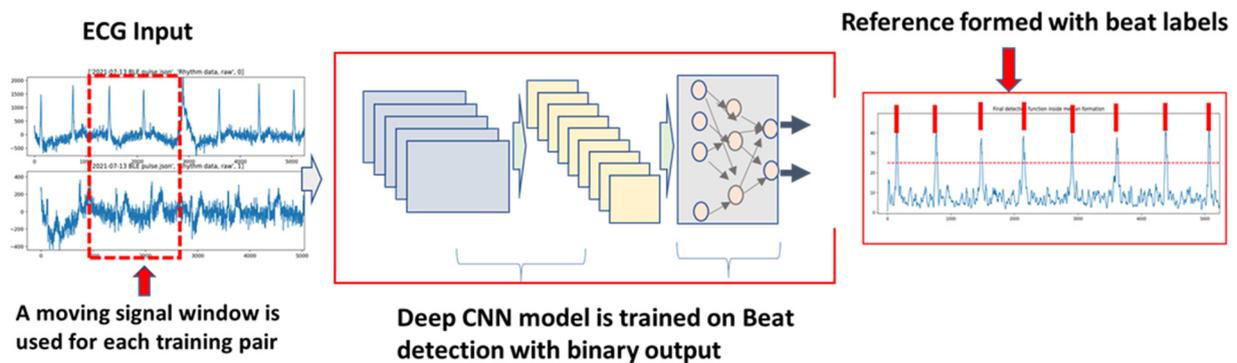
##### 4.1.2. Time Series Analysis

By nature, AECG is a time-sequence signal, meaning it has beat cycles. Within each cycle, there are sequences propagating along certain pathways. Therefore, the beat detection and classification also need to take this time-series feature into consideration. For example, when a beat is identified, the following 200–250 ms is called the refractory period when heart tissue cannot be triggered. This logic is easily applied in conventional algorithms, but also needs to be considered for machine learning—deep learning neural network (ML-DNN)-based algorithms.

#### 4.2. ML/DL-Based Beat Detection and Classification

The majority of ML/DL-based AECG studies are beat classifications and arrhythmia detection [7,13]. Among those studies, most DL algorithms are CNN-based supervised learning, while few are RNN-based algorithms trying to fit into time-series features of AECG. Although quite a few papers targeted beat detection and classification, DNN-based algorithms showed a potential of directly classifying arrhythmia events without explicitly marking each beat and its type. In one of the first CNN-based large AECG dataset studies, 12 arrhythmias were classified directly from 1 lead ECG with a selected 30 s segment [10]. Another large study also showed a CNN-based model for classifying multiple arrhythmias and morphology-related abnormalities from multi-lead ECG waveforms directly without beat detection and beat classification [12].

Figure 5 shows a general DL model for beat detection and classifications, where one real value of applying DNN models to AECGs is to possibly skip beat-by-beat analysis altogether, instead obtaining the final event detection from ECG waveform in one large model. This works as long as large training sets are available, and Hannun's study [10] had about 90,000 ECGs while Kashou's study [9] had about 2.4 million ECGs. For the learning data set, one key parameter is the length of the input data for each input-label pair, which can be arranged from 1 s to 10 s, as indicated by the dotted window in Figure 5. A short window with only one QRS beat can be used to teach the model how to recognize various QRS beats, whereas a longer window including 3–4 beats can teach the model for the current QRS beat and its surrounding beats and noise. The longer window is also better for PVC detection by the model since the window includes the current PVC beat and its previous and following beats. However, the longer window for input data usually requires a larger training set since it has more variations of the beat series pattern. Therefore, if the training set is large enough, the optimal input data window is around 3–4 s.



**Figure 5.** DL models can be trained for beat detection and classification directly from ECG waveform. The key is to define the input/reference pair. The label can indicate beat type and noise segment.

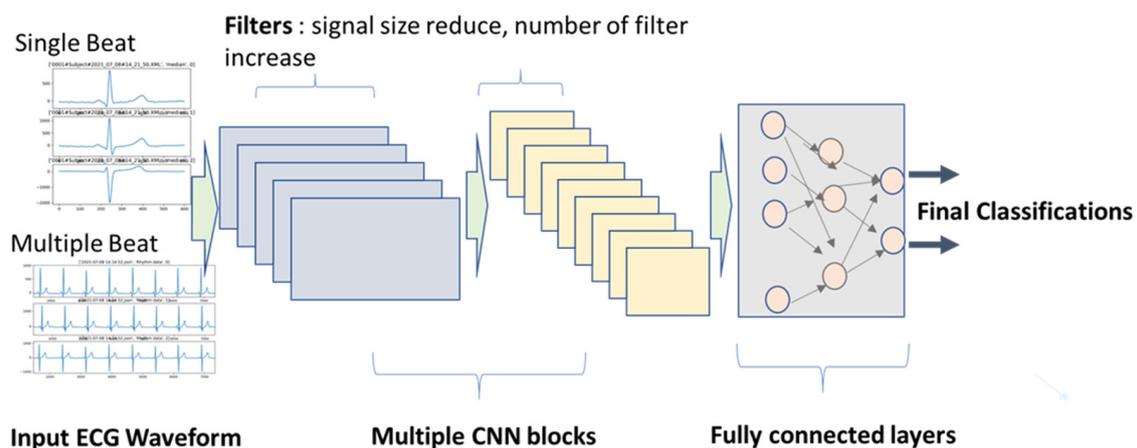
In some AECG applications, especially consumer-based wearable and home-used devices, the one-step 'black box' approach might be more preferred. However, for those AECG applications such as 24 h + Holter and ECG patch, a more comprehensive analysis including detailed beat detection and beat classification might be preferred, since these data are edited and annotated for a final medical report and are also more interpretable.

#### 4.2.1. DL Supervised Learning for Beat Detection and Classification

As shown in the drawing of a general CNN-based ECG learning in Figure 6, the input can be a one-beat or multiple-beat ECG waveform, also being able to have one or multiple leads. Borrowing the idea of image recognition for original CNN applications, we can also call this input an ECG image. For the input of multiple-lead ECG signals, either 2-D CNN or 1-D CNN can be used. However, to handle one lead ECG, 1-D CNN is the better choice. For supervised learning, model structure design also includes selecting a pair of input-targets (reference), in addition to the number of filters, kernel length, and a number of layers. The paper of [6] reviewed several CNN-based beat classifiers by model structure, input data, target class, and other features. All of the studies reviewed took ECG waveforms with different preprocessing as inputs. Some of them also added extracted ECG features such as pre- and post-RR intervals of the current beat. It also stated that the CNN models can perform better for noisy data than conventional feature-based beat detection, and, as shown in the drawing of general CNN-based ECG learning in Figure 6, the input can be a one-beat or multiple-beat ECG waveform, being able to also have one or multiple leads. Borrowing the idea of image recognition for original CNN applications, we can also call this input an ECG image. For the input of multiple-lead ECG signals, either 2-D CNN or 1-D CNN can be used. However, to handle one-lead ECG, 1-D CNN is the better choice. For supervised learning, model structure design also includes selecting a pair of input

targets (reference), in addition to the number of filters, kernel length, and the number of layers. Below are several key parameters for CNN model structure on AECG:

- **Input data, length, and dimension:** Some studies used one beat cycle, and some used multiple beat cycles. The advantage of using multiple beat cycles is to add time-series information. It is also beneficial for differentiating signal and noise. However, using multiple segments require larger training sets, since the variation is increased.
- **CNN kernel filter length and number for each layer:** Length of the kernel filter seems not to affect performance by much, but the number of filters usually increases from layer to layer, such as 16, 32, and 64. For AECG processing, the number of filters increased with the complexity of the input signals. For example, if the input signals are rhythm signals in original sampling frequency, the number of kernel filters needed can go up to 128 for deep layers. However, if the input signals are shorter averaged/median beats, the number of filters can be reduced to 32. A large number of filters generate more latent variables. If there are too many latent variables than needed, the model learning speed can be unnecessarily slow.
- **Regular convolution or residual convolution (ResCNN) layer:** ResCNN with a skip layer for each unit can increase training speed. Therefore, most studies adopted the ResCNN model structure.



**Figure 6.** A diagram of CNN model for AECG classifications. The model input can be a single beat or multiple beats. Multiple ECG leads can form an ECG image. If only 1 lead is presented, then 1-D CNN can be used. However, if there are multiple leads as inputs, 2-D CNN can be used. There are multiple CNN blocks to form a deep layer structure. Usually, the number of filters is gradually increased, but the size of the output of each block is reduced by down-sampling after each block of max/average pooling and stride operation. The final block of the model consists of fully connected layers. The final output can be either binary or multiple classifications.

Below are several key strategies for the training of CNN models on AECG:

- **Batch size:** Usually, large batch size is preferred for a big data set and multiple classes, such as 256 or 512. For binary classification and limited data size, a smaller batch size can be used, such as 32 or 64. Another consideration is for GPU memory size, since the whole batch is loaded into GPU memory as one block to speed up the training. The larger batch will take more GPU memory. For AECG processing, the selection of the batch size is also heavily dependent on the available training samples. If the training set is very large, such as 1 million plus, the batch size can be set at 512 to generate a smoother gradient search path by avoiding too much fluctuation of a smaller batch size, and most importantly to avoid being ‘trapped’ in a local minimum. However, if the training set is relatively small, the batch size has to be reduced, with the reduction of the model layers.
- **Loss function and output function:** These strategies need to be clarified to avoid misuse. The classification task can be divided into three categories (as shown in Table 1):

(1) binary classification, e.g., QRS complex vs. noise; 2) multi-class classification that is mutually exclusive, e.g., classify QRS beats into N, S, V, F, Q beat types; (3) multi-class classification but not mutually exclusive, e.g., morphology-related ECG abnormal: LBBB, ischemia, sinus, etc. Different loss functions and output functions are selected accordingly. The suggestions are also listed in Table 1.

- **Balance of different class types in each batch of training samples:** Very often, we can have a very different distribution of classes. For example, there are many more normal beats than PVC or other abnormal beats. If the same distribution is used in the training batch, it is very likely that the sensitivity of PVC beat detection will be poor. Therefore, the number of PVC beats can be augmented in each training batch, which can be in their original form or with variations. Another method is to use weight balancing. Popular machine learning frameworks, such as Keras, provide a ‘class weight’ parameter for this purpose [43].
- **Prevent overfitting:** Theoretically, there are concerns for both underfitting and overfitting. However, in most DNN studies, the model size/layer is so large that we might only need to worry about overfitting problems. The overfitting is usually caused by a lack of training samples with too many model parameters that need to be trained. There are several methods that can be used. The first one is to increase the training; for example, by adding certain noise to the original ECG recording to avoid simple repetition of the same data. The second method is to apply drop-off to the training process, which randomly ‘disconnects’ the weights to the output. The third method is to apply transfer learning [44,45].

**Table 1.** Selection of loss function and target function for different classification tasks.

Classification Tasks	Binary Classification	Multi-Class Classifications (Mutually Exclusive)	Multi-Class Classifications (Mutually Nonexclusive)
Examples:	QRS beat vs. noise	N, S, V, F, Q beat types	LBBB, ischemia, sinus
Loss function	Binary cross-entropy	Categorical cross-entropy	Binary cross-entropy
Target function	One-hot vector (Softmax)	One-hot vector (Softmax)	Sigmoid, scalar target

#### 4.2.2. Unsupervised Learning for Beat Detection and Classification

Unsupervised learning does not need reference/labels for training, and, therefore, more data are available for building the algorithms or feature extraction from the models. The most widely used non-DL algorithms in this category are blind source separation (BSS) and principal component analysis (PCA).

Blind source separation (BSS) techniques such as ICA and PCA have been widely applied as a key signal condition method for ECG and other bioelectric signals [46–48]. Most bioelectrical signals have a specific source of generation, and noise observed in the sensors is generated from independent sources. ECG can be considered as a single localized electrical signal being observed from the body surface, while typical noise for ECG—movement artifact, EMG, electrode noise, etc.—are either non-localized or localized to different locations, making it spatially independent from ECG signal. By the nature of the ECG signal pattern, the common noise signals are also temporally independent from ECG signals. In sum, the lack of correlation, both spatially and temporally, is the foundation of BSS-based ECG noise reduction.

Hidden Markov model (HMM) is a statistical model where the system is assumed to be a Markovian process—each state entirely depends on its immediately previous states. Cardiac electricity is a well-controlled and organized electrical process and, therefore, largely suits the profile of a Markovian state. The application of HMM usually comprises two stages—the training stage, where the statistical model adapts to the series of events considered to be part of a Markovian process, and the application stage, where the ‘trained’ HMM is used to provide hidden states as encoded information of the states or provide an estimation of the incoming state [49–52]. The states can be a subcomponent of an ECG beat

or segment of ECG rhythms or events. Overall, HMM is a stochastic state model based on a previous state, and the probability distribution of the model depends on the training data. The nature of cardiac electricity is deterministic, unlike other bioelectric signals such as electroencephalogram (EEG), electrogastrogram (EGG), or electromyogram (EMG), making it a suitable target for HMM.

However, one should note that most arrhythmias are overwhelmingly low in prevalence in most populations but very critical for clinical purposes. One good example is high-degree AV blocks. Consequent P waves without QRS or T waves are very unlikely in HMM if such a model is trained on the basis of ECG of the general population. This can lead to false negative detection of P waves that are unsynchronized with QRS waves, arriving in incorrect rhythm analysis. In the process of ECG analysis, beat segmentation after, or as part of QRS detection is a key step in a successful ECG beat classification and analysis. HMM has been applied to ECG segmentation [49,50] to encode each ECG beat detected. Gaussian mixture model (GMM) is also applied in ECG segmentation and delineation processes. With GMM built for P, QRS, T, and ECG baseline, clustering methods can be applied to the ECG signal to increase the resolution of ECG segments [53]. Another unsupervised clustering method, self-organizing neural networks through competitive learning, is also used to improve QRS onset and offset detection [54].

Higher-order HMM (HOHMM) is also applied to beat classification in ECG analysis [51]. HOHMM, with a similar philosophy to RNN, expands the dependency of state to further the immediate previous state to allow for more complex dynamics to be modeled.

For beat classification, k-nearest neighbors (k-NN) or fuzzy C-means (FCM) methods are used very often. Clustering methods have been widely used in ECG beat analysis [55–59] without clearly defined labels of beat types. While the implementation of clustering methods varies, the core concept is to group the items in the targeted dataset on the basis of their ‘similarity’. In these clustering methods (KNN, FCM), the said similarity is defined directly through the Euclidean distance in the feature space. The major advantage of the clustering methods is the lack of dependency on labels, a critical bottleneck for all supervised methods. However, knowledge-based inputs are not completely absent, and even item by item labels are no longer needed. For FCM and KCM as examples, the number of clusters is needed for the formulation of the problem, and, therefore, an incorrect number of clusters will result in either duplication of cluster content or unnecessary mixings of different types of beats.

An auto-encoder can also be used for beat classification, which can create a latent representation of ECG with no externally labeled data and, therefore, be able to serve as a feature extraction component for further beat classification [60–62]. Once the auto-encoder reconstructs the ECG beats with proper accuracy, one can assume that the most critical features of the waveform have been captured and consolidated in the latent vector encoded, becoming the ideal feature for either clustering methods or supervised training methods that require fewer data. However, for ECG practices, it is not all sunny. In the training process of auto-encoder methods, the core objective function for reconstruction is usually second-order norm or energy-based. The clear advantage of the energy-based objective function is the simplicity in the generation of first-order derivatives and consequently better efficiency in training. However, this convenience comes at a price. For ECG analysis, the shape and location of P wave, a very small component in amplitude, is critical for the analysis of complicated rhythms such as atrial fibrillation or supraventricular arrhythmias. However, in an energy-based optimization scheme, more efforts will be focused on higher amplitude segments—QRS and T waves due to their size and less on P waves, especially when noise is present. Losing track of P waves may not jeopardize the results of simple tasks such as discriminating normal beats versus ventricular ectopy, but it will affect more complicated analysis involving atrial or junctional activities.

#### 4.2.3. Transfer Learning

Transfer learning is useful when there is not enough training data for a relatively large deep-learning model. The larger-deep model benefits from direct training on ECG waveform instead of extracted features. The main approach for transfer learning is to start with a pre-trained model, adding some padding on the front to adapt to the ECG input, and then adding a couple of layers on the back for final classification. During the training process, most weights from the original model are fixed, while only the weights of newly added layers are trained. In this way, the needed training set is much smaller than if the whole model had been trained. Alternatively, one could use the original deep model for feature extraction or use an additional simple model for final classification with extracted features from the deep model. The study of [45] used a very deep model as automatic feature extraction. The input is the spectrogram of ECG to fit into a 2-D CNN of the original model. The features are then extracted from the deep layer and fed to a support vector machine (SVM) for further training. The performance of classifying normal sinus rhythm, AFIB, VFIB, and ST change is 97%, a very good result considering the limited training samples (a total of 7008 data instances).

#### 4.2.4. Ensemble Learning

Ensemble learning involves combining multiple classification models to form a better performing one. This method has been used for computerized ECG interpretations by statistical learning methods before [63]. Each model can take different or the same input signal/features; the final classification is obtained by a voting method. Ensemble learning is much like multiple experts working on the same problem from different angles, and then the final decision is reached with some ‘consultation’ of the multiple solutions and proposals. Ensemble learning has been used in AECGs [62,64]. In the studies of [62,64], multiple ML algorithms were assembled to form a ‘super’ classifier for AFIB detection. The performance of the final ensemble classifier was better than any individual classifier.

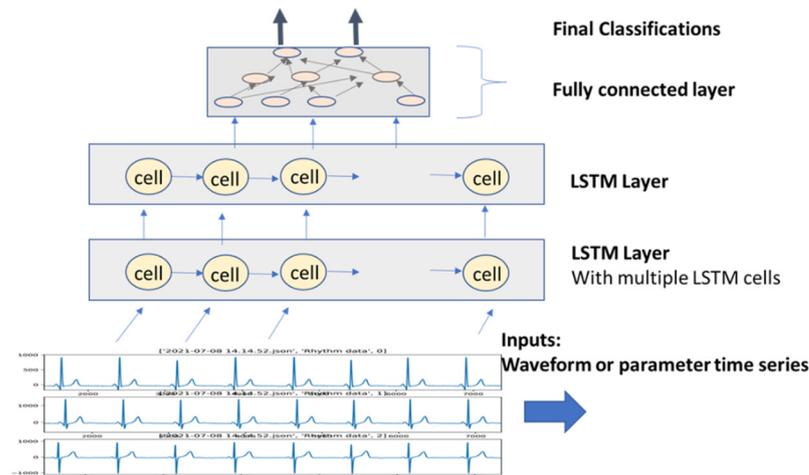
The random forest (RF) algorithm is a very powerful and robust ensemble method too. Instead of working on the ECG waveform, it takes ECG features as input. Inside the model, there are several random trees, each of which works on a random group of input variables in parallel. During the test, the sample is classified with the majority of trees agreed upon. One study used the RF algorithm for AFIB detection, employing 150 ECG time and frequency domain features as input, achieving high performance for AFIB detection when compared to other methods [65].

### 5. AECG Event Detection and Classification

Event detection and classification are the final steps for conventional AECG analysis. Here, the events are the interpretation of rhythm or morphology-related abnormal ECG classifications. For example, in rhythm analysis, ECG events can be classified into normal sinus rhythm, sinus rhythm with sinus arrhythmia, atrial fibrillation (AFIB), atrial flutter (AFLUT), junctional rhythm, ventricular rhythm, etc. Some morphology classifications include left or right bundle branch block (LBBB/RBBB), left ventricular hypertrophy (LVH), ischemia, myocardial infarction (MI), long QT (LQT). An AECG’s event detection is mainly focused on rhythm analysis, while resting ECG analysis usually targets more comprehensive rhythm and morphology analysis.

The conventional AECG analysis relies on the early stage of beat detection and beats classification for the final event detection. However, deep-learning models can potentially include all steps into one model, i.e., from ECG waveform input to event classification output, as we mentioned was previously shown by some very good performance studies [9,10]. At the same time, more AECG analyses need to provide more detailed reports on different aspects of beat-related information, such as number of PVC, PVC couplet, short run/long run VT, AFIB burden, and most crucially the trend of RR intervals. Therefore, the analysis modules of beat detection, beat classification, and event detection are still needed, even for ML-DNN-based algorithms.

RNN model is very useful for AECG time series event detection such as AFIB and PVC/VT. Figure 7 shows a diagram of a general structure of RNN event detection. Here, RNN models can include one or multiple RNN layers with multiple cells in each layer. In this case, long short-term memory (LSTM) cells are used. The cells in the same layer are connected with either one-directional links or dual-directional links. The input of RNN is a series of ECG waveforms or ECG parameters, e.g., R-R intervals or P-R-T components. The last block of the model consists of fully connected layers. The final output can be either binary or multiple classifications.



**Figure 7.** RNN model includes one or multiple RNN layers with multiple cells in each layer. Here, LSTM cells are used. The input of RNN is a series of ECG waveforms or ECG parameters, e.g., R-R intervals or P-R-T components. The last block of the model consists of fully connected layers. The final output can be either binary or multiple classifications.

For the application of unsupervised learning, with proper features extracted, most clustering methods can also be applied to event or rhythm classifications [66–71]. In contrast to beat clustering, rhythm segments contain significantly more information and therefore one of the most critical steps for a successful clustering application is to select proper features. Other than conventional features, DL methods such as auto-encoders have also been applied to provide features for clustering [61] in a similar fashion to ECG beat classifications.

The following are some major event detections of AECG:

### 5.1. AFIB/AFLUT

AFIB is one of the most prevalent arrhythmias, and missed diagnosis can result in possible stroke and heart failure. Early detection can result in the best treatment options. In the center of a conventional AFIB detection algorithm, there are two features most commonly used: R-R sequence and P wave status. For most AECGs, P wave detection is not reliable due to poor SNR, and therefore R-R sequence becomes the main feature used in most AFIB detection algorithms. It is not so difficult to build a high-sensitivity AFIB detection algorithm with RR sequence analysis alone, especially for longer ECG recordings. It is more challenging to achieve a very high specificity at the same time. Again, in a conventional AFIB detection algorithm, high specificity requires the result P wave status detection. For example, some sinus arrhythmias can have similar RR variations as AFIB, but a regular P wave in front of QRS can rule out AFIB to avoid false positive detection. However, for most one-lead AECGs, especially wearable devices, usually only an equivalent of lead I is available, which is not a very good lead for small P wave detection due to heart vector projection angle. In a standard 12-lead, lead II and v1 usually are the best leads for P wave detection.

Can state-of-art DL models detect AFIB better than conventional algorithms? The Mayo Clinic's study [9] showed that by using 12-lead 10 s ECG as input, the deep-learning model achieved an area under the curve (AUC) of 0.999 for AFIB detection with a 2.4 million ECG data set. Although this model uses the resting ECGs that usually have better SNR, especially for P wave, it still can be a benchmark for other deep-learning models that use ECG waveforms as input directly without explicit feature extraction.

In this AFIB detection challenge [64], there are many interesting AFIB algorithms presented, all of which are machine learning-based. Some of them need explicit feature extraction, while others are deep-learning models with ECG waveform as input. The best performance of the challenge was achieved by the method of feeding time sequence of ECG features to an RNN model [72]. It also provided a high degree of interpretability. These are very encouraging results since interpretability is still very important for medical applications. Most deep-learning models achieved good performance by acting as a 'black box'. Several top performances were achieved with random forest method, a powerful statistical machine learning method that combines features of time and frequency domains [65,73]. Furthermore, an ensemble learning is proposed on the basis of the algorithms joined challenge [64]. It demonstrated that by using top performing algorithms or all algorithms, the AFIB detection accuracy can be higher than any single algorithm can achieve.

## 5.2. PVC/VT

The detection of premature ventricular contraction (PVC) is one major task of any AECG algorithm. Although a few isolated PVCs might not have any pathological significance, some more severe arrhythmias such as bigeminy, trigeminy, couplet, and ventricular tachycardia are all associated with PVC detection.

Conventional machine learning AECG algorithms use feature-based template matching for PVC detection [74]. The features can include R-R intervals, the pattern of QRS complex, width of QRS complex, and ST-T wave. Noise removal and detection are still the keys for high-accuracy PVC detection. The pattern matching and clustering technique mentioned previously can be used for grouping similar morphology complexes together. Some most often used algorithms include k-nearest neighbors (k-NN) [75], discrete hidden Markov model [76], support vector machine, Bayesian classification [77], and random forest [78].

Deep-learning PVC detection algorithms have been presented in many recent studies [61,79–83]. Many of these algorithms use ECG waveform directly without conducting explicit feature extraction, although some extract features from DL models and feed them into a non-deep model such as SVM. Since PVC detection requires both morphological pattern information and time sequence information, LSTM models that extract time series features from incoming ECG signals perform well when used properly [61].

The potentials of deep-learning models are not only served to improve pattern classification, but can also enhance the causality analysis of PVC. One study applied the CNN model to detect PVC origin [84]. It combined a ventricular computer model into the training scheme. The training datasets were generated by multiplying ventricular current dipoles derived from single pacing at various locations with a patient-specific lead field. The origins of PVC are localized by calculating the weighted center of gravity of classification returned by the CNNs. Although the testing results are limited by the number of cases, it still is a very interesting direction for deep-learning models.

Severe arrhythmias including ventricular and supraventricular tachycardia (VT, SVT) are the critical target events for many AECG algorithms. DL models have shown the potential of detecting VT and SVT already [10,80]. Most of those models are CNN deep-learning with ECG waveform as inputs, while some are RNN-based models. The varying length of signal segments is used ranges between 1 and 30 s. The longer the segment, the larger the data set needed. Again, the key is to let models learn the difference between noise segment and VT/SVT segment. The noise is a major issue for false-positives of

VT/STV/VF detection in conventional algorithms [85], where both time and frequency domain features are extracted from every 5 s segment to form a time sequence.

### 5.3. QT Analysis

QT interval is one of the most difficult ECG measurements, and therefore most diagnostic QT estimations are based on resting 12-lead ECG. However, AECG can provide continuous QT analysis and trends with their longer recording time. Conventional QT algorithms involve ECG filtering, segmentation, and estimation [86]. The QT algorithm's accuracy has been improved significantly over the previous 12-lead QT algorithm [87].

DL algorithms have been applied for QT estimation. This study takes ECG waveform input directly without feature extraction [88]. It is trained on 2.4 million ECGs from the Mayo Clinic's ECG database, but with only 2 out of 12 leads. The algorithm uses both regression and classification schemes in forming its special cost function. The model consists of three residual layer blocks and two fully connected layers. The test performance against the doctor's annotation is close to that of other major QT algorithms. In addition, the algorithm was also applied to the data collected with the two-lead mobile electrocardiogram device, Kardia 6L<sup>TM</sup>. The results are also within the error range of the measurement standard (IEC 60601-2-25 (diagnostic electrocardiographs)).

### 5.4. Noise Segment

It is worth taking a separate section for the discussion of noise in the event detection, since it is so critical in any AECG processing that it is not surprising to know that half of the codes of an AECG program can be devoted to noise detection and the separation of noise from the signals. Another reason for discussing more about noise is that modern machine learning models have made some significant contributions to noise detection, more than any other processing stages for AECG processing.

Using DNN models to detect noise segments has been proven to be very robust. The method of DNN noise detection includes the CNN classification model [89], where training pair is formed with controlled SNR and the noise labels, the auto-encoder method [90]. The differences between these two types of noise detection deep-learning models are that the first type use CNN to classify noisy ECG into noisy/clean directly, and the second type first uses an auto-encoder to train the encoder to form the latent variables and then use this layer as the input for next noise classification model. Both methods do not need to extract features explicitly as conventional noise detection algorithms.

## 6. ECG Risk Stratification/Prediction

Predictions of severe arrhythmia/cardiac events are always very challenging since the performance of predictions is usually lower than the detections when the events are already happening. The low prevalence of the future event is another reason why the prediction task usually has a very low positive predictive value. Predicting the new onset of AFIB should be meaningful for prevention purposes since adverse outcomes of undetected AFIB can be stroke and heart failure. Previous algorithms for predicting AFIB included P wave averaging analysis [91,92]. The difficulty here is to find the best feature(s) for the prediction.

One study used a large database of 12-lead ECG with 1.6 million ECGs to predict one-year AFIB occurrence probability [93]. The AFIB prediction performance was 0.85 of AUC. A more significant study used a machine learning model to detect the mechanism of AFIB, as well as to guide the best treatment of AFIB ablation [94]. AFIB drives were induced in two computerized atrial models and combined with eight torso models. A total of 103 features were extracted from the signals. A binary decision tree classifier was trained on the simulated data and evaluated using hold-out cross-validation. The classifier yielded 82.6% specificity and 73.9% sensitivity for detecting pulmonary vein drivers on the clinical data.

One study focused on risk stratification of mortality of patients with acute myocardial infarction [95]. It used a large data registry of Korean acute myocardial infarction (AMI),

including 22,875 AMI patients from 59 hospitals. The study took 36 hospitals' data for the training set and 23 hospitals' data for the testing set. A DL model with ECG and other clinical parameters was used. The results with the DL model achieved an AUC (area under the curve) of 0.905 vs. the 0.85 GRACE score.

These studies built prediction models for more life-threatening events, cardiac arrest, and ventricular fibrillation through DL. DL-based models extract features from ECG waveforms directly. The advantages might be that the feature extraction is automatic, but in the meantime, it is a non-transparent model approach, and therefore the mechanism under hidden features is not clear. In building the prediction models for risk stratification, understanding the underline mechanism of the risks is also very important as, we learned from T wave alternans (TWA)-related risk stratification for sudden cardiac death (SCD) [96], where researchers found TWA is not only closely associated with an increased risk of SCD, but its mechanism is also connected to perturbations in calcium transport processes. Therefore, TWA may play a role not only in risk stratification but also in the pathogenesis of ventricular tachyarrhythmia events. This is what we hope DL-based models can also contribute to in terms of more such causal prediction and analysis.

## 7. Discussion

This review covers a very wide range of the AECG application of machine learning-based algorithms, especially DL models. The development and efforts in the AECG algorithms have never been so great and so fast, partially due to the fast advancement of machine learning fields in parallel and interdisciplinary paths.

For conventional AECG algorithms, feature extraction is one of the most important steps, and also one of the most demanding tasks. There are two main reasons to explain why feature extraction is so critical. The first one is that most features used in the algorithm have links to the underlying physiological structures and conditions. For example, P waves are associated with atrial excitation of the heart, QRS is associated with ventricular depolarization, and ST-T segments are related to the repolarization process. The second reason is that most currently used interpretive algorithms are expert system-based, following the same logic and analysis that physicians are using to analyze pathological changes of the heart.

As we have indicated, among the most noticeable advantages of DL models is their ability to handle ECG waveform directly without an explicit feature extraction process. At the same time, this waveform-to-event-detection process of DL models also brings the arguments of the 'black box' model versus a more transparent model of an expert system-based algorithm. Is there a way to take advantage of both approaches to have a more automatic and transparent model? There have been some efforts in this regard. One study used an input sensitivity analysis for trained neural networks for acute MI detection [97]. This feature sensitivity analysis was also extended to DL models, and thus it was made clear as to which part of ECG segments was more significant for their contribution to the final output. For example, for an ML model that differentiates sinus rhythm from atrial fibrillation, it would be helpful to understand if the model makes the detection only on the basis of R-R interval information, or whether it actually also uses P wave information itself. There are some other efforts to make DL models more interpretable [98].

There is already much research to build risk stratification algorithms from AECGs, such as T wave alternans (TWA), heart rate turbulence (HRT), heart rate variability (HRV), and QT interval dynamicity. The center point of those algorithms is to identify the features related to the physiological system. For example, TWA is linked to the substrate of myocardial tissues, and HRT and HRV are related to the control of the autonomous system to the heart condition. Those features help us to understand the mechanism of the changes of ECG. Thus far, most DL models are still only performing a better job on pattern recognition. It is important to improve the accuracy of detecting cardiac events with AECG, but it shall also be meaningful to reveal the causal relationships between surface ECG and underlying

mechanisms. With the large data sets and large models, it might help us to find more relevant features.

One topic not discussed here is so-called online learning, or adaptive learning, meaning that an event detection model can be updated with the incoming data and new labels that can be edited by physicians [99]. There have been few to no successful online learning models in AECG. This is partly due to the technical challenge of balancing pre-trained models with the new data and is partly due to the difficulty of meeting regulatory constraints, which require a model to be fully tested and verified to pass certain standards. However, in this era of deep learning and big data, it is reasonable to take more action in this direction. One big advantage of machine learning is that the model can be built quickly and automatically with the available big data. The technical challenge is how to balance the previously trained data set and the new data set. Some methods and ideas from transfer learning can be borrowed, such as fixing most parts of an already trained model, only retraining, or adding a few layers for the new data. However, an automatic verification process also needs to be built into the process to meet the regulator requirements.

It is no secret that DL models heavily rely on the availability of big data sets for training. The model overfitting is caused by not enough training data for a large model so that the performance for independent test data is poor. This is what we called ‘model generalization’. To some extent, the test and verification are more important for deep-learning models than conventional detection algorithms when a smaller data set is used. Many papers and research studies use MIT/BME ECG databases on the Physionet [31], classic data sets that have been used for the last 30 years to test various ECG algorithms. The question is whether these databases are still sufficient for building and testing deep-learning models. If not, perhaps work toward building larger public data sets for deep-learning models of AECG is required? We are so glad to see the efforts in this direction in the studies/challenges organized by Computing in Cardiology [100]. In these challenges, large data sets were donated by different institutes, and multiple algorithms were built and shared in the AECG society. What is even more exciting is the effort of putting all these algorithms in the direction of ensemble learning. This type of collaborative learning can also be very promising for targeting some very complicated AECG tasks in the future.

**Author Contributions:** L.Y. contributed to the portion of unsupervised learning; J.X. contributed to the rest of the paper. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Acknowledgments:** The authors would like to acknowledge discussions of AECG and machine learning with Gari Clifford and Bioinformatics group of Emory University and machine learning group of Alivecor Inc.

**Conflicts of Interest:** Joel Xue is a research fellow with AliveCor Inc.

## References

1. Steinberg, J.S.; Varma, N.; Cygankiewicz, I.; Aziz, P.; Balsam, P.; Baranchuk, A.; Cantillon, D.J.; Dilaveris, P.; Dubner, S.J.; El-Sherif, N.; et al. 2017 ISHNE-HRS expert consensus statement on ambulatory ECG and external cardiac monitoring/telemetry. *Heart Rhythm* **2017**, *14*, 55–96, Epub 2017 May 8. [[CrossRef](#)] [[PubMed](#)]
2. Rosenberg, M.A.; Samuel, M.; Thosani, A.; Zimetbaum, P.J. Use of a Noninvasive Continuous Monitoring Device in the Management of Atrial Fibrillation: A Pilot Study. *Pacing Clin. Electrophysiol.* **2013**, *36*, 328. [[CrossRef](#)] [[PubMed](#)]
3. Isakadze, N.; Martin, S.S. How useful is the smartwatch ECG? *Trends Cardiovasc. Med.* **2020**, *30*, 442–448. [[CrossRef](#)]
4. Hall, A.; Mitchell, A.R.J.; Wood, L.; Holland, C. Effectiveness of a single lead AliveCor electrocardiogram application for the screening of atrial fibrillation: A systematic review. *Medicine* **2020**, *99*, e21388. [[CrossRef](#)] [[PubMed](#)]
5. Pan, J.; Tompkins, W.J. A Real-Time QRS Detection Algorithm. *IEEE Trans. Biomed. Eng.* **1985**, *32*, 230–236. [[CrossRef](#)] [[PubMed](#)]
6. Manisha; Dhull, S.K.; Singh, K.K. ECG Beat Classifiers: A Journey from ANN to DNN. *Procedia Comput. Sci.* **2020**, *167*, 747–759. [[CrossRef](#)]
7. Somani, S.; Russak, A.J.; Richter, F.; Zhao, S.; Vaid, A.; Chaudhry, F.; De Freitas, J.K.; Naik, N.; Miotto, R.; Nadkarni, G.N.; et al. Deep learning and the electrocardiogram: Review of the current state-of-the-art. *Europace* **2021**, *23*, 1179–1191. [[CrossRef](#)]

8. Ribeiro, A.L.P.; Ribeiro, M.H.; Paixão, G.M.M.; Oliveira, D.M.; Gomes, P.R.; Canazart, J.A.; Ferreira, M.P.S.; Andersson, C.R.; Macfarlane, P.W.; Meira, W.; et al. Automatic diagnosis of the 12-lead ECG using a deep neural network. *Nat. Commun.* **2020**, *11*, 1760. [CrossRef]
9. Kashou, A.H.; Ko, W.-Y.; Attia, Z.I.; Cohen, M.S.; Friedman, P.A.; Noseworthy, P.A. A comprehensive artificial intelligence-enabled electrocardiogram interpretation program. *Cardiovasc. Digit. Health J.* **2020**, *1*, 62–70. [CrossRef]
10. 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]
11. Lu, W.; Shuai, J.; Gu, S.; Xue, J. Method to Annotate Arrhythmias by Deep Network. In Proceedings of the 2018 IEEE International Conference on Internet of Things (iThings), Halifax, NS, Canada, 30 July–3 August 2018. [CrossRef]
12. Hong, S.; Zhou, Y.; Shang, J.; Xiao, C.; Sun, J. Opportunities and Challenges of Deep Learning Methods for Electrocardiogram Data: A Systematic Review. *Comput. Biol. Med.* **2020**, *122*, 103801. [CrossRef] [PubMed]
13. Ebrahimi, Z.; Loni, M.; Daneshlab, M.; Gharehbaghi, A. A review on deep learning methods for ECG arrhythmia classification. *Expert Syst. Appl.* **2020**, *7*, 100033. [CrossRef]
14. Serhani, M.A.; El Kassabi, H.T.; Ismail, H.; Navaz, A.N. ECG Monitoring Systems: Review, Architecture, Processes, and Key Challenges. *Sensors* **2020**, *20*, 1796. [CrossRef] [PubMed]
15. Wasimuddin, M.; Elleithy, K.; Abuzneid, A.S.; Faezipour, M.; Abuzaghlh, O. Stages-based ECG signal analysis from traditional signal processing to machine learning approaches: A survey. *IEEE Access* **2020**, *8*, 177782–177803. [CrossRef]
16. Hoffmann, J.; Mahmood, S.; Fogou, P.S.; George, N.; Raha, S.; Safi, S.; Schmailzl, K.J.; Brandalero, M.; Hubner, M. A Survey on Machine Learning Approaches to ECG Processing. In *Signal Processing—Algorithms, Architectures, Arrangements, and Applications*; IEEE: Piscataway, NJ, USA, 2020; pp. 36–41. [CrossRef]
17. Hastie, T.; Tibshirani, R.; Friedman, J. *The Elements of Statistical Learning—Data Mining, Inference, and Prediction*; Springer Series in Statistics; Springer: New York, NY, USA, 2008.
18. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016; Available online: <http://www.deeplearningbook.org> (accessed on 17 July 2021).
19. Xue, J.Q.; Hu, Y.H.; Tompkins, W.J. Neural-Network-Based Adaptive matched Filtering for QRS Detection. *IEEE Trans. Biomed. Eng.* **1992**, *39*, 317–329. [CrossRef] [PubMed]
20. Xuel, J.Q.; Hu, Y.H.; Tompkins, W.J. Training of ECG signals in nueral network pattern recognition. In Proceedings of the Internaional Conference of the IEEE Engineering in Medicine and Biology Society, Philadelphia, PA, USA, 1–4 November 1990; Volume 12, pp. 1465–1466.
21. Xue, Q.; Reddy, B.R.S. Late Potential Recognition by Artificial Neural Networks. *IEEE Trans. Biomed. Eng.* **1997**, *44*, 132–143. [CrossRef] [PubMed]
22. LeCun, Y.; Haffner, P.; Bottou, L.; Bengio, Y. Object recognition with gradient-based learning. In *Shape, Contour and Grouping in Computer Vision*; Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); Springer: Berlin/Heidelberg, Germany, 1999; Volume 1681, pp. 319–345. [CrossRef]
23. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
24. Arulkumaran, K.; Deisenroth, M.P.; Brundage, M.; Bharath, A.A. A Brief Survey of Deep Reinforcement Learning. *IEEE Signal Process. Mag.* **2017**, *34*, 26–38. [CrossRef]
25. Silver, D.; Huang, A.; Maddison, C.J.; Guez, A.; Sifre, L.; Driessche, G.V.D.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M.; et al. Mastering the game of Go with deep neural networks and tree search. *Nature* **2016**, *529*, 484–489. [CrossRef]
26. Insani, A.; Jatmiko, W.; Sugiarto, A.T.; Jati, G.; Wibowo, S.A. Investigation Reinforcement learning method for R-Wave detection on Electrocardiogram signal. In Proceedings of the International Seminar on Research of Information Technology and Intelligent Systems, Yogyakarta, Indonesia, 5–6 December 2019. [CrossRef]
27. IEC 60601-2-47:2012 | IEC Webstore. Available online: <https://webstore.iec.ch/publication/2666> (accessed on 17 July 2021).
28. IEC 60601-2-25:2011 | IEC Webstore. Available online: <https://webstore.iec.ch/publication/2636> (accessed on 17 July 2021).
29. Thakor, N.V.; Zhu, Y.-S. Adaptive filtering to ECG analysis. *IEEE Trans. Biomed. Eng.* **1991**, *38*, 785–794. [CrossRef]
30. Awal, M.A.; Mostafa, S.S.; Ahmad, M.; Rashid, M.A. An adaptive level dependent wavelet thresholding for ECG denoising. *Biocybern. Biomed. Eng.* **2014**, *34*, 238–249. [CrossRef]
31. PhysioBank Databases. Available online: <https://archive.physionet.org/physiobank/database/> (accessed on 17 July 2021).
32. Arsene, C. Design of Deep Convolutional Neural Network Architectures for Denoising Electrocardiographic Signals. In Proceedings of the IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology, Viña del Mar, Chile, 27–29 October 2020. [CrossRef]
33. Xiong, P.; Wang, H.; Liu, M.; Zhou, S.; Hou, Z.; Liu, X. ECG signal enhancement based on improved denoising auto-encoder. *Eng. Appl. Artif. Intell.* **2016**, *52*, 194–202. [CrossRef]
34. Xiong, P.; Wang, H.; Liu, M.; Liu, X. Denoising auto-encoder for eletrocardiogram signal enhancement. *J. Med Imaging Health Inform.* **2015**, *5*, 1804–1810. [CrossRef]
35. Xiong, P.; Wang, H.; Liu, M.; Lin, F.; Hou, Z.; Liu, X. A stacked contractive denoising auto-encoder for ECG signal denoising. *Physiol. Meas.* **2016**, *37*, 2214–2230. [CrossRef] [PubMed]

36. Garus, J.; Pabian, M.; Wisniewski, M.; Sniezynski, B. Electrocardiogram Quality Assessment with Auto-encoder. In Proceedings of the International Conference on Computational Science, 2021 ICCS, Krakow, Poland, 16–18 June 2021; Springer: Cham, Switzerland, 2021; pp. 693–706. [\[CrossRef\]](#)
37. Antczak, K. A Generative Adversarial Approach to ECG Synthesis and Denoising. *arXiv* **2020**, arXiv:2009.02700v1.
38. Yildirim, O.; Tan, R.S.; Acharya, U.R. An efficient compression of ECG signals using deep convolutional auto-encoders. *Cogn. Syst. Res.* **2018**, *52*, 198–211. [\[CrossRef\]](#)
39. Wang, F.; Ma, Q.; Liu, W.; Chang, S.; Wang, H.; He, J.; Huang, Q. A novel ECG signal compression method using spindle convolutional auto-encoder. *Comput. Methods Programs Biomed.* **2019**, *175*, 139–150. [\[CrossRef\]](#)
40. Bekiryazici, T.; Gurkan, H. ECG Compression method based on convolutional auto-encoder and discrete wavelet transform. In Proceedings of the 2020 28th Signal Processing and Communications Applications Conference (SIU 2020), Gaziantep, Turkey, 5–7 October 2020. [\[CrossRef\]](#)
41. Arsene, C. Complex Deep Learning Models for Denoising of Human Heart ECG signals. *arXiv* **2019**, arXiv:1908.10417.
42. Antczak, K. Deep Recurrent Neural Networks for ECG Signal Denoising. *arXiv* **2018**, arXiv:1807.11551v3.
43. Johnson, J.M.; Khoshgoftaar, T.M. Survey on deep learning with class imbalance. *J. Big Data* **2019**, *6*, 27. [\[CrossRef\]](#)
44. Weimann, K.; Conrad, T.O.F. Transfer learning for ECG classification. *Sci. Rep.* **2021**, *11*, 5251. [\[CrossRef\]](#)
45. Salem, M.; Taheri, S.; Yuan, J.-S. ECG Arrhythmia Classification Using Transfer Learning from 2-Dimensional Deep CNN Features. In Proceedings of the IEEE Biomedical Circuits and Systems Conference (BioCAS), Cleveland, OH, USA, 17–19 October 2018. [\[CrossRef\]](#)
46. Xue, J.; Gao, W.; Chen, Y.; Han, X. Study of repolarization heterogeneity and electrocardiographic morphology with a modeling approach. *J. Electrocardiol.* **2008**, *41*, 581–587. [\[CrossRef\]](#)
47. Romeo, I. PCA and ICA applied to noise reduction in multi-lead ECG. In *Computer in Cardiology*; IEEE: Piscataway, NJ, USA, 2011; pp. 613–616.
48. Alickovic, E.; Subasi, A. Effect of Multiscale PCA De-noising in ECG Beat Classification for Diagnosis of Cardiovascular Diseases. *Circuits Syst. Signal Process.* **2015**, *34*, 513–533. [\[CrossRef\]](#)
49. Koski, A. Modelling ECG signals with hidden Markov models. *Artif. Intell. Med.* **1996**, *8*, 453–471. [\[CrossRef\]](#)
50. Andreão, R.V.; Dorizzi, B.; Boudy, J. ECG signal analysis through hidden Markov models. *IEEE Trans. Biomed. Eng.* **2006**, *53*, 1541–1549. [\[CrossRef\]](#)
51. Liao, Y.; Xiang, Y.; Du, D. Automatic Classification of Heartbeats Using ECG Signals via Higher Order Hidden Markov Model. In Proceedings of the IEEE International Conference on Automation Science and Engineering, Hong Kong, China, 20–21 August 2020; pp. 69–74. [\[CrossRef\]](#)
52. Silipo, R.; Deco, G.; Schürmann, B.; Vergassola, R.; Gremigni, C. Investigating the underlying Markovian dynamics of ECG rhythms by information flow. *Chaos Solitons Fractals* **2001**, *12*, 2877–2888. [\[CrossRef\]](#)
53. Hajimolahoseini, H.; Hashemi, J.; Redfearn, D. ECG Delineation for Qt Interval Analysis Using an Unsupervised Learning Method. In Proceedings of the ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing, Calgary, AB, Canada, 15–20 April 2018; pp. 2541–2545. [\[CrossRef\]](#)
54. Tighiouart, B.; Rubel, P.; Bedda, M. Improvement of QRS boundary recognition by means of unsupervised learning. *Comput. Cardiol.* **2003**, *30*, 49–52. [\[CrossRef\]](#)
55. Lagerholm, M.; Peterson, G. Clustering ECG complexes using hermite functions and self-organizing maps. *IEEE Trans. Biomed. Eng.* **2000**, *47*, 838–848. [\[CrossRef\]](#) [\[PubMed\]](#)
56. Zhang, C.; Wang, G.; Zhao, J.; Gao, P.; Lin, J.; Yang, H. Patient-specific ECG classification based on recurrent neural networks and clustering technique. In Proceedings of the 13th IASTED International Conference on Biomedical Engineering (BioMed 2017), Innsbruck, Austria, 20–21 February 2017; pp. 63–67. [\[CrossRef\]](#)
57. Osowski, S.; Linh, T.H. ECG beat recognition using fuzzy hybrid neural network. *IEEE Trans. Biomed. Eng.* **2001**, *48*, 1265–1271. [\[CrossRef\]](#) [\[PubMed\]](#)
58. Chudáček, V.; Petřík, M.; Georgoulas, G.; Čepeck, M.; Lhotská, L.; Stylios, C. Comparison of seven approaches for holter ECG clustering and classification. In Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology, Lyon, France, 22–26 August 2007; pp. 3844–3847. [\[CrossRef\]](#)
59. Korürek, M.; Nizam, A. A new arrhythmia clustering technique based on Ant Colony Optimization. *J. Biomed. Inform.* **2008**, *41*, 874–881. [\[CrossRef\]](#) [\[PubMed\]](#)
60. Nurmaini, S.; Partan, R.U.; Caesarendra, W.; Dewi, T.; Rahmatullah, M.N.; Darmawahyuni, A.; Bhayyu, V.; Firdaus, F. An automated ECG beat classification system using deep neural networks with an unsupervised feature extraction technique. *Appl. Sci.* **2019**, *9*, 2921. [\[CrossRef\]](#)
61. Hou, B.; Yang, J.; Wang, P.; Yan, R. LSTM-Based Auto-Encoder Model for ECG Arrhythmias Classification. *IEEE Trans. Instrum. Meas.* **2020**, *69*, 1232–1240. [\[CrossRef\]](#)
62. Kuznetsov, V.V.; Moskalenko, V.A.; Zolotykh, N.Y. Electrocardiogram Generation and Feature Extraction Using a Variational Auto-encoder. *arXiv* **2020**, arXiv:2002.00254.
63. Rubel, P.; Fayn, J.; Macfarlane, P.W.; Pani, D.; Schlögl, A.; Värrri, A. The History and Challenges of SCP-ECG: The Standard Communication Protocol for Computer-Assisted Electrocardiography. *Hearts* **2021**, *2*, 384–409. [\[CrossRef\]](#)

64. Clifford, G.; Liu, C.; Moody, B.; Lehman, L.-W.; Silva, I.; Li, Q.; Johnson, A.; Mark, R. AF Classification from a Short Single Lead ECG Recording: The PhysioNet/Computing in Cardiology Challenge 2017. In *Computing in Cardiology*; IEEE: Piscataway, NJ, USA, 2017. [[CrossRef](#)]
65. Zabihi, M.; Rad, A.B.; Katsaggelos, A.K.; Kiranyaz, S.; Narkilahti, S.; Gabbouj, M. Detection of atrial fibrillation in ECG hand-held devices using a random forest classifier. In *Computing in Cardiology*; IEEE: Piscataway, NJ, USA, 2017; Volume 44, pp. 1–4. [[CrossRef](#)]
66. Lin, Z.; Ge, Y.; Tao, G. Algorithm for Clustering Analysis of ECG Data. In Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual Conference, Shanghai, China, 17–18 January 2006; pp. 3857–3860. [[CrossRef](#)]
67. Xia, Y.; Han, J.; Wang, K. Quick detection of QRS complexes and R-waves using a wavelet transform and K-means clustering. *Bio-Med Mater. Eng.* **2015**, *26*, S1059–S1065. [[CrossRef](#)]
68. Özbay, Y.; Ceylan, R.; Karlik, B. A fuzzy clustering neural network architecture for classification of ECG arrhythmias. *Comput. Biol. Med.* **2006**, *36*, 376–388. [[CrossRef](#)] [[PubMed](#)]
69. Stridh, M.; Rosenqvist, M. Automatic screening of atrial fibrillation in thumb-ECG recordings. In Proceedings of the 39th Conference on Computing in Cardiology, Krakow, Poland, 9–12 September 2012; IEEE: Krakow, Poland, 2012; Volume 39, pp. 161–164.
70. Plesinger, F.; Nejedly, P.; Viscor, I.; Halamek, J.; Jurak, P. Automatic detection of atrial fibrillation and other arrhythmias in holter ECG recordings using rhythm features and neural networks. In *Computing in Cardiology*; IEEE: Piscataway, NJ, USA, 2017; Volume 44, pp. 1–4. [[CrossRef](#)]
71. Tan, W.W.; Foo, C.L.; Chua, T.W. Type-2 fuzzy system for ECG arrhythmic classification. In Proceedings of the IEEE International Conference on Fuzzy Systems, London, UK, 23–26 July 2007. [[CrossRef](#)]
72. Teijeiro, T.; García, C.A.; Castro, D.; Félix, P. Arrhythmia classification from the abductive interpretation of short single-lead ECG records. In *Computing in Cardiology*; IEEE: Piscataway, NJ, USA, 2017; Volume 44, pp. 1–4. [[CrossRef](#)]
73. Antink, C.H.; Leonhardt, S.; Walter, M. Fusing QRS detection and robust interval estimation with a random forest to classify atrial fibrillation. In *Computing in Cardiology*; IEEE: Piscataway, NJ, USA, 2017; Volume 44, pp. 1–4. [[CrossRef](#)]
74. Yu, J.; Wang, X.; Chen, X.; Guo, J. Automatic Premature Ventricular Contraction Detection Using Deep Metric Learning and KNN. *Biosensors* **2021**, *11*, 69. [[CrossRef](#)]
75. Kaya, Y.; Pehlivan, H. Feature selection using genetic algorithms for premature ventricular contraction classification. In Proceedings of the ELECO 2015—9th International Conference on Electrical and Electronics Engineering, Bursa, Turkey, 26–28 November 2015; pp. 1229–1232. [[CrossRef](#)]
76. Boublenza, S.; Chikh, A.; Bouchikhi, M.A. Discrete hidden Markov model classifier for premature ventricular contraction detection. *Int. J. Biomed. Eng. Technol.* **2015**, *17*, 371–386.
77. Casas, M.M.; Avitia, R.L.; Gonzalez-Navarro, F.F.; Cardenas-Haro, J.A.; Reyna, M.A. Bayesian Classification Models for Premature Ventricular Contraction Detection on ECG Traces. *J. Healthc. Eng.* **2018**, *2018*, 2694768. [[CrossRef](#)]
78. Xie, T.; Li, R.; Shen, S.; Zhang, X.; Zhou, B.; Wang, Z. Intelligent Analysis of Premature Ventricular Contraction Based on Features and Random Forest. *J. Healthc. Eng.* **2019**, *2019*, 5787582. [[CrossRef](#)]
79. De Marco, F.; Finlay, D.; Bond, R.R. Classification of Premature Ventricular Contraction Using Deep Learning. In *Computing in Cardiology*; IEEE: Piscataway, NJ, USA, 2020. [[CrossRef](#)]
80. Murugesan, B.; Ravichandran, V.; Ram, K.; S.P., P.; Joseph, J.; Shankaranarayana, S.M.; Sivaprakasam, M. ECGNet: Deep Network for Arrhythmia Classification. In Proceedings of the IEEE International Symposium on Medical Measurements and Applications (MeMeA), Rome, Italy, 11–13 June 2018. [[CrossRef](#)]
81. Novotna, P.; Vicar, T.; Ronzhina, M.; Hejc, J.; Kolarova, J. Deep-Learning Premature Contraction Localization in 12-lead ECG from Whole Signal Annotations. In *Computing in Cardiology*; IEEE: Piscataway, NJ, USA, 2020. [[CrossRef](#)]
82. Javadi, M.; Ebrahimpour, R.; Sajedin, A.; Faridi, S.; Zakernejad, S. Improving ECG classification accuracy using an ensemble of neural network modules. *PLoS ONE* **2011**, *6*, e24386. [[CrossRef](#)]
83. Teplitzky, B.A.; McRoberts, M.; Ghanbari, H. Deep learning for comprehensive ECG annotation. *Heart Rhythm* **2020**, *17*, 881–888. [[CrossRef](#)]
84. Yang, T.; Yu, L.; Jin, Q.; Wu, L.; He, B. Localization of Origins of Premature Ventricular Contraction by Means of Convolutional Neural Network from 12-Lead ECG. *IEEE Trans. Biomed. Eng.* **2018**, *65*, 1662–1671. [[CrossRef](#)] [[PubMed](#)]
85. Prabhakararao, E.; Manikandan, M.S. Efficient and robust ventricular tachycardia and fibrillation detection method for wearable cardiac health monitoring devices. *Healthc. Technol. Lett.* **2016**, *3*, 239–246. [[CrossRef](#)] [[PubMed](#)]
86. Xue, J.Q. Robust QT Interval Estimation—From Algorithm to Validation. *Ann. Noninvasive Electrocardiol.* **2009**, *14* (Suppl. 1), S35. [[CrossRef](#)]
87. Hnatkova, K.; Gang, Y.; Batchvarov, V.N.; Malik, M. Precision of QT interval measurement by advanced electrocardiographic equipment. *Pacing Clin. Electrophysiol. PACE* **2006**, *29*, 1277–1284. [[CrossRef](#)] [[PubMed](#)]
88. Giudicessi, J.R.; Schram, M.; Bos, J.M.; Galloway, C.D.; Shreibati, J.B.; Johnson, P.W.; Carter, R.E.; Disrud, L.W.; Kleiman, R.; Attia, Z.I.; et al. Artificial Intelligence-Enabled Assessment of the Heart Rate Corrected QT Interval Using a Mobile Electrocardiogram Device. *Circulation* **2021**, *143*, 1274–1286. [[CrossRef](#)]

89. Ansari, S.; Gryak, J.; Najarian, K. Noise Detection in Electrocardiography Signal for Robust Heart Rate Variability Analysis: A Deep Learning Approach. In Proceedings of the 2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Honolulu, HI, USA, 18–21 July 2018.
90. Sasmal, B.; Roy, S. ECG Guided Automated Diagnostic Intervention of Cardiac Arrhythmias with Extra-Cardiac Noise Detection using Neural Network. In Proceedings of the 7th International Conference on Optimization and Applications (ICOA), Wolfenbüttel, Germany, 19–20 May 2021. [[CrossRef](#)]
91. Dhala, A.; Underwood, D.; Leman, R.; Madu, E.; Baugh, D.; Ozawa, Y.; Kasamaki, Y.; Xue, Q.; Reddy, S. Signal-Averaged P-Wave Analysis of Normal Controls and Patients with Paroxysmal Atrial Fibrillation: A Study in Gender Differences, Age Dependence, and Reproducibility. *Clin. Cardiol.* **2002**, *25*, 525. [[CrossRef](#)] [[PubMed](#)]
92. Budeus, M.; Hennersdorf, M.; Felix, O.; Reimert, K.; Perings, C.; Sack, S.; Wieneke, H.; Erbel, R. Prediction of atrial fibrillation in patients with cardiac dysfunction: P wave signal-averaged ECG and chemoreflexsensitivity in atrial fibrillation. *Europace* **2007**, *9*, 601–607. [[CrossRef](#)] [[PubMed](#)]
93. Raghunath, S.; Pfeifer, J.M.; Ulloa-Cerna, A.E.; Nemani, A.; Carbonati, T.; Jing, L.; Vanmaanen, D.P.; Hartzel, D.N.; Ruhl, J.A.; Lagerman, B.F.; et al. Deep Neural Networks Can Predict New-Onset Atrial Fibrillation from the 12-Lead ECG and Help Identify Those at Risk of Atrial Fibrillation-Related Stroke. *Circulation* **2021**, *143*, 1287–1298. [[CrossRef](#)] [[PubMed](#)]
94. Luongo, G.; Azzolin, L.; Schuler, S.; Rivolta, M.W.; Almeida, T.P.; Martínez, J.P.; Soriano, D.C.; Luik, A.; Müller-Edenborn, B.; Jadidi, A.; et al. Machine learning enables noninvasive prediction of atrial fibrillation driver location and acute pulmonary vein ablation success using the 12-lead ECG. *Cardiovasc. Digit. Health J.* **2021**, *2*, 126–136. [[CrossRef](#)] [[PubMed](#)]
95. Kwon, J.-M.; Jeon, K.-H.; Kim, H.M.; Kim, M.J.; Lim, S.; Kim, K.-H.; Song, P.S.; Park, J.; Choi, R.K.; Oh, B.-H. Deep-learning-based risk stratification for mortality of patients with acute myocardial infarction. *PLoS ONE* **2019**, *14*, e0224502. [[CrossRef](#)] [[PubMed](#)]
96. Merchant, F.M.; Sayadi, O.; Moazzami, K.; Puppala, D.; Armoundas, A.A. T-wave Alternans as an Arrhythmic Risk Stratifier: State of the Art. *Curr. Cardiol. Rep.* **2013**, *15*, 398. [[CrossRef](#)]
97. Xue, J.; Aufderheide, T.; Wright, R.S.; Klein, J.; Farrell, R.; Rowlandson, I.; Young, B. Added value of new acute coronary syndrome computer algorithm for interpretation of prehospital electrocardiograms. *J. Electrocardiol.* **2004**, *37*, 233–239. [[CrossRef](#)] [[PubMed](#)]
98. Goh, G.S.W.; Lopuschkin, S.; Weber, L.; Samek, W.; Binder, A. Understanding Integrated Gradients with SmoothTaylor for Deep Neural Network Attribution. *arXiv* **2020**, arXiv:2004.10484.
99. Hoi, S.C.H.; Sahoo, D.; Lu, J.; Zhao, P. Online Learning: A Comprehensive Survey. *Neurocomputing* **2021**, *459*, 249–289. [[CrossRef](#)]
100. PhysioNet Index. Available online: <https://physionet.org/challenge/> (accessed on 17 July 2021).