Resource-efficient and Reliable Long Term Wireless Monitoring of the Photoplethysmographic Signal

Sidharth Nabar[*], Ayan Banerjee[†], Sandeep K.S. Gupta[†] and Radha Poovendran[*]
[*]Network Security Lab (NSL), University of Washington, Seattle, USA
[†]IMPACT Lab, Arizona State University, Tempe, USA
snabar@uw.edu, abanerj3@asu.edu, sandeep.gupta@asu.edu, rp3@uw.edu

ABSTRACT

Wearable photoplethysmogram (PPG) sensors are extensively used for remote monitoring of blood oxygen level and flow rate in numerous pervasive healthcare applications with diverse quality of service requirements. These sensors operate under severe resource constraints and communicate over an adverse wireless channel with human body-induced path loss and mobility-caused fading. In this paper, we take a generative model-based data collection approach towards achieving energy-efficient and reliable PPG monitoring. We develop two models that can generate synthetic PPG signals given a set of input parameters. These generative models are then used to design and implement a resource-efficient, reliable data reporting method for wireless PPG sensors. We investigate the performance of our method under realistic wireless channel error models and provide methods to improve accuracy at a marginal energy cost. We implement the proposed technique using existing sensor platforms and evaluate its performance on two datasets: MIMIC database and data collected using commercial wearable sensors. Results show significant bandwidth and energy savings due to data transmission reduction. The average reduction in data size for wearable sensor-based data is 300:1, while maintaining a diagnostic accuracy above 94%.

1. INTRODUCTION

Photoplethysmogram (PPG) is a non-invasive optical technique to measure changes in blood volume in a peripheral artery or tissue. It provides valuable information about the cardiovascular system and is widely used in several clinical applications such as pulse oximetry (SpO2), blood flow or pressure measurements and heart rate monitoring. Since PPG measurement requires only an LED and a photodetector, it is highly suitable for implementation with miniature, unobtrusive devices and is used in home health care as well as in hospitals.

Recently, several wireless, wearable PPG sensors have been developed, which continually collect PPG data and transmit it to a device such as laptop or smartphone, thus providing unobtrusive, continuous monitoring. Such PPG sensors are highly constrained in terms of available energy and onboard memory. Given that PPG is typically sampled at around 100 - 250 Hz to ensure clinical accuracy, a large amount of data is collected at the sensor. Locally storing this data is infeasible due to memory limitations, while transmitting it over the radio to a remote server consumes significant energy. Further, due to form factor requirements, PPG sensors use miniaturized antennas which makes their transmissions sensitive to errors caused by antenna-body coupling, radio wave scattering and signal attenuation occurring in the wireless medium around the human body. In spite of these resource consumption and transmission error issues at the sensor, a certain level of accuracy must be maintained in the PPG data reported to the physician. Thus, there is a need for a resource-efficient PPG monitoring technique that is robust to wireless channel errors, and maintains the accuracy of the reported PPG.

One approach towards resource efficiency in physiological signal monitoring is to use generative models. This approach uses a generative model of the physiological signal of interest at both, the server and the sensor. The server uses this model to generate a synthetic signal closely resembling the real physiological signal. The sensor transmits data updates whenever the model-generated signal does not match the sensed data, thus capturing temporal variations in the sensed signal. Such a model based data collection technique has been shown to achieve high energy savings for continuous ECG monitoring [17]. Using this technique for wireless PPG monitoring could enable significant energy and bandwidth savings, while maintaining the data accuracy. However, in order to achieve this, two research problems need to be solved: a) Development of a generative model for PPG signal, that can generate realistic synthetic PPG signals; and b) Investigating the performance of the model-driven monitoring scheme under wireless channel errors and adding error-handling methods to improve accuracy.

In this paper, we address the above mentioned challenges to achieve resource-efficient and reliable wireless PPG monitoring. In doing so, we observe that the PPG signal is a manifestation of the blood pumping action of the heart and the behavior of the human vascular system. Further, there have been attempts in literature to analytically model the vascular system and relate it to the PPG waveform [10]. We use these models as a starting point for developing generative models for PPG. Further, we enhance the basic generative model-based scheme by adding well-understood wireless communication techniques to handle transmission errors.

Our main contributions in this paper are:

- We develop two generative models for PPG: a differential equation-based model, denoted as *DE-PPG* and a template-based model called *Tem-PPG*. The *Tem-PPG* model has much lower complexity but still achieves diagnostic accuracy of PPG comparable to *DE-PPG*, at the cost of a nominal increase in memory consumption.

Wireless Health '11, San Diego, USA
Copyright 2011 ACM 978-1-60558-989-3 ...$10.00.

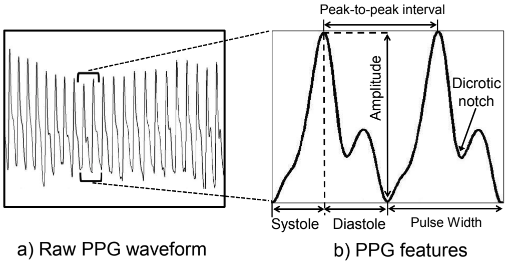


Figure 1: A typical PPG waveform with baseline variation is shown in a) and b) shows salient features of a PPG pulse

- Based on the *Tem-PPG* and *DE-PPG* models, we design and implement a PPG monitoring method that achieves high energy efficiency, while preserving the diagnostic accuracy of the PPG signal.
- We evaluate the performance of our proposed PPG monitoring technique under realistic wireless channel models and develop methods to handle transmission errors. These methods improve the reliability of PPG reporting, at a marginal increase in energy consumption.
- We implement the proposed method on existing sensor platforms to verify its feasibility, and evaluate its energy consumption and accuracy on high-quality real-life PPG data from the MIMIC database, as well as data obtained using wearable PPG sensors. The use of wearable sensor-based data validates the performance of the proposed approach in presence of unpredictable artifacts and noise.

The rest of the paper is organized as follows. Section 2 presents background and related work. Section 3 describes the generic model driven data collection approach used in this paper. The proposed generative models *DE-PPG* and *Tem-PPG* are presented in Section 4. Section 5 describes the proposed PPG monitoring method and Section 6 presents evaluation results. In Section 7, we discuss ways to improve the monitoring performance in the presence of wireless errors. Section 8 concludes the paper.

2. BACKGROUND AND RELATED WORK

A typical PPG waveform, shown in Figure 1 a), contains a high frequency component, called pulsatile component, superimposed on a low frequency baseline variation. The fundamental frequency of the pulsatile component depends on the heart rate and is generally around 1 Hz. The baseline variation is due to respiration, thermoregulation and other physiological factors [3]. Most diagnostic applications based on PPG filter out the baseline variation and use only the pulsatile component [21]. As a result, in this paper, we focus only on the pulsatile component of PPG.

Each pulse of the PPG signal, as shown in Figure 1 b), contains a rising edge corresponding to the systole (contraction of ventricles) and a falling edge corresponding to the diastole (relaxation of ventricles). These edges are hence referred to as the *systolic edge* and *diastolic edge* respectively in this paper. The diastolic edge often has a notch, called dicrotic notch, which is indicative of healthy arteries [3]. A single PPG pulse is characterized by its height, width and shape. These are called the morphology features of PPG, and are expected to remain fairly constant over short time scales for a given person [16]. The temporal characteristics of PPG are studied using the peak-to-peak intervals among pulses. These intervals have been shown to correspond directly to the inter-peak intervals of ECG [22], and hence can be characterized using well-known ECG inter-beat features [14]. These include temporal features such as mean and standard deviation of heart rate and frequency domain features such as LF/HF ratio. In this paper, we consider these three features as inter-pulse features of PPG. The morphology and inter-pulse features together completely characterize a PPG signal.

In existing literature, the relation between the PPG waveform and the vascular system has been modeled using Windkessel models [10]. Their equivalent circuit representations have been developed, where the circuit elements represent properties of the vascular system, while the current represents the PPG signal. The *DE-PPG* model developed in this paper is derived from such circuit representations.

2.1 Related Work

The use of PPG in 24 hour remote monitoring applications has led to the development of several low form factor wearable sensors which measure PPG in an unobtrusive manner [5, 12, 15]. Long term monitoring applications require these sensors to be thrifty in power usage, with recent endeavors towards their operation using harvested energy [13]. In such an energy constrained scenario, periodic transmission of the entire sensed data imposes a major problem since it implies frequent usage of the power hungry radio.

To address this issue, data compression schemes based on compressive sensing [6] and Fourier analysis [20] have been proposed for PPG. These schemes decrease energy consumption by reducing the amount of data transmitted by the sensor. However, data is still transmitted periodically, thus limiting the energy and bandwidth savings. Such periodic transmission is unsuitable for sensors running on scavenged energy, which have intermittent availability of energy. Further, each sensor transmission is susceptible to errors such as random bit errors, fading, and burst errors that commonly occur in on-body wireless channels [4, 9, 26]. Thus, the main challenge in this regard is to minimize sensor transmissions while maintaining the clinical accuracy of the reported PPG signal.

3. GENERATIVE MODEL BASED PHYSIOLOGICAL DATA MONITORING

Generative models are parametric mathematical models that can generate a signal given a set of input parameters. Such models have previously been used in several diverse areas such as Wireless Sensor Networks [11] and music [8]. In our prior work [17], we demonstrated the use of such models for resource-efficient physiological data collection, through an ECG monitoring application. We now describe the basic generative model-driven data reporting methodology and outline the challenges involved in using such a technique for reliable wireless monitoring of PPG.

The overall architecture of the generative model-based data reporting method consists of a server module, referred to as M_S, and a lightweight sensor module denoted as M_{LITE}. M_S consists of a generative model $\mathcal{G}$ of the physiological signal being monitored. This generative model is used at the physician end to generate a synthetic signal from a set of input parameters corresponding to characteristic features of the signal. The sensor module M_{LITE} is designed for wearable sensors and contains a lightweight version of $\mathcal{G}$. It also includes algorithms to compare the actual sensed signal with the synthetic one generated by $\mathcal{G}$.

The initialization of the system includes training the model $\mathcal{G}$ in M_S. This training process consists of measuring a patient's real physiological data and using it to derive suitable input parameter values for $\mathcal{G}$. These values are stored on the sensor as well as the server and are used to generate a waveform closely matching the patient's real physiological signal. Once the initialization is complete, the sensor can be deployed for use by the patient. During the normal system operation, the sensor periodically compares the measured physiological signal to that generated by $\mathcal{G}$. It transmits data updates to the server only when the deviation between these two signals exceeds a certain pre-defined threshold. Such signal comparisons and data updates are of two types:

- Feature comparison and updates: The sensor derives representative features from the sensed signal and compares them to the parameters of $\mathcal{G}$. If the values differ significantly, the parameter values of $\mathcal{G}$ in M_{LITE} are updated, and transmitted as *feature updates*. These updates are used to accordingly update the M_S version of $\mathcal{G}$. This approach is suitable for signal features that can be calculated on the computationally limited sensor.

- Raw signal comparison and updates: For some other parameters of $\mathcal{G}$, the corresponding features may be hard to compute on the sensor. In this case, the sensed signal is directly compared to the signal generated by $\mathcal{G}$ in M_{LITE}. If the mismatch exceeds a given threshold, the raw signal is transmitted to the server, which records it as the patient's real data. Such updates are referred to as *raw signal updates* and are used by the server to derive new parameter values for $\mathcal{G}$. These values are then transmitted back to the sensor to update its model. Raw signal updates typically contain a large number of signal samples and hence are significantly longer and consume more energy than feature updates.

Thus, the model $\mathcal{G}$ is continually updated to capture the temporal variations in the sensed physiological signal. As a result, the model-generated signal closely follows the patient's real data in terms of the diagnostic content. At the server, if raw data is received from the sensor, it is recorded as the patient's real data. For the remaining time intervals, the model $\mathcal{G}$ is used to generate a synthetic waveform. These two signals are then temporally aligned and constitute the final output signal available to the physician or caregiver for analysis. This technique was used for ECG monitoring in our prior work [17], where an existing generative model (ECGYSN [14]) was used as $\mathcal{G}$. Evaluation with real ECG data from MIT-BIH [1] database showed significant energy and memory savings at a minimal loss in diagnostic accuracy. However, the effects of wireless channel errors or wearable sensor-induced signal artifacts on the performance of the method were not investigated.

Generative model-based wireless PPG monitoring: Developing a generative model-based technique for long-term wireless PPG monitoring could significantly reduce data transmission at the body-worn sensor, thus providing bandwidth and energy savings. Memory consumption can be reduced by storing PPG data in the form of model parameters rather than actual signal samples. The stored parameter values can be used at a later instant as inputs to $\mathcal{G}$ to recreate the signal for the corresponding time interval. Further, the key diagnostic features of PPG as well as a reconstructed PPG waveform are available at the physician end, thus enabling the use of various automatic PPG analysis tools as well as visual inspection by the physician. However, in order to use this technique for wireless PPG monitoring, the following three key challenges must be overcome:

1. *Developing a generative model for PPG:* The main prerequisite for using the above data reporting technique for PPG monitoring is the existence of a generative model $\mathcal{G}$ for PPG. Such a model must be able to generate a synthetic PPG signal from a given set of feature values as input parameters. Further, a learning technique must be available to learn suitable input values for a given patient's PPG signal. To the best of our knowledge, no such model currently exists in literature.

2. *Handling wireless channel errors:* The model-driven method depends on error-free transmission of feature updates and raw signal updates from the sensor to the server in order to capture temporal variations in the patient's PPG. However, the wireless medium in which body-worn PPG sensors operate is highly adverse, with path loss caused by human body and mobility-induced fading. Further, since these sensors typically operate in the unlicensed 2.4 GHz band, their transmission suffers from high bit error rates due to interference with other wireless devices using 802.11 or Bluetooth. In order to ensure consistent clinical accuracy of the PPG reported to the physician, it is necessary to improve the basic generative model-based technique, by incorporating methods to handle wireless channel errors.

3. *Verifying performance with wearable sensor-based data:* The use of wearable sensors to collect physiological data typically introduces undesired artifacts and noise into the sensed signal. This can lead to mismatch between model-generated signal and sensed data, causing unnecessary signal transmissions at the power-constrained sensor. Thus, along with verifying the correctness of the data reporting method on clean, medical grade data, it is also necessary to evaluate its performance on wearable sensor-based data.

In the following sections, we focus on addressing the above challenges, and develop a resource-efficient PPG reporting method that is robust to typical wireless channel errors.

4. GENERATIVE MODELS FOR PPG

We present two generative models for PPG: a differential equation model, which is based on human vascular physiology, thus representing the true origin of the PPG signal; and a simplified template-based model, which interprets the PPG as a series of individual pulses, each having a certain shape. For each model, we describe the set of input parameters, the process of synthetic PPG generation and the method used to learn parameter values from a given raw PPG signal as input. As discussed in Section 2, the models focus only on the pulsatile component of PPG.

4.1 Differential Equation PPG (DE-PPG)

This model characterizes the shape of a PPG pulse using differential equations, and is based on a Windkessel model of the human vascular system [10]. In these models, the vascular system is represented as an electrical circuit, with the left ventricular pressure (LVP) as input voltage and the PPG waveform as the output current. Solving for the current in this circuit representation gives the relationship between the LVP wave and the PPG waveform in the form of a differential equation. In [10], the LVP input is assumed to be zero during diastole, and a Laplace transform representation for the resultant circuit is derived. Solving for the output current gives the following time domain equation for PPG during diastole:

$$PPG_{dias}(t) = a_1 + a_2 e^{-a_3 t} + a_4 e^{-a_5 t} cos(a_6 t + a_7)$$

In order to effectively model the dicrotic notch, we modify the last term to include a logistic function term, thus giving the final equation as:

$$PPG_{dias}(t) = a_1 + a_2 e^{-a_3 t} + \frac{1}{a_4 + e^{(-a_5 t - a_6)}} cos(a_7 t + a_8) \quad (1)$$

For the systole, an analytical driving LVP waveform is consid-

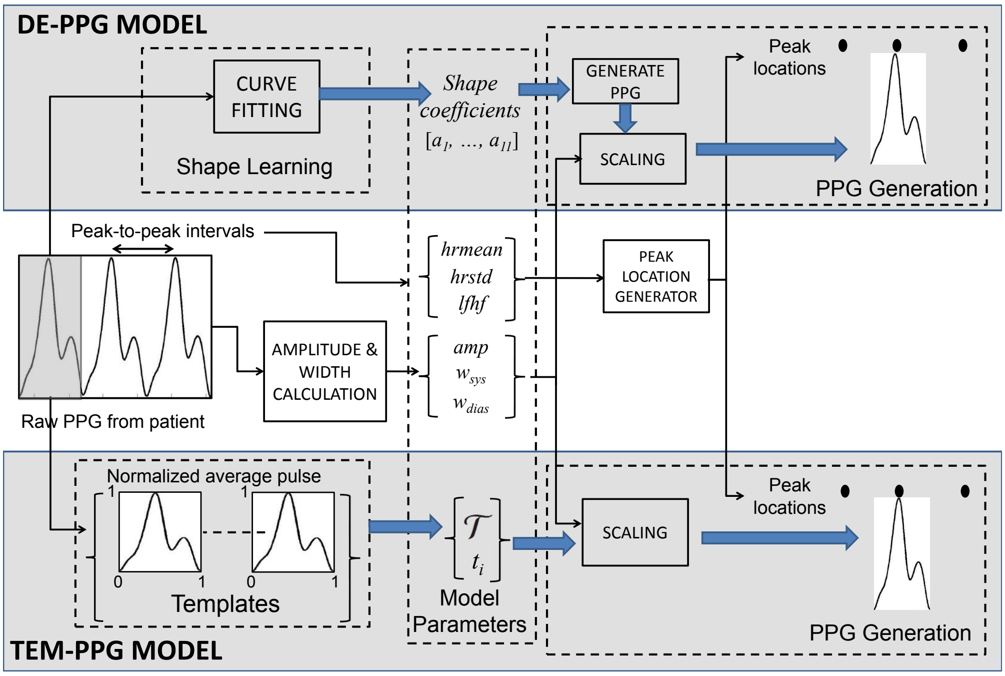


Figure 2: Illustrations of the model learning method, input parameters and PPG generation process for the *DE-PPG* and *Tem-PPG* models. The unshaded region represents extraction of inter-pulse and scaling features from PPG and is common to both models.

ered unknown in [10], and was approximated using ten sinusoidal harmonics in [23]. Using such an input for LVP would lead to a temporal equation with 36 coefficients, which would be very hard to learn. Instead, as an approximation, we represent the systolic edge of the PPG as a single logistical function, as :

$$PPG_{sys}(t) = \frac{1}{a_9 + e^{(-a_{10}t - a_{11})}} \qquad (2)$$

This approximation agrees very well with real PPG data, as discussed in Section 6. The coefficients [a_1, a_2, ... , a_{11}] from (1) and (2) are used to model the PPG pulse shape in the *DE-PPG* model. We now discuss the remaining input parameters, synthetic PPG generation process and learning methodology for the *DE-PPG* model. These are illustrated in Figure 2.

Input parameters: The input parameters of the *DE-PPG* model are divided into two groups: (a) Inter-pulse parameters and (b) Morphology parameters, corresponding to inter-pulse and morphology features respectively, as discussed in Section 2. For the inter-pulse parameters, we use the mean heart rate (denoted as *hrmean*), standard deviation of heart rate (*hrstd*) and the LF/HF ratio (*lfhf*) features. For pulse morphology, we consider the systolic and diastolic phases separately. Accordingly, we use the systolic width, denoted by w_{sys}, and the diastolic width (w_{dias}) as separate input parameters. However, the amplitude is assumed to be equal for both the phases, and is represented by a single amplitude parameter *amp*. The shapes of the systolic and diastolic phases are characterized by the two temporal equations shown in (1) and (2). The coefficients [$a_1, a_2, \ldots, a_{11}$] completely determine the pulse shape and are considered as the shape-related input parameters. Thus, the final set of input parameters for the *DE-PPG* model is: $\{hrmean, hrstd, lfhf, amp, w_{sys}, w_{dias}, [a_1, a_2, \ldots, a_{11}]\}$.

PPG generation: The first step in creating a PPG signal using the *DE-PPG* model is to use the *hrmean*, *hrstd* and *lfhf* parameters to generate a peak-to-peak process which gives time instants corresponding to the occurrence of pulse peaks in the PPG. This is performed using a method proposed in [14], where the peak-to-peak process of ECG was calculated from these parameters[1]. The coefficients [a_1, ..., a_{11}] are used, along with Eq. (1) and (2) to obtain the systolic and diastolic edges of a single pulse. Each edge is horizontally scaled by the respective widths (w_{sys} and w_{dias}) and concatenated to form a single pulse. This pulse is then scaled by the specified amplitude (*amp* parameter) and multiple copies are stitched together to form the final output signal.

Learning methodology: The learning methodology describes how the *DE-PPG* model learns a set of input parameter values from a given raw PPG signal. The first step is filtering the signal to eliminate the low frequency baseline drift and extract the pulsatile component. Then, the peaks and troughs of the signal are detected, which correspond to the pulse peaks and the pulse endpoints respectively. From a set of peak-to-peak intervals, the *hrmean* and *hrstd* parameters are obtained through averaging and standard deviation calculations. To obtain the *lfhf* parameter (LF/HF ratio), a set of 128 peak-to-peak interval values is obtained and the Power Spectral Density (PSD) of this set is computed. The Low Frequency (LF) and High Frequency (HF) components are obtained by integrating this PSD over (0.04Hz - 0.15Hz) and (0.15Hz - 0.4Hz) respectively. The ratio between these components gives the value of the *lfhf* parameter.

The PPG signal is then segmented into individual beats using the endpoints detected during peak detection. The average height for a set of consecutive pulses is computed as the amplitude parameter. The average widths of the systolic and diastolic edges are set as w_{sys} and w_{dias} respectively. Finally, for learning the shape, a single average pulse is obtained, and then normalized to unit amplitude and unit width. Least squares error curve fitting is performed for the systolic and diastolic edges of this normalized pulse to obtain

[1] As discussed in Section 2, the peak-to-peak process of PPG is identical to that of ECG.

the values of the coefficients $[a_1, \ldots, a_{11}]$.

Although the *DE-PPG* model completely characterizes a PPG signal using a small set of input parameters, it requires complex curve fitting techniques to learn the coefficients $[a_1, \ldots, a_{11}]$ for a given pulse shape. Although accurate, this leads to excessive computation and energy consumption at the server. As a result, we also develop a simpler model, which is based on using a template approach to characterize PPG pulse shape.

4.2 Template-based PPG (Tem-PPG)

This model is based on the observation that the shape of a pulse in the PPG waveform is not expected to vary significantly within short time scales [16]. As a result, a fixed set of shape templates is used as a lookup table or codebook to characterize the shape of individual pulses in a PPG signal. This eliminates the need to perform curve fitting, hence reducing model learning complexity.

Input parameters: The input parameters of the *Tem-PPG* model are the same as *DE-PPG*, except for the pulse shape coefficients. In *Tem-PPG*, the shape of a pulse is characterized using a set $\mathcal{T}$ of N diverse shape templates. Each template is a set of samples representing a pulse normalized to unit amplitude and unit width. A template index t_i $(1 \leq t_i \leq N)$ is used to reference a given template in the set $\mathcal{T}$, and acts as a shape-related input parameter to the *Tem-PPG* model. Thus, the input parameters to the *Tem-PPG* model are: $\{hrmean, hrstd, lfhf, amp, w_{sys}, w_{dias}, t_i, \mathcal{T}\}$.

PPG Generation: To generate a PPG signal using the *Tem-PPG* model, we first use the *hrmean*, *hrstd* and *lfhf* parameters to determine the pulse peak locations as in the case of the *DE-PPG* model. The template index t_i is then used to select a template from set $\mathcal{T}$, and this template is scaled according to the amplitude and width parameters to form individual pulses. These pulses are then aligned so that their peaks coincide with the peak occurrence time instants. Finally, the signal is stitched together by connecting the end of each pulse to the start of the next.

Learning methodology: As shown in Figure 2, the learning methods for parameters *hrmean*, *hrstd*, *lfhf*, *amp*, w_{sys} and w_{dias} for *Tem-PPG* are the same as described in Section 4.1 for the *DE-PPG* model. For learning the shape, an average pulse is obtained and then normalized to unit amplitude and width. Its correlation with each template in $\mathcal{T}$ is then computed, and the index corresponding to the maximum correlation template is chosen as the template index t_i. If the correlation with all templates is below a predefined threshold, the normalized average pulse is added to the set $\mathcal{T}$ as a new template.

The learning and PPG generation operations for both models are performed on a moving window of the raw PPG signal. As a result, the model parameter values and the corresponding synthetic PPG change over time, thus capturing the temporal variations in the original PPG. The above-mentioned models are mainly intended for use in the data monitoring technique described in Section 3, but can also be used independently to generate realistic PPG signals. This could enable other applications such as testing and validation of medical devices or signal processing algorithms intended for PPG analysis.

5. MODEL-BASED PPG MONITORING

In this section, we describe a model-based PPG data collection method based on using the *Tem-PPG* and *DE-PPG* models with the data monitoring technique of Section 3.

5.1 DE-PPG based Monitoring

The initialization of the data monitoring scheme comprises of training the model $\mathcal{G}$ using the patient's real PPG data. For the *DE-PPG* model, this is performed using the learning methodology described in Section 4.1. The derived values of the input parameters, including the coefficients $[a_1, \ldots, a_{11}]$, are stored in the server as well as the sensor. The sensor uses the coefficients to generate a systole and a diastole curve, combines them to form a single pulse and stores it in memory as *ModelPulse*. The sensor is then deployed for use by the patient. The overall operation of the proposed scheme is summarized in Figure 3.

5.1.1 Sensor Module

During regular operation of the system, the sensor samples the PPG at a predefined sampling rate F_s and stores it into a B second long buffer. Once the buffer is full, the stored PPG samples are sent to the processing stage, which involves filtering, peak detection and pulse extraction. We use a high-pass IIR filter with cutoff frequency of 1 Hz to eliminate low-frequency baseline variations. The filtered PPG is passed to a peak detection algorithm which detects peaks and troughs in the signal, corresponding to pulse peaks and pulse endpoints respectively. The pulse endpoints are used to segment the signal into individual pulses.

Feature comparison and updates: The sensor uses the feature comparison and update approach discussed in Section 3 for the *hrmean*, *hrstd*, *lfhf*, *amp*, w_{sys} and w_{dias} parameters. The mean and standard deviation of the heart rate are calculated using averaging and standard deviation calculations on peak-to-peak intervals. The LF/HF ratio is computed as discussed in Section 4.1, and the required Fourier transform is performed using an efficient TinyOS implementation developed in our prior work [7]. For calculating the amplitude and width-related features, multiple pulses are averaged to form a *MeanPulse*, and its height, systolic width and diastolic width are measured. Once all the features are calculated, they are compared to the corresponding *DE-PPG* parameters (Figure 3). If the difference between the measured feature values and the model parameters exceeds predefined threshold values, the corresponding model parameters are updated. These updated values are then timestamped and transmitted to the server as feature updates.

Raw signal comparison and updates: Since deriving the shape coefficients $[a_1, \ldots, a_{11}]$ requires curve fitting and cannot be performed at the computationally constrained sensor, the raw signal comparison approach is used for the pulse shape. Following the amplitude and width extraction, the *MeanPulse* is normalized to a unit height and width. This normalized pulse is then compared to the pre-stored *ModelPulse* using a cross-correlation metric, as shown in Figure 3. If the correlation falls below a given threshold, the sensor transmits the entire signal buffer ($B \cdot F_s$ samples) to the server as a raw signal update.

In the absence of any updates, the sensor transmits short HELLO packets to the server once every minute. This helps the server differentiate between sensor failure and absence of updates.

5.1.2 Server module

The server uses the *DE-PPG* model with input parameter values learned during the initialization phase to generate a PPG signal, as described in Section 4.1. Whenever feature updates are received from the sensor, the corresponding parameters are modified and the new values are used for PPG generation. When a raw signal update is received, it is considered as the patient's true PPG and is used to overwrite the model-generated PPG for the corresponding time interval (*Align* block in Figure 3). Further, if multiple raw signal updates are received, it indicates that the PPG pulse shape has deviated from the initial model. The server then uses the received PPG data to learn new coefficient values and transmits these values back to the sensor. The sensor uses these values to generate a new

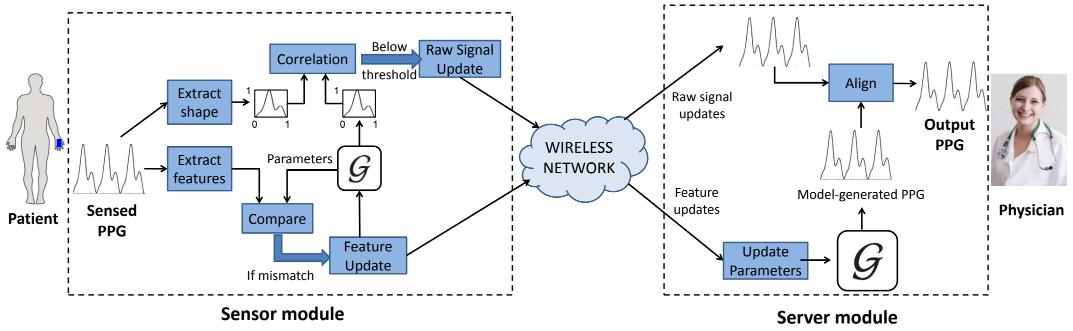


Figure 3: Illustration of the proposed PPG monitoring scheme with *DE-PPG* as the generative model $\mathcal{G}$. The overall architecture is similar for the *Tem-PPG* model, except for the shape comparison, which involves multiple templates stored at the sensor.

ModelPulse. Thus, the *DE-PPG* models at the server and at the sensor are continually updated to capture the temporal variations in the patient's PPG signal.

5.2 Tem-PPG based Monitoring

The overall system operation of the PPG monitoring scheme using the *Tem-PPG* model is the same as the one shown in Figure 3 for the *DE-PPG* case, except for the pulse shape characterization. For the *Tem-PPG* model, during the initialization phase, a set of templates $\mathcal{T}$ are learned from the patient's PPG data, and stored at the server as well as at the sensor. Each template is normalized to unit amplitude and a width of 1 second (F_s samples). The index of the most frequently occurring template is stored as the template index parameter t_i. If the sensor does not have enough memory, only the most frequently occurring template is stored on the sensor. The remaining model parameters are learned as described in Section 4.2 and stored at the server as well as on the sensor.

The sensor module operation for the *Tem-PPG* model is the same as described in Section 5.1.1 for *DE-PPG*, in case of the *hrmean*, *hrstd*, *lfhf*, *amp*, w_{sys} and w_{dias} model parameters. For the shape comparison, the normalized *MeanPulse* (Section 5.1.1) is compared to the template with index t_i. If the correlation is below the threshold, all the other templates are tried, and the one with the highest correlation is selected. Its index is then set as t_i and sent to the server as a feature update. If only one template is stored on the sensor, the sensor compares the stored model template and the normalized *MeanPulse* and sends a raw signal update to the server if the correlation falls below a given threshold.

At the server, the *Tem-PPG* model is used to generate a PPG signal as described in Section 4.2. When feature updates are received from the sensor, the corresponding model parameters are updated. If only one template is stored on the sensor, the server receives raw signal updates when the patient's PPG does not match that template. It then compares the received signal to all the templates in $\mathcal{T}$ and chooses the one with the highest correlation. This template is transmitted to the sensor to update its locally stored copy.

We note that in the *Tem-PPG* based data collection method, storing the entire set of templates or a single template on the sensor is a design choice. Storing the entire set $\mathcal{T}$ consumes memory but enables reduction in data transmission by replacing a raw signal update ($B \cdot F_s$ signal samples) and a subsequent server update (F_s samples) by a single feature update (one parameter value). Thus, this design choice allows a tradeoff between storage memory and energy consumption. In our experiments, the size of the set $\mathcal{T}$ was found to be small enough to be stored in existing sensor platforms. As a result, we store the entire set of templates on the sensor, thus achieving higher energy savings.

Task	Execution Time (s)	Energy Consumption (mJ)
Generate a pulse from coefficients $[a_1, \ldots, a_{11}]$	7.13	106.95
Filtering	3.49	52.35
Peak Detection	0.21	3.15
Amplitude, systole and diastole width calculation	0.005	0.075
Normalize average beat, perform correlation with *ModelPulse*	0.59	8.85
Calculate *hrmean*, *hrstd*	0.058	0.897
Calculate LF/HF ratio	0.318	4.77

Table 1: Execution Time and Energy Consumption for Sensor Module Tasks on TelosB platform

6. PERFORMANCE EVALUATION

We evaluate the proposed scheme in terms of its feasibility, accuracy and energy savings. To verify the feasibility, we implement the sensor module on existing sensor platforms and investigate its power consumption and execution time. The accuracy evaluation includes the learning accuracy for *DE-PPG* shape coefficients, as well as the diagnostic accuracy of the PPG signal reported at the physician's server. For energy savings, we focus only on radio communication, since it is the main factor in energy consumption at the sensor. The data used for evaluation consists of: (a) 10 patients' data from MIMIC database [1]; and (b) Impact dataset, containing 10 patients' data collected using commercial wearable sensors, as a part of this work. The PPG signals in both datasets are sampled at F_s = 125 Hz, and the buffer size at the sensor is B = 5 sec.

Sensor module implementation: The sensor modules for both *DE-PPG* and *Tem-PPG* were implemented in TinyOS and tested on TelosB motes. The measured execution time and energy consumption values for each task are given in Table 1. The processing and feature calculation tasks occur once every 5 seconds, while the pulse shape generation task occurs only once, on initialization of *DE-PPG* shape coefficients. We note that this is a proof of concept implementation to verify feasibility, and operations such as filtering

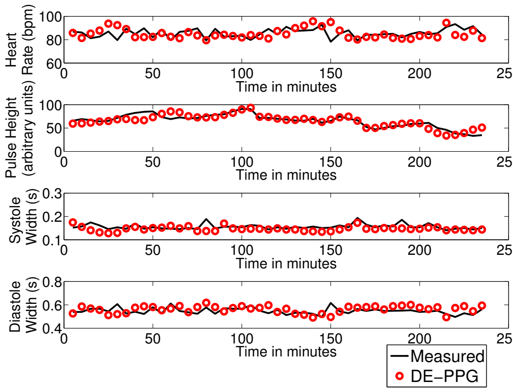


Figure 4: Feature errors over time for *DE-PPG* on a data snippet from MIMIC database

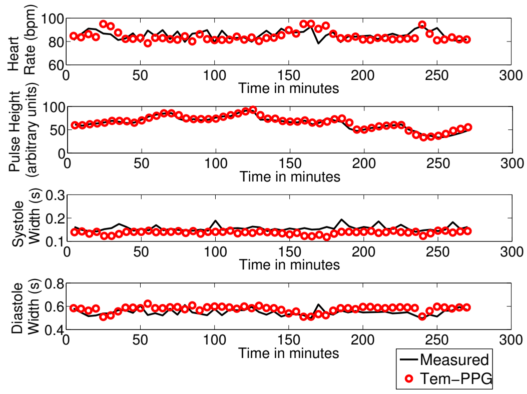


Figure 5: Feature errors over time for *Tem-PPG* on a data snippet from MIMIC database

are better suited for implementing in hardware.

Pulse shape learning accuracy: In this case, we only consider the *DE-PPG* model, since the *Tem-PPG* does not learn any coefficients from a given raw PPG. The learning function was implemented in MATLAB and curve fitting was performed using the *lsqcurvefit.m* function. Starting from a raw PPG, the learning function identifies a minimal set of representative beats from the signal. This set is chosen such that all the remaining beats in the signal are highly correlated (correlation > 0.99) with at least one beat from this set. The coefficients $[a_1, \ldots, a_{11}]$ from Eq. (1) and (2) are learned on this training set, and are used to generate a synthetic PPG pulse, which is compared to the original raw PPG. The average learning accuracy over both datasets was found to be 97.4%.

Diagnostic accuracy of reported PPG: Based on typical clinical applications of PPG, we consider the heart rate estimate, pulse height, systolic width and diastolic width as the key diagnostic features of PPG [21]. The accuracy of the proposed method is calculated by comparing these features over time for the original PPG of the patient and the model-generated PPG at the server[2]. For example, Figures 4 and 5 show such a comparison for Patient 212 in the MIMIC database, for the *DE-PPG* and *Tem-PPG* models respectively. The feature errors averaged over all patients in the MIMIC and Impact datasets are shown in Table 2. We observe that the errors are similar for both models and are within accepted limits [6].

Energy savings: Since the communication energy consumption at the sensor is proportional to the amount of data being transmitted, we measure the energy savings in terms of the reduction in the data sent by the sensor to the server. We define this as the Compression Ratio (CR) to enable comparison with existing compression schemes. Table 2 shows the average compression ratio achieved by the proposed method over the two datasets for the *Tem-PPG* and *DE-PPG* models. Along with energy savings, this compression ratio also represents equivalent bandwidth savings. We note that both the models significantly outperform existing compression schemes for PPG (CR = 40:1 [6], and CR = 12:1 [20]).

6.1 Design Tradeoffs

The proposed PPG monitoring scheme enables a tradeoff between energy savings and diagnostic accuracy by choosing appropriate threshold values for feature and raw signal comparisons at the sensor. Further, the choice between the *Tem-PPG* and *DE-PPG* models is also a tradeoff between data storage and energy consumption at the sensor. We now discuss these tradeoffs and provide the corresponding experimental results.

The threshold values used in the sensor for deciding when to send feature or raw signal updates, can be set *a priori* or during the operation of the system. Choosing tight threshold values would lead to more frequent updates from the sensor, thus increasing the energy consumption. However, this also reduces the feature error in the reported PPG. Conversely, choosing loose threshold values provides higher energy savings at the cost of increased feature errors. This tradeoff can be observed from the results shown in Table 3 for the two datasets. We observe that by relaxing threshold values, a compression ratio as high as 2406:1 can be achieved, at less than 10% feature error. Given a radio energy consumption of 0.05 mJ/sample [6], this translates to reducing the power consumption from 6.25 mW to 2.6 μW, which would make the sensor sustainable using existing energy scavenging techniques [13].

As discussed in Section 5.1 the *DE-PPG* model requires only the coefficients $[a_1, \ldots, a_{11}]$ and the *ModelPulse* to be stored at the sensor. However, variations in the pulse shape of the sensed PPG trigger a raw signal update of $B \cdot F_s = 625$ samples. Conversely, in the *Tem-PPG* case, storing the entire set of templates consumes storage memory on the sensor, but captures shape variations using a single feature update of the template index t_i parameter. This leads to an improved compression ratio, as seen in Table 2.

7. HANDLING WIRELESS CHANNEL ERRORS

The results in the previous section were obtained assuming that the wireless channel is error-free and all the updates sent by the sensor are received at the server. In several Body Sensor Network (BSN) application scenarios, however, the wireless medium is expected to introduce errors in the sensor transmission, which may affect the performance of the proposed data monitoring scheme. In this section, we investigate the robustness of the proposed scheme to typical wireless channel errors and present two methods to tradeoff energy consumption with the reported PPG signal quality. We first discuss the channel error models considered in our evaluation.

7.1 Errors in On-body Wireless Channels

The wireless environment around the human body is characterized by significant path loss, scattering and multipath effects [9]. Several error models have been proposed in the literature for body

[2] A direct signal comparison metric such as Mean Square Error is not applicable here, since the goal is not to transmit the exact PPG waveform. Instead, the similarity in the diagnostic content of the model-generated PPG and the original PPG is of interest.

Generative Model	Database	Compression Ratio	Feature Error (%)			
			Heart Rate	Pulse Height	Systole Width	Diastole Width
Tem-PPG	MIMIC	407.2	5.68	3.54	4.35	4.87
	Impact	230.61	4.3	4.44	5.48	8.2
DE-PPG	MIMIC	262.95	5.77	3.52	4.30	4.82
	Impact	178.13	4.4	4.6	5.5	8.4

Table 2: Average compression ratio and feature errors for *Tem-PPG* and *DE-PPG* for Impact and MIMIC databases

Threshold	Database	Compression Ratio	Feature Error (%)			
			Heart Rate	Pulse Height	Systole Width	Diastole Width
Tight	MIMIC	335.1	5.72	3.53	4.32	4.84
	Impact	204.37	4.35	4.52	5.49	8.3
Moderate	MIMIC	985.65	6.38	4.13	4.56	6.13
	Impact	429.92	6.13	4.64	5.8	8.23
Loose	MIMIC	2406.1	6.68	4.23	4.61	6.93
	Impact	980.75	9.7	4.98	6.69	8.53

Table 3: Trade-off between feature errors and energy savings or compression ratio for different threshold values

sensors operating in various frequency bands [4, 24]. In this paper, we use the 2.45 GHz ISM band, and consider the following channel error models:

1. Random bit errors: Random bit errors are typically caused by path loss or interference from other wireless devices leading to low Signal-to-Noise ratio (SNR) at the receiver. These are especially high in the unlicensed 2.45 GHz band, due to interference from wireless devices such as laptops and smartphones [4]. These errors cause bits in a packet to be randomly flipped with a certain probability, called Bit Error Rate (BER). For a given BER value, the probability of a packet being erroneous is: $PER = 1 - (1 - BER)^L$ where PER is the packet error rate and L is the length of the packet in bits. We use this model to represent errors due to path loss and wireless interference, and use BER values 10^{-6}, 10^{-4} and $10^{-2.5}$ to represent low, medium and high error rates respectively.

2. Channel fading: Fading is commonly observed in BSNs due to the mobility of the user [9]. It introduces a time-varying component into the path loss between the transmitter and receiver nodes, thus causing a time-varying BER [9]. Fading in BSNs has been widely studied and multiple models have been proposed. In this paper, we use a Ricean flat fading model given by IEEE 802.15 TG 6, as a part of the channel model 8.2.6B for on-body BSN communication [2]. In this model, the parameter K of the Ricean fading distribution is path loss-dependent, and is given by: $K_{dB} = K_0 - m_k P_{dB} + \sigma_k n_k$, where K_{dB} is the K parameter in dB, $K_0 = 30.6dB$, $m_k = 0.43dB$, $\sigma_k = 3.4dB$, P_{dB} is the path loss in dB and n_k is a zero mean unit variance Gaussian variable. We use this model, with path loss varying from 10 dB to 70 dB.

3. Burst errors: These errors occur in BSNs due to the dynamic nature of on-body wireless channels, caused by body movements [26]. These movements could be intermediate, such as ambulation, or continuous, such as from breathing. Burst errors affect a string of bits, thus leading to corrupted packets. In literature, burst errors have typically been modeled using Fritchman models which consider Markovian transitions between multiple error-free states and a single error state [18]. In this paper, we used the three-state Fritchman model proposed in [26], which considers one error state and two error-free states. We use the transition matrices given in [26] for thresholds -14 dB, -10 dB and -6 dB to represent low, medium and high burst errors respectively.

7.2 Performance of Proposed Method

The effects of wireless channel errors on the energy savings and accuracy of the proposed method were studied through MATLAB simulations. As discussed in Section 6, diagnostic accuracy is evaluated using heart rate, pulse height, systolic width and diastolic width features of PPG. For ease of representation, we define *average feature error* as the average of the error in these features. We compare our model-based method to a basic periodic transmission case, where PPG samples are buffered at the sensor and periodically transmitted to the server. To represent existing compression schemes [6, 20], we assume that the transmissions are reduced in size by compression ratios of 12 and 40. As discussed in Section 2, such a periodic transmission model is representative of most existing PPG monitoring schemes.

Figure 6 shows the average feature error in the proposed method under the three types of network errors discussed in the previous section. The results are compared with the periodic signal update case for compression ratios 1, 10 and 40. With increasing BER, burst or fading errors, the accuracy of both the proposed and the periodic signal update methods deteriorates. However, at high error rates the proposed method outperforms the periodic signal update method in feature accuracy. This is because the model-based method transmits much lesser data and hence has lesser probability of experiencing an error burst or a strong fade. Further, most of the communication occurs through short feature update packets, rather than long packets used to transmit signal samples. Since the probability of packet corruption at a given BER increases with packet length, the periodic data transmission suffers a much higher packet loss at high BER values. Thus, the proposed method provides better accuracy than periodic signal transmission, while still reducing energy consumption by a factor of approximately 300:1.

We note, however, that the average feature error is around 10%, which may be unacceptable for some applications. Further, since a single feature update determines a significant chunk of the reconstructed PPG, a lost feature update could introduce significant error. It could also lead to a mismatch between the models at the server and sensor, thus affecting future reconstructed PPG values as well. As a result, we investigate methods to improve transmission reliability while maintaining low energy consumption.

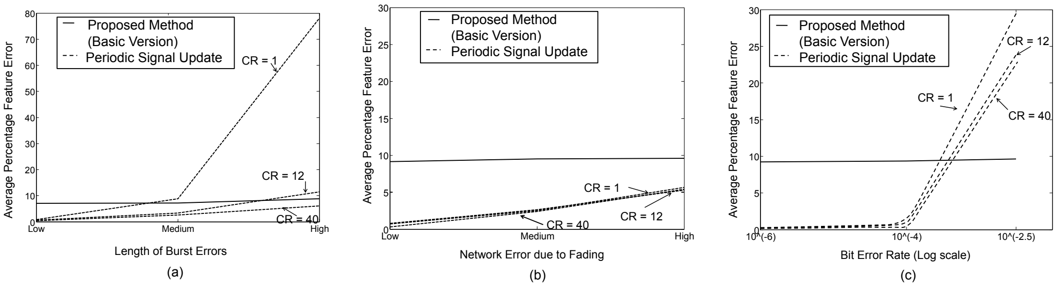
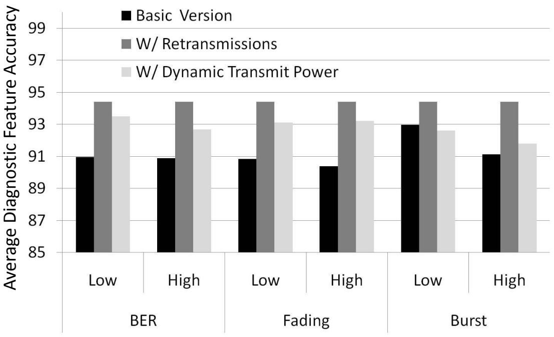


Figure 6: Feature Errors with increasing network errors, for the basic version of the proposed scheme and for periodic transmission schemes: (a) Burst errors, (b) Fading errors and (c) Random bit errors. We note that the proposed scheme achieves accuracy comparable to the periodic transmission case, in spite of reducing energy consumption by a factor of approximately 300.



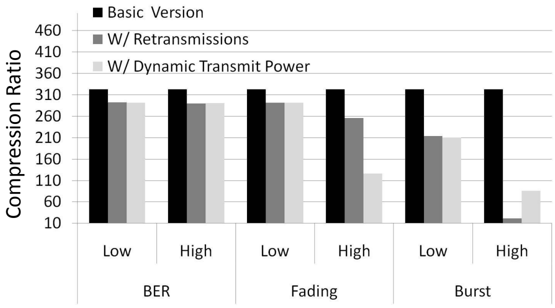


Figure 7: Accuracy comparison for the basic model-based scheme and the two improved versions, under various wireless channel errors. The retransmission method guarantees delivery of all updates thus achieving highest accuracy.

Figure 8: Compression ratio comparison for the basic model-based scheme and the two improved versions, under various wireless channel errors. The basic version has a constant CR since it does not modify its performance based on errors.

7.3 Enhancing transmission reliability

In order to ensure reliable transmission of sensor updates to the server, we investigate the following two approaches:

1. Reliable transmission protocol: In this method, the feature updates sent by the sensor are acknowledged by the server with a short 8-bit ACK packet. The server checks all received updates for errors, and sends an ACK only if the packet is error-free. If no ACK is received within a certain timeout, the sensor retransmits the update. Since feature updates are very short packets, this retransmission incurs marginal energy cost, but could provide significant increase in the accuracy of the reported PPG signal. We do not use this technique for raw signal updates since it would adversely affect the energy savings. Further, an error in a raw signal packet typically affects only a few samples of the reported PPG.

2. Dynamic transmit power control: Another approach to improve the reliability of sensor updates is to increase the transmit power. This improves the SNR at the receiver, thus lowering the BER. However, constantly transmitting at high power would cause increased energy consumption and undesirable radiation effects on the human body. As a result, we use dynamic transmit power control, where the transmit power is increased only if the channel error probability is high. This is typically achieved through link quality estimation where the transmitting node obtains path loss and noise information values for the channel. We assume that such a link quality estimator is available at the sensor and can inform the choice of transmit power. Several such link quality estimation and transmit power control mechanisms are available for BSNs [19,25]. In this paper, we use a scheme where two transmit power levels ($T_{DEFAULT}$ and T_{HIGH}) are defined. Whenever the expected BER exceeds a predefined threshold, the sensor switches the transmit power from $T_{DEFAULT}$ to T_{HIGH}. This scheme is used only for feature updates in order to limit the increase in energy consumption. We note that this approach decreases BER but does not guarantee error-free updates like the retransmission method.

These two methods were added on to the basic data monitoring scheme of Section 5 and the performance of the improved versions was evaluated in the presence of random bit errors, burst errors and fading errors. For the retransmission method, we consider the retransmitted updates as additional data being sent, and accordingly calculate the compression ratio. For the dynamic transmit power case, the compression ratio is calculated by taking a ratio of the total energy consumption of the proposed method to that of the periodic signal update case. As seen in Figure 7, the retransmission and dynamic transmit power methods improve the accuracy of the proposed model-driven scheme. The retransmission method achieves the best accuracy since all updates are guaranteed to reach the server. However, this increase comes at a cost of decreased CR due to the additional overhead in feature update retransmissions. As shown in Figure 8, the decrease in CR is highest under burst errors since long bursts also affect the retransmitted updates. However, in most cases, the retransmission and dynamic transmit power control methods reduce the error to approximately 5%, while still providing significant energy savings.

8. CONCLUSION

In this paper, we proposed two generative models for PPG: *DE-PPG* and *Tem-PPG*, and used them to achieve highly resource-efficient and reliable wireless PPG monitoring. Both models provide equal level of energy savings and accuracy, with *Tem-PPG* having much lower model learning complexity but slightly higher memory consumption than *DE-PPG*. Evaluation on wearable sensor based PPG data showed an average compression ratio of 300:1 with an accuracy of 94% even in the presence of wireless errors. In an error-free medium, with suitable threshold values, the compression ratio increases to 2406:1, while still preserving diagnostic accuracy. Such high compression ratios drastically reduce sensor energy consumption, and represent a significant step towards powering PPG sensors using scavenged energy.

9. REFERENCES

[1] http://www.physionet.org/.

[2] http://math.nist.gov/mcsd/savg/papers/15-08-0780-09-0006-tg6-channel-model.pdf.

[3] J. Allen. Photoplethysmography and its application in clinical physiological measurement. *Physiological measurement*, 28:R1, 2007.

[4] A. Alomainy et al. Statistical analysis and performance evaluation for on-body radio propagation with microstrip patch antennas. *Antennas and Propagation, IEEE Transactions on*, 55(1):245–248, 2007.

[5] H. Asada et al. Mobile monitoring with wearable photoplethysmographic biosensors. *Engineering in Medicine and Biology Magazine, IEEE*, 22(3):28–40, 2003.

[6] P. Baheti and H. Garudadri. An ultra low power pulse oximeter sensor based on compressed sensing. *2009 Body Sensor Networks*, pages 144–148, 2009.

[7] A. Banerjee, K. Venkatasubramanian, and S. Gupta. Challenges of implementing cyber-physical security solutions in body area networks. In *Proceedings of the Fourth International Conference on Body Area Networks*, page 18. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2009.

[8] D. Birchfield. Generative model for the creation of musical emotion, meaning, and form. In *Proceedings of the 2003 ACM SIGMM workshop on Experiential telepresence*, pages 99–104. ACM, 2003.

[9] S. Cotton and W. Scanlon. An experimental investigation into influence of user state and environment on fading characteristics in wireless body area networks at 2.45 GHz. *Wireless Communications, IEEE Transactions on*, 8(1):6–12, 2009.

[10] V. Crabtree and P. Smith. Physiological models of the human vasculature and photoplethysmography. *Electronic Systems and Control Division Research, Department of Electronic and Electrical Engineering, Loughborough University*, pages 60–63, 2003.

[11] A. Deshpande et al. Model-driven data acquisition in sensor networks. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, page 599. VLDB Endowment, 2004.

[12] R. Fletcher et al. iCalm: Wearable sensor and network architecture for wirelessly communicating and logging autonomic activity. *Information Technology in Biomedicine, IEEE Transactions on*, 14(2):215–223, 2010.

[13] B. Gyselinckx et al. Human++: autonomous wireless sensors for body area networks. In *Custom Integrated Circuits Conference, 2005. Proceedings of the IEEE 2005*, pages 13–19. IEEE, 2005.

[14] P. McSharryet al. A dynamical model for generating synthetic electrocardiogram signals. *Biomedical Engineering, IEEE Transactions on*, 50(3):289–294, 2003.

[15] Y. Mendelson, R. Duckworth, and G. Comtois. A wearable reflectance pulse oximeter for remote physiological monitoring. In *Engineering in Medicine and Biology Society, 2006. EMBS'06. 28th Annual International Conference of the IEEE*, pages 912–915. IEEE, 2006.

[16] S. Millasseau et al. Contour analysis of the photoplethysmographic pulse measured at the finger. *Journal of hypertension*, 24(8):1449, 2006.

[17] S. Nabar et al. GeM-REM: Generative Model-driven Resource efficient ECG Monitoring in Body Sensor Networks. In *Proceedings of International Conference on Body Sensor Networks (BSN)*. IEEE, 2011 http://ee.washington.edu/research/nsl/papers/GeMREM.pdf.

[18] C. Pimentel and I. Blake. Modeling burst channels using partitioned Fritchman's Markov models. *Vehicular Technology, IEEE Transactions on*, 47(3):885–899, 1998.

[19] M. Quwaider, J. Rao, and S. Biswas. Body-posture-based dynamic link power control in wearable sensor networks. *Communications Magazine, IEEE*, 48(7):134–142, 2010.

[20] K. Reddy et al. Use of fourier series analysis for motion artifact reduction and data compression of PPG signals. *Instrumentation and Measurement, IEEE Transactions on*, 58(5):1706–1711, 2009.

[21] A. Reisner et al. Utility of the photoplethysmogram in circulatory monitoring. *Anesthesiology*, 108(5):950, 2008.

[22] N. Selvaraj et al. Assessment of heart rate variability derived from finger-tip photoplethysmography as compared to electrocardiography. *Journal of Medical Engineering & Technology*, 32(6):479–484, 2008.

[23] S. Shirouzu, E. Shirouzu, Y. Tsuda, and H. Sugano. Circulatory parameter extraction from digital plethysmogram. i. waveform analysis of digital plethysmogram. In *Engineering in Medicine and Biology Society, 1998. Proceedings of the 20th Annual International Conference of the IEEE*, volume 6, pages 3087 –3089, 1998.

[24] K. Takizawa, T. Aoyagi, and R. Kohno. Channel Modeling and Performance Evaluation on UWB-Based Wireless Body Area Networks. In *Communications, 2009. ICC'09. IEEE International Conference on*, pages 1–5. IEEE, 2009.

[25] S. Xiao et al. Transmission power control in body area sensor networks for healthcare monitoring. *Selected Areas in Communications, IEEE Journal on*, 27(1):37–48, 2009.

[26] B. Zhen et al. Characterization and Modeling of Dynamic On-Body Propagation at 4.5 GHz. *Antennas and Wireless Propagation Letters, IEEE*, 8:1263–1267, 2009.