4-1-2019

Towards a Pre-Processing Algorithm for Automated Arrhythmia Detection

Sarah Miller
University of Dayton

Towards a Pre-Processing Algorithm for Automated Arrhythmia Detection

Honors Thesis

Sarah Victoria Miller

Department: Electrical and Computer Engineering

Advisor: Timothy Reissman, Ph.D.

May 2019

Towards a Pre-Processing Algorithm for Automated Arrhythmia Detection

Honors Thesis

Sarah Victoria Miller

Department: Electrical and Computer Engineering

Advisor: Timothy Reissman, Ph.D.

May 2019

Abstract

There are a variety of different wearable fitness/cardiac monitoring devices that are currently used in many people's day to day life. The primary cardiac function of these devices is to monitor heart rate, however we believe that they could be utilized to detect different forms of arrhythmia. In order to categorize and identify different forms of arrhythmia, we are utilizing published EKG data sets from existing databases as a basis for machine learning. The challenge that comes from the existing data sets is that the format they present the data in does not lend itself to machine learning, which requires data to be in a vector. This makes the process of converting the existing data sets into workable vectors long and tedious. Therefore, we are working to develop an algorithm that will be able to vectorize the data from multiple different data sets so we, and anyone who wishes to use machine learning on these signals, are able to quickly and accurately use now workable, prior data sets.

Table of Contents

Background

"An arrhythmia is a problem with the rate or rhythm of the heartbeat" that can cause the heart to beat too fast, too slow, or with an irregular rhythm [9]. There are several different causes of arrhythmia, including damage from disease, injury, and genetics, that cause changes in heart tissue and activity [9]. Figure 1 shows an example of several different arrhythmia.

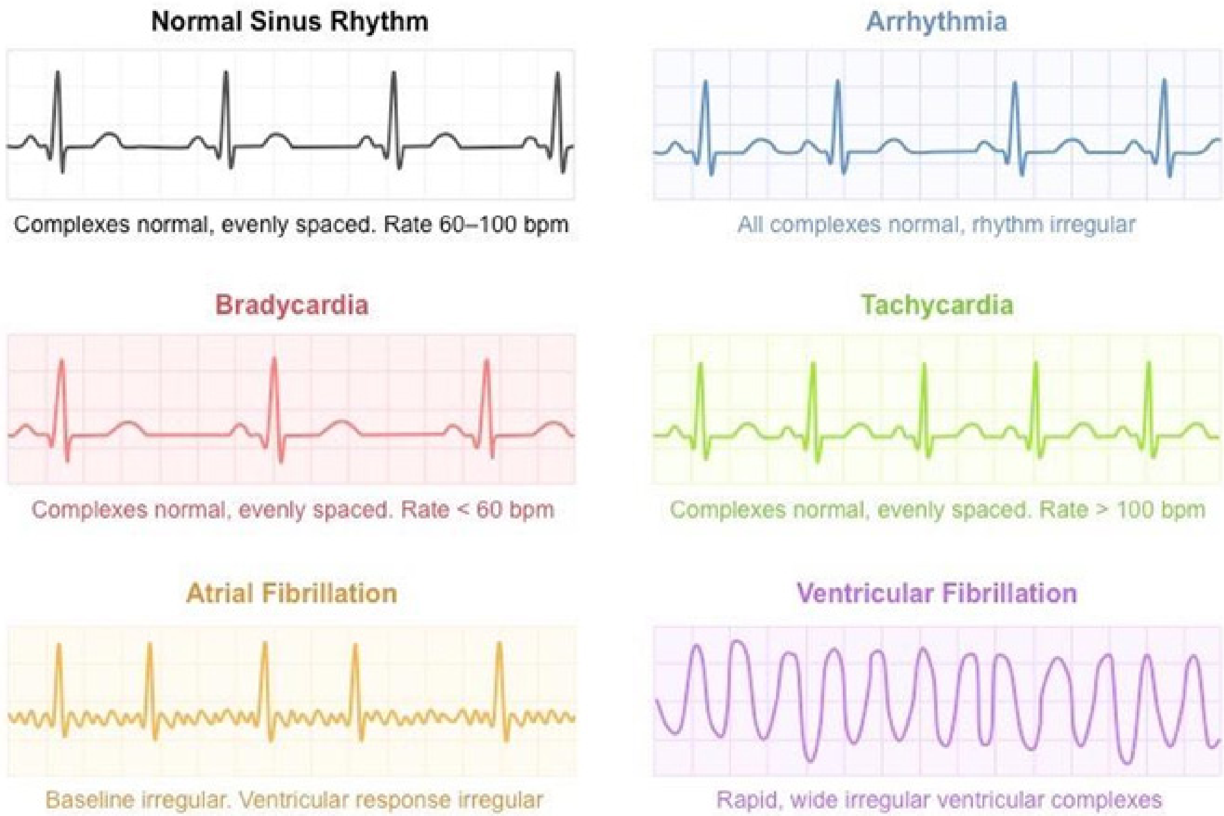

Figure 1: Different Forms of Arrhythmia [11]

The normal sinus rhythm shown in Figure 1 represents what a healthy heart beat would look like on an electrocardiogram (EKG or ECG). The complexes (peaks) are evenly spaced and the resting heart rate represented is between 60-100 beats per minute. Bradycardia, shown below the normal sinus rhythm, is an arrhythmia where the heart beats too slowly (typically a resting heart rate below 60 beats per minute) but maintains normal, evenly spaced complexes. Opposite of bradycardia is tachycardia, a condition in which the heart beats too fast (typically a resting heart rate above 100 beats per minute) but maintains normal, even complexes. Below bradycardia in Figure 1 is atrial fibrillation, the most common sustained arrhythmia that can lead to heart failure, hypertension, and valvular and ischemic heart disease [2]. Opposite of atrial fibrillation is an example of ventricular fibrillation. Ventricular fibrillation is a very serious abnormal rhythm that is responsible for around "75-85% of sudden deaths in persons with heart problems" [3]. There are several other arrhythmia that occur and are categorized by an abnormal heart rate, complex position, and/or ventricular response.

These various forms of heart arrhythmia affect millions of people around the world, but despite their high prevalence, their diagnosis has proven to be challenging [1]. This is due to the fact that many different forms of arrhythmia are not felt by those afflicted as they can be intermittent, short-lasting, and/or asymptomatic [1]. Detection of arrhythmia, however, can be lifesaving. The early detection of arrhythmia such as ventricular tachycardia and ventricular fibrillation can be critical in preventing sudden cardiac death [1]. Sudden cardiac death is a natural death that is marked by an abrupt loss of consciousness approximately one hour after the onset of acute symptoms [1]. It is typically "considered as the final result of a ventricular arrhythmia" and "accounts for approximately 300,000 deaths in the United States per year" [6,3]. Detection of ventricular arrhythmia and other more common arrhythmia, such as atrial fibrillation which affects an estimated 2.7 - 6.1 million people in the United States, is crucial in prolonging the life and wellbeing of those afflicted [10]. "The most common test used to diagnose an arrhythmia is an electrocardiogram" [9]. There are currently three different categories of devices that are utilized to detect different forms of arrhythmia: non-looping devices, external looping devices, and implantable looping devices.

Non-looping devices are those which are intermittently applied and do not continuously gather data [1]. They are typically "cordless devices that are either handheld devices or worn on the wrist" and are activated by the patient when their symptoms occur [1]. These are useful for the "investigation of sustained symptoms that are long enough to record the heart rhythm by applying the recorder" [1]. Non-looping devices that utilize smartphone technology and electrodes to record and transmit a rhythm strip, such as the Apple KardiaBand and FitBit, have gained popularity with those with known heart arrhythmias and those who are concerned they may have them [1]. These, and other non-looping devices have several benefits. The devices are easily accessible, relatively affordable, and avoid the skin irritation that is "associated with the electrodes required for the looping event recorders" [1]. However, non-looping devices are unable to capture the onset of events, which can be valuable information, because they are only applied after the development of symptoms [1]. This is where looping devices are helpful. Figure 2 shows an example of a non-looping device.

Figure 2: Non-Looping Device [6]

External looping devices are those which "are continuously worn and only removed for bathing and showering" [1]. These devices are typically used for patients who have short, frequent symptoms that occur at least once a week [1]. One of the most common forms of an external looping device is the Holter Monitor. The Holter monitor is a portable recording device connected to either 2,3, or 12 ECG-channels that provides continuous, real time monitoring for 24-48 hours [1]. Figure 3 shows an example of a 12 channel Holter Monitor.

Figure 3: 12-Channel Holter Monitor [7]

As seen in Figure 3, a Holter Monitor consists of electrodes that are placed on the chest wall [1]. The Holter Monitor can be very beneficial in several ways. It has a "widespread availability, simple application, complete data capture (not just events) and the independence of patient-activated event recording" [1]. The downsides of the Holter Monitor and other external looping devices, however, are low diagnostic yield in patients with infrequent symptoms due to the short monitoring time frame, the inconvenience of wearing and transporting the system, and the skin irritation that can be caused by the electrodes [1]. Other, less common, external looping devices are initiated by the patient or automatically triggered based on pre-programmed criteria [1]. Patient activated devices are limited by the compliance of the patients and their usefulness can be hindered by improper operation [5].

When patients have very infrequent symptoms, less than once a month, or it is determined that more than 48 hours of continuous monitoring is necessary, an implantable looping device is used [1]. "Implantable cardiac monitors are small leadless, long-term rhythm monitoring devices that are implanted under the skin of the left parasternal/left precordial chest wall" [1]. Figure 4 shows an example of an implantable looping device.

Figure 4: Implantable Looping Device [8]

While implantable looping devices are the 'gold standard' of event recorders, they come with several disadvantages. Implantable monitors carry the risk of pocket infections and have a tendency to 'under-sense' and 'over-sense' events leading to difficulty distinguishing between similar arrhythmia such as ventricular tachycardia and supraventricular tachycardia [5].

Each of the event recorders described gathers data that is then analyzed by a cardiologist in order to determine what, if any, arrhythmia a patient has. We believe that we can expedite this process by coming up with a pre-processing algorithm that turns the raw data gathered by these devices into a vector that can be utilized with machine learning. This would allow neural networks to be created that would be able to classify different arrhythmia without the need of a cardiologist. In order for this to work, however, the data must be in a form that is acceptable for machine learning.

Machine learning is a way for a computer to "learn without being programmed" [12]. In order for machine learning to be successful, two different datasets are needed, a training set and a test set [12]. The training dataset is the largest used, typically consisting of hundreds of thousands of categorized samples [13]. This dataset allows the neural network to determine how to weight different features and allows for the network to distinguish between different cases and arrhythmia [13]. The test dataset is much smaller than the training set and is utilized after the training dataset [13]. This dataset is used to see if the network is able to accurately distinguish and identify the data presented based off of the patterns formed through the training dataset [13].

We believe that if we are able to transform ECG waveforms into vectors utilizing the existing data from open share sources such as Physionet that we will be able to create a comprehensive training set that will allow for easier detection of arrhythmia. It is imperative that the data be in a standardized vector format in order for linear algebra operations to be applied and deep learning to be successful.

Manual Method

Typical datasets provide the timestamp and corresponding voltage at varying points along the EKG. For machine learning to be successful, we first standardize voltage recordings at 100 Hz. This in of itself is challenging as different datasets provide different variations in their timestamps given, ranging from random intervals to set intervals of 300 Hz to 1kHz. To retain dataset accuracy, we filter as close to 0.01 second accuracy as possible. From there, we take the voltage values that are associated with each of these standardized times and map them into a concise vector form. This form includes those voltages and a set spacing between each value to create an array-like unit. It is important that datasets be mapped in this way because machine learning algorithms require the imported dataset format to allow for linear algebra operations.

The process of manually converting a dataset into a workable vector form is long and tedious, taking approximately six hours to convert each ten second EKG into a usable format. The process begins with the standardization of voltage recordings at 100 Hz. To do this, the given values are sorted through to find timestamps that most consistently have a 0.01 second time difference between the them. If a given waveform does not have timestamps perfectly spaced at a 100 Hz rate, linear extrapolation is used to infer the value that would appear at the desired timestamp. This process is repeated until all 1,000 standardized data points are determined. From here, the standardized data is typed out into the proper matrix format. An overview of the manual process can be seen in Figure 5a-5c below.

Time (hh:mm:ss.mmm	Date dd/mm/yyyy)	irect_1 (uV)
[00:00:00.000	01/01/2011]	28.800
[00:00:00.001	01/01/2011]	26.700
[00:00:00.002	01/01/2011]	24.800
[00:00:00.003	01/01/2011]	23.800
[00:00:00.004	01/01/2011]	23.600
[00:00:00.005	01/01/2011]	23.700
[00:00:00.006	01/01/2011]	23.700
[00:00:00.007	01/01/2011]	23.700
[00:00:00.008	01/01/2011]	23.800
[00:00:00.009	01/01/2011]	24.400
[00:00:00.010	01/01/2011]	25.400
[00:00:00.011	01/01/2011]	26.500
[00:00:00.012	01/01/2011]	27.400
[00:00:00.013	01/01/2011]	27.900
[00:00:00.014	01/01/2011]	28.300
[00:00:00.015	01/01/2011]	28.600
[00:00:00.016	01/01/2011]	28.900
[00:00:00.017	01/01/2011]	28.900
[00:00:00.018	01/01/2011]	28.500
[00:00:00.019	01/01/2011]	27.800
[00:00:00.020	01/01/2011]	27.600
[00:00:00.021	01/01/2011]	28.400
[00:00:00.022	01/01/2011]	30.400
[00:00:00.023	01/01/2011]	32.901
[00:00:00.024	01/01/2011]	34.701
[00:00:00.025	01/01/2011]	35.001
[00:00:00.026	01/01/2011]	33.901
[00:00:00.027	01/01/2011]	32.200
[00:00:00.028	01/01/2011]	31.500
[00:00:00.029	01/01/2011]	32.801
[00:00:00.030	01/01/2011]	35.901

Figure 5a: Example Partial Online ECG Data Set

Time (hh:mm:ss.mmm	Date dd/mm/yyyy)	irect_1 (uV)
[00:00:00.000	01/01/2011]	28.800
[00:00:00.010	01/01/2011]	25.400
[00:00:00.020	01/01/2011]	27.600
[00:00:00.030	01/01/2011]	35.901
[00:00:00.040	01/01/2011]	46.001
[00:00:00.050	01/01/2011]	38.501
[00:00:00.060	01/01/2011]	37.101
[00:00:00.070	01/01/2011]	30.100
[00:00:00.080	01/01/2011]	32.300
[00:00:00.090	01/01/2011]	33.501
[00:00:00.100	01/01/2011]	36.401
[00:00:00.110	01/01/2011]	33.101
[00:00:00.120	01/01/2011]	39.301
[00:00:00.130	01/01/2011]	31.200
[00:00:00.140	01/01/2011]	40.101
[00:00:00.150	01/01/2011]	41.701
[00:00:00.160	01/01/2011]	41.601
[00:00:00.170	01/01/2011]	46.601
[00:00:00.180	01/01/2011]	116.902
[00:00:00.190	01/01/2011]	43.501
[00:00:00.200	01/01/2011]	1.500

Figure 5b: Standardization

```
[28.800   25.400   27.600   35.901   46.001   38.501
 37.101   30.100   32.300   33.501   36.401   33.101
 39.301   31.200   40.101   41.701   41.601   46.601
 116.902   43.501   1.500   34.201   39.301   37.901
 38.601   39.701   41.201   44.601   42.401   ...     ]
```

Figure 5c: Partial Finalized Vector

Our manual method of pre-processing the online EKG datasets is able to create a vector that can be imported into machine learning algorithms. However, due to the large value of data sets required in order to successfully train and test a neural network, the manual method is not a feasible way to gather data sets. This led to us creating an automated process that is able to successfully carry out the same standardization and vectorization in under a second.

Automated Method

To create an automated version of the pre-processing algorithm, first a base code is created utilizing Java and the Abdominal and Direct Fetal ECG Database [14,15]. The base code consists of two elements: the main class and the ReadFile class. The ReadFile class is common to all datasets, while the main class is individualized based on which dataset is being read in. Figure 6 shows the ReadFile class that is utilized with the Abdominal and Direct Fetal ECG Database pre-processing algorithm.

AbdominalAndDirectFetalECGDatabase.java × ReadFile.java ×

Source History

```java
/*
 * This file is used to read in the information
 * from the datasets in order for them to be processed
 * by the main code
 */
package abdominal.and.direct.fetal.ecg.database;

import java.io.IOException;
import java.io.FileReader;
import java.io.BufferedReader;

public class ReadFile {

    private String path;

    public ReadFile (String file_path){
        path = file_path; // create a path for the file to be read into
    }

    public String[] OpenFile() throws IOException{
        FileReader fr = new FileReader(path); //create a new file reader
        BufferedReader textReader = new BufferedReader(fr);// create a new buffer reader

        int numberOfLines = readLines(); // determine number of lines of text in the data set
        String[] textData = new String[numberOfLines]; // create an array of strings

        int i; // create a variable to parse through data set

        for (i=0; i<numberOfLines;i++){ // as long as there are still lines in the text, continue
            textData[i] = textReader.readLine();// read in the data from each line in the text
        }// end for

        textReader.close();// close the text reader
        return textData;// provide the data from the text
    }// end OpenFile

    int readLines() throws IOException{
        FileReader file_to_read = new FileReader(path); // create a new file reader
        BufferedReader bf = new BufferedReader(file_to_read);// create a new buffer reader

        String aLine; // create a variable to hold a single line of data
        int numberOfLines = 0; // start the number of lines at 0

        while((aLine = bf.readLine()) != null){// while there are still lines of data
            numberOfLines ++;// increase the count of the number of lines
        }// end while
        bf.close();// close the buffer reader

        return numberOfLines;// return the number of lines of data
    }// end readLines
}
```

Figure 6: ReadFile for Abdominal and Direct Fetal ECG Database

The purpose of the ReadFile class is to allow the main code to read in and classify the data that is provided by physionet [15]. To do this, the user selects a dataset, record, and signal from the input section of the PhysioBank ATM, then selects ‘show samples as text’ from the toolbox section of the PhysioBank ATM. This provides the timestamp and corresponding voltage value for the selected ECG. The user then copies and saves the provided values into a document, which is read into the pre-processing algorithm through the ReadFile class. The ReadFile class analyzes the data line-by-line and stores it in a format that is accessible by the main class. Figure 7 shows the main class for the Abdominal and Direct Fetal ECG Database.

```java
/*
 * This is the code that is used to create a vector
 * from the Abdominal and Direct Fetal ECG
 * datasets standardized at 100 Hz
 */
package abdominal.and.direct.fetal.ecg.database;

import java.io.IOException;

/**
 *
 * @author staff
 */
public class AbdominalAndDirectFetalECGDatabase {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) throws IOException {
        //throws IOException means that the code will alert the user if the file is not found
        //put the location of whatever you have your data saved as in the quotes below
        String file_name = "C:\\Users\\Sarah\\Documents\\NetBeansProjects\\AbdominalAndDirectFetalECGDatabase/adfecgdb_0_10.txt";

        try{
            ReadFile file = new ReadFile(file_name); //Read in the data set
            String[] aryLines = file.OpenFile(); // Transform the data set into an array of strings

           String f = "["; //Begin standardized vector


            int i; // create a variable to parse through lines in the data set
            for(i=2; i<aryLines.length; i= i+10){ // as long as there is data in the data set, continue standardization
                //System.out.println(aryLines[i]);
                // Uncomment the above line if you would like to ensure that the data is standardized to 100 Hz
                String st1 = aryLines[i]; // grab a single line of data
                String[] str1 = st1.split("\t", 0); // break the line of data into the timestamp and the voltage value

                String volt = str1[1].toString(); // grab just the voltage value
                f += volt + "  "; // add the voltage value and a set amount of spaces to the standardized vector

                i = i+10; // proceed to next relevant time entry
                //System.out.println(aryLines[i]);
                // Uncomment the above line if you would like to ensure that the data is standardized to 100 Hz
                st1 = aryLines[i];// grab the relevant line of data
                str1 = st1.split("\t", 0);// break the line of data into the timestamp and voltage value

                volt = str1[1].toString();// grab just the voltage value
                f += volt + "  "; // add the voltage value and a set amount of spaces to the standardized vector

            }// end for
            f += "]";// close off standardized vector
            System.out.print(f);// display final standardized vector

        }// end try

        catch(IOException e){
            System.out.println(e.getMessage()); // if the file is not found, alert the user
        }// end catch
    }
}
```

Figure 7: Main Class for Abdominal and Direct Fetal ECG Database

The main class begins by creating a file name based on where the user has the aforementioned document saved. This information is then utilized by the ReadFile class to ensure that the main class is able to work with the timestamp and voltage value data provided. From here, the main code standardizes the given timestamp and voltage value data to 100 Hz. In the case of the base code, the data is provided at a 1000 Hz so all this entails is creating a new vector containing the voltage values from every 10th sample. The vector is then printed out for the user in a format that is compatible with machine learning. It is found that the automated version of the pre-processing algorithm is able to produce a vector that is a perfect match for the manual method in under a second. Figure 8 shows an example of a partial vector produced using data from the Abdominal and Direct Fetal ECG Database along with the runtime for the total vector creation.

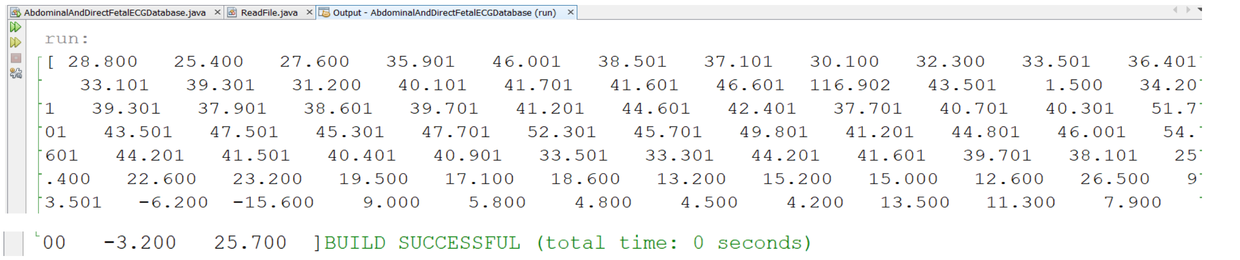

Figure 8: Partial Finalized Vector and Runtime

From Figure 8 it can be seen that the creation of the standardized vector took less than a second to create.

A unique version of the pre-processing algorithm is created for each of the existing datasets utilizing this base code. To do this, first the pattern in the timestamps of a given database is analyzed manually. Once the pattern is identified, the for loop of the base main code is altered to create a vector that correctly standardizes the data to 100 Hz. It is found that the automated pre-processing algorithm is able to quickly and accurately create workable vectors for each of the databases it is designed for.

Future Recommendations

We believe that with the successful creation of the pre-processing algorithm, the data from Physionet will be able to be utilized to create a successful training dataset for machine learning. Following the creation of the training set, it is recommended that data from varying monitors - Fitbit, Apple Kardiaband, chest straps, holter monitors, etc. - be analyzed within the neural network as the test dataset. The results of the test data should be compared with cardiologist diagnosis to ensure that the neural network is functioning as intended.

We believe that our pre-processing algorithm will be able to help others interested in the field of automated arrhythmia detection. There are several different conferences, such as Computing in Cardiology, that are relevant to the work that we have accomplished and show the widespread interest in the subject. Due to this interest, we recommend that our pre-processing algorithm(s) be shared openly with the public.

References

1. P. Kowey et al. (eds.), *Cardiac Arrhythmias, Pacing and Sudden Death*, Cardiovascular Medicine, DOI 10.1007/978-3-319-58000-5_5
2. P. Kowey et al. (eds.), *Cardiac Arrhythmias, Pacing and Sudden Death*, Cardiovascular Medicine, DOI 10.1007/978-3-319-58000-5_2
3. H. Saleh et al., *Self-powered SoC Platform for Analysis and Prediction of Cardiac Arrhythmias*, Analog Circuits and Signal Processing, DOI 10.1007/978-3-319-63973-4_1
4. P. Kr¨omer et al. (eds.), *Proceedings of the Fifth Intern. Conf. on Innov. in 375 Bio-Inspired Comput. and Appl. IBICA 2014*, Advances in Intelligent Systems and Computing 303, DOI: 10.1007/978-3-319-08156-4_37
5. Tanno, Kaoru. "Use of implantable and external loop recorders in syncope with unknown causes." Journal of Arrhythmia, vol. 33, no. 6, 2017, pp. 579-582. OhioLINK Electronic Journal Center, doi:10.1016/J.JOA.2017.03.006.
6. Guest. "12-Lead Digital Holter - Spacelabs Healthcare." *1pdf.Net*, 1PDF.NET, 1 Jan. 1970, 1pdf.net/12-lead-digital-holter-spacelabs-healthcare_586225efe12e89b626f23216.
7. "Reveal LINQ Insertable Cardiac Monitoring System." *EvergreenHealth*, www.evergreenhealth.com/reveal-linq-insertable-cardiac-monitoring-system.
8. "Arrhythmia." *National Heart Lung and Blood Institute*, U.S. Department of Health and Human Services, www.nhlbi.nih.gov/health-topics/arrhythmia.
9. "Atrial Fibrillation Fact Sheet|Data & Statistics|DHDSP|CDC." *Centers for Disease Control and Prevention*, Centers for Disease Control and Prevention, www.cdc.gov/dhdsp/data_statistics/fact_sheets/fs_atrial_fibrillation.htm.
10. "Brent Cornell." *BioNinja*, ib.bioninja.com.au/standard-level/topic-6-human-physiology/62-the-blood-system/electrocardiography.html.
11. "Machine Learning." *GeeksforGeeks*, www.geeksforgeeks.org/machine-learning/.
12. *How Neural Networks Are Trained*, ml4a.github.io/ml4a/how_neural_networks_are_trained/.
13. Jezewski J, Matonia A, Kupka T, Roj D, Czabanski R. >Determination of the fetal heart rate from abdominal signals: evaluation of beat-to-beat accuracy in relation to the direct fetal electrocardiogram. *Biomedical Engineering/Biomedizinische Technik 2012* Jul;57(5):383-394. doi:10.1515/bmt-2011-0130.
14. Goldberger AL, Amaral LAN, Glass L, Hausdorff JM, Ivanov PCh, Mark RG, Mietus JE, Moody GB, Peng C-K, Stanley HE. PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals. *Circulation*101(23):e215-e220 [Circulation Electronic Pages;http://circ.ahajournals.org/content/101/23/e215.full]; 2000 (June 13).