A Wearable Smartphone-Based Platform for Real-Time Cardiovascular Disease Detection Via Electrocardiogram Processing

Joseph J. Oresko, *Student Member, IEEE*, Zhanpeng Jin, *Student Member, IEEE*, Jun Cheng, Shimeng Huang, Yuwen Sun, Heather Duschl, and Allen C. Cheng, *Member, IEEE*

***Abstract*—Cardiovascular disease (CVD) is the single leading cause of global mortality and is projected to remain so. Cardiac arrhythmia is a very common type of CVD and may indicate an increased risk of stroke or sudden cardiac death. The ECG is the most widely adopted clinical tool to diagnose and assess the risk of arrhythmia. ECGs measure and display the electrical activity of the heart from the body surface. During patients' hospital visits, however, arrhythmias may not be detected on standard resting ECG machines, since the condition may not be present at that moment in time. While Holter-based portable monitoring solutions offer 24–48 h ECG recording, they lack the capability of providing any real-time feedback for the thousands of heart beats they record, which must be tediously analyzed offline. In this paper, we seek to unite the portability of Holter monitors and the real-time processing capability of state-of-the-art resting ECG machines to provide an assistive diagnosis solution using smartphones. Specifically, we developed two smartphone-based wearable CVD-detection platforms capable of performing real-time ECG acquisition and display, feature extraction, and beat classification. Furthermore, the same statistical summaries available on resting ECG machines are provided.**

***Index Terms*—Arrhythmia detection, cardiovascular disease (CVD) detection, ECG processing, machine learning, smartphone, windows mobile.**

I. Introduction

CARDIOVASCULAR disease (CVD) is the single leading cause of death in both developed and developing countries, and encompasses a variety of cardiac conditions, including heart attack and hypertension. According to the American Heart Association, in the United States alone, 80 000 000 people are estimated to have one or more forms of CVD and nearly 2400 Americans die of CVD each day [1]. Cardiac arrhythmia, defined as abnormal heart rhythms, is a very common type of CVD and is thought to be responsible for most of the sudden cardiac deaths that occur every year.

The most common test for a cardiac arrhythmia is an ECG. The ECG measures the electrical impulses of the heart via electrodes on the skin's surface. However, it is difficult to diagnose many arrhythmias with a standard resting ECG, because it can only provide a snapshot of the patient's cardiovascular activity in time. An intermittent arrhythmia can go unnoticed, and physicians must rely on self-monitoring and symptoms reported by patients to support their final diagnosis. In some cases, ambulatory recording of ECG data, collected over extended periods of time, may be taken in an attempt to acquire data during an occurrence of an intermittent arrhythmia. However, existing solutions for this type of recording are limited. Although they can lead to a diagnosis and therapy that may greatly improve the quality of life for the patient, they can be inconvenient for both the patient and the physician.

Three types of ECG solutions are possible, which are as follows: 1) those that can store information to be diagnosed offline after data collection is complete; 2) those that use remote connections to provide real-time diagnosis via a separate server; and 3) those that perform real-time diagnosis within the device itself. Among the first type of systems, Holter monitors and event recorders stand out, such as GE's SEER (GE Healthcare, Waukesha, WI), Philips's DigiTrack (Philips Healthcare, Andover, MA), and Midmark's IQmark (Midmark Corporation, Versailles, OH), among others. These devices only provide recording and monitoring capabilities and no real-time classification of ECGs because the classification is performed offline. The second type utilizes telemedical functionalities via a remote real-time monitoring system [2]–[4]. Most of them make use of mobile phones/personal digital assistants (PDAs) to collect the ECG data and send them to a monitoring center, where the analysis and classification are performed, thus depriving the user of real-time feedback. For the third type of systems, researchers have proposed some intermediate level of local real-time classification, such as the classification of heart beats, by using up-to-date smartphones or PDAs [5]–[8], but these do not provide a complete CVD diagnosis solution. The continued development of powerful microprocessors allows researchers to develop applications for these handheld devices that deliver comparable performance to that of a desktop computer only a few years ago.

Manuscript received July 21, 2009; revised October 22, 2009 and February 3, 2010. First published April 12, 2010; current version published June 3, 2010. This research was supported by Microsoft Research (Cell Phone as a Platform for Healthcare Award) and the National Science Foundation grant (NSF#0832990) awarded to Allen C. Cheng.

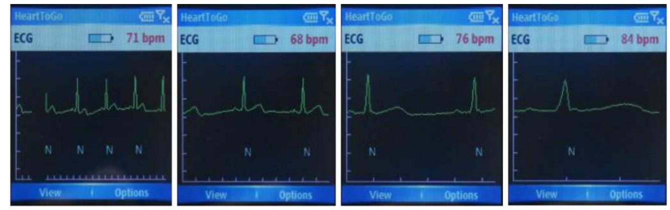
J. J. Oresko, Z. Jin, J. Cheng, S. Huang, Y. Sun, and H. Duschl are with the Department of Electrical and Computer Engineering, University of Pittsburgh, Pittsburgh, PA 15261 USA (e-mail: jjo5@pitt.edu; zhj6@pitt.edu; juc25@pitt.edu; shh61@pitt.edu; yus25@pitt.edu; hld11@pitt.edu).

A. C. Cheng is with the Departments of Electrical and Computer Engineering, Computer Science, Bioengineering, and Neurological Surgery, University of Pittsburgh, Pittsburgh, PA 15261 USA (e-mail: acc33@pitt.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITB.2010.2047865

Fig. 1. *HeartToGo* experimental prototype consisting of an Amoi Windows Mobile 5 Smartphone and a single-channel Alive ECG sensor.



(a) (b) (c) (d)

Fig. 2. Real-time ECG display at different magnifications: (a) 1×, (b) 2×, (c) 4×, and (d) 8×.

With people becoming more active in monitoring their own health via assistive diagnosis platforms, there exists a need to implement a real-time, user friendly CVD monitoring system. We seek to provide the user with an enriched interface with which they can monitor their ECG in real time. The contribution of this research lies on in-depth analysis of ECG signals and the development of a portable smartphone-based CVD monitoring and assistive diagnosis platform. To this end, we implemented two smartphone-based wearable CVD-detection platforms: a machine-learning and rapid prototyping platform and a plug-in-based GUI platform. Each performs real-time ECG acquisition and display, feature extraction, and beat classification. Furthermore, the same statistical summaries available on resting ECG machines are provided, which include: RR, P, and QRS durations; PR, QT, and QTc intervals; and average, high, and low heart rates.

II. System Design Framework

The system design framework for the machine-learning and plug-in-based GUI platform is presented in this section. Incorporating machine learning into the platform is an effective way to introduce self-adaptable ECG processing and CVD detection to account for the user's unique physiological characteristics, while a plug-in-based platform allows for new disease-detection rules to be integrated without changing the main program. The prototype consists of a Windows Mobile Smartphone and a single-channel Alive ECG sensor, as shown in Fig. 1.

A. Plug-In-Based GUI Platform

1) Data Acquisition: To acquire real-time ECG signals, we employed Alive Technology's (Alive Technologies Pty. Ltd., Robina, Qld., Australia) state-of-the-art wireless ECG heart monitor, which is a lightweight (60 g with battery), low-power (60 h of operation with continuous wireless transmission) wearable single-channel ECG-sensing device capable of recording 300 8-bit samples per second. It is equipped with a class-1 Bluetooth transmitter, which can send its data to smartphones or other wireless devices. Also, the monitor is equipped with a three-axis accelerometer. Furthermore, the ECG signal from the monitoring session can be recorded to a secure digital (SD) card that plugs into the sensor, allowing for optional offline analysis by a physician similar to that of a Holter monitor. Note that the recording length varies with the size of the SD card used (e.g., a 1 GB card could store 40 days worth of continuous ECG data).

HeartToGo, the multithreaded C# application we developed for the real-time ECG display, processing, and cardiac summary reports, runs on Windows Mobile Smartphones. The Alive heart monitor communicates with HeartToGo using a Bluetooth serial port profile (SPP) connection. HeartToGo uses a dedicated thread to process the incoming Bluetooth data stream, which is made up of variable length packets containing both ECG data samples and acceleration data samples. Once the input data is read, parsed, and verified, thread delegates manage the sharing of the new ECG samples between the display and feature-extraction threads in order to avoid cross-threading errors.

2) Real-Time ECG Display: As real-time ECG data arrives at the phone, the Bluetooth communication thread passes the data to the display thread for plotting on the screen. Different threads are used for display and data acquisition to obtain a responsive GUI and increase thread-level parallelism.

The ECG signal is plotted on the fly on the phone as data arrives at a sampling rate of 300 Hz, as shown in Fig. 2. Four different levels of magnification (1×, 2×, 4×, and 8×) were implemented to allow for a close-up examination of the ECG signal and are shown in Fig. 2. The axis for the ECG plot conforms to the clinical standard of a resting ECG machine: the scale for the vertical voltage axis is 0.5 mV per tick, and the scale of the horizontal time axis is 200 ms per tick. Moreover, each beat is classified and an annotation is shown below each QRS complex; this is shown in Fig. 2 with the "N" marking to signify a normal beat. If the beat had been a premature ventricular contraction (PVC) beat, then, "V" would be displayed below the beat. Also, besides the real-time ECG signal, the average heart rate and battery life level of the heart monitor are displayed. Furthermore, if desired, the user can also switch between plotting the three-axis acceleration trace instead of the ECG signal.

In addition to plotting the real-time ECG signal, we focus specifically on recognizing PVCs, an arrhythmia that occurs when the ventricles of the heart contract early. PVC beats are the most common ventricular arrhythmia, and can originate from an existing cardiac condition, such as a heart arrest, valvular heart disease, or cardiomyopathy, or can be caused by a noncardiac stimulus, such as caffeine, alcohol and other drugs, electrolyte imbalances, or infection. PVCs are fairly common in the general population, even amongst healthy individuals; these beats are

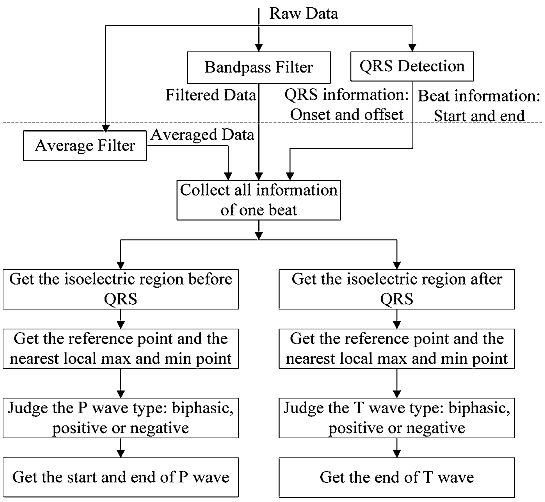


Fig. 3. ECG feature-extraction workflow.

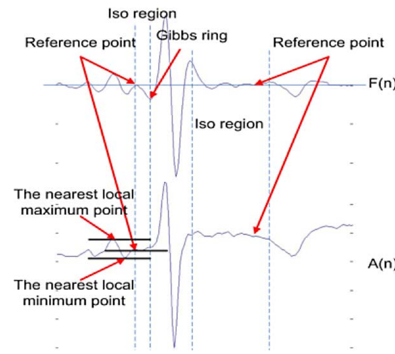


Fig. 4. Comparison of the BP FIR filter result and average filter result. $F(n)$ is the corresponding BP FIR filter result, and $A(n)$ is the corresponding average filter result.

found in approximately 60% of healthy people in ambulatory ECGs [9]. This prevalence makes PVC a very suitable candidate for the first-classified CVD of our functional prototype.

3) Feature Extraction and Classification: Feature extraction and classification are implemented on the smartphone as a separate dynamic-linked library (dll) from the main HeartToGo application and runs on its own thread. Due to the computationally intensive math required, it is written in C++ to improve execution speed, as opposed to C#, which was used for the GUI, since it has a streamlined implementation in Windows Mobile using the .NET Framework. Using this approach, different dll plug-ins can be created for different CVDs without changing the main program, which is responsible for data acquisition and display.

Fig. 3 shows the workflow for ECG feature extraction. To get all the features of the ECG, the first step is to detect the QRS. We utilize the algorithm proposed by Hamilton [10] to get the onset and offset of each QRS. The implementation details can be found in Section II-B when we describe the machine-learning framework. The original algorithm in [10] used a bandpass (BP) finite-impulse response (FIR) filter to remove noise, which is also mentioned in [11] and [12]. In our design, to identify the Gibbs rings in the filtered results, we add an averaging filter that calculates the average value of every six neighbor points. The averaging filter also helps us to judge the polarity of the P and T waves. Although we can exclude most of the obvious Gibbs rings, we may not find the main peaks of the P or T waves in the filtered result directly, if their amplitudes are too low. Therefore, we need to know the polarity of the P and T waves before identifying their locations. First, we have to determine the isoelectric region before and after the QRS in order to find the reference point for the P and T waves. Then, we look for the nearest local maximum and minimum points before the reference P wave, as well as the nearest local maximum and minimum points after the T wave. By adding a threshold, we can judge the polarity of the P and T waves. Finally, we can get the start and end of the P wave and the end of the T wave. What follows is the justification for choosing a BP FIR filter and the details of our design.

BP filters can reduce the baseline wander significantly, but a Gibbs ringing phenomena is introduced into the Q and S waves, which manifests as distortions with an amplitude larger than the P wave [13]. However, compared with wavelet filters, BP filters can conserve a lot of computational resources, which is especially important for real-time mobile phone systems. Therefore, we propose to identify the Gibbs rings in the filtered signal by sending the original signal to an averaging filter and comparing the filter result $A(n)$ with the BP filter result $F(n)$ to identify the Gibbs rings from the ECG signals.

Although averaging filters can remove high-frequency noise without creating Gibbs rings, the drift line still exists. We have to find a reference point in the signal and use the local difference of the amplitude of the signal waves to judge the polarity. To find the reference point for the P wave, we first find the isoelectric region in $A(n)$ before the QRS. In this region, the change of the signal amplitude stays in a small area. We choose one point in $F(n)$, which is equal to zero, or close to zero, in this region, as the reference point. Similarly, the isoelectric region is after the QRS, but it may not be a straight line because of the drift line. The change of the slope could stay in a small area. In this region, $F(n)$ should have several points equal to zero, and we choose the one, which is closest to the T wave, as the reference point. Then, we calculate the height difference between the reference point and both the nearest maximum and minimum points in $A(n)$ to judge the polarity of the P and T waves. In this example, the P wave is biphasic, and the T wave is negative. When we get the polarity of the P and T waves, we can easily find the onset and offset of the P wave and the end of the T wave in $F(n)$. Fig. 4 illustrates these reference points, which were used to identify the P and T waves.

4) Cardiac Summary Reports: The same statistical cardiac summary reports provided by standard resting ECG machines are implemented in HeartToGo. Both cardiac statistics and features are extracted from the ECG signal in real time. Fig. 5(a) shows the cardiac statistics, consisting of the average, high, and low heart rate, and the total number of beats, as well as the number of normal and PVC beats. Fig. 5(b) presents the cardiac feature report, which includes the RR duration, P duration, QRS

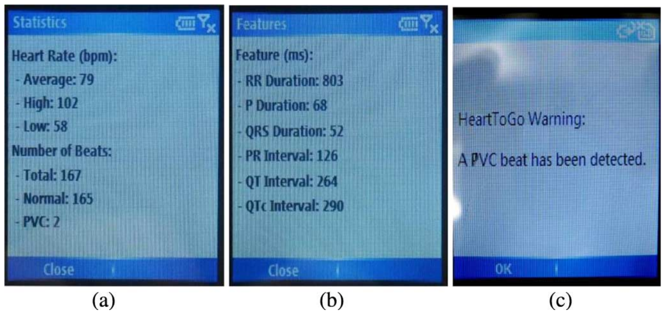


Fig. 5. Cardiac summary reports. (a) Statistics. (b) ECG features. (c) Pop-up alarm message for abnormal beat occurrences.

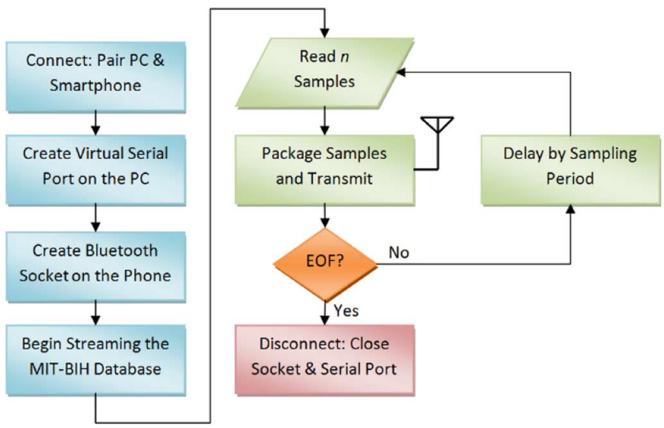


Fig. 6. Workflow for streaming ECG data via Bluetooth for heart monitor simulation.

duration, PR interval, QT interval, and QTc interval. Furthermore, an alarm is triggered when abnormal beats occur so that the user does not have to continuously monitor the cardiac summary reports. The alarm chosen is a pop-up message warning that an abnormal beat has occurred, as shown in Fig. 5(c), and an audible/vibratory notification on the phone to alert the user than an abnormal beat has occurred.

5) Streaming Database Verification: In order to verify the proper functionality of the cardiac summary reports and beat classification algorithm, a virtual heart monitor emulator was created with a PC. Instead of transmitting the user's ECG data from the Alive heart monitor, the PC transmitted the signal from an ECG database using a Bluetooth dongle paired to the smartphone. Using MATLAB, a virtual communication port was established and ECG data was read from a database file (e.g., the MIT-BIH Arrhythmia Database from PhysioNet [14]), packaged, and then, transmitted to the phone. Furthermore, a pause was added; therefore, the packages would arrive at the same sampling rate as the Alive heart monitor. This workflow is shown in Fig. 6.

B. Machine-Learning-Based Platform

1) Data Acquisition: As previously described, we again employed Alive Technology's wireless ECG heart monitor to acquire real-time ECG signals. The connection between Alive's monitor and smartphone was established via Bluetooth and the data were acquired in our LabVIEW implementation via an SPP.

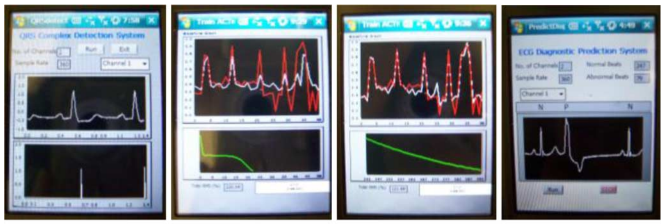
In order to verify the proper functionality and accuracy of the proposed system, we also emulated the real-time data of heart activities and reported the results based on the widely used MIT-BIH Arrhythmia Database. The MIT-BIH database contains 48 30-min ambulatory ECG recordings. The original data files were directly stored in the smartphone and a specific signal exporter was developed to convert the original "format 212" of MIT-BIH database to a binary format readable by LabVIEW.

2) ECG Feature Extraction: The classical Pan–Tompkins QRS-detection algorithm [11] and its errata were adopted and implemented in our ECG processing system because of its proven sensitivity of 99.69% and positive prediction of 99.77% when evaluated with the MIT-BIH arrhythmia database.

The QRS-detection algorithm consists of a BP filtering stage that uses a set of cascaded filters. The first two stages consist of a low-pass filter with a cutoff frequency at about 11 Hz and a gain of 36 mV/mV with a filtering processing delay of six samples, and a high-pass filter with a cutoff frequency of about 5 Hz with a unity gain and a processing delay of 16 samples. The derivative stage differentiates signals to obtain information about the slope of the QRSs. The squaring stage identifies the slope of the frequency response curve of the derivative and restricts false positives caused by T waves with higher spectral energy. The moving-window integration stage is used to obtain waveform feature information in addition to the information about the slope and the width of the QRS complex. Finally, the rising edge of the integration waveform corresponds to the desired QRS complex. A fiducial mark for the QRS complex feature can be represented by the temporal location of the peak of the R wave, which is determined according to the waveform excerpt located within the range of this rising edge. As detailed earlier, we first implemented this six-stage QRS-complex-detection framework on a PC test bench in LabVIEW, which successfully reduces the effects of irregular distance between peaks, irregular peak forms, and the presence of a low-frequency component in the ECG due to patient's breathing. Next, the PC implementation was converted to a smartphone implementation, using the LabVIEW Mobile Module, to allow this fast feature-extraction algorithm to run on a Windows Mobile Smartphone, as illustrated in Fig. 7(a), which shows the ECG signal plotted on the fly as well as the QRS complex that has been identified. As part of this work-in-progress, we have extended the diagnosis capability to include additional features, such as the RR interval, the QRS width, the R peak, and the beat width.

3) Machine-Learning-Based ECG Classification: Among the numerous machine learning paradigms, we focus on the feedforward multilayer perceptron (MLP) artificial neural network (ANN), which is one of the best-known techniques used in pattern recognition and classification, time-series modeling, nonlinear control, and system identification. In this study, we exploit potential applications of the MLP structure to perform an important ECG processing task: QRS beat pattern classification, on a state-of-the-art smartphone.

QRS beat classification is a crucial task in ECG diagnosis, and most existing methods rely highly on various discrete



(a) (b) (c) (d)

Fig. 7. (a) QRS-complex-detection system using the LabVIEW Mobile Module. (Top) ECG signal. (Bottom) Identified QRS complex. (b) Initial ANN training (top) red—the target results and white—the predicted results; (bottom) the rms, which was continuously decreased as the training progressed from (b) initial training to (c) gradual refinement. (d) ECG diagnostic predication system: beats are marked above their respective ECG waves.

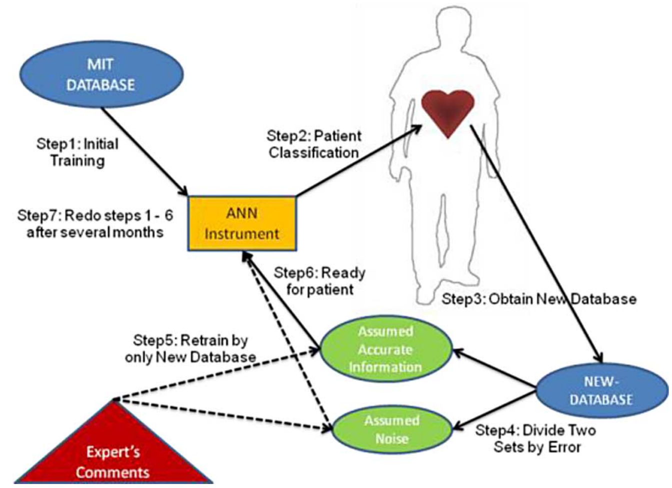


Fig. 8. Adaptive patient-specific ECG predication diagram.

signal features extracted or measured from the ECG waveform, such as the cardiac interbeat intervals. In this study, we use the morphologies of the heartbeats rather than their discrete features to detect cardiac abnormalities and identify potential arrhythmia. Given the QRS complex and fiducial information obtained from the feature-extraction stage, the input to the MLP ANN is the original QRS morphological beat pattern with an 11-bit resolution over a 10 mV range. Each input template contains 51 samples centered on the annotated fiducial mark in the recording, which approximately represents a time segment of 150 ms; each output is associated with a particular class of arrhythmia conditions. Fig. 7(b) and (c) presents two different training phases of our ANN implementation on the smartphone, where the red and white waveforms in the upper window represent the target values and the training values, respectively, and the green line in the lower window depicts the gradually decreasing training error (the exact value is shown in the text box at the bottom of the GUI) between targets and trained values. It is worth noting that we dynamically rescale the error axis in order to enlarge and show the training performance, since the training errors will become small and negligible as the training progresses compared to the errors in the initial training epochs. It is shown in Fig. 7(b) that, during the first 100 training epochs, the training error plunges significantly. After that, the network is gradually tuned to find a more optimized set of parameters, as shown in Fig. 7(c). Correspondingly, the trained waveform (white) continuously adapts itself and gets closer to the target waveform from Fig. 7(b) to (c). Furthermore, the whole training process is guided and assessed by the validation set of inputs using cross-validation strategy [15] that is able to help in preventing overfitting. The training process is stopped when the error in a validation set starts growing. However, the error trend of the validation set is not shown in the current GUI due to the additional computation resources required and energy savings considerations.

4) Adaptive ANN-Based Prediction: Compared to traditional approaches, patient-specific classification approaches analyze the ECG waveform characteristics more precisely on a patient-by-patient basis. Nevertheless, real-time applications often require faster training techniques that can be applied to the detector in advance. Thus, we proposed a hybrid adaptive strategy, which is more suitable for the real-time mobile environment by utilizing both patient-specific information and established ECG medical databases.

As shown in Fig. 8, the proposed hybrid training strategy can be broken down into the following steps:

1) using an established ECG database to train the ANN;
2) testing the trained ANN on the prospective user;
3) collecting a new ECG dataset M from this user;
4) dividing this new dataset M into two groups, named A and B, whose testing errors are, respectively, within or beyond the predetermined threshold;
5) retraining the ANN based on the dataset M combined with expert's annotation, or purely based on the dataset A;
6) the individualized ANN can perform more accurate CVD assessment by accounting for physiological characteristics that are specific to the target user.

Starting from a generic ECG processing structure and following this adaptive training process, the proposed ANN will be able to classify the end user's individual ECG data more accurately.

The final user interface of the proposed CVD classification and prediction framework is shown in Fig. 7(d), where each beat is classified into the normal set or one of the 13 arrhythmias with the detection result shown above it. In this snapshot, two "N" markers correspond to the normal beats and a PVC is identified as a "P."

III. Implementation

As previously described, the objective of this research seeks to develop smartphone-based platform technologies for wearable CVD detection, which are capable of performing real-time ECG acquisition and display, feature extraction, and beat classification. To demonstrate the feasibility of this idea and to conduct real-time performance characterization on real devices, we developed two proof-of-concept prototypes to show that the proposed techniques are applicable for both high-end and low-end smartphones.

A. Plug-In-Based GUI Platform

An Alive Bluetooth ECG heart monitor, an Amoi E72 Microsoft Windows Mobile 5 Smartphone, MATLAB R2008b (used for initial algorithm creation and validation), and Microsoft Visual Studio 2008 were utilized in the implementation and testing of our plug-in-based real-time CVD monitoring system. The application is multithreaded and is written in C# with a feature-extraction plug-in written in C++. For data acquisition and display threads, C# code was used to take advantage of the efficient integration between C# and Windows Mobile. On the other hand, algorithmic code was written in C++ to increase execution speed of the algorithms.

B. Machine-Learning-Based Platform

An Alive Bluetooth ECG heart monitor and an HTC Microsoft Windows Mobile 6 Smartphone were utilized in the creation of the machine-learning platform. MATLAB R2008b was utilized to develop and verify our ANN-based ECG processing and CVD classification algorithms prior to converting to LabVIEW. Next, a platform making use of the LabVIEW Mobile Module (version 8.5), which allows a smartphone to run a LabVIEW binary executable, was developed to deploy the verified algorithms onto the Windows Mobile Smartphone to assess their real-time performance in the target mobile environment.

IV. Results

Both offline and online verification of the algorithms was performed using the MIT-BIH database. In particular, this paper proposes and uses the novel streaming of the MIT database to the smartphone for real-time verification. This allows us to verify the detection of arrhythmias that we could not otherwise detect using normal subjects wearing a heart monitor. To the best of our knowledge, this streaming-based verification technique for CVD detection on smartphone is new.

Furthermore, the machine-learning platform investigated 5421 QRS complex templates, which cover the normal beats and four arrhythmia conditions from the MIT-BIH arrhythmia database. To estimate the classification accuracy, we adopted the three-way cross-validation method that is designed to minimize the variations due to the random sampling of finite-size data samples [16]. We partitioned each class of the dataset randomly into three disjoint subsets of approximately equal size denoted by trial A, trial B, and trial C. The training and testing were performed three times each with each one of the three subsets as the training set and the other two as the testing sets. We did such partitioning using MATLAB and the details are summarized in Table I.

We performed the experiment in which a 51-30-5 MLP ANN structure, based on the structure established for ECG signal detection and classification in [16] was used to classify all beats into five classes. Table II summarizes the results. A prediction accuracy of greater than 90% was achieved, except for the 81% prediction accuracy for fusion of paced and normal beat (PFUS), which could be attributed to the varying morphologies of fusion complexes, which change depending on the portion of the ventricles depolarized by each of the activation fronts [17].

TABLE I
DISTRIBUTION OF SAMPLES IN EACH PARTITION

	No. of Beats	Trial A	Trial B	Trial C
Normal	1856	600	600	656
RBBB	1781	450	550	781
PVC	99	25	30	44
PACE	1685	450	550	685
PFUS	30	7	11	12
Total	5421	1532	1741	2178

TABLE II
PREDICATION ACCURACY OF NORMAL AND FOUR ABNORMAL BEATS
(UNIT IN PERCENTAGE)

Train	A		B		C		Average
Test	B	C	A	C	A	B	
Normal	99.7	100	99.7	95.5	99.5	100	99
RBBB	98.2	98.7	97.8	98.6	97.2	97.6	98
PVC	93.3	93.2	88.0	95.5	92.0	93.3	92.6
PACE	96.4	97.1	95.6	96.4	95.3	95.5	96
PFUS	81.8	83.3	71.4	91.7	85.7	72.7	81

*RBBB: right bundle branch block beat, PVC: premature ventricular contraction, PACE: paced beat, and PFUS: fusion of paced and normal beat.

V. Conclusion and Future Work

Two smartphone-based platforms for the continuous monitoring and recording of a patient's ECG signal successfully perform real-time CVD detection and generate personalized cardiac health summary reports. Not only can the ECG signal be recorded for offline analysis similar to Holter monitors, but we have also provided the user an enriched interface that provides real-time CVD monitoring. The same statistical information as resting ECG machines is generated, except on a Windows Mobile Smartphone instead of a large, bulky contemporary ECG machine. The plug-in-based platform currently diagnoses PVC beats, which are an extremely common arrhythmia, while the machine-learning platform diagnoses right bundle branch block beat (RBBB), PVC, paced beat (PACE), and PFUS beats.

These platforms allow users to perform assistive diagnosis solutions, such as establishing a baseline level of abnormal beats. They can further utilize the system to monitor their daily number of abnormal beats and investigate on their own if lifestyle changes, such as increasing exercise, diet management, reducing caffeine intake, etc., which can decrease the number of uncomfortable and potentially dangerous arrhythmic beats.

We are currently working on increasing the number of detectable CVDs as well as more sophisticated diagnostic algorithms. Furthermore, code parallelizations are being performed to increase concurrency. In future phones, multicore processors would be able to take advantage of the current multithreaded application and the additional parallelization in order to increase the capacity of the smartphone-based CVD-detection solutions, allowing patients to become more involved with monitoring their own health.

ACKNOWLEDGMENT

The authors would like to thank Dr. S. Zurek at Cardiff University (United Kingdom) for the contribution and help in the ANN implementation on LabVIEW.

REFERENCES

[1] D. Lloyd-Jones *et al.*, “Heart disease and stroke statistics 2009 update. A report from the American Heart Association statistics committee and stroke statistics subcommittee,” *Circulation*, vol. 119, no. 3, pp. e21–e181, Jan. 2009.

[2] J. M. Cano-García, E. González-Parada, V. Alarcón-Collantes, and E. Casilari-Pérez, “A PDA-based portable wireless ECG monitor for medical personal area networks,” in *Proc. IEEE MELECON*, Benalmádena, Spain, 2006, pp. 713–716.

[3] W. Chung, C. Yau, K. Shin, and R. Myllyla, “A cell phone based health monitoring system with self analysis processor using wireless sensor network technology,” in *Proc. 29th IEEE EMBC*, Lyon, France, 2007, pp. 3705–3708.

[4] T. Lee, J. Hong, and M. Cho, “Biomedical digital assistant for ubiquitous healthcare,” in *Proc. 29th IEEE EMBC*, Lyon, France, 2007, pp. 1790–1793.

[5] X. Chen, C. T. Ho, E. T. Lim, and T. Z. Kyaw, “Cellular phone based online ECG processing for ambulatory and continuous detection,” *Comput. Cardiol.*, vol. 34, pp. 653–656, Sep. 2007.

[6] K. Goh, J. Lavanya, Y. Kim, E. K. Tan, and C. B. Soh, “A PDA-based ECG beat detection for home cardiac care,” in *Proc. 27th IEEE EMBC*, Shanghai, 2005, pp. 375–378.

[7] J. Rodríguez, A. Goni, and A. Illarramendi, “Real-time classification of ECGs on a PDA,” *IEEE Trans. Inf. Technol. Biomed.*, vol. 9, no. 1, pp. 23–34, Mar. 2005.

[8] Z. Jin, Y. Sun, and A. Cheng, “Predicting cardiovascular disease from real-time electrocardiographic monitoring: An adaptive machine learning approach on a cell phone,” in *Proc. 31st IEEE EMBC*, Minneapolis, MN, 2009, pp. 6889–6892.

[9] M. Gertsch, *The ECG Manual*. London: Springer-Verlag, 2009, p. 267.

[10] P. Hamilton, “Open source ECG analysis,” *Comput. Cardiol.*, vol. 29, no. 1, pp. 101–104, Sep. 2002.

[11] J. Pan and W. Tompkins, “A real-time QRS detection algorithm,” *IEEE Trans. Biomed. Eng.*, vol. BME-32, no. 3, pp. 230–236, Mar. 1985.

[12] P. S. Hamilton and W. J. Tompkins, “Quantitative investigation of QRS detection rules using the MIT/BIH arrhythmia database,” *IEEE Trans. Biomed. Eng.*, vol. BME-33, no. 12, pp. 1157–1165, Dec. 1986.

[13] G. D. Clifford, F. Azuaje, and P. McSharry, *Advanced Methods and Tools for ECG Data Analysis*. Boston, MA: Artech House, 2006, pp. 146–148.

[14] A. L. Goldberger *et al.*, “PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals,” *Circulation*, vol. 101, no. 23, pp. e215–e220, Jun. 2000.

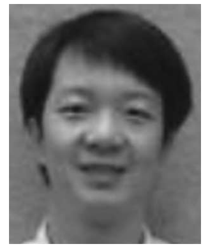
[15] R. Kohavi, “A study of cross-validation and bootstrap for accuracy estimation and model selection,” in *Proc. 14th Int. Joint Conf. Artif. Intell.*, Montreal, QC, Canada, 1995, pp. 1137–1143.

[16] Y. Hu, W. J. Tompkins, J. L. Urrusti, and V. X. Afonso, “Application of artificial neural networks for ECG signal detection and classification,” *J. Electrocardiol.*, vol. 26, pp. 66–73, 1994.

[17] B. Surawicz and T. K. Knilans, *Chou’s Electrocardiography in Clinical Practice*. Philadelphia, PA: Saunders Elsevier, 2008, p. 607.

Joseph J. Oresko (S’04) received the B.S. degree in electrical engineering technology and the B.S. degree in mechanical engineering technology from the University of Pittsburgh, Johnstown, PA, where he is currently working toward the M.S. degree from the Department of Electrical and Computer Engineering.

He was an Assistant Electromechanical Engineer in the Department of Modeling and Simulation, Concurrent Technologies Corporation, Johnstown, PA. His current research interests include biomedical and bioimplantable computing systems and embedded systems.



Zhanpeng Jin (S’08) received the M.S. degree in computer science from the Northwestern Polytechnical University, X’ian, China. He is currently working toward the Ph.D. degree from the Department of Electrical and Computer Engineering, University of Pittsburgh, Pittsburgh, PA.

His current research interests include energy-efficient embedded systems, biomedical systems and devices, reconfigurable hardware systems, hardware/software codesign, integrated circuit and very-large-scale integration system design, computer architecture, and microprocessors.

Mr. Jin is a Student Member of the IEEE Computer Society.

Jun Cheng received the M.S. degree in signal and information processing from the University of Electronic Science and Technology, Chengdu, China. He is currently working toward the M.S. degree from the Department of Electrical and Computer Engineering, University of Pittsburgh, Pittsburgh, PA.

His current research interests include image processing, biosignal processing, biological engineering, and instrumentation.

Shimeng Huang received the B.S. degree in automation from Tsinghua University, Tsinghu, China. She is currently working toward the Ph.D. degree from the Department of Electrical and Computer Engineering, University of Pittsburgh, Pittsburgh, PA.

Her current research interests include computer architecture, biomedical and bioimplantable computing systems, and brain–computer interface technology.

Yuwen Sun received the B.S. degree in information science and electrical engineering from Zhejiang University, Hangzhou, China. He is currently working toward the Ph.D. degree from the Department of Electrical and Computer Engineering, University of Pittsburgh, Pittsburgh, PA.

His current research interests include biomedical computing systems, computer architecture, very-large-scale integration, and field-programmable gate array prototyping.

Heather Duschl is currently working toward the B.S. degree in computer engineering and the B.A. degree in Japanese language and literature from the Department of Electrical and Computer Engineering, University of Pittsburgh, Pittsburgh, PA.

She was at Chonbuk National University, Jeonju, Korea, where she was involved in a display laboratory focused on liquid crystals. She was with Dynon Avionics and SAE International during summers. She is currently an Undergraduate Student in the Department of Electrical and Computer Engineering, University of Pittsburgh. Her current research interests include biomedical computing systems and embedded systems.

Allen C. Cheng (M’98) received the Ph.D. degree in computer science and engineering from the University of Michigan, Ann Arbor.

He is currently an Assistant Professor in the Departments of Electrical and Computer Engineering, Computer Science, Bioengineering, and Neurological Surgery, University of Pittsburgh, Pittsburgh, PA, where he also directs the Advanced Computing Technology Laboratory. His research interests include the interdisciplinary confluence of computer engineering, computer science, neural engineering, biomedical engineering, and medicine.

Dr. Cheng is a member of the Association for Computing Machinery and the American Association for the Advancement of Science.