

ECG Anomaly Detection via Time Series Analysis

M. Chuah, F. Fu

Department of Computer Science & Engineering

Lehigh University

chuah@cse.lehigh.edu, fef205@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 propose an anomaly detection scheme that 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%.

Keywords—anomaly detection; ECG; assisted-living; wireless sensor network

I. 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. Providing early detection is an economical way of reducing the burden of disease treatment on the existing healthcare system. Thus, recently 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],[10], 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 are run on battery. One possible way is to send only abnormal data to the physician while archiving the normal data at the local computer. The normal data can be sent periodically (say once a month) via wide area network (without the power and limited bandwidth

constraints) to a physician only if he/she requests for such information.

Three important vital signs that are usually collected in such a medical sensor monitoring system are heartbeats, pulse rates and oxygen saturation. As described earlier, it is more energy efficient 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 paper, we propose an adaptive window-based discord discovery (AWDD) scheme to detect abnormal heart beats 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 heart beats 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 summarized related work. In Section III, we describe both the BFDD and the AWDD schemes. In Section IV, 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 V.

II. RELATED WORK

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]. However, these detectors have been tested using data collected from expensive medical sensors. We are focusing on detection of the second group events.

In our project [10], we are building a medical information system that uses cheaper medical sensors which may generate noisy heart beat readings. In addition, due to the limited power constraint, we need to use an anomaly detection scheme that does not consume too much computational power. 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. For example in Figure 1, the discord coincides with the location annotated by a cardiologist as containing an anomalous heartbeat. 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 and the Adaptive Window Based Discord Discovery (AWDD) scheme that we design will be described in more details in Section III.

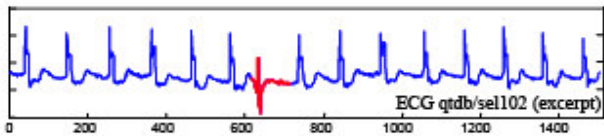


Figure 1: The time series discord in an excerpt of electrocardiogram qtdb/sel102 (marked in bold) which coincides with a premature ventricular contraction [4].

III. OVERVIEW OF THE BFDD AND AWDD SCHEMES

A. 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 $T = t_1, \dots, t_m$ 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, $C = t_p, \dots, t_{p+n-1}$ 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 C_p .

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: $Dist(Q, C) = \text{sqrt} [\sum (q_i - c_i)^2]$.

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 M_D of D , and non-self match M_C of C , $\min(Dist(D, M_D)) > \min(Dist(C, M_C))$.

B. 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 2. $RR-i$ in Figure 2 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 distance function, since it is meaningless to compare time series with different offsets and amplitudes [8]. Figure 3 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.

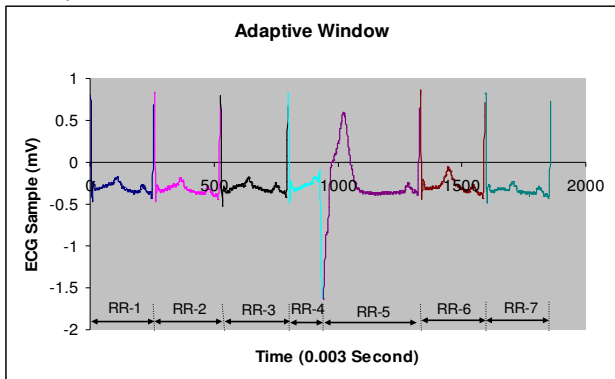
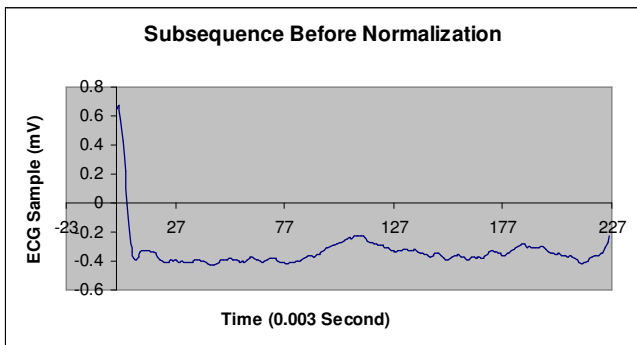
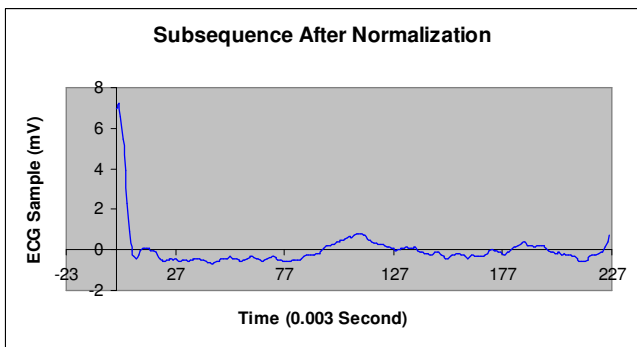


Figure 2. Adaptive Window



(a) Time-Series Subsequence: Before Normalization



(b) Time-Series Subsequence: After Normalization

Figure 3. Normalization of Time Series Subsequence

For clarity, the pseudo code of the BFDD algorithm is shown in Figure 4(a), and our enhanced algorithm is shown in Figure 4(b). Since we are using adaptive windows, we make two more changes to determine the

subsequence with the largest distance. First, the two subsequences that need to be compared may be of different lengths. To overcome this, we compress the longer subsequence to match the shorter one. The subsequences are normalized before any potential compression takes place.

```

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 ( lp ... lp+n-1, lq ... lq+n-1 ) < nearest_neighbor_dist
                nearest_neighbor_dist = Dist ( lp ... lp+n-1, lq ... lq+n-1 )
            end
        end
    end
    if nearest_neighbor_dist > best_so_far_dist
        best_so_far_dist = nearest_neighbor_dist
        best_so_far_loc = 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 lp 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 lp ... lp+outer_len to have a length of inner_len
        end
        if outer_len < inner_len
            compress lq ... lq+inner_len to have a length of outer_len
        end
        if |p - q| ≥ min ( outer_len, inner_cnt )
            if Dist ( lp ... lp+outer_len, lq ... lq+inner_len ) < nearest_neighbor_dist
                nearest_neighbor_dist = Dist ( lp ... lp+outer_len, lq ... lq+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_loc = p
    end
    p = peak_loc [ ++outer_cnt ]
end
return [ best_so_far_dist, best_so_far_loc ]

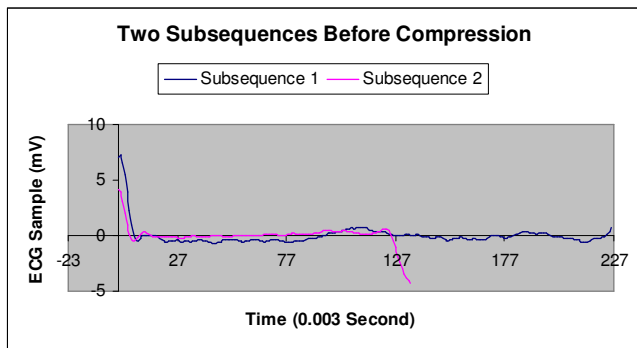
```

(b) Pseudo-Code for the AWDD scheme

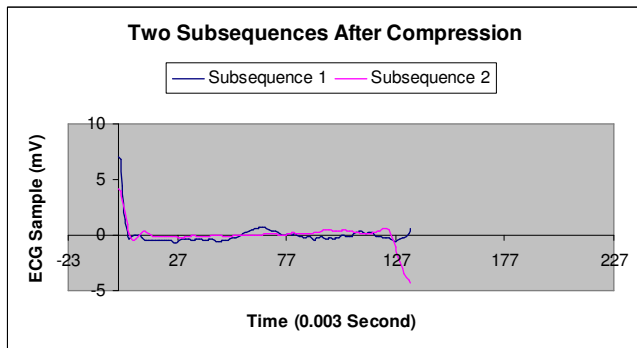
Figure 4. Discord Discovery Schemes

Figure 5 illustrates the effect of the compression on a subsequence of record 205. Next, the number of samples in the compressed subsequence pair may still be different from another subsequence pair. Thus, we

use a scaling approach to deal with this problem. We basically scale all distances such that they correspond to the distance computed using the same number of samples.



(a) Normalized Time-Series Subsequence: Before Compression



(b) Normalized Time-Series Subsequence: After Compression

Figure 5. Compression of Normalized Time Series Subsequence

Figure 6 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 7 illustrates the two subsequences, where the discord and the nearest non-self match reside.

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.

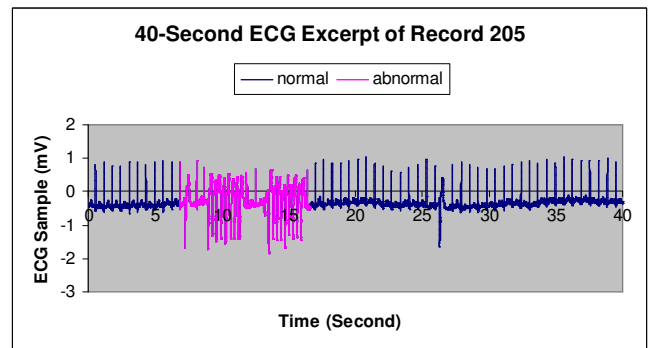
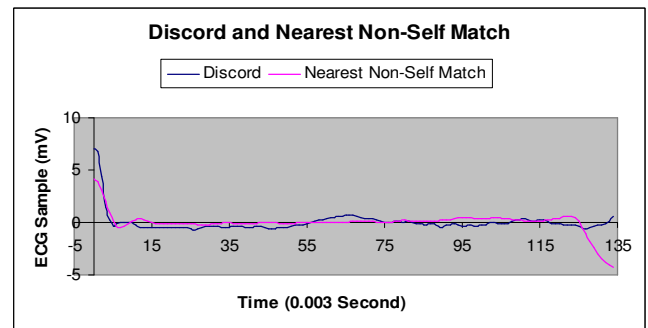
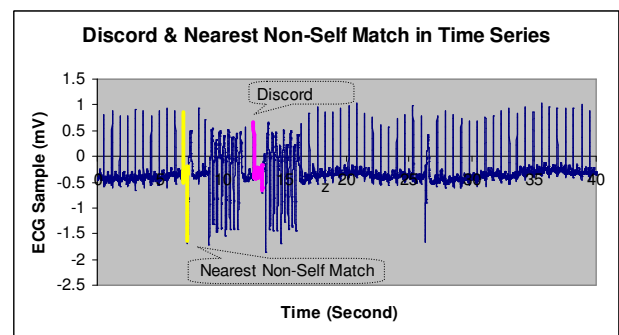


Figure 6. A Sample 40-Second ECG Excerpt from MIT-BIH Record 205



(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 7. 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 there should be a big difference between the distance for the discord in an abnormal ECG time series and the one in a normal ECG time series. The one for the abnormal time series should have a bigger value than the one for the normal 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 IV.

IV. EVALUATION RESULTS

A. 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], these twenty-three 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], 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.

B. 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.

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

Index of excerpts	start point - end point (second)	1 st anomaly's location (second)
1	80-120	90.741
2	430-470	445.783
3	700-740	710.886
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

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 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	7.211	No
2	430-470	445.783	440.611	467.917	6.557	No
3	700-740	710.886	728.25	731.222	7.141	Yes
4	960-1000	965.98	985.361	968.528	6.708	Yes
5	1040-1080	1048.75	1051.944	1056.861	6.782	Yes
6	0-40	na	na	Na	6.928	Na
7	200-240	na	na	Na	3.317	Na
8	600-640	na	na	Na	6.164	Na
9	1320-1360	na	na	Na	3.317	Na
10	1380-1420	na	na	Na	5.000	Na

Table 2(b). 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	160-200	160.233	178.167	160.528	5.831	No
2	900-940	902.436	900.972	921.944	7.0	No
3	1200-1240	1203.213	1201.194	1207.25	5.9161	Yes
4	1420-1460	1435.497	1421.639	1446.694	6.8557	Yes
5	1600-1640	1614.7	1637.139	1627.222	5.0	Yes
6	270-310	na	na	Na	4.359	Na
7	320-360	na	na	Na	4.123	Na
8	380-420	na	na	Na	6.403	Na
9	500-540	na	na	Na	4.58	Na
10	560-600	na	na	Na	6.708	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	3.166	Yes
2	430-470	445.783	452.653	446.261	2.125	Yes
3	700-740	710.886	725.464	729.858	20.847	Yes
4	960-1000	965.98	970.131	973.117	9.377	Yes
5	1040-1080	1048.75	1057.783	1043.897	7.408	Yes
6	0-40	na	na	na	1.360	na
7	200-240	na	na	na	0	na
8	600-640	na	na	na	1.488	na
9	1320-1360	na	na	na	1.379	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 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	160-200	160.233	177.4	179.36	17.875	Yes
2	900-940	902.436	914.467	916.017	8.721	Yes
3	1200-1240	1203.213	1218.761	1220.578	9.9	Yes
4	1420-1460	1435.497	1446.622	1459.036	9.153	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	2.084	na
9	500-540	na	na	na	0	na
10	560-600	na	na	na	1.062	na

C. 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)

V. CONCLUSION

In this paper, we have described 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-second 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 heart beat time series of the patients.

We are currently building the Code-Blue mote-based medical sensors designed by Harvard [2]. Once we successfully test those Code-Blue sensors, we will collect heart beat data from several volunteers. Then, we will apply the AWDD scheme to these more noisy heart beat 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 a PDA. Software to display excerpts of medical sensor data with anomalies will also be developed.

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.

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 Morphology 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, pp 611-626,1994
- [8] R. Clayton, A. Murray, R. Campbell,"Recognition of ventricular fibrillation using neural networks", Med Biological Engineering and Computing, Vol 32, pp 217-220, 1994
- [9] K. Minami, H. Nakajima, T. Toyoshima, "Real-time discrimination of ventricular tachyarrhythmia with Fourier-transform neural network", IEEE Transaction on Biomedical Engineering, Vol 46, pp 179-185, Feb 1999.
- [10] Lehigh's Sensor-Based Medical Information System (SBMIS)
<http://www.cse.lehigh.edu/~chuah/research.html>