Identifying Usage Anomalies For ECG-based Sensor Nodes

Lei Chen and Iain Bate
Department of Computer Science, University of York
Deramore Lane, Heslington, York, YO10 5GH, UK
lc920@york.ac.uk, iain.bate@york.ac.uk

***Abstract*— Body Sensor Networks (BSNs) are being used across a wider range of applications including healthcare ones where sensors may be attached to the body to sense certain properties including Electrocardiogram (ECG). The dependability of the systems is a key concern and is affected by the way in which it is used. For example, if the leads are loosely attached then the resulting signal will not be useful. It has been reported that the rate of such error is around 4% in the intensive care unit [8] when operating medical devices by trained professionals. The problem is made worse as the users of the systems are often not trained professionals. Some work has been performed on detecting anomalous signals. However, all of it has concentrated on anomalies caused by medical conditions (e.g arrhythmia). That is, to the best of our knowledge, no prior work has looked at anomalies caused by incorrect usage. In this paper a range of usage anomalies are defined in conjunction with a cardiologist and a lightweight algorithm is developed that achieves a high identification rate.**

Keywords: Dependability, ECG, Anomaly detection and identification

I. Introduction

It's been proposed that the healthcare service in the future may shift from centralised services (e.g stay in hospital) with expert caregivers (e.g doctor) to distributed services (e.g stay at home) with informal caregivers (e.g families, friends etc.) model [11]. Systems such as CodeBlue [9], ALARM-NET [11] and AMON [2] have been proposed to support the distributed healthcare services. The particular focus for this paper are BSNs featuring ECG sensors to capture longitudinal data for remote monitoring or later analysis. A key part of these BSNs systems are comparatively low cost compared to the systems used in hospitals.

For the data from all these systems to be useful, it is of paramount importance that the sensed signals have sufficient quality. These devices feature several sensor pads that are each connected to a node via a wire. For the signal to be of sufficient quality it is important that the sensors are attached in the correct place, that the sensors are securely connected with medical tape so that the signal noise is acceptable (e.g. static noise is below a certain level), and that the leads are then connected to the right port of the node. The correct use of the device is difficult enough for trained professionals. According to Rudiger et al. [8], between 0.4% and 4% of ECG signals which were recorded at the outpatient clinic and at the intensive care unit respectively are with erroneous usage (e.g misplacement). It is certain that for the average patient or informal caregiver the likelihood of erroneous usage increases.

To date, none of the available ECG-based remote monitoring systems advise the user when the system has either not been fitted correctly or has become incorrectly fitted through use. Equally, to the best of our knowledge, there are no other published papers that address the problem of identifying usage anomalies for BSNs. Existing works focus on the detection of anomalous signal caused by heart conditions. However, heart condition anomalies and usage anomalies share some similar features. Both type of anomalies affect the appearance of ECG signals in the time domain. As a result, existing work may serve as inspiration and comparison.

In [4], Keogh et al. proposed an algorithm called brute force discord detection (BFDD) and in their application case study, they applied BFDD to some ECG signals in the MIT-BIH Arrhythmia database [5]. They suggest that the BFDD is able to locate the anomalies. No further evaluation is carried out in terms of detection accuracy. Chuah et al. [3] evaluated BFDD showing that it can only detect between 40% to 70% of anomalies in the MIT-BIH Arrhythmia database. Chuah et al. then proposed another anomaly detection algorithm called adaptive window discord detection (AWDD) based on BFDD and they show that AWDD can achieve a detection accuracy between 70% and 90%.

In this paper, we define and demonstrate how the erroneous usage can corrupt the ECG signal, and the possibility of identifying the anomalous signals on a BSN mote at run-time so that the captured signals have sufficient quality. The contributions of the paper are:

1) In conjunction with a cardiologist, agree what types, and size, of usage anomalies can render the signal not to be useful to them especially for BSN devices
2) Design of a lightweight algorithm, which can operate on a BSN node, to detect when there is an anomaly and identify which type it is.
3) Evaluating the algorithm with real patient information from the MIT-BIH ECG Arrhythmia database to show its performance and resource efficiency

The rest of the paper is organised as follow. Section II will define and demonstrate the usage related anomalies in an ECG system. Section III will introduce the proposed algorithm for detecting and identifying the usage anomalies of an ECG based remote healthcare monitoring system. Section IV presents the evaluation. Finally, the work in this paper will be concluded in Section V.

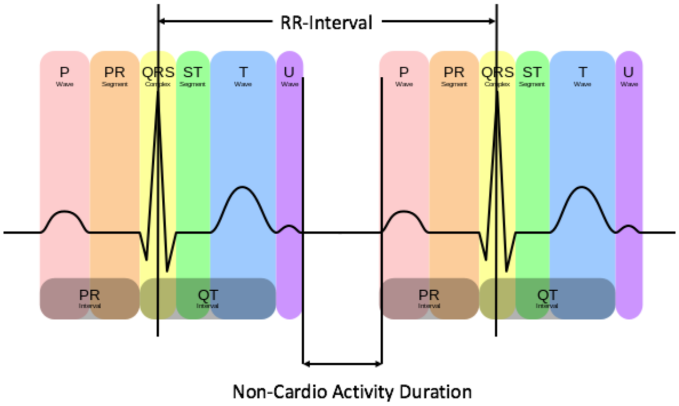


Fig. 1: Illustration of a typical ECG signal

II. Understanding the ECG Signal and Potential Anomalies

The purpose of this section is three fold: to understand what constitutes a correct signal; to understand what the features are of an incorrect signal affected by usage anomalies; and to introduce the ECG signal database that is used by this research.

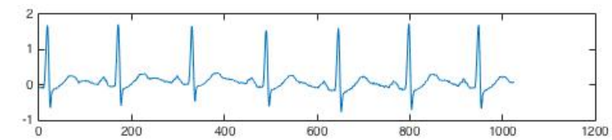
A. *The characteristics of a correct ECG signal*

An ECG is the electrical signal generated by the activity of the heart. In a normal medical grade ECG device, the signal is sampled at minimum of 250 Hz (e.g the signals in MIT-BIH Arrhythmia database are sampled at 360 Hz). It's composed of several segments which reflect different phases of heart activity. Figure 1 illustrates a typical ECG signal for two heart beats. The components of an ECG signal are defined as P-Wave, QRS-Complex, T-Wave, and U-Wave, although the U-Wave is not visible in 50% to 75% of ECG signals [10]. The duration after T-Wave and the next P-Wave (the signal without any annotation in Figure 1) is called the *Non-Cardio Activity duration*. As an ECG records the electrical signal of each heart-beat, the period of ECG signal is the interval between two heart-beats which can subsequently be converted to a heart-rate. Generally, the heart-rate is obtained through calculating the length of the *RR-Interval* which is the time interval between consecutive QRS-Complex peaks (R-Peak). The RR-interval's length may change depending on the subject's activity pattern. The characteristics of the components of the ECG signals remain sufficiently similar unless a heart condition occurs.

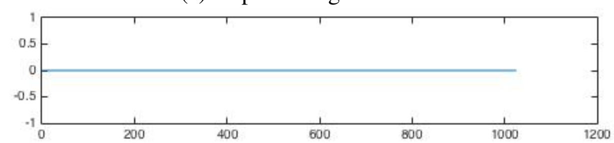
B. *The characteristics of an ECG signal affected by usage anomalies*

The most important aspect of this work is that the resulting signal is of use to a cardiologist. Therefore extensive consultations were held with a cardiologist to understand the ways in which the correct signal can be corrupted so that it was not usable and the characteristics of those incorrect signals. According to the cardiologist, the following three main types of anomalies may affect its usefulness:

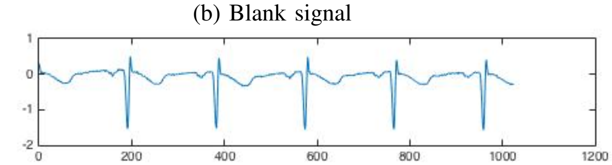
1) *No Signal* - This would be caused by the leads either not being attached or becoming loose. The characteristics of the signal are shown in Figure 2b. No ECG component is detectable under this situation.
2) *Inverted Signal* - This would be caused by the wrong lead being attached to the wrong port on the node. The characteristics of this signal are shown in Figure 2c. The QRS-Complex is visible but the top peak is the S-Peak instead of the R-Peak which becomes the lower peak.
3) *Noisy Signal* - This would be caused by the sensor not being correctly attached to the body, or the connection of the lead to the sensors or node being loose. The affects of a noisy signal are shown in Figures 2d and 2e which represent a minor and severe case respectively. The rest of this section discusses this anomaly as it is the most complex to detect.



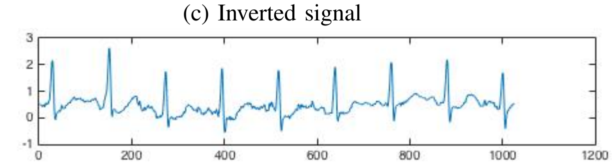
(a) Expected signal



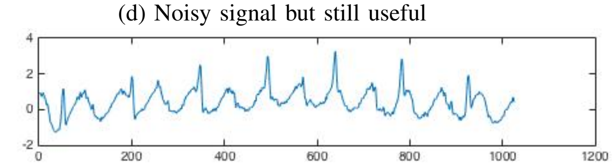
(b) Blank signal



(c) Inverted signal



(d) Noisy signal but still useful



(e) Noisy signal and not useful

Fig. 2: Signal illustration

When a signal is noisy, it doesn't necessarily mean the signal is useless. Cardiologist can tolerate a certain level of noise. According to our cardiologist, if the signal's components are visible, clear and the duration of each component can still be measured, the signal is still useful to them. For example, Figure 2d illustrates a period of signal that's noisy. In this signal, the general appearance of each RR-Interval is still clear enough and those key components are still measurable according to our cardiologist. On the other hand, the signal in Figure 2e is not useful according to our cardiologist. Although the R-Peak is still clear, the exact locations of QRS-Complex and T-Wave are no longer easily observable or measurable, which makes it hard or impossible to perform medical diagnosis.

C. MIT-BIH Database

ECG signals from MIT-BIH Arrhythmia database [5] is widely used by other researchers. The MIT-BIH database contains 48 half-hour ambulatory ECG signals which are obtained from 47 patients. These signals are sampled at 360Hz with 11-bit resolution. The signals in MIT-BIH Arrhythmia database have been professionally annotated although these annotations aim at heart conditions instead of usage errors.

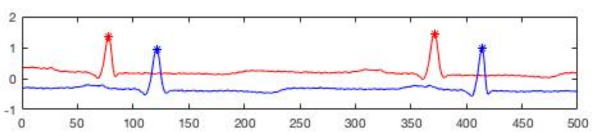
III. Algorithm Design

In this section, we propose a lightweight algorithm called *AID* which runs on a BSN mote and performs the usage anomaly detection at run-time. The algorithm can be considered in two parts: the approach for detecting the RR-peak and whether it is normal (section III-A), and the mechanism by which the algorithm is tuned (section III-B).

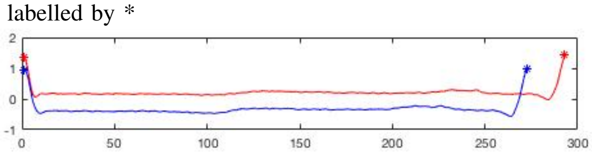
A. Detecting RR-Peaks And Whether They Are Normal

To better explain how *AID* works, following terms are defined for *AID*:

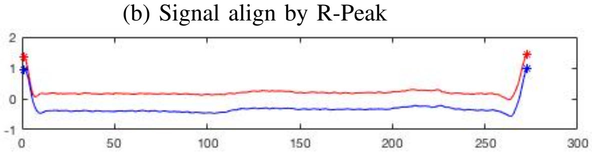
1) *Correct RR-Interval* - A correct signal as judged by a cardiologist
2) *New RR-Interval* - The new signal captured during the operation of the BSN.
3) *Predicted RR-Interval* - The predicted signal based on the *correct RR-Interval*. It's used to justify whether the *New RR-Interval* is correct



(a) Correct RR-Interval (Red) and New RR-Interval (Blue) with R-Peak labelled by *



(b) Signal align by R-Peak



(c) Size match via RR-Interval Prediction

Fig. 3: Algorithm illustration

The three main stages of the algorithm are as below and illustrated in Figure 3. The following sub-sections then details them further.

1) **Stage 1** - *R-Peak Detection* - The R-Peak is the most significant point of an ECG signal and it is the easiest ECG component to detect. Figure 3a shows the *Correct RR-Interval* (Red) and *New RR-Interval* (Blue) with the detected R-Peak labelled by $*$. *AID* then uses the first R-Peak of each signal to align the signal (as shown in Figure 3b). If the R-Peak can not be detected within a certain duration (set by cardiologist), it triggers the *No Signal* anomaly.
2) **Stage 2** - *RR-Interval Prediction* - The *Predicted RR-Interval* is generated by compressing or extending the Non-Cardio Activity duration of the *Correct RR-Interval*. As Non-Cardio Activity duration contains no heart activity signal, modifying the signal in this duration won't compromise the detail of ECG components. As shown in Figure 3b and 3c, the second R-Peak of the *Correct RR-Interval* in Figure 3b is at 293 whereas the second R-Peak of the *Predicted RR-Interval* (Red) in Figure 3c is at 273 which is the same as *New RR-Interval* (Blue).
3) **Stage 3** - *Signals Comparison* - Compare the *Predicted RR-Interval* and the *New RR-Interval* using correlation coefficient approach. As shown in Figure 3c, the *Predicted RR-Interval* (Red) and the *New RR-Interval* (Blue) have an amplitude difference. By using correlation coefficient to compare two signal, the amplitude difference won't affect the detection accuracy.

1) Stage 1 - R-Peak Detection: There are plenty of algorithms (e.g Pan Tompkins [7]),which can reliably detect the R-Peak under various condition, available. These algorithm require either domain transform or filters, which are very complex operation and won't fit on a mote.

In *AID*, a simple algorithm is used. For each new sample captured, the algorithm compares the difference to the previous sample. If the absolute difference exceeds T_R, it is the possible candidate for R-Peak. For each R-Peak candidate, the algorithm will measure the difference between its previous sample and its next sample. If the difference is below T_R, the R-Peak is confirmed.

By using this approach, the complexity is reduced compared to other existing approaches. When the captured signal is normal, it can detect the R-Peak reliably as the R-Peak is unique and most significant. When the *No Signal* anomaly happens, no R-Peak exists. As a result, it's expected that the algorithm doesn't report any R-Peak being detected. When the *Inverted Signal* anomaly happens, the highest peak becomes the lowest peak. As our algorithm compares the absolute difference, it will either report the highest peak (correct R-Peak) or lowest peak (inverted R-Peak) as long as they are significant. As a result, the algorithm will work under this situation.

When the *Noisy Signal* anomaly happens, depending on the strength of the noise, the algorithm behaves differently. When the noise is much weaker than the R-Peak, the peaks caused by the noise won't affect the detection due to the thresholds. As shown in Figure 2d, there are a lot of peaks but they are all much smaller than the R-Peak. On the other hand, if the noise is similar or stronger than the R-Peak (as shown in Figure 2e), the peaks caused by the noise may be reported as R-Peak by this algorithm. In other words, the algorithm may not be able to reliably and accurately detect the R-Peak under this condition. However, it's won't affect *AID* as the R-Peak unrecognisable is one of the features of *Noisy Signal*.

2) Stage 2 - RR-Interval Prediction: Inspired by the AWDD, *AID* performs RR-Interval Prediction by modifying the length of the correct signal and returns the modified signal as the *predicted signal*. In AWDD, the longer signal is sub-sampled to match the length of shorter signal. However, the sub-sampling process may compromise the detail of the ECG signal. In contrast, *AID* only processes the Non-Cardio Activity duration of each RR-Interval as it contains no information related to heart activity. By doing this, the predicted signal matches the length of captured signal without losing the useful information related to heart activity in an ECG signal.

For each *New RR-Interval*, the algorithm measures its length. When the *Correct RR-Interval* is shorter than the *New RR-Interval*, the algorithm will perform linear interpolation to the Non-Cardio Activity duration of the correct signal until the whole signal matches the length of the captured signal. On the other hand, if the *Correct RR-Interval* is longer than the *New RR-Interval*, the algorithm will shrink the correct signal by removing the samples from the Non-Cardio Activity duration. The modified signal is then used as the *Predicted RR-Interval*.

3) Stage 3 - Signal Comparison: AID uses the correlation coefficient to compare the time domain similarity between *Predicted RR-Interval* and *New RR-Interval*. More specific, Pearson's correlation coefficient is used in *AID*. As Pearson's correlation coefficient can normalise the difference in two signals' amplitude, the result will not be affected by the amplitude difference.

B. Tuning the Algorithm

There are three principal ways of tuning an algorithm: supervised, semi-supervised and un-supervised. In consultation with cardiologists, it was decided the supervised approach may not be suitable for ECG applications as it's hard to comprehensively build the training data set. An unsupervised approach may also be unsuitable. This is because when the usage anomaly appears, it is unlikely that it can go away without any user intervention. That means the majority data is anomalous. In which case, the unsupervised approach will ignore the usage anomalies. Therefore this work adopts *AID* a semi-supervised approach to perform the detection as it has the advantages of the supervised approach without the disadvantages of the un-supervised approach.

The tuning of the algorithm has two key parts:

1) **Part 1** - *Algorithm Setup* - Set up the *Correct RR-Interval* and parameters for *AID* by cardiologist as every cardiologist has their own tolerance of usage anomalies
2) **Part 2** - *Signal Classification* - The signal is classified based on threshold, T_c, which is set in the first step by the cardiologist according to their preference. Above the threshold means no anomaly is detected. Below the threshold means an anomaly is detected. Subsequently, each detected anomaly is identified based on its feature.

In the following sub-section, each part will be discussed in detail.

C. Part 1 - Algorithm Setup

AID needs to know what is the correct signal from this specific signal source and what level of noise is accepted by cardiologist. During the first time deployment, cardiologist can check the quality of the captured signal until the signal quality meets their requirement. A RR-Interval, which meets cardiologist's best quality requirement (e.g no noise, fully correct ECG signal), will be picked and stored on the mote as the *Correct RR-Interval*. *AID* can then measure the amplitude of the R-Peak and its adjacent samples of the *Correct RR-Interval* to set-up the threshold, T_R, for R-Peak detection. The control of acceptable level is done by setting up the correlation coefficient threshold T_c. The cardiologist can manipulate the deployment (artificially creating noise to the signal) until the signal quality meets their lowest requirement (the worst signal that a cardiologist can interpret). *AID* then calculates the correlation coefficient between the *Correct RR-Interval* and the minimum quality signal which is used as the threshold T_c for future detection.

D. Part 2 - Signal Classification

A threshold, T_c, is used to classify the captured signal. If the *New RR-Interval* is highly correlated to the *Correct RR-Interval* chosen in Stage 1, the signal is declared as anomaly free. On the other hand, if the *New RR-Interval* is uncorrelated to the *Correct RR-Interval*, it's most likely that it's the anomalous signal. By using this approach, the signal can be efficiently classified.

For each anomalous signal detected, the algorithm extracts the related features and matches with those anomalies' feature. For the *No Signal* anomaly, the key feature is that there is no R-Peak detectable. AID constantly measures the RR-Interval length. If the length of a RR-Interval exceeds a preset threshold, the signal is classified as *No Signal* anomaly.

If the correlation coefficient between predicted signal and captured signal is below a negative threshold, the two signals are inverse correlated. As a result, the captured signal is identified as an *Inverted Signal*. For all the captured signals that have no correlation to the *predicted signal*, they are identified as a *Noisy Signal* anomaly.

IV. Evaluation

In this research, the evaluations have been carried out in Matlab and on a mote. The purpose of the Matlab-based evaluation is to determine how reliably usage anomalies are detected and the mote-based evaluation uses Shimmer2r to determine whether the algorithm is practical from a resource perspective. *AWDD* has also been implemented in the Matlab-based evaluation to give some form of comparison.

A. Evaluation Metrics

To evaluate the detection accuracy, true-positive rate (abbreviated as TP), true-negative rate (abbreviated as TN), false-positive rate (abbreviated as FP), and false-negative rate (abbreviated as FN) are used as the evaluation metrics. TP is defined as the number of true positive detections divided by the total number of anomalies. TN is defined as the number of true negative detections divided by the number of correct signal. FP is defined as the number of false positive detections divided by the total number of signals. FN is defined as the number of false negative detections divided by the total number of signals. In addition, the anomaly identification accuracy is defined as the number of correct identified anomalies divided total number of true positive detections.

B. *Matlab Evaluation Setup*

In order to control the test environment, it is important to know exact location of where the anomaly occurs, and the type of the anomaly. The fault injection technique is used to construct the evaluation signals. By doing this, the evaluation metrics can be calculated accurately. Twenty signals from MIT-BIH Arrhythmia database [5] are manually picked as source signal, which have no or nearly no existing usage related anomalies. The usage anomalies are then injected into the source signals. Each type of anomalous signal will be injected into the source signal individually with the number of injected anomalies, anomaly location, and anomaly duration recorded. For each anomaly type, the simulation performs the detection for a minimum of 4,000 RR-intervals with a minimum of 200 anomalies being injected.

The following methods are used to obtain each type of usage anomaly signals. For the *No Signal* anomaly, it is captured by Shimmer2r [1] without attaching any electrode. The data capture parameters in Shimmer2r (e.g sampling frequency) are set exactly the same as in MIT-BIH database. *Inverted Signal* anomaly is obtained by inverting part of the signal from MIT-BIH database. For *Noisy Signal* anomaly, noise signals from MIT-BIH Noise Stress Test database [6] are used. As the noise strength can play an important role in the result, noisy signals with SNR of $-10dB$, $-6dB$, $0dB$, and $6dB$ are injected to the source signals.

C. *Matlab Simulation Result*

The result from Matlab simulation is shown in Table I. As *AWDD* doesn't have the ability to identify anomaly type, only the identification rate of *AID* is shown in the table.

When *No Signal* anomaly is injected, *AID* can improve the TP and TN rate by 66.67% and 5% respectively compared with *AWDD*. *AID* also reduces the FP and FN rate by 6.29% and 0.39% respectively. It can correctly identify all injected *No Signal* anomalies.

Similar improvement can also achieved when *Inverted Signal* anomaly being injected. *AID* improves the TP and TN rate by 80.07% and 12.86% respectively, and reduces FP and FN rate by 6.39% and 7.49% respectively compared to *AWDD*. The results show that *AID* is significantly better than *AWDD*. However, there are occasionally some false result (FP and FN). Through manually checking each incorrect detection, the causes are mainly due to the source signals' quality fall below the accepted level of our cardiologist.

For *Noisy Signal* anomaly, the results show that when the noise is much stronger than the signal, *AID* can achieve over 80% of TP rate whereas AWDD can only manage around 30%. When the noise strength decreased, the TP rate decreases an FN rate increases. In the extreme case when the noise is significantly weaker than the signal (6dB), the performance of *AID* falls behind the *AWDD*. An extra analysis has been carried out to investigate the false results. Through plotting those false results and manually checking the data quality one by one, it shows that most of the signals from the FN result are actually acceptable by cardiologist (confirmed by cardiologist). With the SNR increased, the signals' quality improved. As the signal's quality is accepted by cardiologist, *AID* does not categorise it as anomaly. This behaviour complies with the way in which the *AID* is designed to work. For the FP result, the investigation result is categorised into 4 groups (shown in Table II). The most causes of FP result is the wrong correct signal. In *AID*, when each new signal is categorised as correct signal, it will be used to replace the stored correct signal. However, as the threshold setting allows some tolerance, the signal error may accumulate with each update. As a result, the stored correct signal may become incorrect. By disabling the stored correct signal update, the result shows that it's able to remove those 88 FP result. Subsequently, the performance of *AID* is further improved. However, the long-term side effect of disabling the stored correct signal update is not investigated so far due to lack of long-term signal.

D. *On-mote Evaluation*

In this work, the on-mote evaluation mainly looks into whether limitations of motes (e.g. no floating point unit) will lead to a degradation in accuracy and the overhead in terms of computation, memory (both ROM and RAM), and firmware code size. The *AID* is implemented based on Contiki OS. To perform the on-mote evaluation for accuracy, 1000 seconds of signal from MIT-BIH#100 with anomalies injected are segmented into five hundreds of 2-second signals due to the on mote memory limitation. Each 2-second signal is then pre-loaded on the mote and run the test. The same data is also fed to Matlab. The same evaluation metrics to Matlab evaluation are used to compare the result from on-mote evaluation and Matlab evaluation.

In total, 1264 detections have been performed on mote and 621 anomalies have been injected at random location. Overall, *AID* achieved 98.60% TN rate, 98.71% TP rate, 0.63% FN rate and 0.71% FP rate. The same evaluation was also carried out in Matlab for comparison. The result is shown in Table III. For TN rate, FP rate and FN rate, the results from mote is comparable to the result from Matlab.

	Result from Matlab	Result from on-mote
True-Positive Rate	99.19%	98.71%
True-Negative Rate	99.53%	98.60%
False-Positive Rate	0.24%	0.71%
False-Negative Rate	0.4%	0.63%

TABLE III: Detection accuracy comparison between the Matlab-based evaluation and the Mote-based evaluation

Injected Anomaly		TP	TN	FP	FN	Recognition rate	Total RR-Interval	Total anomalies
No Signal	AWDD	33.33%	93.01%	6.81%	0.39%	-	4047	200
	AID	100%	98.01%	0.52%	0%	100%	4047	200
	Improvement	66.67%	5%	6.29%	0.39%	-	-	-
Inverted Signal	AWDD	18.45%	84.73%	7.02%	7.75%	-	4019	271
	AID	98.52%	97.59%	0.66%	0.26%	96.31%	4019	271
	Improvement	80.07%	12.86%	6.39%	7.49%	-	-	-
Noisy Signal (-10dB)	AWDD	31.91%	91.18%	6.25%	2.32%	-	4038	293
	AID	87.94%	94.96%	2.58%	0.77%	99.19%	4038	293
	Improvement	56.03%	3.78%	3.67%	1.55%	-	-	-
Noisy Signal (-6dB)	AWDD	32.33%	90.76%	6.66%	2.33%	-	4038	293
	AID	84.21%	94.84%	2.59%	0.9%	99.11%	4038	293
	Improvement	52.3%	4.08%	4.07%	1.42%	-	-	-
Noisy Signal (0dB)	AWDD	29.81%	90.11%	7.42%	2.44%	-	4038	293
	AID	51.92%	92.96%	2.64%	2.83%	100%	4038	293
	Improvement	22.11%	2.85%	4.78%	−0.39%	-	-	-
Noisy Signal (6dB)	AWDD	27.08%	89.69%	7.68%	2.72%	-	4038	293
	AID	9.38%	90.67%	2.65%	5.3%	100%	4038	293
	Improvement	−17.7%	0.98%	5.03%	−2.58%	-	-	-

TABLE I: Simulation results for each type of anomaly being injected

	Count	Total Detection Count
Heart Condition	18	2570
Wrong Correct signal	88	2570
Inaccurate R-Peak detection	9	2570
Noisy source signal	5	2570

TABLE II: The causes of FP detection

When compiling the firmware without the code of *AID*, the .ihex file size is 61.7kB. When integrating *AID* to the firmware, the compiled .ihex file increases to 64kB. That means it generates 2.3kB (3.72%) of overhead in terms of code size. During the run-time, depending on the heart rate (HR) and sampling frequency (Fs), the storage of new captured signal and correct signal requires $(HR/60 * Fs) * 2$ Bytes each. Assuming the lowest acceptable heart rate is 30 beats per minute (bpm) and the signal is sampled at 256Hz, The worst case RAM consumption of the algorithm will be around 2 kByte (20% of total RAM on Shimmer2r or T-Mote Sky). It's also measured that *AID* requires around 80 ms to check one RR-Interval whose typical duration is around 1 second (heart rate around 60 bpm). That means the *AID* can finish checking each RR-Interval before next RR-Interval arrived. As a result, the *AID* can perform detection on a BSN mote at run-time.

V. Conclusion & Future work

In this paper, we've proposed a lightweight anomaly detection algorithm *AID* for healthcare BSNs. Through the evaluation, *AID* has shown its ability to detect and identify those anomalies caused by usage that would affect the usefulness of the signals from the perspective of cardiologists. Meanwhile, through the complexity analysis of the algorithm, it's been shown that the proposed algorithm is lightweight and can be run on a mote at run-time.

References

[1] Shimmer sensing: http://www.shimmersensing.com/. Website.

[2] U. Anliker, J. A. Ward, P. Lukowicz, G. Troster, F. Dolveck, M. Baer, F. Keita, E. B. Schenker, F. Catarsi, L. Coluccini, A. Belardinelli, D. Shklarski, M. Alon, E. Hirt, R. Schmid, and M. Vuskovic. AMON: A Wearable Multiparameter Medical Monitoring and Alert System. *IEEE Transactions on Information Technology in Biomedicine*, 8(4):415–427, Dec. 2004.

[3] M. Chuah and F. Fu. ECG Anomaly Detection via Time Series Analysis. In *Frontiers of High Performance Computing and Networking ISPA 2007 Workshops*, volume 4743 of *LNCS*, pages 123–135. Springer Berlin Heidelberg, 2007.

[4] E. Keogh, J. Lin, and A. Fu. HOT SAX: efficiently finding the most unusual time series subsequence. In *Proceedings of the Fifth IEEE International Conference on Data Mining*, pages 8 pp.–233, Nov. 2005.

[5] G. B. Moody and R. G. Mark. The impact of the MIT-BIH Arrhythmia Database, May 2001.

[6] G. B. Moody, W. E. Muldrow, and R. G. Mark. A noise stress test for arrhythmia detectors. In *Computers in Cardiology*, pages 381–384, 1984.

[7] J. Pan and W. J. Tompkins. A Real-Time QRS Detection Algorithm. BME-32(3):230–236, Mar. 1985.

[8] A. Rudiger, J. P. Hellermann, R. Mukherjee, F. Follath, and J. Turina. Electrocardiographic artifacts due to electrode misplacement and their frequency in different clinical settings. *The American Journal of Emergency Medicine*, 25(2):174–178, Feb. 2007.

[9] V. Shnayder, B.-r. Chen, K. Lorincz, T. R. F. Fulford-Jones, and M. Welsh. Sensor Networks for Medical Care. Technical report, Division of Engineering and Applied Sciences, Harvard University, 2005.

[10] R. A. Wolthuis, V. F. Froelicher, A. Hopkirk, J. R. Fischer, and N. Keiser. Normal electrocardiographic waveform characteristics during treadmill exercise testing. *Circulation*, 60(5):1028–1035, Nov. 1979.

[11] A. Wood, G. Virone, T. Dona, Q. Cao, L. Selavo, Y. Wu, L. Fang, Z. He, S. Lin, and J. Stankovic. ALARM-NET: Wireless Sensor Networks for Assited-Living Residential Monitoring. Technical report, Department of Computer Science, University of Virginia, 2006.