



Energy-efficient collection of wearable sensor data through predictive sampling

Wassila Lalouani ^a, Mohamed Younis ^{a,*}, Ian White-Gittens ^b,
Roland N. Emokpae Jr. ^c, Lloyd E. Emokpae ^c

^a CSEE Dept., Univ. of Maryland, Baltimore County, Baltimore, MD, USA

^b Harlem Hospital, NY, 506 Lenox Ave, New York, NY, 10037, USA

^c LASARRUS Clinic and Research Center LLC., Baltimore, MD, USA

ARTICLE INFO

Keywords:

Wearable devices
Energy optimization
Body area sensor network
Predictive sampling
In-network data processing

ABSTRACT

Wearable sensors have emerged as viable and attractive solutions for monitoring the health of people under risk of major problems such as hypertension, heart attacks, and athletes overstressing their bodies. These devices would report on the status of certain body organs to a gateway node over wireless links. A major challenge for effective use of these miniaturized devices is sustaining their operation using a limited energy supply. Therefore, minimizing energy consumption is a crucial design goal. In this paper we propose a novel approach for reducing the volume and frequency of data transmission through data sample prediction. Our methodology is based on applying advanced machine learning techniques to determine when data transmissions are skipped, and by implicitly making the gateway aware of the omitted samples in order to achieve accurate signal reconstruction. The paper also presents a data quantization technique for increased throughput and reduced energy overhead while sustaining desired medical assessment accuracy. Furthermore, a packet formation algorithm is proposed to leverage the available buffering space to improve bandwidth and energy utilization subject to latency constraints. The effectiveness of our approach is validated using publicly available Electrocardiography and Electromyography datasets and is shown not only to outperform conventional data compression methods but can also be applied in conjunction with them.

1. Introduction

The major breakthrough in developing wearable medical devices and the emergence of the Internet of Things (IoT) has revolutionized the healthcare industry (Riazul Islam et al., 2015; Nadeem et al., 2015; Niknejad et al., 2020). Particularly, these advanced technologies have enabled the development of effective and economic solutions for remote and continuous monitoring of patients with medical conditions. For example, the heartbeat of individuals can be measured to detect cardiac unrest and automatically call for emergency assistance. Such a monitoring service has traditionally been possible only through hospitals or specialized clinics, and consequently deemed both expensive for insurance companies and inconvenient for patients and their families. For health insurance providers reducing the cost is paramount in order to maintain affordable premiums. Moreover, wearable sensors are invaluable for monitoring the body conditions under stress, e.g., while exercising or playing sports. The architecture of such a real-time health

* Corresponding author.

E-mail address: younis@umbc.edu (M. Younis).

monitoring system consists of single or multi-modality sensing devices to collect relevant measurements and transmit them through a gateway node to storage centers, either cloud-based or private, to be accessible to caregivers.

Since the sensing devices will be wearable, i.e., attached on the body or as part of an apparel, they ought to be miniaturized and require low maintenance. Therefore, powering wearable sensors needs to rely on very small batteries or even scavenged energy from ambient sources. However, the operation of these wearable devices involves significant energy consumption due to the wireless transmission and high sampling rates required for collecting physiological data such as Electrocardiography (ECG), electromyography (EMG), acoustic cardiography (ACG) and acoustic myography (AMG). Therefore, minimizing energy consumption is a major goal to the proper operation of a wearable sensor system. Contemporary optimization schemes can be classified into three categories: (i) low power circuit design ([Karray et al., 2018](#)), (ii) energy-aware communication protocols ([Ali et al., 2019](#)), and (iii) in-network data processing ([Dehkordi et al., 2020](#)). The latter is generally an application-dependent methodology yet proved to be quite effective in the context of wearable sensors ([Abiodun et al., 2017](#)). Intuitively, a tradeoff is involved between the growth in computation overhead and the decrease in communication-related energy consumption. Compressive sensing is an example of such methodology where data reduction is sought to cut on the number of transmissions at the expense of increased on-node signal processing ([Al Disi et al., 2018](#)).

In this paper, we focus on reducing the number of transmissions through in-network data processing. The objective is to conserve the energy of a wearable device by skipping some samples without degrading the accuracy of the physiological data so that the correctness of the diagnostics is sustained. Unlike contemporary schemes that consider certain signal characteristics and exploit similarities among samples, our approach opts to limit the overhead of frequent sample transmission by enabling the sensor and the gateway to accurately forecast the upcoming signal samples. We employ recurrent neural networks using long short-term memory (LSTM) at the communicating pair, i.e., sensor and gateway, in order to predict the next physiological data samples. By setting a certain variation threshold, the sensor will decide on skipping the transmission if the difference from the previous data sample is negligible. By running the same LSTM model at the gateway, missing certain samples can be expected and the corresponding data can be estimated. A major advantage of our approach is that the error bound for a reproduced data sample is easily controlled by adjusting the variation threshold and thus our approach can be applied to a wide range of sensor modalities. Furthermore, our approach does not suffer from distortion of each reconstructed signal segment as the errors are handled sequentially; any violation of the bound necessitates the transmission of the sample and consequently restoring the accuracy for further upcoming sample prediction.

In order to further reduce energy consumption and boost the transmission throughput, we exploit the trade-off between the condition assessment accuracy for the monitored individual and the size and latency of the transmitted data. Basically, the quantization accuracy, i.e., number of bits, of the data, is optimized so that the fewest bits are included on a packet payload corresponding to the data samples. The device buffering capacity is also leveraged to pack as many data samples within a single packet and fully utilize the maximum allowed packet size and consequently reduce the energy consumed per transmitted sample. This also improves the bandwidth utilization for the wireless links. Overall, our approach can be seen as complementary for any compressive sensing algorithm and may be generically applied to various sensor modalities. The effectiveness of our sample prediction based energy optimization (SPEO) approach is validated through simulation using publicly available ECG datasets. The simulation results demonstrate that our SPEO approach outperforms a prominent data compression scheme. In summary the contribution of the paper is as follows:

- Employing advanced machine learning techniques to enable accurate signal reconstruction in the absence of some data samples.
- Developing a protocol to minimize energy consumption in collecting data from wearable devices by predicting the sample values at both communications ends.
- Devising a data quantization technique for increased throughput and reduced energy overhead while sustaining desired medical assessment accuracy.
- Developing a mechanism for packet formation that exploits the tradeoff between buffer size and data delivery latency to reduce overhead.
- Demonstrating the synergistic effect of the proposed approach to contemporary data reduction techniques such as compressive sensing.
- Validating the proposed techniques using large ECG datasets.

The reminder of the paper is organized as follows. The next section discusses the related work. Section 3 goes over our system model and architecture. Section 4 presents the machine learning model for sample prediction. The packet formation optimization is described in Section 5. Section 6 discusses the validation environment and performance results. Finally, the paper is concluded in Section 7.

2. Related work

Conserving energy has been a popular design goal for wearable systems in order to increase longevity. Many of the published schemes have focused on the networking aspect of the system and focused on energy consumption optimization at the various level of communication protocol stack, particularly at the network layer ([Qureshi et al., 2020](#); [Raj & Chinnadurai, 2020](#), [Yan et al., 2020](#)), and link layer ([Yang et al., 2018](#); [Beltramelli et al., 2021](#)). Some work has also exploited the selective activation of sensors in order to keep them in the low-power sleep mode for the longest time ([Kaur & Sood, 2017](#); [Raval et al., 2021](#)). Variants of the published studies have considered achieving the longevity goal while considering potential energy scavenging ([Ma et al., 2020](#)). Given the contribution of the paper, we focus on work that exploited signal processing and compression.

Compression of IoT Data: Compressive sensing (CS) is widely used for extending the battery lifetime of wearable devices ([Al Disi et al., 2018](#)). In general, CS aims to digitize the sampled signal using fewer measurements than the Shannon-Nyquist rate. CS

approaches in the literature either apply transformations or direct compression to the original signal (Qaisar et al., 2013; Djelouat, Amira, & Bensaali, 2018). Transformation based schemes change the signal representation using popular techniques such as Fourier descriptors, Walsh transform, Karhunen–Loeve transform and wavelet transform. Meanwhile, direct compression applies encoding techniques such as amplitude zone time epoch, the turning point, the coordinate reduction time, differential pulses modulation. Although direct compression yields better reconstructed signals, transformation based schemes achieve a higher compression ratio. A survey and detailed taxonomy of CS in biomedical application can be found in (Khosravy & Duque, 2020). There is also continuous development of CS reconstruction algorithms that aim to reduce error-rates or address special remote sensing applications. Zhang and Rao (2013) have proposed a reconstruction algorithm that capitalizes on the signal structure and its intra-block correlation. Again, SPEO can be viewed as a complementary, rather than competing, optimization scheme to CS. In Section 6, we compare the performance of SPEO to CS, individually and when applied together.

Deep learning for Biomedical IoT: In connected healthcare applications, data collection and data processing are usually the two main sensor-related functional components. Deep learning techniques have been traditionally used for data processing, but are rarely exploited for optimizing the data collection process. Most published work in such an area of research focuses on applying deep learning techniques for information extraction, behavior modeling, and phenotyping (Zemouri et al., 2019). For example, a deep belief network-based algorithm has been proposed in (Wu et al., 2016) to learn features from ECG data of arrhythmias patients. A. Abrishami et al. (2018) have focused on extracting spatial features from ECG signals to ensure automatic segmentation of heartbeat signals. Alternatively, some work considers the temporal features for disease or anomaly classification. This has been the approach taken by Yildirim et al. (2018), where deep bidirectional LSTM network-based wavelet sequence encoding and decoding are exploited for decomposing and classifying ECG signals in order to determine symptoms of alcoholism in a person. In general, ECG signal decomposition and classification do not allow accurate reconstruction of the original signal and consequently hinder analyzing the patient health status at the gateway node.

Energy-aware signal reconstruction and analysis: IoT based architectures have been leveraged for numerous medical applications, such as remote health monitoring, fitness programs, and elderly care (Riazul Islam et al., 2015). There have been varying studies that look at architectures for real-time monitoring of patient health conditions. Some studies have employed gateway nodes to establish connectivity to medical devices to cloud computing platforms (Moustafa et al., 2016; Rahmani et al., 2018). Meanwhile others rely on portable devices with limited-capabilities. For example, Grndl et al. (2012) have demonstrated the feasibility of real-time ECG QRS detection, feature extraction, and heartbeat classification on smart portable devices. SPEO focuses on the energy optimization for collecting sensor samples by reducing the number of wireless transmissions, and can be integrated with any computation platform for conducting medical assessment.

Energy efficient reconstruction of sensor signals has been the objective of a number of studies; yet the focus has been on conducting the analysis at the gateway node rather than at the sensor node. For example, an iterative shrinkage-thresholding algorithm is employed in (Kanoun et al., 2011) to reconstruct ECG signals on smartphones. Other work considers real-time energy efficient reconstruction using specifically designed microcontrollers. The idea is to exploit the distribution of information in sparse signals in order to reduce the number of samples that are needed for reliable data acquisition (Al Disi et al., 2018). While the computation cost is factored in such a category of work, the communication overhead is ignored. SPEO focuses on reducing the communication energy at the sensor side and can be viewed as complementary to the aforementioned computation energy reduction techniques.

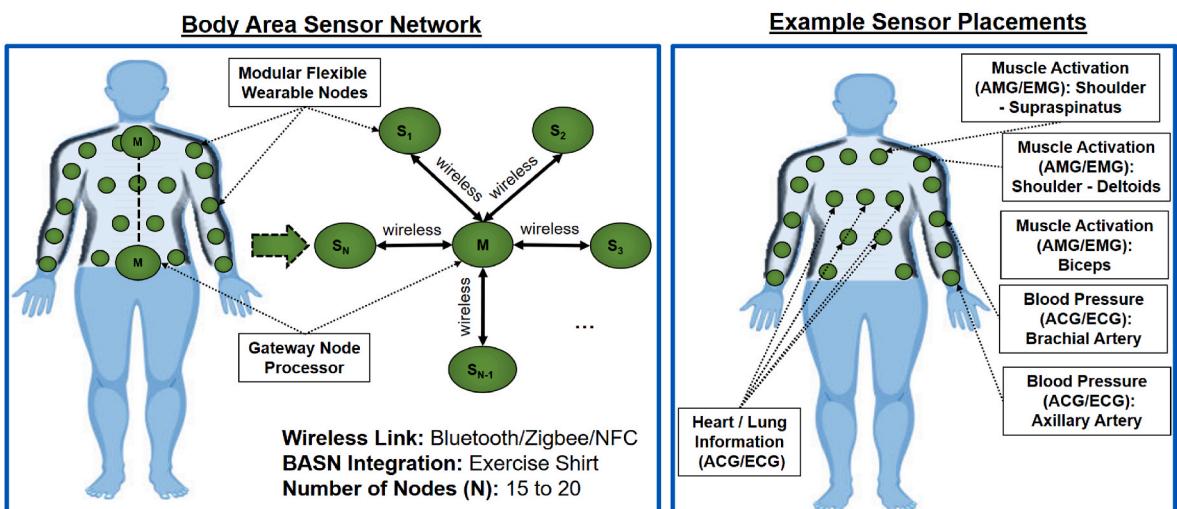


Fig. 1. Image to the left shows the body area sensor network, which integrates sensor nodes for health analysis. Image to the right illustrates an example of the sensor node placement corresponding to the physiological measurement.

3. System model and solution overview

3.1. System architecture

We consider a body area sensor network (BASN), which incorporates a mesh of sensors that are networked to measure full torso range of motion, muscle activation, and body vitals in the form of ECG, EMG, ACG and AMG. The sensor data is disseminated to a gateway node that either processes the data or forwards it to a remote healthcare facility. The system architecture is shown in Fig. 1. The BASN is assumed to employ a robust MAC protocol that ensures delivery of packets, e.g., by sending data packet acknowledgements. In this paper, we are assuming that each sensor node in the BASN will utilize the Digi XBee® 3 Zigbee standard for establishing wireless links.

BASNs have the potential for improved health monitoring by aggregating multiple interconnected nodes, on a human body, for sensorimotor measurements. For example, BASN would provide a patient with a quantitative measure of progress that currently does not exist in the practice of conventional rehabilitation, e.g., physical therapy. Without any quantitative data, the patient may face motivational challenges when improvements are subtle and gradual. Furthermore, without knowing the details of progress, it can be difficult to set and reach goals, for example a goal to improve shoulder abduction by 5 % within 6 months. With the considered BASN, each modular wireless sensor node incorporates multimodal sensors (electrical, MEMS, and acoustics) for measuring both torso range of motion and the physiological state of the user at the specific integration point (shoulder, biceps, chest, back, etc.).

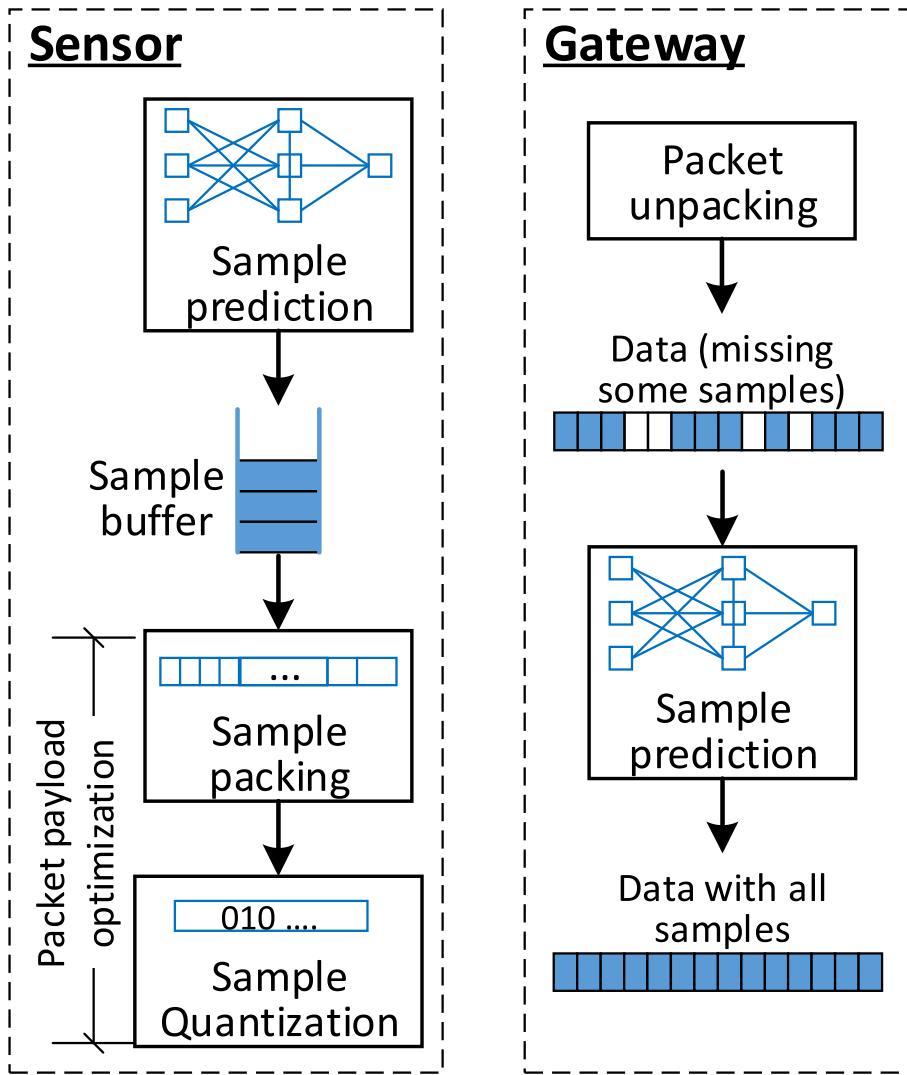


Fig. 2. High level block diagram of the proposed SPEO approach.

3.2. Approach overview

Nodes in a BASN are required to operate on very limited battery capacities; therefore, minimizing energy consumption is very critical to the lifetime of the nodes and network. In that regard, radio transmissions impose the most energy overhead for the BASN nodes and is thus a prime target for optimization. SPEO opts to conserve communication-related energy by: (1) avoiding transmission of data samples that can be implicitly inferred given prior sample and based on the specific sensor modality, and (2) reducing the number of sent packets by optimized packet payload formation. Fig. 2 provides an overview of the steps at both the wearable sensor and gateway nodes. The sensor node will employ a machine learning model to assess whether the current sample can be predicted using the previous ones, subject to a certain inaccuracy bound. Samples that can be accurately predicted are skipped, while the rest are buffered to be sent to the gateway node.

The buffered samples are in effect the payload of the transmitted packets. Payload formation is further optimized to conserve energy. The optimization here is based on two techniques:

- (i) sample packing to fully utilize the maximum packet size, where our SPEO approach strives to include as many samples as feasible as possible while informing the gateway about any skipped samples. In essence, data sample buffering constitutes a tradeoff between data delivery latency and energy conservation. To elaborate, assume that the size of a packet header is α bits and a data sample needs β bits; hence, sending two samples within the same packet will require transmission of $\alpha+2\beta$ bits rather than $2\alpha+2\beta$ bits if the samples are sent in two different packets and consequently consume less energy. To combine these two samples on the same packet payload necessitates buffering of the samples rather than rushing their transmission and hence increased data delivery latency. Upon receiving a packet, the gateway will use the embedded information in the packet to determine the number and order of samples in the payload. The gateway then applies the sample prediction model employed by the sensors, in order to estimate the missing samples.
- (ii) determining the minimum quantization accuracy (number of bits) for representing the samples without diminishing the application effectiveness. Quantization accuracy is not the only factor that affects medical assessment accuracy. Nonetheless, SPEO does not control the signal processing and reconstruction and assumes that the desired quantization accuracy is determined based on an application-level metric covering the end-to-end process. Determining the quantization accuracy would in practice be based on consultation with physicians, e.g., showing reconstructed signals for various quantization accuracies to the physicians to decide on the least quantization accuracy for which they can make the correct assessment of the patient conditions.

While our proposed sample prediction technique is not specific to certain sensing modality, the use of quantization accuracy could vary for example based on whether ECG or ACG data are being disseminated from the sensor to the gateway. The next section describes the two aforementioned techniques in detail.

4. Sample prediction technique

To cut on the number of transmissions and conserve the energy of wearable devices, we propose a data sample prediction mechanism based on advanced machine learning models. The key idea is based on the observation that under no serious health conditions, there are little variations in monitored physiological attributes and consequently the collected data. Thus, the transmission of some data samples could be skipped as long as no negative effect will be inflicted on the application, i.e., no health alerts will be missed. However, skipping samples raises two main issues:

- (1) An assessment of the utility of the sample has to be conducted at the sensor level. This conventionally requires modality-specific sensor data processing that often involves multiple prior samples;
- (2) The gateway node should be aware of the skipped samples. Fundamentally the samples should be time-stamped so that they can be correctly ordered and processed. A missed sample could be attributed to packet loss over noisy wireless links or to failure at the sender (sensor) side. Gateway awareness is particularly a more pressing issue if skipping samples is not done on a regular basis, i.e., sporadic over time, and if multiple consecutive samples could be skipped. Therefore, the decision by a sensor to not transmit a sample should be expected and mitigated by the gateway.

To tackle the aforementioned issues, our SPEO approach employs a machine learning model at the sensor side; such a model is also replicated on the gateway node. Our model identifies the possible predictable set of samples that may be inferred by the gateway. Indeed, the analog sensing data from different modalities like ECG, EMG and AMG exhibits some known patterns constituting time series. We exploit such propriety in order to forecast the next sample from previous measurements; the accuracy is then assessed when the next sample is collected and a decision is made on whether a transmission is needed, i.e., if the actual sample significantly deviates from the predicted one to warrant transmission. SPEO leverages Recurrent Neural Networks (RNN), which proved to be very effective for time series datasets (Olah, 2015). Particularly, we employ a Long Short-Term Memory (LSTM) network, which is a special type of RNN that is trained using backpropagation through time and overcomes the vanishing gradient issue of ordinary recurrent networks (Djelouat et al., 2020).

Instead of neurons, LSTM networks employ memory blocks known as gates that are connected through layers. Gates operate upon an input sequence and use a specific activation function to control whether a prior input is to be factored in (remembered) or be

deemed outdated (forgotten). This in effect will control changes in the LSTM state and consequently the information flow. Those conditional gates can be either forget, input or output gates. The forget gates decide which information to throw away from the block. Input Gate decides which values from the input to update the memory state. Output Gate decides what to output based on input and the memory of the block. Each gate has weights that are learned during the training process. Fig. 3 shows the architecture of the LSTM deep network that we employ to predict sensor samples.

Our network is constructed of one input layer, single hidden layer with multiple LSTM cells, and output layer. Our activation function is the sigmoid activation. The following are the operations performed in each prediction (Bianchi et al., 2017; Olah, 2015).

$$i_t = \sigma(W_i \cdot [H_{t-1}, x_t] + b_i) \quad (1)$$

$$f_t = \sigma(W_f \cdot [H_{t-1}, x_t] + b_f) \quad (2)$$

$$\check{C}_t = \tanh(W_C \cdot [H_{t-1}, x_t] + b_C) \quad (3)$$

$$C_t = C_{t-1} \cdot f_t + i_t \cdot \check{C}_t \quad (4)$$

$$o_t = \sigma(W_o \cdot [H_{t-1}, x_t] + b_0) \quad (5)$$

$$h_t = o_t \cdot o(c_t) \quad (6)$$

where:

- x_t is the input sequence (vector) in at time epoch t , which constitutes the n previous samples.
- C_t is the new LSTM state at time epoch t , and depends on prior state and input.
- H_t is the output for time epoch t , which is dependent on C_{t-1} and x_t .
- i_t , f_t , o_t are input, forget, and output gate sub-tensors for time epoch t .
- \check{C}_t is a new cell candidate in the input sequence at t .
- b is the bias for appropriate input sub-tensor.
- W_i , W_o , and W_C are the weight vectors for the three LSTM layers, respectively. They are determined through training.

First, the forget gate determines what information is to be discarded based on H_{t-1} and the input vector, i.e., samples, and generates the output f_t . Then, the memory gate decides on what to store in the cell state. The input gate subtensor calculates the new value of i_t , while the \tanh layer creates a vector of new candidate values, \check{C}_t , to be included in the state. The new value of the state is the aggregation of the previous state multiplied by the forget gate output and add $i_t \cdot \check{C}_t$. We need to decide the output based on filtered cell state. To do so, we use a sigmoid layer to determine the parts of the cell state to output. Then, we pass the cell state through \tanh and multiply it by the output of the sigmoid gate, so that we only output the relevant parts.

Both the wearable sensor and the gateway node need to have the same LSTM architecture. The wearable sensor captures the sample of the waveform for a given snapshot (time window) whose length depends on the buffering size and the tolerable reporting delay by the application, as discussed in the next section. For each snapshot, the wearable device forecasts the actual sample using LSTM based on the previous n previous samples in the time series, regardless whether these samples were reported or predicted. When the prediction matches the real value or falls within some acceptable interval based on the specific sensor modality, the sample is not sent to the gateway. Otherwise the sample will be scheduled for transmission based on our packet formation algorithm described in the next.

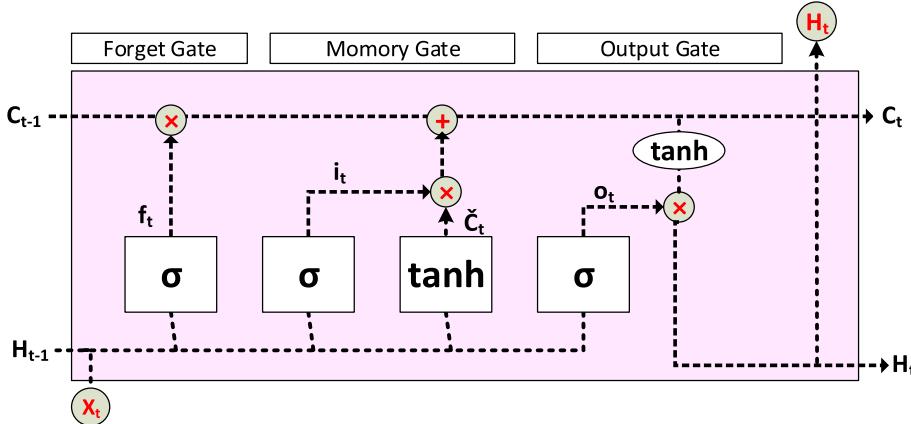


Fig. 3. Showing the detailed design of the LSTM cell. A LSTM model consists of multiple cells; in SPEO the use of four cells provided the best results when considering ECG data.

By running the same LSTM, the gateway will estimate the samples that are skipped by the specific sensor. The LSTM is device dependent and needs to be trained accordingly. Using physiological signals, we collect samples over time. The collected samples serve as a training dataset where each sample is labeled with the corresponding timestamp. The LSTM correlates each sample s_k^t at time t for device k with the previous n samples ($s_k^{t-n}, \dots, s_k^{t-1}$) and determines the weights for the deep network.

5. Packet formation optimization

To take advantage of skipping the transmission of some samples, the gateway ought to know what sample is not sent in order to apply the LSTM model and estimate it. In essence, each packet from a sensor node should provide information to indicate the order (sequence number) of the samples that are included. Incorporating such information introduces additional overhead and should be thus optimized. In addition, a packet typically consists of two parts, namely, header and payload. In almost all protocols, such as Zigbee and Bluetooth, the header size is usually constant regardless how big the payload is. In practice, the payload size is capped based on the medium access control protocol and the condition of the wireless channel. Thus, it is desirable to pursue the maximum packet size and pack as many samples as possible in each packet while avoiding unutilized space in the payload, e.g., the left bits in the payload are not sufficient for fitting a data sample.

5.1. Sample packing optimization

Let S be the number of bits needed for a sample. Generally, S depends on the range of the data values and fundamentally depends on the sensing modality. Let P_{max} be the largest allowed payload in bits, which corresponds to the maximum packet size for the underlying communication protocol. Since some samples are skipped in SPEO, a sequence indicator I for each sample needs to be provided in the packet so that the gateway knows what samples are included and what samples are to be estimated using the LSTM model. Such an indicator normally would not be necessary if all samples are sent. Thus, the total number of required indicator bits will be $(M \times I)$, where M is the number of samples in the packet. Thus, for any packet:

$$P_{max} \geq M \times I + M \times S \quad (7)$$

To be able to pack multiple samples per packet, multiple of them need to be buffered. We refer to the time window corresponding to multiple samples as a snapshot, denoted as T , and is measured in terms of the number of samples covered by the time window. The objective of the optimization is to find T to minimize the packet count N for sending the samples, keeping in mind that some of the T samples may be skipped. Sample buffering will impose delivery latency for the sensor data. Usually such a latency should be capped, which will constrain the size of the snapshot. Assuming that a sensor generates a sample every τ time units, i.e., the sampling rate is $1/\tau$, and that the maximum tolerable delay in sending a sample is Δ , the snapshot T should not exceed $\lfloor \Delta / \tau \rfloor$. Assuming a relatively large Δ and unlimited buffer size, we can determine the optimum number of data packets that fully utilize P_{max} within each packet. However, a wearable sensor node typically has limited onboard memory and thus T is also constrained by the available buffering space B . Thus, the optimization problem can be mathematically specified as follows:

$$\text{Minimize } T, N \text{ and } W$$

such that:

The total size for all transmitted samples and the corresponding overhead should not exceed the maximum combined data capacity of the N packets, i.e.,

$$N \geq \frac{\sum_{0 < i \leq N} (M_i I + M_i S)}{P_{max}} \quad (8)$$

The samples and overhead with each individual packet should be less than maximum allowed by the communication protocol.

$$P_{max} \geq \max_{0 < i \leq N} (M_i I + M_i S) \quad (9)$$

Leaving out unutilized space with a packet should be avoided. Such utilized space, W , is to be minimized and is calculated by:

$$W = P_{max} - \max_{0 < i \leq N} (M_i I + M_i S) \quad (10)$$

The snapshot is constrained by the timeless requirement of the application and the buffer space availability onboard a sensor. The total number of sent samples will be at most equal to the snapshot size if no sample can be predicted and skipped.

$$T \leq \lfloor \Delta / \tau \rfloor \quad (11)$$

$$T \leq B \quad (12)$$

$$T \geq \sum_{0 < i \leq N} M_i \quad (13)$$

Before discussing the solution of the abovementioned optimization, we analyze how to note the sequence and missing samples in

the packet. If I is taken to be the sample sequence number, a total of $(M \times I)$ bits will be needed. Assume that a total of Q samples are covered by a packet, i.e., both included and skipped samples. In such a case, I will be $\log_2 Q$ since we have to factor in the number of skipped samples and not only the M that are included in the packet. Thus, the overhead for adding the sample sequence number in the packet will be $(M \log_2 Q)$. Instead, SPEO pursues a better representation, where the packet payload is to have fields for: (i) the number of samples, Q , covered by the packet, and (ii) a bit for each of the Q samples; the bit will be set to one only if the sample is included, i.e., M out of the Q bits will be ones. For example, assume a packet covers 10 samples of which #1, 2, 5, 6, 7, 9, and 10, are included in the payload, i.e., $Q = 10$. In that case, the indicator overhead consists of 4 bits for reporting the sample count and 10 bits provided with the value 1100111011, reflecting the included and skipped samples, such that the least significant bit corresponds to the first sample and the most significant bit is for the last sample. On the other hand, using sample sequence numbers will imply 28 bits.

SPEO applies an additional optimization in order to minimize wasted space and facilitate parsing the packets by the gateway. Basically, the combined payload of the N packets is first formed and then gets split among the individual packets. In other words, the samples along with the necessary overhead are first packed in $N \times P_{max}$ bits. The rationale is that Eq. (10) above can still yield wasted space up to $(S-1)$ bits per packet. Considering the samples of a snapshot collectively will enable bringing the average wasted space per packet to a maximum of $(S-1)/N$. Fig. 4 shows the proposed format, where the sample size is set to S unless optimized quantization is possible, as discussed in the next subsection. The sample count equals the number of samples, m , among a snapshot T that cannot be skipped. Thus, this field will have $\log_2 T$ bits in the worst case where no sample could be correctly predicted. The skip indicator has one bit for each of the T samples. The remaining space is solely used for the data samples. Once formed, such payload will be divided among the N packets and re-combined at the gateway to be decoded and retrieve the data samples. Thus, Eqs. (9) and (10) above are replaced by:

$$W = [N \times P_{max} - (\lceil \log_2 S \rceil + T + \lceil \log_2 T \rceil) - T \times S] / N \quad (14)$$

It is worth mentioning that SPEO assumes guaranteed packet delivery, e.g., by using acknowledgements, and thus the gateway will not miss the transmitted samples. Such reliable delivery ensures synchronization of the LSTM of both the sensor and gateway. A new packet will not be transmitted until the previous packet is acknowledged. This will prevent divergence of the LSTM models and the sensor and gateway, even when experiencing burst fading of wireless links. Basically, no new packet will be sent during the burst where transmission retrial will persist for the current packet until successful delivery. The only negative effect will be the latency, which is a byproduct of the burst fading rather than SPEO. We also note that the snapshot size T is not fixed and will be a function of the success of the LSTM model in predicting samples. Therefore, SPEO will keep buffering samples until finding T that minimizes N and W for the considered data samples. Upon transmission of the N packets, the process is repeated again for the next T . In other words, SPEO applies a heuristic to solve the multi-objective optimization. We also note that the value of S in Eq. (14) will be subject to optimization as we explain next.

5.2. Sensor sample quantization

The binary representation of floating point numbers is usually subject to length and precision tradeoff. The number of bits S for representing a data sample generally depends on the sensor modality and the required precision. In essence, determining S is a quantization process for an analog signal that opts to enable reconstructing the signal of the various digitized samples such that key properties such as peaks, oscillations, etc., are preserved. Reducing S not only limits the buffer size requirement, but also cuts the communication overhead in terms of bandwidth and energy (Ali et al., 2019). Basically, more samples can be included in a packet if fewer bits per sample are used. In the case of SPEO, the quantization has to factor in the precision of the predicted samples at the gateway side. In other words, the value of S should not only preserve the sensor signal properties when reconstructed at the gateway node, but also enable the gateway to accurately estimate the skipped samples.

While the quantization accuracy can be predetermined (fixed) based on the specific signal, e.g., ECG, SPEO exploits possible reduction of S , per the individual snapshots. The rationale is that determining S is in essence a data-driven process and thus the value of S can be adaptively set based on the signal properties that ought to be preserved. SPEO applies the following:

- (1) Denormalized floating point representation is used in order to sustain maximum precision. To enable doing so, the entire dataset is biased based on the range of values. For example, ECG signals are usually in the millivolts (mV) range and can be biased by multiplying by 1000 at the sensor nodes to boost accuracy; such a bias becomes implicit where the gateway scales back the data, i.e., divides by 1000, when reconstructing the ECG signal.
- (2) A default value of S is determined to maintain clinically acceptable signal reconstruction quality. This depends on the sensor modality and is picked in consultation with the physicians.



Fig. 4. Showing the format for combined payload of the N consecutive packets needed to send a snapshot T .

- (3) While determining the snapshot, possible reduction of S is considered for the buffered samples. The reduction is gauged by both the impact on the samples that are to be skipped. If the accuracy does not diminish, i.e., the predicted samples stay below the tolerated error limit, S is decremented. The process is conducted incrementally, consecutive reduction by one bit is considered.

We note that changing S at the sensor side, requires informing the gateway in order for accurate parsing of a transmitted data packet. Thus, we will add a few overhead bits to inform the gateway on the amount of bits used per sample size, γ . Since $\gamma \leq S$, the number of bits for specifying is $\log(S)$. Overall the quantization-based optimization can yield reduction of payload requirement of $(S -) \cdot m$, where m is the number of samples that must be transmitted.

The pseudo code of the SPEO algorithm is provided in Algorithm 1. The algorithm employs greedy heuristics to pack the maximum number of samples in the minimum number of packets. We aim to achieve this by minimizing the average wasted (unutilized) space within the payload of a packet. First, we determine the maximum snapshot size based on the allotted buffer size and the data latency requirement of the application and then iterate over them to accumulate the remaining space in each packet and track the minimum value. The number of samples that corresponds to the minimum wasted space is the best T and the associated packet count corresponds to N . The first three lines in Algorithm 1 are for training the LSTM. Lines 4–10 checks whether each sample in the buffer B , can be predicted using the LSTM. Using the maximum buffer size B , we try to find the number of samples corresponding to the least wasted packet space in lines 12–27, according to Eq. (14). Those samples are then used to form the N packets. The loop of lines 19–26 are for applying the optimized sample size quantization process described above.

Alg. 1. Pseudo code summary of the SPEO algorithm

```

1. Prepare the dataset and normalize it
2. Train the model
3. Extract the weights of the network
4. For each sample in  $B$ :
5.   Predict the sample using the LSTM.
6.   If the accuracy < Threshold:
7.     Insert the actual sample
8.   else
9.     Mark the sample predicted
10. End for
11. Rescale the data
12.  $T_{best} = B$  // the most-suited snapshot is initialized
13. For each snapshot  $T$  in  $[1, B]$ 
14.   Find smallest number of bits  $S_{min}$  (quantization)
      that meets min accuracy for predicted samples
15.    $V = T + \lceil \log_2 T \rceil$  // Sample specification overhead
16.    $V = V + \lceil \log_2 \gamma \rceil$  // Quantization overhead
17.    $W_{min} = 1$  // least avg wasted space per packet
18.    $\gamma_{min} = S$  // Default quantization (sample size)
19.   For each  $\gamma$  in  $[S_{min}, S]$ 
20.      $N = \left\lceil \frac{V + T \times \gamma}{P_{max}} \right\rceil$  // minimum packet count  $N$ 
21.      $W = (N \times P_{max} - V - T \times \gamma) / N$  // Waste/packet
22.     If ( $W \leq W_{min}$ ) // If it is a better snapshot
23.        $W_{min} = W$ 
24.        $T_{best} = T$ 
25.        $\gamma_{min} = \gamma$ 
26.   End for
27. End for
28. Form the packets using  $T_{best}$  and  $N$  and send the data

```

Alg. 1: Pseudo code summary of the SPEO algorithm.

6. Validation experiments

To validate the effectiveness of our SPEO approach, we use a popular dataset from PhysioNet (Novak et al., 2010). The dataset contains 24 h ECG and EMG measurements, collected during patient monitoring. In the validation, we use the data for one patient. We compare the performance of SPEO with a popular compressive sensing technique from the literature (Rajoub, 2002). In fact, SPEO is deemed to be complementary, rather than competing, with data compression approaches. The main objective of the comparison is to

gauge the effectiveness of SPEO. In addition, we show the results when using data collected using the wearable multimodal electroacoustic (WearME) sensors. WearME is developed by LASARRUS LLC (Emokpae, 2020). It includes a digital stethoscope for acoustic cardiology (ACG), ECG monitor, temperature sensor and body posture tracker, the technical specifications of the WearME sensor are shown in the left portion of Fig. 5. The sensor is used to collect about 190,000 multi-modal ECG and ACG samples as shown in Fig. 5. In this paper, we primarily use the ECG recordings in our performance evaluation. The validation environment and performance results are discussed in the balance of this section.

6.1. Simulation environment and experiment setup

In order to study the performance of SPEO, the dataset is divided into two subsets for supporting the training and test phases. Assuming the current time is t , we want to predict the value at the next time epoch ($t + 1$) given the measurements for current and n previous epochs. The LSTM is trained for 100 epochs with a batch size of 1024 measurements of a single patient. After training LSTM, we extract the parameters of the model, specifically the weights W_i , W_o , and W_C . We have experimented with the number of LSTM cells and found the incorporation of 4 cells does yield the best results. We have also evaluated the effect of n , and observed no significant variation of the prediction accuracy when more than three samples are considered. Therefore, the reported results in this section are based on $n = 3$, i.e., the LSTM predicts s_k^{t+1} for a device k using $(s_k^{t-2}, s_k^{t-1}, s_k^t)$.

The performance of SPEO is compared with compressive sensing (Rajoub, 2002), which serves as a baseline. Such a compressive sensing approach is based on the discrete wavelet transform (DWT) with five decomposition levels. The DWT coefficients are divided into 3 groups; a threshold is set for each group based on a desired energy packing efficiency. To retain 95 % of the signal energy, the thresholds are set to: 99.9 % for the approximated band coefficients of level five, 97 % for the detail band coefficients of level five, and 85 % for detail sub-bands coefficients of levels 1–4. To determine the most significant coefficient for each level i , we: (i) calculate the energy of all coefficients, $EC_{i,j}$, (ii) sort $EC_{i,j}$ in descending order, and (iii) add $EC_{i,j}$ in the sorted list progressively until the desired thresholded energy corresponding to the respective coefficient level i (i.e., energy \times threshold $_i$) is reached. The remaining coefficients are below the threshold and thus will be insignificant. A binary significance map is then generated by scanning the wavelet decomposition coefficients and outputting a binary one if the scanned coefficient is significant, and a binary zero otherwise. Compression is achieved using direct binary representation of the significant coefficients. The reconstruction is obtained by uncompressing the signals using the map and computing an inverse discrete wavelet transform. Unless varied, the default window (buffer) size for the compression algorithm is set to 3000 ECG/EMG samples.

As noted earlier, SPEO can be viewed as a complementary approach to data compression techniques rather than a competitor. Therefore, we show the performance of the SPEO alone and in conjunction with data compression (baseline). The simulation opts to capture the effect of:

- **Buffer size:** This reflects the number of samples considered in the packet formation optimization.
- **Tolerable inaccuracy:** this is the prediction error threshold, which determines when a sample may be skipped. If the difference between a predicted sample and the actual data exceeds such a threshold, the sample needs to be transmitted

We measure the effectiveness of SPEO in terms of the following metrics:

- **The compression ratio:** it reflects the percentage of correctly predicted samples among the overall test set.
- **The number of required data packets:** This metric assesses the effectiveness in terms of reduced packet transmissions, which translate to energy saving and bandwidth efficiency.

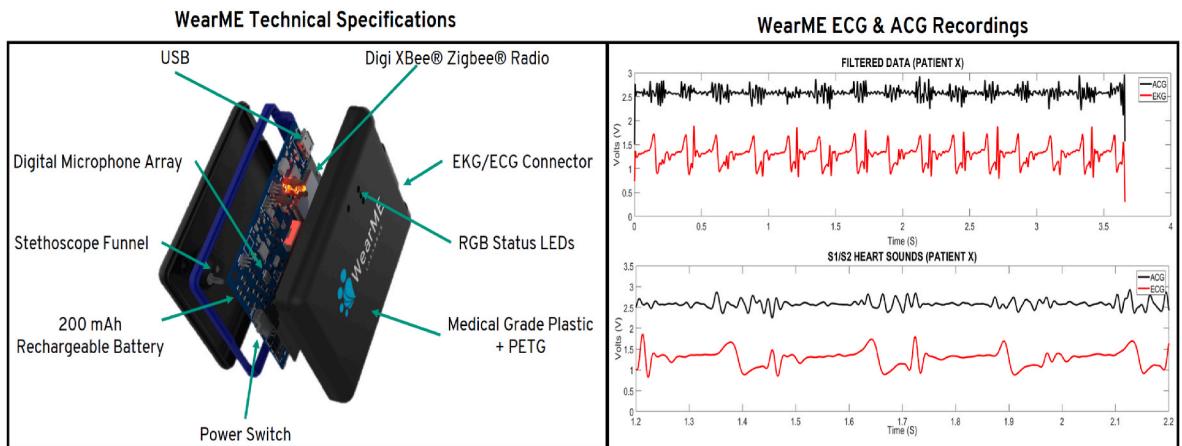


Fig. 5. Technical Specifications of each WearME sensor, which can be used for ECG, ACG, EMG and AMG (left image). The right image shows data recording of both ECG and ACG from a healthy human, which shows synchronous recording of electrical activity with S1 and S2 heart sounds. In this paper, we mainly use the ECG recordings for our SPEO validations.

- **The cross correlation ratio:** This metric opts to capture the effect of inaccuracy caused by sample prediction and quantization by measuring the maximum divergence of the reconstructed signal from the original. To calculate such a measure, we use the cross-correlation function of MATLAB.
- **Energy Consumption:** The energy consumption of nodes in the network is mainly due to communication and data processing. We can define the energy (in joules or J) required for transmission as the change in power ΔP time's change in time Δt . This is effectively the sum of the transmission electrical energy $E_{ELEC-TX}$ and the reception energy $E_{ELEC-RX}$, i.e.

$$E_{TX} = \Delta P \cdot \Delta t = E_{ELEC-TX} + E_{ELEC-RX} \quad (15)$$

The Digi XBee 3 Zigbee radios assumed in our simulation have: (i) a transmit power of 90 mW, (ii) maximum raw data throughput of 250 k Bits per second, and (iii) 84 Bytes data payload size, which is the maximum for broadcast messages. Thus,

$$\text{The energy per bit} = 90mW/250,000 \text{ bit/s} = 360 \text{ nJ/bit} \quad (16)$$

$$\text{The energy per packet} = 360nJ \times 8(\text{from byte to byte}) \times 84(\text{packets size in bytes}) \times \#\text{packets} \quad (17)$$

The computation energy is based on using an Arduino platform. The Arduino microcontroller used in our experiments has an active current of 1.23 mA when clocked at 16 MHz. The average power consumed during processing is approximately 5 mW, which is an order of magnitude less than the power consumed due to communication. We have used valgrind profiler with verrou tools to estimate the set and the number of instructions for the applied algorithms while handling the same number of samples (e.g., 50,000 ECG samples). Such instruction count is further multiplied by the published figures for the number of cycles per instruction to estimate the runtime. Finally, the computation energy is calculated by multiplying the runtime by the power.

6.2. Simulation results

SPEO performance: Fig. 6 reports the achieved compression ratio for SPEO for the ECG and EMG datasets under varying buffer sizes and for four settings of tolerable inaccuracy, namely, 10^{-4} , 10^{-5} , 10^{-6} , and 10^{-7} . Note that the original ECG signal also contains motion artifacts as is represented in the fourth peak after the QRS interval. Since SPEO can work for different modalities, we have applied the method of (Harshada et al., 2018) to suppress these artifacts. Fig. 6 also captures the performance of compressive sensing when applied along and in conjunction with SPEO. Overall, increasing the threshold for tolerable errors enables SPEO to skip more samples and achieve higher compression ratios. Particularly, requiring the predicted sample to deviate by no more than 10^{-7} is quite restrictive and limits sample transmission reduction to only 28 % and 50 % for ECG and EMG, respectively. In fact, the reconstructed ECG signal under 10^{-7} accuracy constraint, matches exactly the real ECG signal, as shown in Fig. 7(a). Tolerating inaccuracy up to 10^{-5} boosts the ECG sample skipping effectiveness of SPEO to about 80 %. The difference between the performance of 10^{-5} and 10^{-4} is not much, yet can be leveraged if acceptable to the application, i.e., does not hinder the physician's ability in assessing the patient's condition. SPEO proved to be even more effective for EMG samples when comparing Fig. 6(a) and (b). In fact, for error tolerance of 10^{-5} , the effectiveness of SPEO for EMG improves to 98 %, which is quite remarkable. Such reduction ratio diminishes significantly with constrained accuracies. It is important to note here that in the considered dataset, the standard deviation of values for an EMG signal is significantly less than that of the ECG samples; when scaling into [0, 1], the difference between readings in the 4th and even the 5th decimal is not much and consequently SPEO will skip many samples and the effectiveness becomes very high. This is unlike the case of ECG where the signal range is much higher, as seen in Fig. 7 and hence the tolerable error is more impactful. Fig. 6 (c) shows the results when applied to ECG samples collected by the WearME sensors. For this experiment, 160,000 and 30,000 samples are used for training and testing, respectively. Again, SPEO enables a significant compression ratio and yields consistent results with the published ECG dataset; we note that the sampling mismatch between our WearME sensors to published ECG data sets led to a reduction in

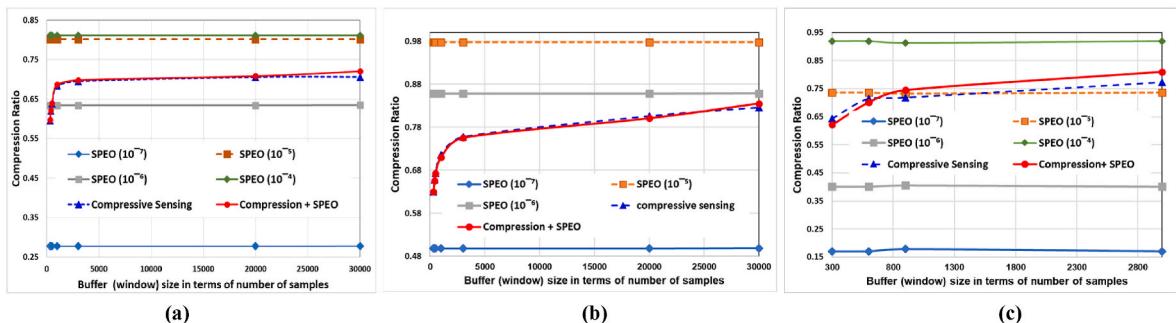


Fig. 6. The effect of buffer size and tolerable inaccuracy on the ability of SPEO to predict and skip data samples for the: (a) ECG, and (b) EMG publicly available datasets. The combination of compressive sensing and SPEO is based on tolerance level of 10^{-7} . Part (c) shows the results for the ECG data we collected with the WearME sensors.

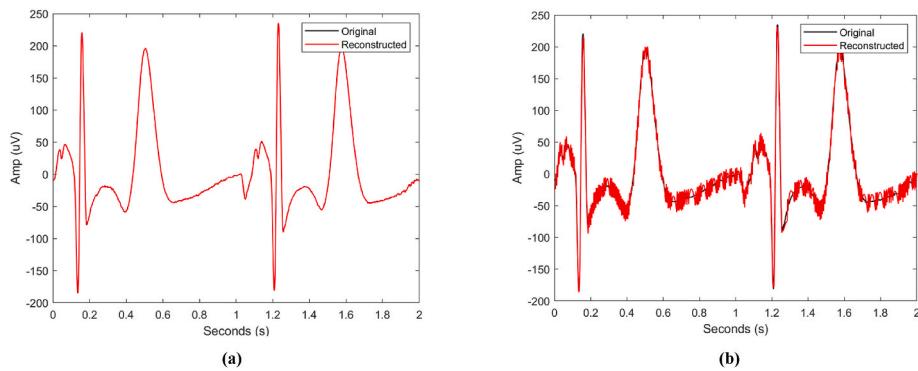


Fig. 7. When the accuracy has to be maintained up to: (a) 10^{-7} , SPEO enables reconstructing an exact match of the original ECG signal, yet with reduced transmission overhead; on the other hand, with (b) 10^{-4} the reconstructed signal keeps most characteristics of the original signal.

processing precision. Our WearME sensor operates at a sampling rate of 500 Hz, which is more than sufficient for ECG analysis (Pizzuti et al., 1985), while published data is sampled at rates greater than or equal to 1000 Hz. Fig. 7(b) shows the reconstructed signal for tolerable inaccuracy of 10^{-4} , which still resembles the original ECG signal. Fig. 8 compares the quality of the reconstructed signal relative to the original one by showing the cross-correlation ratio. As indicated by the figure, the accuracy requirement affects the quality of the reconstructed signal where an accuracy of up to 10^{-7} would ensure exact match, as also demonstrated in Fig. 7(a). It is interesting to note that setting the tolerable inaccuracy to 10^{-5} is quite effective in terms of compression ratio (Fig. 6) and still achieves a very high cross-correlation ratio. Similar conclusions could be made about the reconstructed EMG signals.

To further confirm the effectiveness of SPEO, we have generated 30 reconstructed ECG signals using SPEO with tolerable inaccuracy of 10^{-5} , and shown them to two physicians for review. Upon providing the physicians with the original ECG signals, the patterns have been 100 % consistent.

Comparison with Compressive Sensing: Fig. 6 also demonstrates the superiority of SPEO relative to contemporary compressive sensing. SPEO distinguishes itself with its ability to leverage application level inaccuracy tolerance. As shown in the figure, with sufficient tolerance SPEO is more effective than compressive sensing. When tolerating inaccuracy up to 10^{-5} SPEO outperforms compressive sensing by about 15 % for ECG; for the EMG dataset SPEO outperforms compressive sensing even with 10^{-6} . In addition, Fig. 8 shows that the quality of the reconstructed ECG signal in SPEO surpasses that of compressive sensing when the accuracy of predicted samples is high. Such superior quality can also be inferred when comparing the waveforms of Figs. 7 and 9. Again Figs. 6 and 8 collectively promote SPEO with a 10^{-5} and 10^{-6} prediction accuracy threshold as the best choice for ECG and EMG data, respectively. The results of Fig. 6 also indicate that SPEO is complementary to compressive sensing and boosts its performance, especially for large window sizes. Using both compressive sensing and SPEO with error tolerance of 10^{-7} , we could achieve the same compression ratio of compressive sensing, yet with reduced loss (as shown in Fig. 8). To apply the SPEO's sample prediction mechanism over the compressed signal, we have used the LSTM model with the same configuration, despite the fact that it is trained using compressed ECG signals, and predicted the compression coefficients.

Quantization performance: Fig. 10 reports the number of packets generated by SPEO using fixed and dynamic (optimized) quantization of the ECG and EMG data samples. The performance is studied for two settings of tolerable sample prediction inaccuracy,

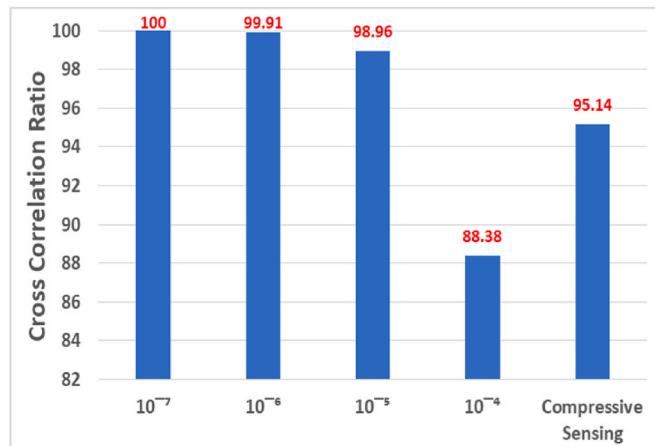


Fig. 8. Comparing the similarity between the original ECG signal and the reconstructed one using SPEO for a prediction accuracy of 10^{-7} , 10^{-6} , 10^{-5} , and 10^{-4} , as well as compressive sensing.

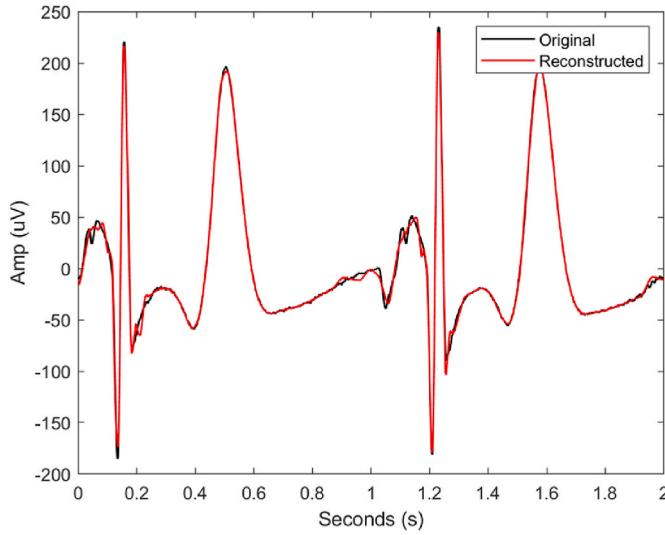


Fig. 9. Comparing the original and reconstructed ECG signal when compressive sensing is applied to retain 95 % of the signal energy.

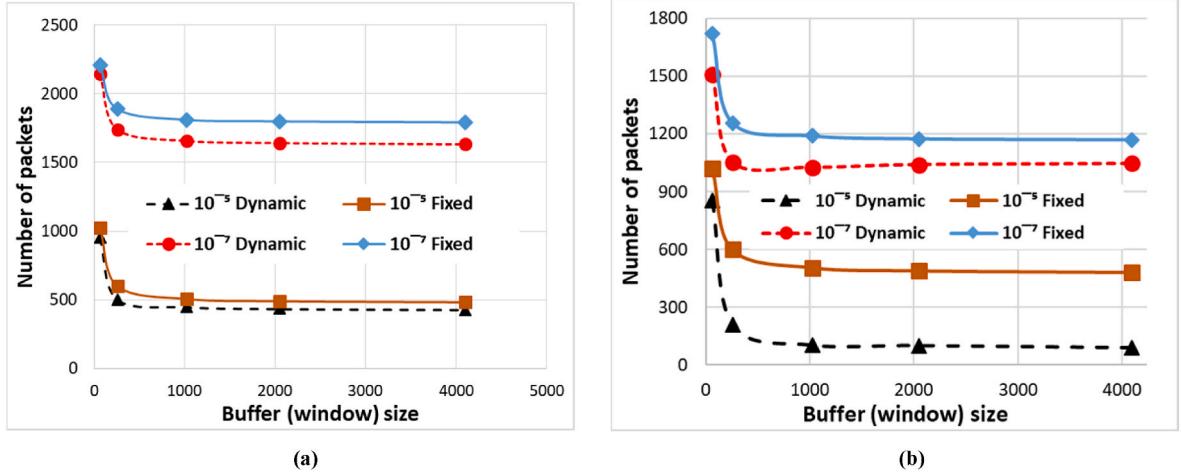


Fig. 10. The impact of the quantization optimization of SPEO on the number of generated packets while varying the buffer size and tolerable sample prediction inaccuracy, for (a) ECG, and (b) EMG datasets.

namely, 10^{-7} and 10^{-5} . For fixed quantization, the number of bits for representing a sample is set to 32 and 28, for error tolerance of 10^{-7} and 10^{-5} , respectively. These settings reflect the least value of S for the considered ECG and EMG datasets and accuracy requirements. With fixed quantization, the size of S is assumed to be known to the gateway by default and is not thus included in the packet. For the dynamic quantization configuration, control information is further included in packets to indicate the number of bits used for quantization, as explained in Section 5. We vary the size of the buffering window and compare the number of packets generated.

As seen in Fig. 10(a), the dynamic quantization setting decreases the packet count for the ECG dataset by about 10 % and 5 % for error tolerance 10^{-7} and 10^{-5} , respectively. Dynamic quantization is more impactful for the EMG dataset, as indicated by Fig. 10(b), given the precision and range of the measurements, which also explains the lower overall packet count. Obviously, the drop in packet count is due to the use of fewer bits to represent data samples. The impact of dynamic quantization is less in the case of 10^{-5} accuracy, since the fixed setting is 28 bits, which is already small. In addition, for the case of 10^{-7} the number of predicted samples is much less than when more error tolerance is allowed in the 10^{-5} configuration, and consequently there are more samples to transmit and benefit from quantization optimization. Finally, we note that the increased buffer size has a positive impact on the number of packets, mainly due to the increased flexibility in packet formation, as noted in Section 5, and the increased compression ratio, as indicated by Fig. 6. It is worth mentioning that the effect of growing the buffer size diminishes after 1000, since the picked snapshot T size almost fully utilizes the packet payload and stays unchanged as long as it is less than the buffer size.

Finally, Fig. 11 reports the consumed energy. The transmission energy is calculated using equations (16) and (17). The computation

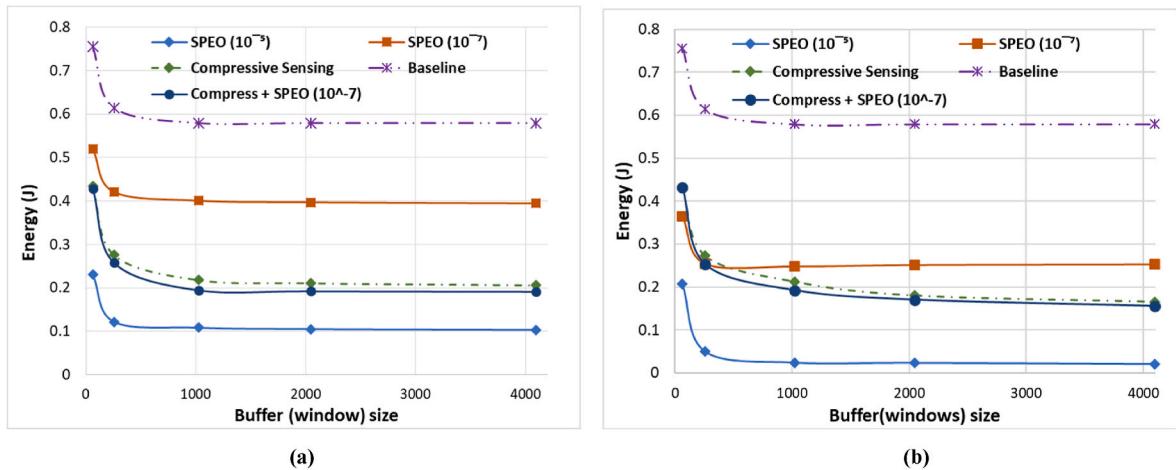


Fig. 11. Capturing the energy savings achieved by SPEO when applied to (a) ECG and (b) EMG dataset, in comparison to compressive sensing and to the baseline case where no optimization is applied, i.e., all samples are transmitted.

energy is estimated based on an Arduino microcontroller with the aforementioned configuration. The estimated runtime duration for predicting the data sample by our LSTM is approximately 0.4496 ms, which is much less than the sampling rate required for ECG and EMG. The runtime is multiplied by the power to calculate the energy consumed in computation. The performance of the SPEO is found to be mainly dominated by the communication overhead. Overall, we have observed that the computational overhead is quite insignificant and is about 1 % of that of communication. This is because the employed LSTM model is simple and only involves 101 trainable parameters. The model is to be trained offline and hence light computational overhead is experienced during BASN usage for patient monitoring. The consumed energy results are consistent with Fig. 10 and confirm the effectiveness of SPEO. Overall, SPEO achieves dramatic energy savings due to reduced data size and optimized quantization, where more than 6 times reduction in communication energy could be made when applying SPEO with tolerable accuracy of 10^{-5} . Fig. 11 also highlights the significant impact of the tolerance inaccuracy on the performance of SPEO, as we pointed out earlier. It is thus recommended for the application developer to exploit the tradeoff between accuracy and energy consumption to take full advantage of SPEO and maximize the network lifetime. It is worth emphasizing that SPEO is not an alternative to compressive sensing and can both be applied, where combining compressive sensing with SPEO (accuracy of 10^{-7}) yields performance that surpasses each of them individually.

7. Conclusions and future work

This paper has presented SPEO, a novel sample prediction based energy optimization approach for miniaturized health monitoring devices. SPEO employs advanced machine learning techniques to predict data samples and consequently reduces the volume and frequency of wireless transmissions. SPEO duplicates the sample prediction mechanism over the wearable nodes and the gateway node in order to achieve consensus on the omitted data while monitoring the accurate signal reconstruction. SPEO also employs an effective packet formation and data sample quantization techniques for reducing the number of required packets, and consequently increasing spectrum and energy efficiency, while sustaining desired medical assessment accuracy. SPEO is validated using popular ECG dataset. The validation results have demonstrated that SPEO outperforms contemporary compressive sensing, yet can be used in conjunction and complement its performance. The results have also demonstrated the effectiveness of SPEO in terms of compression ratio and reduced packet count. Furthermore, the quality of the ECG has been validated using cross-correlation with the original signals and through inspection by a medical expert. In the future, we plan to extend SPEO to exploit the correlation between multimodality sensing in order to further reduce the traffic in BASN networks.

Author statement

Wassila Lalouani: Conceptualization, Methodology, Software, Validation, Formal analysis, Writing – original draft, Mohamed Younis: Conceptualization, Methodology, Formal analysis, Writing – review & editing, Supervision, Funding acquisition, Lloyd E. Emokpae: Conceptualization, Methodology, Formal analysis, Writing – review & editing, Supervision, Funding acquisition, Ian White-Gittens: Data curation, Validation, Roland N. Emokpae: Data curation, Validation, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grants No. 1912945 and 2030629.

References

- Abiodun, A. S., Anisi, M. H., Ali, I., Akhunzada, A., & Khan, M. K. (2017). Reducing power consumption in wireless body area networks: A novel data segregation and classification technique. *IEEE Consumer Electronics Magazine*, 6(4), 38–47.
- Abrishami, H., Campbell, M., Han, C., Czosek, R., & Zhou, X. (2018). P-QRS-T localization in ECG using deep learning. In *The proc. IEEE EMBS international conference on biomedical & health informatics (BHI)* (pp. 210–213).
- Al Disi, M., Djelouat, H., Kotroni, C., Politis, E., Amira, A., Bensaali, F., & Alinier, G. (2018). ECG signal reconstruction on the IoT-gateway and efficacy of compressive sensing under real-time constraints. *IEEE Access*, 6, 69130–69140.
- Ali, M., Moungla, H., Younis, M., & Mehaoua, A. (2019). In H. Ammari (Ed.), *Interference mitigation techniques in wireless body area networks. Book chapter, the philosophy of mission-oriented wireless sensor networks*. Springer.
- Beltramelli, L., Mahmood, A., Österberg, P., Gidlund, M., Ferrari, P., & Sisinni, E. (2021). Energy efficiency of slotted LoRaWAN communication with out-of-band synchronization. *IEEE Transactions on Instrumentation and Measurement*, 70, 1–11.
- Bianchi, F. M., Maiorino, E., Kampffmeyer, M. C., Rizzi, A., & Janssen, R. (2017). *Recurrent neural networks for short-term load forecasting*. SpringerBriefs in Computer Science. <https://www.springer.com/gp/book/9783319703374>.
- Djelouat, H., Al Disi, M., Boukhenoufa, I., Amira, A., Bensaali, F., Kotronis, C., ... Dimitrakopoulos, G. (2020). Real-time ECG monitoring using compressive sensing on a heterogeneous multicore edge-device. *Microprocessors and Microsystems*, 72, 102839, 2020.
- Djelouat, H., Amira, A., & Bensaali, F. (2018). Compressive sensing-based IoT applications: A review. *Journal of Sensor and Actuator Networks*, 7(4), 45–75. October 2018.
- Dehkordi, S. A., Farajzadeh, K., Rezazadeh, J., Farahbakhsh, R., Sandrasegaran, K., & Dehkordi, M. A. (2020). A survey on data aggregation techniques in IoT sensor networks. *Wireless Networks*, 26, 1243–1263.
- Emokpae, L. E. (2020). Body area sensor network bio-feedback system. *U.S. Patent*, 16, 861, 824, filed April 29, 2020 (pending).
- Gradl, S., Kugler, P., Lohmüller, C., & Eskoer, B. (2012). Real-time ECG monitoring and arrhythmia detection using Android-based mobile devices. In *Proc. 34th IEEE annual international engineering in medicine and biology conference (EMBC)* (pp. 2452–2455).
- Harshada, C., Patil, N., & Sinha, V. (2018). Online suppression of motion artifacts during treadmill ECG. *Int. J. Eng. Sci. Comp.*, 8(1), 15965–15967.
- Kanoun, K., Mamaghanian, H., Khaled, N., & Atienza, D. (2011). A real-time compressed sensing-based personal electrocardiogram monitoring system. In *Proc. Design, automation test in europe*. DATE 2011.
- Karray, F., Jmal, M. W., Garcia-Ortiz, A., Abid, M., & Obeid, A. M. (2018). A comprehensive survey on wireless sensor node hardware platforms. *Computer Networks*, 144, 89–110.
- Kaur, N., & Sood, S. K. (2017). An energy-efficient architecture for the Internet of Things (IoT). *IEEE Systems Journal*, 11(2), 796–805.
- Khosravy, M., & Duque, N. D. C. (2020). *Compressive sensing in health care* (1st ed.). Academic Press.
- Ma, D., Lan, G., Hassan, M., Hu, W., & Das, S. K. (2020). Sensing, computing, and communications for energy harvesting IoTs: A survey. *IEEE Communications Surveys & Tutorials*, 22(2), 1222–1250.
- Moustafa, H., Schooler, E. M., Shen, G., & Kamath, S. (2016). Remote monitoring and medical devices control in eHealth. In *The proc. 12th international conference on wireless mobile computing. Network, and. Communication (WiMob)*.
- Nadeem, A., Hussain, M. A., Owais, O., Salam, A., Iqbal, S., & Ahsan, K. (2015). Application specific study, analysis and classification of body area wireless sensor network applications. *Computer Networks*, 83, 363–380.
- Niknejad, N., Ismail, W. B., Mardani, A., Liao, H., & Ghani, I. (2020). A comprehensive overview of smart wearables: The state of the art literature, recent advances, and future challenges. *Engineering Applications of Artificial Intelligence*, 90, 103529.
- Novak, V., Hu, K., Desrochers, L., Novak, P., Caplan, L., Lipsitz, L., & Selim, M. (2010). Cerebral flow velocities during daily activities depend on blood pressure in patients with chronic ischemic infarctions. *Stroke-A Journal of Cerebral Circulation*, 41(1), 61–66.
- Olah, C. (2015). Understanding LSTM networks. Colah's blog. *Github*, 27. Aug. 2015.
- Pizzuti, G. P., Cifaldi, S., & Nolfe, G. (1985). Digital sampling rate and ECG analysis. *Journal of Biomedical Engineering*, 7(3), 247–250.
- Qaisar, S., Bilal, R. M., Iqbal, W., Naureen, M., & Lee, S. (2013). Compressive sensing: From theory to applications, a survey. *Journal of Communications and Networks*, 15(5), 443–456.
- Qureshi, K. N., Dine, S., Jeon, G., & Piccialli, F. (2020). Link quality and energy utilization based preferable next hop selection routing for wireless body area networks. *Computer Communications*, 149, 382–392.
- Rahmani, A. M., Gia, T. N., Negash, B., Anzanpour, A., Azimi, I., Jiang, M., & Liljeberg, P. (2018). Exploiting smart e-health gateways at the edge of healthcare internet-of-things: A fog computing approach. *Future Generation Computer Systems*, 78, 641–658.
- Raj, A. S., & Chinnadurai, M. (2020). Energy efficient routing algorithm in wireless body area networks for smart wearable patches. *Computer Communications*, 153, 85–94.
- Rajoub, B. (2002). An efficient coding algorithm for the compression of ECG signals using the wavelet transform. *IEEE Transactions on Bio-Medical Engineering*, 49, 355–362.
- Raval, M., Bhardwaj, S., Aravelli, A., Dofe, J., & Gohel, H. (2021). Smart energy optimization for massive IoT using artificial intelligence. *Internet of Things*, 13, 100354.
- Riazul Islam, S. M., Kwak, D., Humaun Kabir, M., Hossain, M., & Kwak, K.-S. (2015). The Internet of Things for health care: A comprehensive survey. *IEEE Access*, 3, 678–708.
- Wu, Z., Ding, X., Zhang, G., Xu, X., Wang, X., Tao, Y., & Ju, C. (2016). A novel features learning method for ECG arrhythmias using deep belief networks. In *Proc. IEEE 6th international conference on digital home (ICDH)* (pp. 192–196).
- Yan, P., Choudhury, S., Al-Turjman, F., & Al-Oqily, I. (2020). An energy-efficient topology control algorithm for optimizing the lifetime of wireless ad-hoc IoT networks in 5G and B5G. *Computer Communications*, 159, 83–96.
- Yang, X., Wang, L., & Zhang, Z. (2018). Wireless body area networks MAC protocol for energy efficiency and extending lifetime. *IEEE Sensors Letters*, 2(1), 1–4.
- Yildirim, Ö., Tan, R. S., & Acharya, U. R. (2018). An efficient compression of ECG signals using deep convolutional autoencoders. *Cognitive Systems Research*, 52, 198–211.
- Zemouri, R., Zerhouni, N., & Racooceanu, D. (2019). Deep learning in the biomedical applications: Recent and future status. *Applied Sciences*, 9(8), 1526.
- Zhang, Z., & Rao, B. D. (2013). Extension of SBL algorithms for the recovery of block sparse signals with intra-block correlation. *IEEE Transactions on Signal Processing*, 61(8), 2009–2015.