International Journal of Smart Home
Vol. 2, No. 2, April, 2008

Sensor-Based Medical Information System (SBMIS)

M. Chuah, F. Fu, P. Yang
Department of Computer Science & Engineering
Lehigh University
chuah@cse.lehigh.edu, fef205@lehigh.edu, pey204@lehigh.edu

Abstract

Recently, wireless sensor networks have been proposed for assisted living and residential monitoring. In such networks, physiological sensors are used to monitor vital signs e.g. heartbeats, pulse rates, oxygen saturation of senior citizens. Sensor data is sent periodically via wireless links to a personal computer that analyzes the data. In this paper, we first describe the architecture of a sensor-based medical information system that we are developing. Then, we describe how we deal with security issues in our system. Next, we describe an ECG anomaly detection scheme that we proposed. Our approach is based on time series analysis that will allow the computer to determine whether a stream of real-time sensor data contains any abnormal heartbeats. If anomaly exists, that time series segment will be transmitted via the network to a physician so that he/she can further diagnose the problem and take appropriate actions. When tested against the heartbeat data readings stored at the MIT database, our ECG anomaly scheme is shown to have better performance than another scheme that has been recently proposed. Our scheme enjoys an accuracy rate that varies from 70-90% while the other scheme has an accuracy that varies from 40-70%.

1. INTRODUCTION

Recent report [1] has indicated that an aging baby-boom generation is stressing the US healthcare system. Hospital administrators and other medical care-givers are looking for ways to maintain quality of care at reduced costs. Thus, some researchers [1] have proposed to shift from the familiar centralized, expert-driven, crisis-care model to one that allows senior citizens to live with informal caregivers e.g. family, friends, and community. They propose using wireless sensor networks that can provide capabilities that are valuable for continuous, remote monitoring [1]. In such sensor networks [1],[2],[9], wireless devices are integrated with a wide variety of environmental and medical sensors. Vital sign data can be collected automatically, thus enabling remote medical monitoring and diagnosis. It is envisioned that such a system needs to be designed efficiently since some of these monitoring devices run on battery and thus have limited power constraints. Usually sensor data is collected by some intermediate storage nodes which have higher wireless bandwidth. For better energy efficiency, the intermediate storage nodes can process these real-time streams to identify any abnormality. Once identified, only the abnormal data needs to be sent to the physician for further diagnosis while the rest of the normal data can be archived at the local storage nodes. The local storage nodes can further transfer such normal data to longer term storage units at a slower time scale (e.g. daily). The system can also provide a feature for the physician to request for more detailed immediate data from the local storage nodes or change the frequency of monitoring of the sensor nodes.

In this paper, we first give a high level overview of the system architecture of a sensor-based medical information system that we are developing for a nursing home. Our system uses the medical sensors designed by the CodeBlue [2] project. Base stations and some

mobile data collectors e.g. PDAs owned by care-givers that can communicate with such sensors are deployed all over the nursing home. The base stations are connected to a centralized data server. Several design issues need to be addressed before such a system can be deployed e.g. secure transfer of measurement data from the sensors to a centralized server, power-efficient scheme to monitor the vital signs of the elderly, location tracking of the elderly etc. Next, we discuss some of the potential threats to sensor-based medical information systems. Hackers can eavesdrop on the data or spoof a base station to retrieve data from the deployed sensors or modify the sensor data sent to a base station. To mitigate such threats, our system only allows the sensors to communicate directly with legitimate base stations or mobile data collectors. Medical personnel can only query a patient's data through one or a few base stations but not directly through the sensors. The base stations or the centralized data station will verify the medical personnel before allowing for such queries.

Besides tackling security issue, our initial work focuses on designing a power-efficient scheme for monitoring vital signs and secure transfers of these measurements. The medical sensors run on batteries and communications have been shown to be the primary sources of power consumptions [16]. Several important vital signs that are usually collected in such a medical sensor monitoring system are heartbeats, pulse rates and oxygen saturation. To allow the sensor nodes to last longer, we propose to transmit only abnormal data via wireless links. To decide if a real time sensor data stream contains abnormal data, one needs to use an anomaly detection scheme. In this work, we propose an adaptive window-based discord discovery (AWDD) scheme to detect abnormal heartbeats within a series of heartbeat readings. Our scheme is an enhancement of the Brute Force Discord Discovery (BFDD) scheme proposed in [4]. Using the heartbeat records from the MIT-BIH arrhythmia database [3], we demonstrate that our AWDD scheme provides higher accuracy in distinguishing between normal/abnormal heartbeats within a 40 seconds excerpts of heartbeat readings when compared to the BFDD.

The rest of the paper is organized as follows: In Section II, we first summarize related work on ECG anomaly detection. Then, we summarize related work on the security design of medical information systems. In Section III, we first describe the overall architecture of our system. Then, we describe the security architecture of our system. In Section IV, we describe both the BFDD and the AWDD schemes that are used for ECG anomaly detection. In Section V, we present the training and test results when the two schemes are applied to the records selected from the MIT-BIH arrhythmia database. We conclude in Section VI.

2. RELATED WORK

2.1 ECG Anomaly Detection

Heart arrhythmias result from any disturbance in the regularity, rate, site of origin or conduction of the cardiac electric impulse [1],[2]. There are two groups of arrhythmias [2]: (i) the first group is life threatening and includes ventricular fibrillation and tachycardia, and (ii) the second group is not life threatening but may require medical attention to prevent bigger problems. There are well researched and successful detectors for detection of the first group of arrhythmias. Such detectors have high sensitivity and specificity [3],[4],[5],[6],[7]. However, these detectors have been tested using data collected from expensive medical sensors. In our work, we hope to use cheaper medical sensor nodes which may generate more noisy data. Thus, we are focusing more on the detection of the second group events.

Due to the limited power resources in a sensor-based medical information system, we need to use an anomaly detection scheme that is not computationally expensive. In a seminal paper [4], the authors introduce the new problem of finding time series discords. Time series discords are subsequences of a longer time series that are maximally different to all the rest of the time series subsequences. Time series discords have many uses for data mining including data cleaning, improving quality of clustering and anomaly detection. The authors in [4] propose two discord discovery algorithms, namely the Brute Force Discord Discovery (BFDD) and the Heuristic Discord Discovery (HDD) schemes. The BFDD scheme has a $O(m^2)$ time complexity while the HDD can have an O(m) time complexity where m is the number of samples in the time series. The authors show that their schemes can be used to detect discords that exist within Electrocardiograms (ECGs) (which are a time series of the electrical potential between two points on the surface of the body caused by a beating heart). For example in Figure 1, the identified discord coincides with the location annotated by a cardiologist as containing an anomalous heartbeat. The Adaptive Window Based Discord Discovery (AWDD) scheme that we design in this paper is motivated by the two schemes in [4], and will be described in more details in Section III.

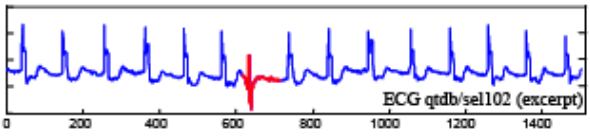


Figure 1: The time series discord in an excerpt of electrocardiogram qtdb/sel102 (marked in bold) which coincides with a premature ventricular contraction [4].

2.2 Related Work on Security Design of Medical Information System

One of the major barriers to deploying security on sensor networks is that the sensors have limited computation and communication capabilities. Early work on sensor network security uses symmetric cryptography e.g. [11] to protect sensitive data. In [12], the authors describe TinySec, a fully-implemented protocol for link-layer cryptography in sensor networks. TinySec provides two options, namely authenticated encryption and authentication only. With authenticated encryption, TinySec encrypts the payload and authenticates the packet with a Message Authentication Code (MAC). The MAC is computed over the encrypted data and the packet header. In authentication only mode, TinySec merely authenticates the entire packet with a MAC. The data payload is not encrypted. However, TinySec has been tested on mica2 motes but we are using micaz motes. In addition, TinySec does not address the key management issue. Recent research [13],[14] has shown that performing public-key computations are viable in the resource-constrained sensor motes.

Several of the existing medical sensor network research projects acknowledge the need for security in their systems but do not provide detailed descriptions of their security design except for [1]. In [1], the authors propose using symmetric Advanced Encryption Standard (AES) cipher but does not elaborate on how key management can be accomplished.

3. Architecture Overview of Our Sensor-Based Medical Information System

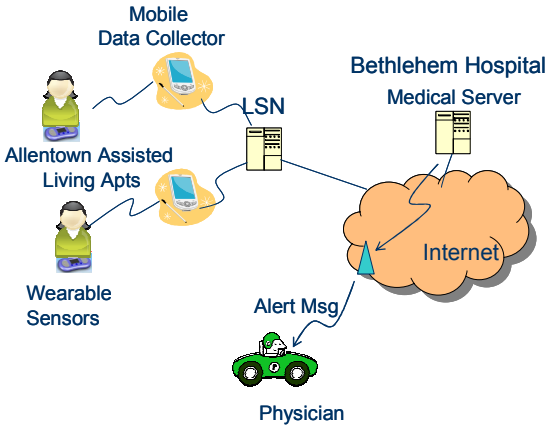


Figure 2. Overall System Architecture of the SBMIS

Figure 2 shows the overall system architecture of our SBMIS. The elderly residents will wear sensors that help to monitor vital signs. Each sensor consists of a mote connected to another medical sensor board e.g. a pulse oximeter or ECG board that collects samples of an elderly's vital signs when it is activated. We will refer to this integrated sensor merely as the medical sensor mote. The medical sensor mote can communicate with any base stations that are deployed all over the nursing home. Base stations (referred to as local storage nodes in Figure 2) are deployed all over the nursing home. These base stations form an ad hoc network and are connected to a centralized data server. In addition, mobile data collectors e.g. PDAs owned by care-givers are also deployed. The base stations and the mobile data collectors run multihop routing protocol. These base stations or the mobile data collectors can issue queries to the medical sensors to retrieve sensor data or issue commands to change the sampling frequency of the medical sensors. However, the mobile data collectors are only allowed to do so after the users are authenticated by the centralized system. All communications are done securely so that outsiders cannot eavesdrop on the data exchange between the base stations and the medical sensors. More discussions of our security design is deferred to the next subsection.

Each sensor mote is pre-loaded with some information that captures the normal profile of the elderly that wears the sensor mote. Each medical sensor mote runs an anomaly detection module and sends only excerpts of abnormal time series of vital signs to the base stations which are relayed to the centralized data server. The data server runs applications that allow alert messages to be generated to the cell phones of physicians that take care of the elderly. In our initial prototype, the ECG anomaly detection software runs either on a PDA that is deployed in the bedroom of the elderly or at the base stations. In the final version, we hope to simplify the ECG anomaly detection module such that it can reside within the medical sensor. We have not implemented the application on the data server yet but we intend to investigate if the one developed in the I-living project [15] can be tailored for our needs.

3.1 Security Challenges

There are two types of threats to the sensor-based medical information system (SBMIS), namely outsider and insider attacks. Outsider attacks are those launched by attackers who do not have control of any legitimate nodes in the system e.g. the medical sensors or base stations. Such attackers can eavesdrop on the communications between the medical sensors and the base stations, between the base stations. They can also masquerade as legitimate base stations to extract data from the medical sensors or change the packets exchanged between base stations. In addition, they can replay some old data packets that are sent from the medical sensors to the base stations. Insider attacks are launched by attackers once they have compromised the nodes within our SBMIS. For example, if they have control of the medical sensors, they can forge false data or extract any stored measurement data. If they control any base stations, they can arbitrarily change the sampling frequency of the medical sensors or issue queries to extract measurement data from the medical sensors. Such physical compromise is more likely in medical sensor networks than in mobile ad hoc networks because the nursing home is often visited by outsiders. Security solution needs to be provided to thwart such attacks. Furthermore, a scalable solution to authorize different personnel e.g. physicians, nurses or care-givers to access different levels of medical information need be provided to ensure the privacy of the data.

3.2 Security Design

In our system, we assume that the centralized data server will authenticate any medical personnel before he/she is allowed to issue queries to retrieve medical measurements stored at the sensors. A submitted query will be routed to a base station that is closer to the sensor worn by a particular elderly. To prevent attackers from using any sensor nodes that do not belong to the network to communicate with legitimate base stations, we propose storing a biometric signature inside any mote that is deployed and worn by an elderly. This signature can be verified by a base station before communication between the base station and the mote can occur. Such biometric signature can be created from an offline fingerprint reader. In addition, the queries and responses between any base station and a mote are encrypted using pairwise symmetric keys shared between the mote and that base station. Such symmetric keys can be negotiated securely at the beginning of their communication. We describe an identity-based key negotiation protocol that allows a mote to negotiate a symmetric key with a base station.

Figure 3 shows the identity-based key negotiation protocol we propose. Before a mote is given to an elderly, it is preloaded with the resident identifier (RID) assigned to that elderly and its private key. One may use the same public key for all base stations or a few different public keys e.g. all base stations in the 1st floor have the same public/private keys. We assume that base stations periodically send beacons advertising their own identifiers which will also be used as their public keys since we are using identity-based cryptography. We further assume that when a query is issued to a particular mote via the mobile reader or base station, the query contains the mote identifier which also acts as the public key of that particular mote. The key negotiation begins by having a base station sends a key_init message. This key_init message is encrypted using the public key of the mote. The mote responds with a key_neg_start message that contains the RID, a session number, and a nounce n_1. This message is encrypted using the public key of the base station. The base station then responds with a key_neg_ack message that contains the mote's nounce n_1, another nounce n_2 generated by the base station and K_m. This message is encrypted using the public key of the mote. The

master key K_m and both nounces are used to create the encryption key and the MAC key for all subsequent messages between the mote and the base station. The mote then returns a a confirm message called key_neg_cfm which contains the RID, the base station's nounce, and K_m. These exchange allow both the mote and the base station to authenticate one another. Note that the encryption key and the MAC key is not sent during this key negotiation process. Thus, unless the nodes are compromised, eavesdroppers cannot figure out what the encryption and MAC keys are since they do not know the functions *f,* and *g*.

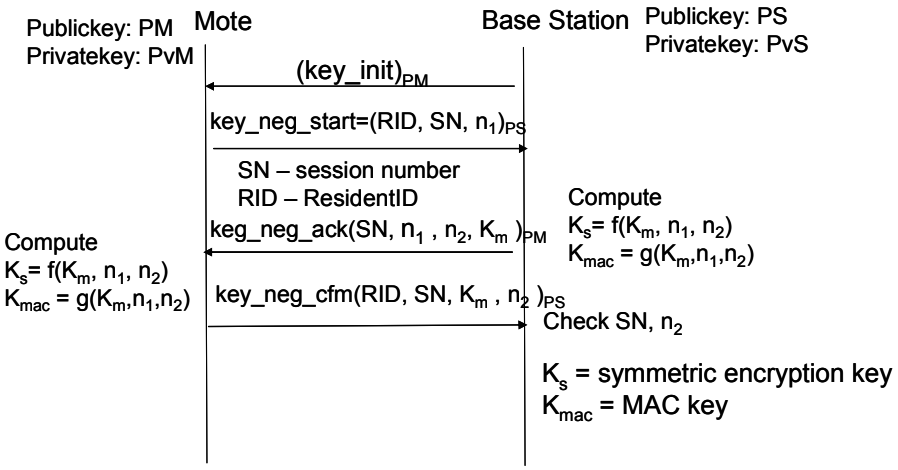


Figure 3. Key Negotiation Process between a Mote and the Base Station

4. OVERVIEW OF THE BFDD AND AWDD SCHEMES

As mentioned earlier, to ensure that the medical sensor can be used for a long time before its battery needs to be changed, the sensor software is designed such that it only sends abnormal measurements to the base station. In order for this feature to work, we need to design efficient anomaly detection module that can run on the motes. In this section, we describe two anomaly detection schemes that can be used to detect abnormal heart beats.

4.1 Notations Used

Before describing both the BFDD and the AWDD schemes that find discords in a time series, we first list the notations we use (which is the same as [4]):

Times Series: *A time series T = t1, …, tm is an ordered set of m real-valued variables. In this project, the real-valued variables are the heartbeat sensor readings.*

Subsequence: Given a time series *T* of length *m*, a subsequence *C* of *T* is a sampling of length $n \leq m$ of contiguous position from *T*, that is, $C = tp, \ldots, tp+n-1$ for $1 \leq p \leq m-n+1$.

Sliding Window: Given a time series *T* of length *m*, and a user-defined subsequence length of *n*, all possible subsequences can be extracted by sliding a window of size *n* across *T* and considering each subsequence *Cp*.

Distance: *Dist* is a function that has *C* and *M* as inputs and returns a nonnegative value *R*, which is said to be the distance from *M* to *C*. For subsequent definitions to work we require that the function D be symmetric, that is, *Dist(C,M) = Dist(M,C)*.

Euclidean Distance: Given two time series *Q* and *C* of length *n*, the Euclidean distance between them is defined as: *Dist(Q,C)* = sqrt [$\sum (qi - ci)^2$].

Non-Self Match: Given a time series *T*, containing a subsequence *C* of length *n* beginning at position *p* and a matching subsequence *M* beginning at *q*, we say that *M* is a non-self match to *C* at distance of *Dist(M,C)* if $| p - q| \geq n$.

Time Series Discord: Given a time series *T*, the subsequence *D* of length *n* beginning at position *l* is said to be the discord of *T* if *D* has the largest distance to its nearest non-self match. That is, $\forall$ subsequence *C* of *T*, non-self match *MD* of *D*, and non-self match *MC* of *C*, min(*Dist*(*D*, *MD*)) > min(*Dist*(*C*, *MC*)).

4.2 Adaptive Window Based Discord Discovery (AWDD) Scheme

The original BFDD algorithm proposed in [4] is a one-pass algorithm which uses a fixed window size and hence a user needs to specify the window size. This algorithm compares a fixed length subsequence with another subsequence of the same length that is obtained by sliding down a given time series one sample at a time. Hence, the original BFDD scheme is very computational expensive. Our AWDD scheme is motivated by the BFDD scheme. The AWDD scheme is a two-pass approach with adaptive window size. In the first pass, we identify the peak points in the 40-seconds excerpts of heartbeat readings. Then, we consider only the subsequence that starts from a peak and ends at the next peak. The size of the sliding window is of one heartbeat's length, as illustrated in Figure 4. In Figure 4, RR-*i* denotes the heartbeat to heartbeat (denoted as RR) interval between heartbeats *i* and *(i+1)*. As in the original BFDD scheme, each subsequence is normalized to have a mean of zero and a standard deviation of one before calling the euclidean distance function, since it is meaningless to compare time series with different offsets and amplitudes [8]. Note that we use only euclidean distance in this work. Figure 5 shows the effect of normalization on a subsequence of time series obtained from the patient record 205.

In the second pass, we consider each possible subsequence, and find the distance between this and its nearest non-self match. The subsequence that has the largest distance is the discord. The location of the discord is accomplished with nested loops, where the outer loop considers each possible candidate subsequence, and the inner loop is a linear scan to identify the candidate's nearest non-self match.

The time complexity of the AWDD scheme will be $O(h^2)$ where h is the number of heartbeats but the technique that is used in HDD to reduce the time complexity to O(m) can be equally applied to the AWDD scheme to produce a scheme with a time complexity of O(h). As far as space is concerned, AWDD only requires an additional array to keep location of peaks.

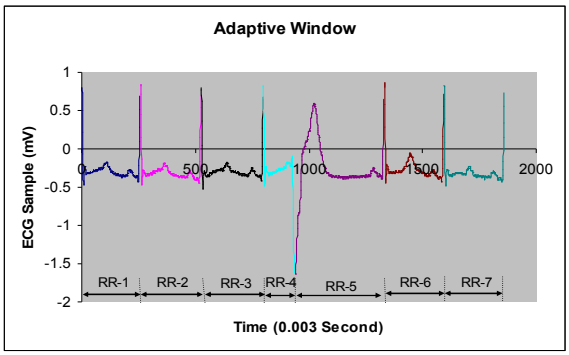
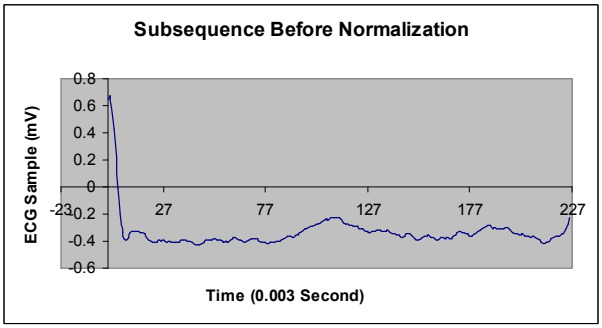
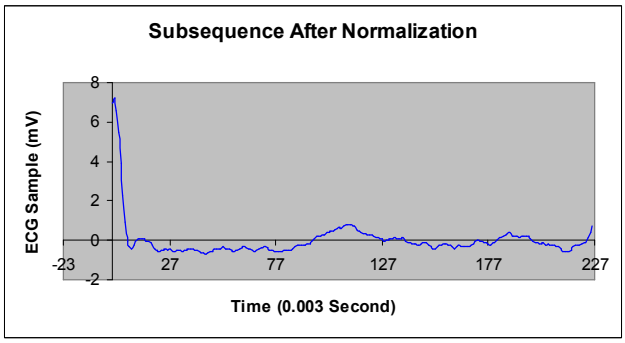


Figure 4. Adaptive Window



(a) Time-Series Subsequence: Before Normalization



(b) Time-Series Subsequence: After Normalization

Figure 5. Normalization of Time Series Subsequence

For clarity, the pseudo code of the BFDD algorithm is shown in Figure 6(a), and our enhanced algorithm is shown in Figure 6(b). Since we are using adaptive windows, we make two more changes to determine and compare the discords. The first change is to ensure that we can compare subsequences of different lengths. To do this, we compress the longer subsequence to match the shorter one. The subsequences are normalized before any potential compression takes place. Figure 7 illustrates the effect of the compression on a subsequence of record 205.

```
Function [dist, loc] = Brute_Force ( T, n )
best_so_far = 0
best_so_far = NaN
outer_cnt = 0

for  p = 1 to | T | - n + 1                    // begin outer loop
        nearest_neighbor_dist = infinity
        for  q = 1 to | T | - n + 1    // begin inner loop
                if  | p – q | ≥ n     // non-self match?
                        if  Dist ( tp ... tp+n-1,  tq ... tq+n-1 ) < nearest_neighbor_dist
                                nearest_neighbor_dist = Dist ( tp ... tp+n-1 ,  tq ... tq+n-1 )
                        end
                end
        end
        if nearest_neighbor_dist > best_so_far_dist
                best_so_far_dist = nearest_neighbor_dist
                best_so_far_dist = p
        end
end
return [ best_so_far_dist , best_so_far_loc ]
```

(a) Pseudo Code for BFDD scheme

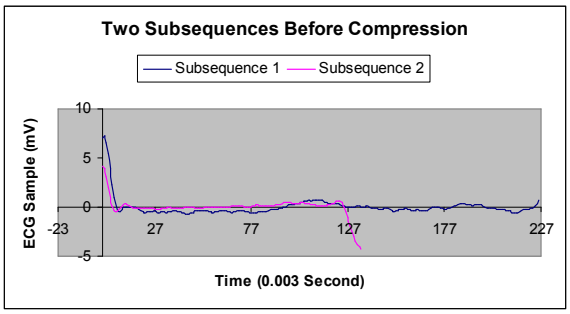
```
Function [dist, loc] = Brute_Force ( T )
best_so_far = 0
best_so_far = NaN
num_of_peaks = 0
p = 1
while  p < | T |                                                    // locate each peak
        if  tp is locally the biggest sample
                peak_loc [ num_of_peaks ++ ] = p
        end
end
outer_cnt = 0
p = peak_loc [0]
while  p < peak_loc [ num_of_peaks – 2 ] + 1                    // begin outer loop
        nearest_neighbor_dist = infinity
        outer_len = peak_pos [ outer_cnt + 1 ] – peak_pos [ outer_cnt ]
        inner_cnt = 0
        q = peak_loc [0]
        while  q < peak_loc [ num_of_peaks – 2 ] + 1          // begin inner loop
                inner_len = peak_pos [ inner_cnt + 1 ] – peak_pos [inner_cnt ]
                if outer_len > inner_len
                        compress  tp ... tp+ outer_len  to have a length of inner_len
                end
                if outer_len < inner_len
                        compress  tq ... tq+ inner_len  to have a length of outer_len
                end
                if  | p – q | ≥ min ( outer_len, inner_cnt )
                        if  Dist ( tp ... tp+ outer_len ,  tq ... tq+ inner_len ) < nearest_neighbor_dist
                                nearest_neighbor_dist = Dist ( tp ... tp+ outer_len ,  tq ... tq+ inner_len )
                        end
                end
                q = peak_loc [ ++ inner_cnt ]
        end
        if nearest_neighbor_dist > best_so_far_dist
                best_so_far_dist = nearest_neighbor_dist
                best_so_far_dist = p
        end
        p = peak_loc [ ++ outer_cnt ]
end
return [ best_so_far_dist , best_so_far_loc ]
```

(c) Pseudo-Code for the AWDD scheme

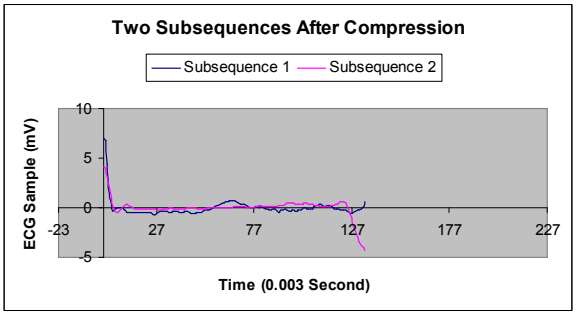
Figure 6. Discord Discovery Schemes

The next change is to deal with the fact that one subsequence-pair may have more samples than the other subsequence-pair and hence we cannot compare the computed distances directly. We overcome this by scaling all distances such that they correspond to the distance computed using the same number of samples.

Figure 8 shows a 40-second ECG excerpt of the patient record 205 with samples from 290th second to 330th second. In this excerpt, abnormal heartbeats start from the 296.875th second and end at the 305.900th second. When feeding this 40-second ECG excerpt to the discord discovery algorithm, our enhanced algorithm locates the discord at the 302.531th second. Its nearest non-self match is at the 296.875th second. Their distance is 7.483. By checking the ECG record annotated by the cardiologists, we can tell that there is indeed an anomaly sitting at the location of the discord found by our algorithm. Figure 9 illustrates the two subsequences, where the discord and the nearest non-self match reside.



(a) Normalized Time-Series Subsequence: Before Compression



(b) Normalized Time-Series Subsequence: After Compression

Figure 7. Compression of Normalized Time Series Subsequence

However, the discord found by the algorithm may or may not be an anomaly of the ECG excerpt. Thus, we use a configurable threshold to decide whether or not a discord is an anomaly. If the distance between the discord and its nearest non-self match exceeds the threshold, we determine that the discord found by our algorithm is an anomaly. Otherwise, our program will not flag this as an anomaly. This threshold is different for each patient and is found by training.

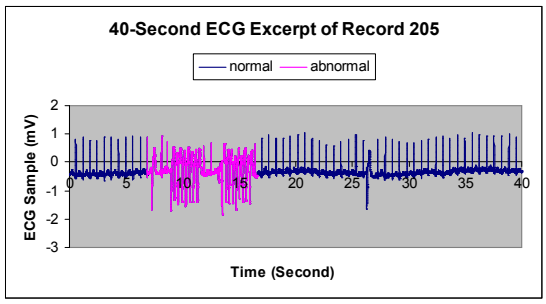
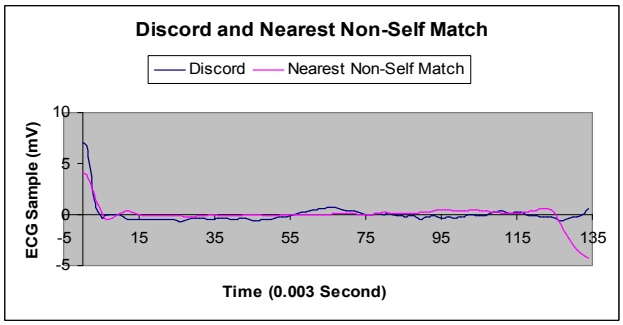
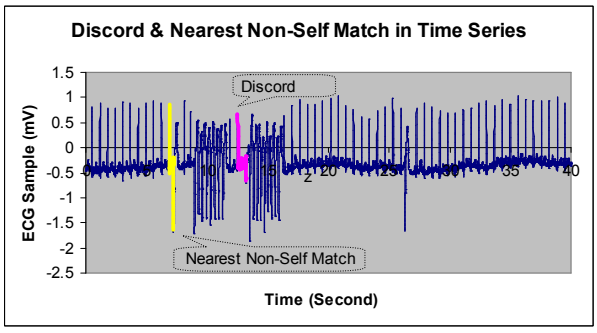


Figure 8. A Sample 40-Second ECG Excerpt from MIT-BIH Record 205



(a) Normalized & Compressed Discord & Nearest Non-Self Match: Distance = 7.843



(b) Discord & Nearest Non-Self Match in the 40-Second ECG Excerpt (the same one as in Figure 6)

Figure 9. Discord & Nearest Non-Self Match

We select some subsets of data from each patient's records as the training data. Each set of training data consists of 5 non-overlapping 40-second excerpts from the same patient, with at least one abnormal heartbeat (an abnormal ECG time series), and another 5 non-overlapping 40-second excerpts from the same patient, which do not contain any abnormal heartbeats (normal ECG time series). Then, we apply the algorithm to each set of training data. Our conjecture is that the distance for the discord in an abnormal ECG time series should be larger than the one in a normal ECG time series. A threshold can then be easily found to allow us to conclude if abnormal heartbeats exist. We will discuss how this threshold is chosen for each patient record and the results of applying this threshold to the test data set in Section V.

5 . EVALUATION RESULTS

5.1 ECG Datasets

Since our medical sensor boards are not ready yet, we use the ECG data from the MIT-BIH Arrhythmia Database [3]. The database contains 48 half-hour excerpts of two-channel ambulatory ECG recordings, obtained from 47 patients studied by the BIH Arrhythmia Laboratory. According to [3],[10], twenty-three of these recordings were chosen at random from a set of 4000 24-hour ambulatory ECG recordings collected from a mixed population of inpatients (about 60%) and outpatients (about 40%) at Boston's Beth Israel Hospital; the other 25 recordings were selected from the same set to include less common but clinically significant arrhythmias that would not be well-represented in a small random sample.

According to [3],[10], the recordings were digitized at 360 samples per second per channel with 11-bit resolution over a 10 mV range. Two or more cardiologists independently annotated each record; disagreements were resolved to obtain the computer-readable reference annotations for each beat (approximately 110,000 annotations in all) included with the database. Out of the 48 half-hour excerpts of two-channel ambulatory ECG recordings, we randomly select 6 half-hour excerpts, which are numbered as records 106, 108, 114, 205, 210, and 219 in the database. In each of the 6 half-hour ECG excerpts, we select 10 40-second excerpts, with 5 of them having abnormal heartbeats inside, and the other 5 having no abnormal heartbeats inside as the training set. We use the first channel ECG recordings, rather than use both channels' ECG recordings. Later, we select another 10 40-second excerpts from the same patient records as the test dataset.

5.2 Training & Testing using Record 106

The 10 40-second ECG excerpts chosen from record 106 for training purposes are listed in Table 1. The first 5 40-second excerpts contain at least one abnormal heartbeat, and the remaining 5 excerpts do not contain any anomaly. The 3rd column indicates the location where the 1st abnormal heartbeat starts. For example, the 1st 40-second ECG excerpt from record 106 starts from the 80th second and ends at the 120th second, with the 1st abnormal heartbeat starting from 90.741th second.

Using the BFDD scheme, the window is shifted by one ECG sample each time in both the inner and outer loops. The discord found in each of the 10 40-second ECG excerpts from record 106 is listed in Table 2(a). The last column tells if the heartbeat that the discord belongs to is an abnormal heartbeat. The distance column, which is next to that last column, tells the distance between a discord and its nearest non-self match. We can see that for excerpts 1-5, which do contain abnormal heartbeats, the reported distance between the located discord and its non-self match exceeds 6.5, and for excerpts 6-10, which do not contain abnormal heartbeats, the reported distance never exceeds 6.5, except excerpt 6. So we could set a distance threshold of 6.5, knowing that excerpt 6 will be misclassified as having an anomaly if similar data appear in the test set.

Table 1. 40-Second ECG Excerpts from MIT-BIH Record 106

Index of excerpts	start point - end point (second)	1st anomaly's location (second)
1	80-120	90.74
2	430-470	445.78
3	700-740	710.89
4	960-1000	965.98
5	1040-1080	1048.75
6	0-40	na
7	200-240	na
8	600-640	na
9	1320-1360	na
10	1380-1420	na

Table 2(b) show the results of applying this threshold to the ten test datasets using the BFDD scheme. We see that with a threshold of 6.5, excerpts 1,3 and 5 will not be classified as abnormal and except 10 will be classified as abnormal. So, our accuracy is only 60% (with 30% false negative and 10% false positive) using the BFDD scheme.

Table 2(a). Discords from the training set of MIT-BIH Record 106 using BFDD scheme

index of excerpts	start point - end point (second)	1st anomaly's location (second)	discord's location (second)	nearest non-self match's location (second)	distance between discord & nearest non-self match	is the discord an anomaly in reality?
1	80-120	90.741	86.611	80.917	7.21	No
2	430-470	445.783	440.611	467.917	6.56	No
3	700-740	710.886	728.25	731.222	7.14	Yes
4	960-1000	965.98	985.361	968.528	6.71	Yes
5	1040-1080	1048.75	1051.944	1056.861	6.78	Yes
6	0-40	na	na	Na	6.93	Na
7	200-240	na	na	Na	3.32	Na
8	600-640	na	na	Na	6.16	Na
9	1320-1360	na	na	Na	3.32	Na
10	1380-1420	na	na	Na	5.00	Na

Table 2(b). Discords from the test set of MIT-BIH Record 106 using BFDD scheme

index of excerpts	start point - end point (second)	1st anomaly's location (second)	discord's location (second)	nearest non-self match's location (second)	distance between discord & nearest non-self match	is the discord an anomaly in reality?
1	160-200	160.233	178.167	160.528	5.83	No
2	900-940	902.436	900.972	921.944	7.0	No
3	1200-1240	1203.213	1201.194	1207.25	5.92	Yes
4	1420-1460	1435.497	1421.639	1446.694	6.86	Yes
5	1600-1640	1614.7	1637.139	1627.222	5.0	Yes
6	270-310	na	na	Na	4.36	Na
7	320-360	na	na	Na	4.12	Na
8	380-420	na	na	Na	6.40	Na
9	500-540	na	na	Na	4.58	Na
10	560-600	na	na	Na	6.71	Na

Next, we train the AWDD scheme using the same training dataset. Table 3(a) shows the discord found in each of the 10 40-second ECG excerpts from the training set of patient record 106. We can see that for excerpts 1-5, which do contain abnormal heartbeats, their distance exceeds 2, and for excerpts 6-10, which do not contain abnormal heartbeats, their distance never exceeds 2. Thus, we could set a distance threshold of 2. If the distance between a discord and its nearest non-self match exceeds 2, we will declare the discovered discord as an anomaly and a cardiologist needs to examine the patient's time series.

Table 3(a). Discords from the training set of MIT-BIH Record 106 using AWDD scheme

index of excerpts	start point - end point (second)	1st anomaly's location (second)	discord's location (second)	nearest non-self match's location (second)	distance between discord & nearest non-self match	is the discord an anomaly in reality?
1	80-120	90.741	116.675	96.828	3.17	Yes
2	430-470	445.783	452.653	446.261	2.13	Yes
3	700-740	710.886	725.464	729.858	20.85	Yes
4	960-1000	965.98	970.131	973.117	9.38	Yes
5	1040-1080	1048.75	1057.783	1043.897	7.41	Yes
6	0-40	na	na	na	1.36	na
7	200-240	na	na	na	0	na
8	600-640	na	na	na	1.49	na
9	1320-1360	na	na	na	1.38	na
10	1380-1420	na	na	na	0	na

In the test set shown in Table 3(b), all of the excerpts 1-5 have abnormal heartbeats. None of the excerpts 6-10 contain abnormal heartbeats. The results indicate that we can identify abnormality in excerpts 1-5 since the reported discord distance is greater than the threshold of 2 (which is chosen based on the training set). For excerpts 6-10 with normal heartbeats, only except 8, will report a discord distance which is slightly larger than the threshold of 2. Thus, we get an accuracy of 90% on this testing dataset using our adaptive window based discord discovery scheme. The false positive rate is 10%.

Table 3(b). Discords from the test set of MIT-BIH Record 106 using AWDD scheme

index of excerpts	start point - end point (second)	1st anomaly's location (second)	discord's location (second)	nearest non-self match's location (second)	distance between discord & nearest non-self match	is the discord an anomaly in reality?
1	160-200	160.233	177.4	179.36	17.88	Yes
2	900-940	902.436	914.467	916.017	8.72	Yes
3	1200-1240	1203.213	1218.761	1220.578	9.9	Yes
4	1420-1460	1435.497	1446.622	1459.036	9.15	Yes
5	1600-1640	1614.7	1636.711	1625.739	2.9	Yes
6	270-310	na	na	na	0	na
7	320-360	na	na	na	0	na
8	380-420	na	na	na	2.08	na
9	500-540	na	na	na	0	na
10	560-600	na	na	na	1.06	na

5.3 Accuracy Comparison

We repeatedly performed the above operations on 20 40-second excerpts selected from patient records 108, 114, 205, 210 and 219. Ten excerpts are used for training purposes and ten excerpts are used for testing purposes. The accuracies of the reported anomalies using the

BFDD and the AWDD schemes for the various patients are summarized in Table 4. In Table 4, the first number is the accuracy, the second number is the false negative rate, and the third number is the false positive rate. A higher false positive rate than the false negative rate is acceptable since it pays to check the patient slightly more frequently than to miss checking abnormal heartbeat events. The results indicate that our AWDD scheme can detect abnormality better than the BFDD scheme.

Table 4. Accuracy Using BFDD vs. AWDD Schemes

record	accuracy using fixed window (%)	accuracy using adaptive window (%)
106	60 (30,10)	90 (0, 10)
108	50 (10,40)	80 (10, 10)
114	70 (10,20)	80 (0, 20)
205	70 (30,0)	80 (10,10)
210	70 (0,30)	90 (0, 10)
219	40 (20, 40)	70 (30, 0)

6. CONCLUSION

In this paper, we have presented a high level overview of a sensor-based medical information system that we are building for a nursing home. We also describe the security solution that we adopt in our system. Our solution only allows sensors to communicate with base stations and all users' queries are authenticated and routed via the base stations. In addition, we describe an adaptive window-based discord discovery (AWDD) scheme for detecting abnormal patterns in the heartbeat related time series. Our scheme is motivated by the BFDD scheme proposed in [4] but we use adaptive rather than fixed windows. Our AWDD scheme uses a simple re-sampling method to compare two subsequences that are of different lengths. We apply both the BFDD and the ADWW schemes to ten 40-seconds excerpts of six patient records from the MIT-BH arrhythmia database. Our results show that the enhanced algorithm can achieve better accuracy in locating anomalies in the heartbeat time series of the patients.

We are currently redesigning the Code-Blue mote-based medical sensors designed by Harvard [2] because the operational amplifier chip used in the original design is phased out by Texas Instrument. Once we successfully build our new ECG sensors, we will collect heartbeat data from several volunteers. Then, we will apply the AWDD scheme to these more noisy heartbeat sensor data. We also hope to analyze the sensor data collected from pulse oximeters using the AWDD scheme to see if it is equally effective in detecting anomalies in time series of oxygen saturation readings. In addition, we intend to optimize this algorithm so that it can be run on the mote. Software to display excerpts of medical sensor data with anomalies on PDAs will also be developed. In addition, we intend to investigate if the server platform designed for the I-living [15] can be tailed to our needs.

Acknowledgment

This work is sponsored by PITA. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of PITA. We would like to thank Dr George Moody from MIT for patiently explaining the MIT-BIH database information and to Prof Keogh for making his code on HotSax available.

REFERENCES

[1] A. Wood et al, "ALARM-NET: Wireless Sensor Networks for Assisted-Living Residential Monitoring", University of Virginia Computer Science Department Technical Report (2006)

[2] V. Shnayer et al, "Sensor Networks for Medical Care", Harvard University Division of Engineering and Applied Sciences Technical Report, TR-08-05, (2005)

[3] MIT-BIH Arrhythmia Database. http://www.physionet.org/physiobank/database/mitdb/

[4] E. Keogh, J. Lin, A. Fu. HOT SAX: Efficiently Finding the Most Unusual Time Series Subsequence. The 5th IEEE International Conference on Data Mining (ICDM), (2005)

[5] M. S. Thaler, "The only EKG book you'll ever need", 3rd Ed., Philadelphia, PA: Lippincott Williams & Wilkins, (1999)

[6] P. Chazal, M. O'Dwyer, R. B. Reilly, "Automatic Classification of Heartbeats Using ECG Morpholgy and Heartbeat Interval Features", IEEE transaction on biomedical engineering, Vol 51, No 7, July, (2004)

[7] S. Evans, H. Hastings, M. Bodenheimer, "Differentiation of beats of ventricular and sinus origin using a self-training neural network", PACE, Vol17, (1994) 611-626

[8] R. Clayton, A. Murray, R. Campbell,"Recognition of ventricular fibrillation using neural networks", Med Biological Engineering and Computing, Vol 32, (1994), 217-220

[9] Lehigh's Sensor-Based Medical Information System (SBMIS) http://www.cse.lehigh.edu/~chuah/research.html

[10] G. B. Moody and R. G. Mark, "The impact of the MIT-BIH Arrhythmia Database", IEEE Eng in Med and Biol Vol 20(3) May-June (2001) 45-50.

[11] A. Perrig et al, "SPINS: Security protocols for sensor networks", Proceedings of the ACM Mobicom, 2001.

[12] C. Karlof et al, "TinySec: A Link Layer Security Architecture for Wireless Sensor Networks", Proceedings of ACM Sensys 2004.

[13] H. Wang, B. Sheng, and Q. Li, "TelosB Impleemntation of elliptic curve cryptography over primary field", Technical Report WM-CS-2005-12, Dept of Computer Science, College of William and Mary, Oct, 2005

[14] A. Liu, P.Kampanakis, and P. Ning, "TinyECC: Elliptic Curve Cryptography for Sensor Networks", http://discovery.csc.ncsu.edu/software/TinyECC/

[15] Q Wang et al, "I-Living: An open system architecture for assisted livign", Proceedings of IEEE SMC, 2006.

[16] Z. Shelby, C. Pomalazza-Raez et al, "Enery Optimization in Multihop Wireless Embedded and Sensor Networks", International Journal of Wireless Information Networks Vol 12, No. 1, January 2005.

Authors

Mooi Choo Chuah received her M.S. and Ph.D. degrees in Electrical Engineering fr om University of California, San Diego in 1988 and 1991 respectively, and her B.En g (1st Class Honors) degree in Electrical Engineering from University of Malaya in 1 984. She is currently an associate professor in CSE department at Lehigh University. Before she joined Lehigh, she spent 12 years at Bell Laboratories, Holmdel, NJ doin g research in wireless networking. She has been awarded 52 US patents, 4 internation al patents and have 10 more pending in various areas e.g. resource management, wire less MAC design, mobility management etc. Her current research interests are design ing next generation internet systems, disruption tolerant network and smart home des igns.

Fen Fu received her bachelor degree in Engineering (CS) from Huazhong University in 2003 and M.S (CS) from Lehigh University in 2006. She worked a software engin eer and an intern at Lucent Technologies from July 2003 to August 2005.

Peng Yang is a Ph.D. candidate in the Department of Computer Science and Enginee ring at Lehigh University. He received his MEng. in Computer Engineering from Na nyang Technological University, Singapore and a BEng. in Computer Engineering fr om the Huazhong University, China. His current research focuses on architectural an d protocol design for Delay/Disruption Tolerant Networks