UNIVERSITY OF CALIFORNIA,
IRVINE

Energy and Bandwidth Efficient Edge Computing for the Internet of Healthcare Things

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Electrical Engineering and Computer Science

by

Delaram Amiri

Dissertation Committee:
Associate Professor Marco Levorato, Chair
Professor Nikil Dutt
Professor Ramesh Jain

2020

DEDICATION

This thesis is lovingly dedicated to my mother, whose patience, encouragement and constant love sustained me throughout my life;
my father who taught me how to be consistent; and my best friend, my sister, who was with me in every step of my life.

TABLE OF CONTENTS

LIST OF FIGURES

Page

LIST OF TABLES

List of Algorithms

ACKNOWLEDGMENTS

I would like to thank my adviser, Professor Marco Levorato for providing guidance and feedback on my research work.

I thank my committee members , Professor Nikil Dutt, and Professor Ramesh Jain. I thank collaborators Professor Amir M. Rahmani, Professor Pasi Liljeberg for their constructive feedback. I thank the colleagues from Department of Future Technologies at University of Turku, and VTT Technical Research Centre of Finland.

My PhD research is based upon work supported partially by the US National Science Foundation (NSF) WiFiUS grant CNS-1702950 and Academy of Finland grants 311764 and 311765. Data used for this research was provided by MIT-BIH Arrhythmia Database and MIT-BIH Supraventricular Arrhythmia Database of Beth Israel Deaconess Medical Center and Massachusetts Institute of Technology.

VITA

Delaram Amiri

EDUCATION

Doctor of Philosophy in Electrical Engineering	**2020**
University of California, Irvine	*Irvine, California*
Master of Science in Electrical Engineering	**2015**
Purdue University	*Indianapolis, Indiana*
Bachelor of Science in Electrical Engineering	**2013**
Shiraz University	*Shiraz, Iran*

RESEARCH EXPERIENCE

Graduate Research Assistant	**2015–2020**
University of California, Irvine	*Irvine, California*
Graduate Research Assistant	**2013-2015**
Purdue University	*Indianapolis, Indiana*

PROFESSIONAL EXPERIENCE

Communications Systems Intern	**Summer 2018**
Maxlinear Inc., Irvine	*Irvine, California*
Communications Systems Intern	**Fall 2017**
Maxlinear Inc., Irvine	*Irvine, California*

TEACHING EXPERIENCE

Teaching Assistant	**2015–2020**
University of California, Irvine	*Irvine, California*
Lecturer	**2016**
University of California, Irvine	*Irvine, California*
Teaching Assistant	**2013–2015**
Purdue University	*Indianapolis, Indiana*
Lecturer	**2015**
Purdue University	*Indianapolis, Indiana*

REFEREED BOOK CHAPTER PUBLICATIONS

Optimizing Energy Efficiency of Wearable Sensors Using Fog-assisted Control **2020**
Fog Computing: Theory and Practice

REFEREED JOURNAL PUBLICATIONS

Optimal Accuracy Bandwidth Tradeoff for ECG Classification via Sequential Hypothesis Testing **2020**
ACM Transactions on Computing for healthcare(under review)

Edge-Assisted Control for Healthcare IoT Edge-Assisted Control for Healthcare Internet-of-Things: A Case Study on PPG-based Early Warning Score **2020**
ACM Transactions on Internet of Things

Context-Aware Sensing via Dynamic Programming for Edge-Assisted Wearable Systems **2019**
ACM Transactions on Computing for healthcare

REFEREED CONFERENCE PUBLICATIONS

Edge-Assisted Sensor Control in Healthcare IoT **Dec 2018**
2018 IEEE Globecom Communications Conference

REFEREED POSTER PRESENTATIONS

Energy Efficiency and Context-awareness in healthcare-Iot using fog computing **Jun 2019**
16th International Systems on Chip Conference (first place in student contest)

Energy Efficient Edge Computing in healthcare IoT **Oct 2018**
2019 Design Automation Conference

Edge-assisted Sensor Control in Health-care IoT **Feb 2018**
Information Theory an Applications Workshop (ITA)

SOFTWARE

Python `https://www.python.org`
An algorithm to solve sequential testing and classification

Markov Decision Processes Toolbox https://doi.org/10.1111/ecog.00888
Matlab tool to solve Linear programming

ABSTRACT OF THE DISSERTATION

Energy and Bandwidth Efficient Edge Computing for the Internet of Healthcare Things

By

Delaram Amiri

Doctor of Philosophy in Electrical Engineering and Computer Science

University of California, Irvine, 2020

Associate Professor Marco Levorato, Chair

Recent advances in the Internet of Things (IoT) technologies have enabled the use of wearables for remote patient monitoring. Wearable sensors capture the patient's vital signs, and provide alerts or diagnosis based on the collected data. Unfortunately, wearables typically have limited energy and bandwidth, making their use challenging for healthcare applications where monitoring must continue uninterrupted long time, without the need to charge or change the battery. Edge computing can alleviate these problems by offloading computationally intensive tasks from the sensor layer to edge servers, thereby not only meeting the sensors' limited computational capacity but also enabling the use of local closed-loop energy and bandwidth optimization algorithms to increase the battery life. By incorporating the patient's contextual information, a desired quality of experience can be achieved by creating a dynamic balance between energy-efficiency and measurement accuracy. In addition, transmitting representation of a signal instead of the raw signal can minimize the bandwidth consumption of the sensor. We first present a run-time distributed control-based solution to find the most energy-efficient system state for a given context while keeping the accuracy of decision making process over a certain threshold. We propose two approaches, *myopic* and Markov Decision Processes (MDP) to consider both energy constraints and risk factor requirements for achieving a two-fold goal: energy savings while satisfying accuracy requirements of abnormality detection in a patient's vital signs. Vital signs, including heart rate, respiration rate and oxygen saturation (SpO_2), are extracted from a Photoplethysmogram (PPG) signal and errors of extracted features are compared to a ground truth that is modeled as a Gaussian distribution. We control the sensor's sensing energy to minimize the power consumption while meeting a desired level of satisfactory detection performance. We present experimental results on realistic case studies using a reconfigurable PPG sensor in an IoT system, and show that compared to non-adaptive methods, *myopic* reduces an average of 16.9% in sensing energy consumption with the maximum probability of abnormality misdetection on the order of 0.17 in a 24-hour health monitoring system. In addition, during four weeks of monitoring, we demonstrate that our MDP policy can extend the battery life on average of more than 2x while fulfilling the same average probability of misdetection comparing to *myopic* method. We illustrate results comparing, *myopic*, MDP and non-adaptive methods to monitor 14 subjects during one month. Implementation of *myopic*, and Markov Decision Processes scheme are based upon implementation of optimization schemes using Matlab coding. We then, propose a framework to dynamically optimize the resolution of an electrocardiogram (ECG) signal transferred to an edge server to detect heart diseases in real-time. Our objective is two fold: maximizing the detection probability of abnormal ECG cycles while satisfying requirements on wireless channel usage. Based on sequential hypothesis testing and hierarchical wavelet representation, the framework establishes a control loop between the sensor and the edge server to iteratively determine the subset of wavelet coefficients of the ECG heart cycle based on the signal itself. Numerical results show that the sequential hypothesis testing selects different subsets for normal and abnormal ECG cycles. In addition, we show that the signal representation should be personalized, as the system selects different sets of coefficients for different patients. Results over 14 subjects indicate that average channel usage is reduced an average of 43% over non-adaptive optimization to achieve the same accuracy of classification. This work is implemented using Convolutional Neural Network using TensorFlow library and Python programming to achieve the goal.

Chapter 1

Introduction

Wearable devices have enabled out of hospital monitoring of patients in everyday settings. In fact, the use of health monitoring post-operative patients can predict and prevent deterioration and death in chronic patients and improve care outcomes. Industrial technologies can only fulfill the needs of in-hospital settings, where the patients are monitored in a controlled environment and situation. However, when the patients are discharged from the hospital, they are left vulnerable without proper monitoring. Achieving clinical-level monitoring in everyday settings would have a tremendous impact on patients health, but is a technological challenge that has not been solved yet.

Internet of healthcare Things (IoT) technologies have been proposed to provide continuously monitoring subjects, acquiring a variety of biosignals [20, 42, 28]. However, these technologies have several limitations that make continuous high-quality sensing difficult. In fact, these biomedical sensors have limitations in terms of storage, computation load, bandwidth consumption and energy supply. Furthermore, different from hospital environments, in everyday settings the activities the monitored subject engages may cause a degradation of the quality of the signals due to the movement between the skin and the sensor. Some wearable sensors, such as, Photoplethysmogram (PPG), and Electrocardiogram (ECG) are particularly influenced by this effect and may suffer a significant loss of accuracy [5].

Interestingly, different activities may cause a different degree of degradation. For instance, motion artifacts affecting the measured signal when the subject is "Running" are much larger compared to those generated when the person is "Sitting" or "Sleeping". Achieving the same Signal-to-Noise Ratio (SNR) in all the activities and, thus, the same detection quality, requires the tuning of the sensing power to that of the noise. In other words, higher sensing energy could be used when the patient is "Running", and a lower energy could be used when the patient is "Sitting". Nonetheless, wearable sensors are manufactured in industry for worst case scenario in terms of noise level, corresponding to a large sensing power, which results in a high energy consumption and a short battery lifetime. The layered and pervasive IoT infrastructure can be used to support solutions enabling the adaptation of sensing parameters to the joint person-technology system state. Herein, we extend this notion across the layers of the IoT infrastructure, and focus patient's activity as a main driver of adaptation due to the specific application domain.

In the computing literature, the general notion of context captures the state of the system, including any descriptor of the user's state. User's activity is considered as a subclass of context information, whose estimation and tracking requires the acquisition of signals from sensors and the implementation of inference algorithms. Intuitively, imposing a further burden on the sensor to have them connect with each other and implement algorithms is not a suitable option. Therefore, the needed "semantic" support necessarily comes from the higher layers of the IoT infrastructure, which has sufficient resources to perform signal fusion and processing. In particular, fog and edge computing resources [53, 52], which are connected to the mobile devices through 1-hop low latency wireless links, are particularly indicated to host compute-intense tasks informing system-level control. Implementing optimization algorithms controlling sensors on the edge devices can avoid imposing overheads to the sensor layer while being able to rapidly respond to changes thanks to the local control. Edge computing is an architecture that takes advantage of sensors interoperating with the edge layer to store, and process the data and coordinating with the sensor regularly to send proper configuration settings. Settings such as, sampling rate, sensing energy level, duration of sensing, and scheduling sleeping times are among the sensor settings that can be configured to impact the energy efficiency.

In this thesis, we focus on two aspects of “bandwidth” and “energy” efficiency of sensors in healthcare IoT. We propose approaches using the real-time connection between the sensors and the edge device to build a context-aware control loop determining the sensing power used by wearable sensors for PPG signal to optimize “energy” of sensors. To this aim, we develop two distinct strategies. The first strategy is *myopic*, meaning that the sensing power is selected solely based on the current activity. The linear time complexity of *myopic* results in a minimal communication between the edge and sensor layers as well as a lightweight computation at smart gateways making the local control at the edge significantly agile. Conversely, the second strategy builds a statistical model of person’s activities and battery State of Charge (SoC) dynamics, which are then used to locate decisions optimal over an extended period of time. Our proposed framework includes the following:

- A hardware prototype of a reconfigurable wireless PPG sensor node
- An edge-based adaptation algorithm for wearable sensors in the context of abnormality detection
- A Gaussian model of the sensing accuracy as a function of activity and input sensor current from real-world measurements to calculate the probability that an abnormal signal is not detected
- Formulation of an optimization problem that enables a trade-off between detection of abnormalities, and sensing energy consumption in the sensor.

In summary, this thesis makes the following contributions:

- An edge-based adaptation algorithm for wearable sensors in the context of abnormality detection
- A Gaussian model of the detection accuracy as a function of activity and input sensor current from real-world measurements.
- A myopic strategy optimizing the trade-off between detection of abnormalities and sensing energy consumption.
- A Markov Decision Processes (MDP) solution to track the time-evolving, physical activity of an individual along with the state of the battery.
- Extensive evaluation and results based on a real-world trial involving 20 pregnant women for seven months (ethically approved human subject trial).

In addition, we propose a form of adaptive compression for ECG signal to optimize "bandwidth" efficiency of ECG sensors. This work is novel since we are implementing a semantic methodology, meaning that the representation of ECG signal is tuned to the classification goal, rather than aiming at the accurate reconstruction of of the signal.

In summary, this thesis makes the following contributions regarding "bandwidth" efficiency in ECG sensors:

- We propose a an iterative and "semantic" form of compression guided by a Sequential Hypothesis Testing core and based on a wavelet representation of an ECG signal.
- We develop a Convolutional Neural Network model to detect 4 types of arrhythmia from the ECG signal harvested by a sensor. The CNN classifier achieves 4% of False Positive Rate in the detection of abnormal heart cycles when using the original ECG signal.

- We present and discuss extensive results based on real-world ECG trials.Two sets of data, the MIT-BIH arrhythmia database and MIT-BIH Supraventricular arrhythmia database(MIT-SVDB), are used to evaluate the system we propose. These data sets involve 63 subjects to train the Sequential Testing model and 14 subjects to test the proposed system. Numerical results show that the sequential hypothesis testing selects different sets of coefficients for normal and abnormal ECG cycles. In addition, we show that the signal representation is personalized, as the system selects different sets of coefficients for different patients. Results over 14 subjects indicate that average channel usage is reduced by up to 43% to non-adaptive optimization to achieve the same accuracy of classification.

Chapter 2 discusses state of the art in efficiency of IoT systems. Chapter 3 discusses the challenges in monitoring patients out of hospital, the IoT system architecture, and the benefits of edge computing in healthcare IoT. Chapter 4 is based upon work introduced as efficiency in bandwidth using edge computing [3, 7]. Chapter 5 is discussing the topic of bandwidth efficiency in edge computing [6]. Chapter 6 concludes the dissertation and discusses the future work.

Chapter 2

Literature Review

In this chapter, we focus on reviewing the current literature on energy and bandwidth efficiency methods. Wearable sensors are necessarily battery-powered devices. Thus, one of the key problems in continuous monitoring of vital signs acquired by these sensors is their limited energy supply. Algorithms to reduce the energy consumption of sensors have been proposed in previous work and widely studied. Recent research has begun to study the energy efficiency of IoT architectures. Scheduling sleeping intervals [70, 75, 74, 71, 29], determining duty cycles [1] in the sensor layer and implementing packet routing [68] are among the methods to deliver energy efficiency to the sensor networks. However, the state-of-the-art lacks the attention towards designing adaptive healthcare IoT architectures capable of controlling system level parameters based on the context of environment and the state of patients. In 2017, Kaur et al. [33] presented a solution to determine sleep intervals of sensors based on their residual battery level, usage history, and quality of the measured signal. In 2017, Tunic et al. [66] proposed to model as a Markov chain the charging/discharging process of sensors equipped with energy harvesting units. Based on the resulting model, the authors computed the outage probability in a finite time horizon.

Techniques to implement energy efficient networks for bio-sensors known as, Wireless Body Area Network (WBAN) are proposed. Chang et. al proposed a routing protocol for WBAN, considering expected transmission count and remaining energy units for optimal path selection [13]. Pradhan et. al compared energy consumption in four protocols, 802.15.4, IEEE 802.15.6, SMAC and TMAC as hybrid MAC protocols in healthcare [51].

All the above are exemplars of contributions addressing energy efficiency for IoT sensors. However, none of the authors focused on context-aware models. Herein, we propose a context-aware approach to optimize the energy consumption of the sensors without compromising the probability of detecting anomalous signals.

In 2012-2014, Zois et al. [76, 62, 77] proposed MDP based approaches to detect the activity of a monitored subject by means of imperfect observations acquired by noisy wearable sensors. The framework proposed in the papers, which is based on Partially Observable Markov Decision Processes, enable to explore the trade-off between energy consumption, sampling rate and estimation accuracy. Herein, we use a Markov chain [12] to model the activity, but use that information to build an optimal detector of anomalies in biosignals. Note that the two approaches could be merged to obtain a holistic activity estimation and anomaly detection.

There is a broad literature review on methods of saving bandwidth in wearable devices. Data compression in wearables has been studied widely in the literature. Hooshmand *et al.* [27] uses a lossy compression algorithm to decrease energy consumption in ECG sensors. Raju *et al.* [54] proposes a lossless data compression method to improve the compression ratio in wearable sensors. However, the aforementioned work does not study the impact of compression on the performance of abnormality detection in ECG signal. Many solutions to classify arrhythmia automatically already exist. Sayatan *et al.* [21] used Gaussian-Bernoulli deep belief network with support-vector machine (SVM), where three separate classification models were used to detect different kinds of arrhythmia types. Xu *et al.* [73] introduced using deep neural networks both in ECG classification and feature extraction. In this thesis, we focus on the detection/classification of arrhythmia in real-time and study methodologies to determine an optimal representation of signal sections by means of a collaboration across the system's layers.

Sequential Hypothesis Testing, the engine of the framework proposed here, is a decision-making problem formulated in a way that the agent can sequentially choose between several hypotheses based on the outcome of experiments. Karktik [32], proved theoretically that the agent can update the belief on a set of hypotheses by performing a fixed number of experiments. Naghshvar *et al.* [48, 49] proposed an M-ary active sequential testing using a strategy based on Dynamic Programming. Their aim was to minimize the probability of making a wrong decision while minimizing the time needed to make the decision, i.e., the stopping time. In the framework, a representation based on Partially Observable Markov Decision Processes (POMDP) is used. All the above examples focus on a single agent minimizing the probability of misclassification. In this thesis, we build on these results to optimize the communication strategy on a sensor-edge server system whose task is to classify an ECG signal. Importantly, the strategy is a function of the signal itself (e.g., normal vs abnormal), and signal statistics (influenced by the subject), thus resulting in a "personalized" decision maker making specific choices for different signal sections. Connected to our problem, Thakor [63] introduced Sequential Hypothesis Testing to detect abnormality by creating a sequence of binary representation based on a threshold. In other words, the ECG signal is compared to a threshold and a sequence of zeros would be created. Whenever the corresponding ECG signal takes values higher than the corresponding threshold, ones will be added to the sequence. Later, the sequential testing is used based on the binary sequence to detect abnormal ECG cycles from normal cycles. However, in this thesis, we are focusing on using the optimal stopping time to determine the optimal number of coefficients to represent an ECG signal as it is forwarded to an edge server and we denoise the signal with three different filters.

Sequential testing has been used in sensor networks, for instance to detect security threats to local networks. Wei [72] proposes a method to detect rogue Access Points in a Wireless Local Area Network (WLAN) by performing sequential testing on packet-header data. The use of Sequential Hypothesis Testing has been proposed in sensor networks to determine the most informative sensor. Li [37] sets the end goal to minimize the accumulated samples to the constraints of error probabilities. Li *et al.* [38], consider a setting where each sensor may observe different samples, and use a sequential test to determine how to accumulate information from neighboring sensors and performs. To the best of our knowledge, our work is the first using Sequential Hypothesis Testing in the context of healthcare applications.

Chapter 3

Design Challenges

We proposed our energy efficient system architecture based on PPG sensors. Herein, we discuss the limitations and challenges of implementing an IoT architecture based on a PPG sensor. As we discussed earlier, a remote patient monitoring system is expected to make the patients' vital signs available and collectible for healthcare professionals wirelessly. Furthermore, it was mentioned that the PPG signal is a good source for at least three vital signs. Even though the use of PPG signal is a good source of data, it imposes the system to several challenges. The first challenge is the behavior of the PPG signal in different contexts. The typical PPG signal displayed in Figure 3.1 is recordable in the hospital setting when the patient is sitting or lying on the bed without movement. This typical PPG signal consists of two parts: AC part which oscillates with each heartbeat and DC part which forms the baseline of the signal. The AC part is the result of the changes in the amount of oxygenated hemoglobin in the blood and the changes in the DC part are due to the pressure applied from other body tissues to the blood vessels. For example, during the respiration process, the change in the size of lungs applies pressure to blood vessels and causes a low-frequency oscillation in the DC part. These kind of changes in the DC part enable us to measure the respiration rate from the DC part oscillation frequency by filtering the AC part. Such

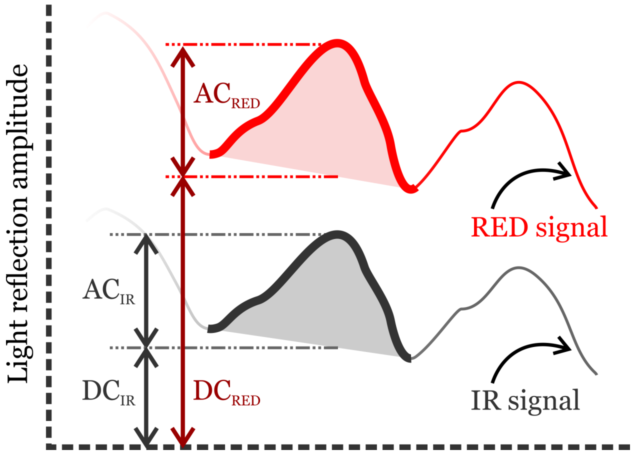


Figure 3.1: PPG waveforms and the four features extracted for SpO_2 calculation

changes in the DC part creates a challenge in signal processing when the patient has some activities. Each body movement causes a change in the DC part and more intense activities make larger changes in the signal baseline. When the amplitude of changes is larger than the amplitude of AC part, the detection of heartbeat peaks would be more difficult. In addition, when the body movements are rhythmic (e.g. walking, jogging, running) with a frequency close to respiration, the calculation of respiration rate is also rather difficult.

The other challenge in the PPG signal acquisition is the amount of noise in the signal. Ambient light diffuses to the exposed body tissues close to the sensor spot and causes a level of noise to the recorded signal. Although increasing the brightness of LEDs in the PPG sensor reduces the effect of ambient light noise, it increases also the power consumption in the sensor node.

The last PPG-related challenge is that the most parts of signal processing are not possible to carry out with low power microcontrollers of wearable sensor nodes. The sensor node should send the raw signal to a gateway or cloud server for further processing. This, in turn, requires more power for radio transmission.

In addition, we implemented bandwidth efficient algorithms using an ECG sensor. The ECG sensor is primarily used to record the heart cycle of a user. The different

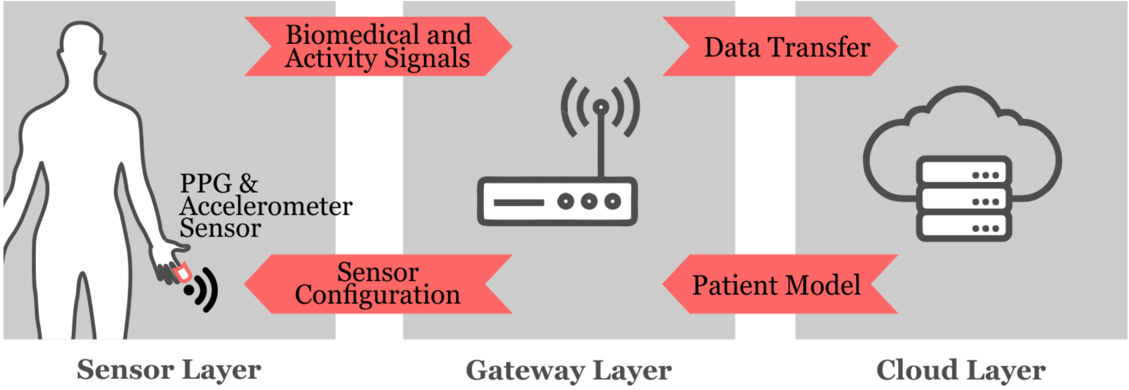


Figure 3.2: IoT system architecture

In the following sections, we describe the solutions to cover the mentioned challenges.

3.1 IoT System Architecture

Internet of Things is a term for describing methods that enable us to sense and control a variety of parameters and objects wirelessly through the Internet. Developing a system based on IoT has several benefits in comparison to traditional wireless sensors and remote control approaches. The most significant characteristic of IoT is its well-organized distribution of energy sources, data storage, and processing power. As shown in Figure 3.2 a common architecture of IoT-based remote health monitoring application consists of three layers. The first layer belongs to the sensor nodes that are recording and reporting the patient state including medical parameters, activity level, posture, location, and environment properties. Sensor nodes send the collected data to one or more gateway devices which are placed close to the patients so that energy cost of radio communication between sensors and gateways remains at a low level. In the gateway layer, the devices have their own storage and processing units powerful-enough to perform pre-processing and fog-computing actions before transferring data to the cloud server. The pre-processing actions may include data filtering, data fusion, data analysis, compression, and encryption. Fog-computing methods may be used to offload a portion of cloud-based tasks to the gateway device at the edge of the network to reduce the bandwidth need, data size, and server load. In the cloud layer, the server receives collected data from several gateways and stores the data recorded from all patients. Such a huge amount of data enables the cloud server to compare patients with their earlier conditions, with other patients in the same condition, and the consequences of the current condition in other patients. The server then would be able to learn from the patients' history and predict the future of patients' health.

In our setup, we use a battery-powered microcontroller connected to a digital I2C PPG sensor, a 3D accelerometer sensor, a temperature sensor, and a WiFi transmission module as our sensor devices. The amount of power required to drive LEDs in the PPG sensor is configurable, the sensors node is able to receive configurations from the gateway device remotely. The other configurable parameters are the recording and hibernation durations. The gateway is a WiFi-enabled Linux machine which receives the recorded data, performs fog-assisted optimization algorithms, decides about the configuration of the next recording period, and sends the new settings back to the sensor device. The cloud server gets updated with all collected data and updates the RMSE values for the gateway over longer periods of time.

3.2 Edge Computing in healthcare IoT

Fog computing leverages the concept of Geo-distribution of networks at the edge, enabling local/hierarchical data analysis, decision making, and storage [11, 52, 53]. Prevalence of connected devices and IoT-based systems demands an intermediary layer of computation, in which the local solutions provide low-latency responses, load balancing, and adaptivity for system behavior.

With the growth of IoT-based systems, a rapid increase in the number of connected devices has led to a massive volume of data that needs to be processed [15, 16]. Cloud computing has, thus far, provided scalable and on-demand storage and processing resources to fulfill the requirement of IoT. However, most recent IoT-based applications require mobility, low-latency response, and location-awareness [57]. Moreover, the latency of data transmission between the edge and the cloud is unsatisfactory especially in latency-sensitive systems such as health monitoring [16]. In this regards, Cisco states that "*Today's cloud models are not designed for the volume, variety, and velocity of data that the IoT generates.*" [59]. Therefore, fog computing can be considered as a complementary solution for the cloud computing paradigm to enable such latency demanding applications [40] as it can relocate location dependent, time-dependent, massive scale, and latency-sensitive tasks from the cloud server to the edge of the network [26].

Edge computing provides several lightweight services at the edge of the network, locally analyzing data collected from heterogeneous connected devices. Depending on the computational [55, 56] capacity of the edge servers or gateway devices, such edge-based services can include not only conventional tasks such as protocol conversion but also local data processing applications, some of which are outlined as follows. There is a variety of applications such as data filtering and data fusion to ensure high-level data quality at the edge, improving the data accuracy and performing data abstraction [53, 9]. Such applications can decrease the amount of data that should be sent to the cloud server and subsequently save external bandwidth. Moreover, local decision-making is a solution at the edge by which the system's availability and reliability are increased particularly when the Internet connection is poor [8]. Adaptive sensing and actuation is another application that intelligently tunes the system's configuration at the edge according to the context information [4]. Such a dynamic reconfiguration can considerably improve the system-driven quality attributes such as energy efficiency. Certain security related services can be also performed at the edge, protecting data from unauthorized access (e.g., authentication, data encryption/decryption, anomaly detection, etc.) [53, 58].

Chapter 4

Energy Efficient Edge-Assisted Control

The development of wearable systems for clinical-level continuous monitoring of patients in everyday setting presents some critical challenges: *(a)* the energy supply available to the sensors is severely constrained due to weight and size constraints; and *(b)* motion artifacts due to body movements diminishes the quality of the acquired signals [46]. Problem (b) especially affects sensors measuring vital signs such as Photoplethysmogram (PPG), Electrocardiogram (ECG), and Electromyography (EMG). The Signal-to-Noise-Ratio (SNR) of the signals can be improved by increasing the sensing energy. This strategy can further shorten the lifetime of the sensors.

The overarching objective of this chapter is to mitigate the issues mentioned above by proposing an edge computing-based architecture supporting a context-aware form of control of the sensors. Specifically, we observe that noise differently affects the collected signal as the monitored subject engages in different activities. For instance, if the person is sleeping, the noise power is much smaller compared to that affecting the same signal acquired when the person is running. Intuitively, knowing the current activity of the person would allow to set a corresponding sensing energy level to reach a desired SNR level.

Intuitively, wearable sensors are not suitable to host such complex reasoning, as detecting the activity would require to process another set of signals – for instance from accelerometers. Therefore we propose to use the edge servers as the central components of a distributed context-aware system: the edge server assisting the sensors will collect their signals, extract the context – in our case the activity of the monitored person – and control the sensing parameters using predefined messages. By adapting the sensing energy used by the sensors to the activity, the system can prolong the lifetime of the sensors, thus improving the integrity of the acquired signal over long periods of time, without compromising the quality of the signal. Based on this general concept and architecture, we fully develop a specific application whose objective is to detect abnormalities in vital signs extracted from PPG sensors [61, 4, 7]. PPG is a low-cost and miniaturized optical sensor widely used in medical and wearable sensors (e.g., fitness trackers, smart rings, smart earrings) which can continuously capture several vital signs such as heart rate, heart rate variability, respiration rate, and blood oxygenation [46]. Based on real-world data, we build a model for normal and abnormal signals, and define corresponding regions in a feature space. In this context, the edge server will detect the current activity of the monitor person and adjust the power used by the PPG sensor to ensure that the misdetection probability is below a predetermined threshold while maximizing the lifetime of the sensor.

To this aim, we develop two distinct strategies. The first strategy is *myopic*, meaning that the sensing power is selected solely based on the current activity. The linear time complexity of *myopic* results in a minimal communication between the edge and sensor layers as well as a lightweight computation at smart gateways making the local control at the edge significantly agile. Conversely, the second strategy builds a statistical model of person's activities and battery State of Charge (SoC) dynamics, which are then used to locate decisions optimal over an extended period of time. Our proposed framework includes the following:

- A hardware prototype of a reconfigurable wireless PPG sensor node
- An edge-based adaptation algorithm for wearable sensors in the context of abnormality detection
- A Gaussian model of the sensing accuracy as a function of activity and input sensor current from real-world measurements to calculate the probability that an abnormal signal is not detected
- Formulation of an optimization problem that enables a trade-off between detection of abnormalities, and sensing energy consumption in the sensor.

In summary, this chapter makes the following contributions:

- An edge-based adaptation algorithm for wearable sensors in the context of abnormality detection
- A Gaussian model of the detection accuracy as a function of activity and input sensor current from real-world measurements.
- A myopic strategy optimizing the trade-off between detection of abnormalities and sensing energy consumption.
- A Markov Decision Processes (MDP) solution to track the time-evolving, physical activity of an individual along with the state of the battery.
- Extensive evaluation and results based on a real-world trial involving 20 pregnant women for seven months (ethically approved human subject trial).

4.1 Edge-assisted Adaptation Control

Based on the edge-assisted architecture described in the previous section, the IoT system we propose uses information related to the context of the monitored subject – the activity in the case study considered here – and the model at the cloud layer to adjust sensing energy. The optimization formulation aims at the *minimization of cost measured as energy expense of the sensor node and maximization of the probability an abnormality will be detected.* As the first step to formulate the abnormality detection, we model the accuracy of sensor output as a function of the different activities and energy levels.

In Section 4.1.1, the accuracy model as a function of Gaussian distribution for vital signs is derived. Section 4.1.2 discusses the stochastic model considering both the activity and the battery model. Section 4.1.2 presents a *myopic* method, which only considers the current patient's activity to determine the sensing configuration. We build a model for the dynamics of the State of Charge (SoC) of the sensor's battery in Section 4.1.2. In addition, we present a MDP formulation and algorithm which optimizes the current level over a finite horizon based on the activity model stored at the cloud layer.

4.1.1 Problem Formulation

Based on the biometric PPG signal generated from the pulse oximeter sensor, the Gateway calculates vital signs of the monitored person including heart rate, respiration rate and SpO_2. We remark that the current level U and the activity X will determine the SNR of the signal used to calculate the features. In order to evaluate the accuracy of the PPG sensor, we use an ECG sensor as a reference for heart rate, an airflow sensor for respiration, and another PPG sensor with higher signal quality – and therefore energy consumption – as a reference for SpO_2.

Based on the reference signals, we compute the error vectors of the three features: heart rate error $e_1(U, X)$, respiration rate error $e_2(U, X)$ and oxygen saturation error $e_3(U, X)$. We calculate the weighted total error vector $e(U, X) = \{\gamma_1 e_1(U, X) + \gamma_2 e_2(U, X) + \gamma_3 e_3(U, X)\}$. The variables $\gamma_1, \gamma_2, \gamma_3$ are positive weights such that γ_1+γ_2+γ_3=1.

We denote as $\rho(e(U, X) \mid U = u, X = x)$ the Probability Density Function (PDF) of the total error as a functions of extracted features during an activity state,

$$X \in \{\text{Sleeping, Sitting, Walking, Jogging, Running}\}$$

, and current level $U \in \{U_1, U_2, ..., U_5\}$.

For practicality reasons, we assume that the PDF is Gaussian, that is $\rho(e(U, X) \mid U = u, X = x) \sim \mathcal{N}(0, \sigma(U, X))$.The variance of error $\sigma(U, X)$ can be derived from the error vector e given current level U and activity state X. The error probability is calculated as the tail probability of the normal distribution. Threshold T corresponds to the maximum total $RMSE$ tolerance in the estimated vital signs. Therefore, the error probability is the following,

$$\mathcal{P}_{\text{error}}(T) = \int_{-\infty}^{T} \rho_E(e(U, X) \mid U = u, X = x) de \tag{4.1}$$

The formulation derived for error probability, can be used to calculate the abnormality misdetection probability. In fact, the abnormality misdetection probability is a function of sensor's error probability (e.g. $P_{\text{error}}(T)$ in Eq. 4.1) and probability of abnormal vital sign regions. The abnormality misdetection probability formulates the probability that an abnormal event in vital signs is not captured considering the error margin in the sensor's measurements. Error margin in the sensor's measurements can be controlled and adjusted by the threshold T in $P_{\text{error}}(T)$. for instance, smaller value of T results in accepting smaller values of error probability in the system. Thus, the abnormality misdetection probability will take smaller values forcing the system to decide more conservative on the optimal power level in the sensor which may increase the sensor's energy consumption.

To formulate the abnormality misdetection probability, we define regions of abnormal vital signs. We assume that the PDF of the vital signs – heart rate, y_1, respiration rate, y_2 and oxygen saturation y_3 – is Gaussian both in normal and abnormal conditions. We calculate the mean of normal vital signs $\mu_n = \sum_{i=1}^{3} \mu_{n,i}\gamma_i$ and variance $\sigma_n^2 = \sum_{i=1}^{3} \sigma_{n,i}^2\gamma_{n,i}^2$. Note that these parameters are calculated based on the combined features $y = \gamma_1 y_1 + \gamma_2 y_2 + \gamma_3 y_3$, we assume that the combined features follow a Gaussian distribution. Abnormal vital signs also follow a Gaussian distribution with mean $\mu_a = \sum_{i=1}^{3} \mu_{a,i}\gamma_i$ and variance $\sigma_a^2 = \sum_{i=1}^{3} \sigma_{a,i}^2\gamma_{a,i}^2$.

In order to detect the normal vital signs from abnormal ones we set the threshold $\tau = \sum_{i=1}^{3} \tau_i\gamma_i$ over the thresholds $\{\tau_1, \tau_2, \tau_3\}$ of each individual feature. We, then, define $\mathcal{P}_{\text{DE}}$ as when the abnormality misdetection probability is detecting the abnormal vital signs as normal. Similar formulation considering normal regions of vital signs can be used to model the misdetection of normal vitals as abnormal. In summary, the probability density functions of abnormal vital signs (e.g. $f_a(y|U, X)$) and error (e.g. $\rho(e|U, X)$) follow independent Gaussian distributions.

Let $\mathcal{P}_{\text{DE}} = \mathcal{P}(\alpha, \beta, \eta)$ be the abnormality misdetection probability defined as the probability of the following events:

- sensor's error tolerance $\alpha = \{e(U, X) < \mathrm{T}\}$,
- region of abnormal vital signs $\beta = \{y > \tau\}$, and
- activity state and current level $\eta = \{U = u, X = x\}$.

Using the chain rule, the joint misdetection probability can be written as,

$$\mathcal{P}_{\text{DE}} = \mathcal{P}(\alpha \mid \beta, \eta)\mathcal{P}(\beta \mid \eta)\mathcal{P}(\eta). \tag{4.2}$$

Consider the following upper bound for $\mathcal{P}(\eta)$, with no prior knowledge about U and X,

$$\mathcal{P}_{\text{DE}} \leq \mathcal{P}(\alpha \mid \beta, \eta)\mathcal{P}(\beta|\eta). \tag{4.3}$$

Intuitively, although the distribution of abnormal vital signs is a function of the activity, it is independent with respect to the sensor's current level. Thus,

$$\mathcal{P}_{\text{DE}} \leq \mathcal{P}(\alpha \mid \eta)\mathcal{P}(\beta \mid X = x) = \mathcal{P}_{\text{UB}}. \tag{4.4}$$

Assuming $\mathcal{P}(\alpha \mid \eta) \sim \mathcal{N}(0, \sigma)$ and $\mathcal{P}(\beta \mid X = x) \sim \mathcal{N}(\mu_a, \sigma_a)$, we obtain

$$\mathcal{P}_{\text{UB}} = \int_{-\infty}^{T} \rho_E(\alpha \mid \eta) de \int_{\tau}^{\infty} f_a(\beta \mid X = x) dy. \tag{4.5}$$

By substituting the error probability in Eq. 4.1, we finally obtain

$$\mathcal{P}_{\text{UB}} = \mathcal{P}_{\text{error}}(T) \int_{\tau}^{\infty} f_a(\beta \mid X = x) dy. \tag{4.6}$$

We use this form of $\mathcal{P}_{\text{UB}}$ along with the energy consumption in PPG sensor as the main optimization parameters in section 4.1.2 and 4.1.2.

4.1.2 Stochastic Model

The PPG sensor is equipped with a battery of finite capacity. A key part of our model tracks the dynamic of the battery's SoC to enable energy-aware optimization of the used current level. Herein, we consider that Δt is a time slot equal to 15 minutes.

The battery capacity is uniformly quantized to create discrete battery states. Specifically, the SoC in a given time Q_k takes values in $Q = \{Q_1, ..., Q_K\}$. In the following, we consider the maximum number of battery states K to be 30. The sensing energy consumption is a function of the used sensing current level, which are determined by the control U taking values in the action set $\{U_0, U_1, U_2, ..., U_5\}$.

We set the states so that the sensor consumes 1 charge level of the battery in 1 hour if the lowest sensing current level U_1 is used. Thus, the maximum number of hours the sensor can continuously function is equal to 30. The highest current level U_5 consumes 5 levels of battery charge in an hour. Therefore, the battery will be drained in 6 hours using the highest current level. Action U_0 indicates that the sensor is in sleeping mode, during which the SoC remains the same.

The dynamics of the battery state can be written as,

$$Q'_k = max\{Q_k - E(U), 0\}, \tag{4.7}$$

where $E(U) \in E = E_1, ..., E_5$ indicates the number of battery levels consumed at a given time. Denoting the power consumption of the sensor as $\mathcal{C}_{\text{TX}}(U)$, Eq. 4.7 can be rewritten as,

$$Q'_k = max\{Q_k - \mathcal{C}_{\text{TX}}(U)\Delta t, 0\} \tag{4.8}$$

We consider that sensing starts with the maximum full charged battery at time 0, that is, Q_{30}. Choosing an optimal solution for the current level, the sensor consumes the corresponding energy budget. The discharging probability in the battery depends on the current level chosen as an optimal action. The sensor consumes energy changing the state of battery from Q_k to $Q_{k'}$ where $k' < k$. Therefore the transition probability during discharging can be defined as,

$$q_{k/k'} \triangleq \mathcal{P}\left(Q'_k|Q_k, U\right) \tag{4.9}$$

The sensor is forced to choose a lower current level U during the critical situation when the battery level at a given time is less than the optimal action . For instance, consider that the sensor's optimal current level was chosen to be U_4 but the battery's critical battery level Q_2 doesn't allow sensing for the next hour. Therefore, the optimal action will be forced to be U_2 to afford the continuation of sensing.

We consider that sensing stops when battery is drained to the lowest level Q_1 forcing the sensor to choose sleeping action. The user charges the battery only if they are not "Walking", "Jogging" or "Running", and the sensor's optimal action would be sleeping mode U_0. During charging state, the sensor goes to a sleep mode by choosing the current level to be U_0. During charging, the state of battery changes from $Q_{k'}$ to Q_k where $k' > k$. Therefore, the transition probability during charging will be,

$$q_{k/k'} = \mathcal{P}\left(Q_{k'}|Q_k, U_0\right) \tag{4.10}$$

Considering the probability of charging and discharging defined above, the dynamics of the battery during the time can be modeled as a Markovian process with states defined with temporal evolution in Figure 4.1.

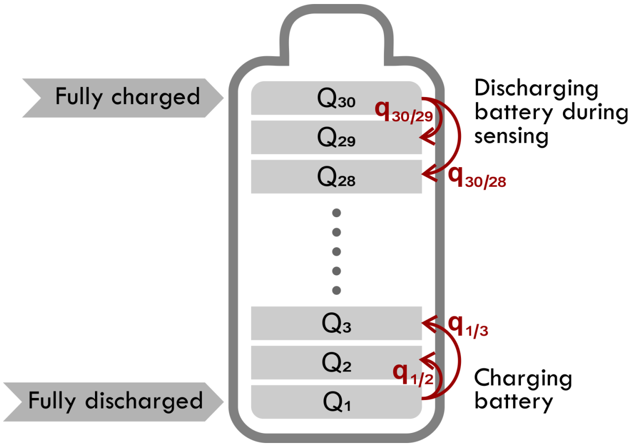


Figure 4.1: Markov chain of battery states during charging and discharging modes

The stochastic model considers the activity of the monitored person. In particular, we define a Markov process X whose state space is $\{$Sleeping, Sitting, Walking, Jogging, Running$\}$. Figure 4.2 shows the connectivity structure of the Markov chain governing the dynamics of X.

Importantly, the statistics of activity dynamics change over time following daily cycles. We capture this characteristic by making the transition probabilities of X. Specifically, we build four different models for daily activities each corresponding to a 6 hour period. Based on the time of the day, the corresponding Markov chain computed from the history of the activity of the subject will be used to calculate the optimal solution. Assume that each model corresponding to each period of time is denoted by $i \in \{1, 2, 3, 4\}$ during a 24 hour activity. For instance, transition probability from activity "Sitting" to "Walking" for first period is denoted as $\mathcal{P}(X_3(1)|X_2(1))$. Transition probabilities during period i from activity $X_j(i)$ to $X_{j'}(i)$ is defined as,

$$p_{j/j'}(i) = \mathcal{P}\left(X_{j'}(i)|X_j(i)\right) \tag{4.11}$$

Markov model corresponding the ith period of 6 hour per day(e.g. $i \in \{1, 2, 3, 4\}$), will be used to solve the optimization problem. Note that the dynamics of activity change over time, and the model needs to be periodically updated. This becomes particularly apparent in the considered case-study, where daily activities are measured in pregnant women. Not only optimization needs to be context-aware, but also needs to be specific to the person and the change over time of the person's habits. The idea of this model demonstrates the necessity of self-aware system to adapt its dynamics throughout the time. In view of the foregoing, our proposed model in adapting the context-aware system to the activity of the subject results in personalization of the system.

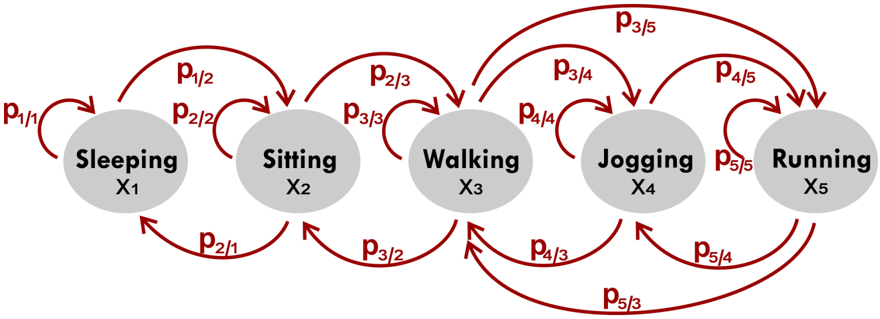


Figure 4.2: Markov chain of activities of an individual during one period of 6 hour activity

The overall system state Z is defined to be the composition of the battery state Q_k and the patient's activity state $X_j(i)$,

$$Z(i) = (X_j(i), Q_k), \tag{4.12}$$

with $Z \in \mathcal{Z} = \mathcal{X} \times \mathcal{Q}$. Note that the dynamics of the battery are deterministic and fully defined by the action U and the current SoC Q_k. During the time period i , it is assumed that the transition probabilities between activities are static but then the transition probability will be updated entering the new period. Since the transition probability between activity

X and battery level Q are independent, the joint transition probabilities can be written as,

$$\begin{gathered}\mathcal{P}(Z'(i)|Z(i),U) = \mathcal{P}\left(Q_{k'}|Q_k,U\right) \times \mathcal{P}\left(X_j(i)|X_{j'}(i)\right). \\ \text{or, equivalently,} \\ \mathcal{P}(Z'(i)|Z(i),U) = p_{j/j'}(i) \times q_{k/k'}\end{gathered} \tag{4.13}$$

where, $\mathcal{P}\left(Q_{k'}|Q_k,U\right)$ captures the the dynamics of the battery as a function of Q_k and the chosen action U and $\mathcal{P}\left(X_j(i)|X_{j'}(i)\right)$ captures the dynamics of activity. We remark that the probability $\mathcal{P}\left(X_j(i)|X_{j'}(i)\right)$ depends on the time period i. At the end of each time horizon i a new model of Markovian chain with different transition probabilities will be calculated. Therefore, the total transition probability of joint battery and activity states updates in each time slot. In addition, transition probability $\mathcal{P}(Z'(i)|Z(i),U)$ depends on chosen action U. For instance, if the user is “Sitting” and the battery state is fully discharged (e.g. $Z_2 = (X_2, Q_1)$ in Figure 4.2), the system transitions to the user with “Walking” and battery state Q_2(denoted as $Z_8 = (X_3, Q_2)$) given the possible action U_0, with probability $q_{1/2} \times p_{2/3}$. An example of resulting Markov chain is illustrated in Figure 4.3. The transition probabilities $\mathcal{P}(X_j(i)|X_{j'}(i))$ are specific to an individual, meaning that the Markov chain is built with the statistics collected over time at the cloud layer. For simplicity, here we assume that the probabilities $\mathcal{P}(X_j(i)|X_{j'}(i))$ are known. Figure 4.4 demonstrates the dynamics of

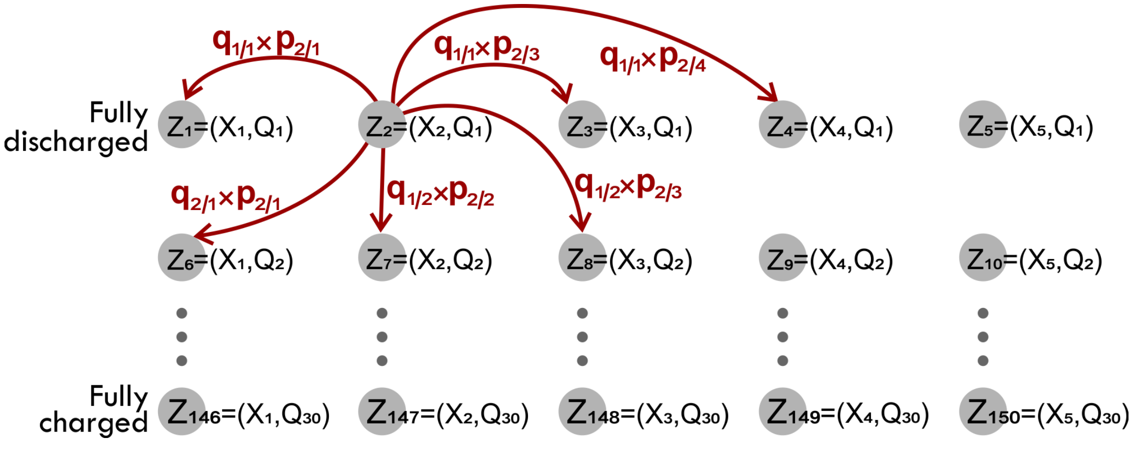


Figure 4.3: Markov chain of joint activity and battery state during one period of 6 hours

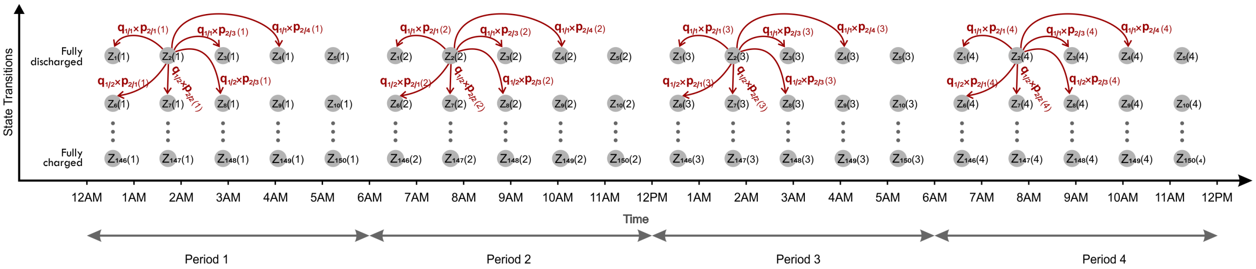

State Transitions
Fully discharged
Fully charged
q1/1×p2/1(1)
q1/1×p2/3(1)
q1/1×p2/4(1)
q1/2×p2/1(1)
q1/2×p2/2(1)
q1/2×p2/3(1)
Z1(1) Z2(1) Z3(1) Z4(1) Z5(1)
Z6(1) Z7(1) Z8(1) Z9(1) Z10(1)
Z146(1) Z147(1) Z148(1) Z149(1) Z150(1)
q1/1×p2/1(2)
q1/1×p2/3(2)
q1/1×p2/4(2)
q1/2×p2/1(2)
q1/2×p2/2(2)
q1/2×p2/3(2)
Z1(2) Z2(2) Z3(2) Z4(2) Z5(2)
Z6(2) Z7(2) Z8(2) Z9(2) Z10(2)
Z146(2) Z147(2) Z148(2) Z149(2) Z150(2)
q1/1×p2/1(3)
q1/1×p2/3(3)
q1/1×p2/4(3)
q1/2×p2/1(3)
q1/2×p2/2(3)
q1/2×p2/3(3)
Z1(3) Z2(3) Z3(3) Z4(3) Z5(3)
Z6(3) Z7(3) Z8(3) Z9(3) Z10(3)
Z146(3) Z147(3) Z148(3) Z149(3) Z150(3)
q1/1×p2/1(4)
q1/1×p2/3(4)
q1/1×p2/4(4)
q1/2×p2/1(4)
q1/2×p2/2(4)
q1/2×p2/3(4)
Z1(4) Z2(4) Z3(4) Z4(4) Z5(4)
Z6(4) Z7(4) Z8(4) Z9(4) Z10(4)
Z146(4) Z147(4) Z148(4) Z149(4) Z150(4)
12AM 1AM 2AM 3AM 4AM 5AM 6AM 7AM 8AM 9AM 10AM 11AM 12PM 1AM 2AM 3AM 4AM 5AM 6AM 7AM 8AM 9AM 10AM 11AM 12PM
Time
Period 1
Period 2
Period 3
Period 4

Figure 4.4: Markov chain of joint activity and battery states from period 1 to 4 with 6 hours of duration

joint activity and battery states during 24 hour. In this case, the transition probability between activity states $p_{j/j'}(i)$ is shown as a function of period i. The activity model for the corresponding period i can be used to calculate the total transition probability between joint states. Therefore, dependency of $p_{j/j'}(i)$ on period i results in dependency of total transition probabilities between states on period i as well. This model allows the system to evolve in time based on dynamics of the user's context.

Myopic **Strategy**

Based on the upper bound for the misdetection probability provided in Eq. 4.6, we define the optimization problem controlling the trade off between sensing power consumption $\mathcal{C}_{\mathrm{TX}}$ and probability of misdetection $\mathcal{P}_{\mathrm{UB}}$. The *myopic* optimization problem trades off between sensing power consumption $\mathcal{C}_{\mathrm{TX}}$ and the probability $\mathcal{P}_{\mathrm{UB}}$ in each time slot as follows

$$\begin{aligned}
&\underset{U}{\text{minimize}} && \mathcal{C}_{\mathrm{TX}}(U) \\
&\text{subject to} && \mathcal{P}_{\mathrm{UB}} \leq \theta \\
& && \text{or, equivalently,} \\
& && \mathcal{P}(\alpha \mid \eta) \leq \frac{\theta}{\mathcal{P}(\beta \mid X = x)} = \zeta.
\end{aligned} \tag{4.14}$$

We define the Lagrangian multiplier λ to solve the equivalent optimization problem,

$$\mathcal{L}(U, \lambda) = \mathcal{C}_{\mathrm{TX}}(U) + \lambda(\mathcal{P}(\alpha \mid \eta) - \zeta). \tag{4.15}$$

Taking the derivative with respect to the sensor's current level,

$$\frac{\partial \mathcal{C}_{\mathrm{TX}}(U)}{\partial U} + \lambda \frac{\partial \mathcal{P}(\alpha \mid \eta)}{\partial U} = 0. \tag{4.16}$$

We obtain a linear relation between power consumption and current level:

$$a_U + \lambda \frac{\partial \mathcal{P}(\alpha \mid \eta)}{\partial \alpha} \frac{\partial \alpha}{\partial U} = 0. \tag{4.17}$$

Given the Gaussian PDF of α and linearity between α and U, we have

$$a_U + \lambda b_U \sum_{i=1}^{N} \mathbb{1}[U = U_i] \frac{\partial \Phi_i(\alpha)}{\partial \alpha} = 0 \tag{4.18}$$

The optimal (U^*, λ^*) can be then calculated as

$$\sum_{i=1}^{N} \mathbb{1}[U = U^*] \frac{\partial \Phi_i(\alpha)}{\partial \alpha} = -\frac{a_U}{\lambda^* b_U} \tag{4.19}$$

Using this formulation, we developed the Algorithm 1 to determine the optimal current level in each slot. If the number of possible current levels is equal to N, the complexity this search is $O(N)$.

Optimization Over a Finite Horizon

The *myopic* method solves the optimization problem with linear time complexity. In this section, we propose the second method to optimize in longer perspective. In our prototype

Algorithm 1 *myopic* method to control sensor's current

```
procedure SOLUTION($\tau, T$)        ▷ $\tau$ and $T$ are thresholds at a given time t
    Extract Activity level $X$
    for $U \in \{U_1, ..., U_5\}$ do
        Calculate the variance of the error $e(U, X)$.
        Calculate tail Gaussian $\mathcal{P}(e \mid U = u, X = x)$
        Estimate $\mathcal{P}_{\text{DE}}$ using $\mathcal{P}_{\text{UB}}$.
        if  $\mathcal{P}_{\text{UB}} < \tau$ then return $u$
        else
            continue
    until the system is terminated
```

system, the edge layer keeps track of patient's activity state along with the battery state of PPG sensor. The cost function in this method is defined as a function of the power consumption and the upper bound for misdetection probability (e.g. Eq. 4.6). The two fold goal is to *minimize the cost while tracking the system and patients' states.* To this end, we must propose a method that determines the sensor's current level to minimize the accumulative cost over a finite time horizon.

Optimal Strategy: The Markov process defined earlier in Section 4.1.2 allows the computation of strategies optimizing the current level over a temporal horizon, which is moved in a sliding window fashion. Optimality is defined over a cost jointly capturing sensing power consumption and misdetection probability in the upper bound form provided in the previous section. In fact, misdetection probability $\mathcal{P}_{\text{UB}}$ as a function of activity state X and current level U directly directly influence the total cost. Therefore, the total cost $\mathcal{C}(Z, U)$ can be written as a function of joint state Z and current level, U,

$$\mathcal{C}(Z, U) = \omega \mathcal{P}_{\text{UB}}(U, X) + (1 - \omega)\mathcal{C}_{\text{TX}}(U) \tag{4.20}$$

Given an initial state $X(0) \in X$ and $Q(0) \in Q$, and the finite horizon $\mathcal{T}$, we calculate an average expected reward under the action U,

$$\mathcal{J}(Z,U) = \frac{1}{\mathcal{T}}\left[\sum_{t=0}^{\mathcal{T}-1}\gamma^t C(Z,U)|Z(0)\right] \tag{4.21}$$

where we find the expected function under action $\mathcal{U} : \mathcal{Z}' \rightarrow \mathcal{Z}$. Note that discount factor $0 \geq \gamma < 1$ is typically chosen close to one.

Optimal Action: The goal is to minimize $\mathcal{J}$ over a finite time horizon with a given initial state $Z(0)$ that achieves the minimum cost

$$U^* = \underset{U}{\text{argmin}}\mathcal{J}(Z,U) \tag{4.22}$$

Using this optimization formulation, we developed Algorithm 2 to determine the optimal current level in a given time. In other words, the edge processor determines the current level such that the total cost as a function of misdetection probability and the sensor's power consumption will be minimized.

Algorithm 2 MDP method to control sensor's current

```
**procedure** SOLUTION($\tau, T$)        ▷ $\tau$ and $T$ are thresholds at a given time t
    Extract Activity level $X$
    Estimate battery level $Q$
    Calculate the variance of the error $e(U, X)$.
    Calculate tail Gaussian $\mathcal{P}(e \mid U = u, X = x)$
    Estimate $\mathcal{P}_{\text{DE}}$ using $\mathcal{P}_{\text{UB}}$.
    Find the transition matrix $\mathcal{P}_{\text{z'z}}(U)$ based on time $t$ on the portion of the day
    **for** state $Z \in \{Z_1, ..., Z_{150}\}$ **do**
        **for** state $U \in \{U_0, ..., U_5\}$ **do**
            Use Equation 4.20 to compute cost $\mathcal{C}(Z, U)$ for transition matrix $\mathcal{P}_{\text{z'z}}(u)$
            Use Equation 4.22 to find optimal action $U^*$
    **until** the system is terminated
```

4.2 Setup and Data

In this section, we first describe in detail the system setup in Section 4.2.1 and, then, in Section 4.2.2, describe the clinical trial at the base of this study and its output data.

4.2.1 Setup

We now provide some specifics of the IoT system we developed. The sensing device, besides the actual PPG sensor, is equipped with a microcontroller to read data from the sensor, a Flash memory to store data temporarily, and a wireless data transmission module to send the recorded data. We use the ESP8266-12E board which integrates all the mentioned components in a single board. In the specific board we used, a full TCP/IP stack is integrated with an L106 32-bit RISC microprocessor core running at 80 MHz, with 96KB on-chip SRAM and 4MB external Flash memory. For the sensor, we use a MAX-REFDES117 PPG sensor board which is highly configurable and provides a digitized signal through I2C communication. We program the microprocessor using C programming language to configure the sensor board internal registers. The sensing device connects to the Edge device via a WiFi connection. It has a permanent configuration memory which records the user information and credentials of three different WiFi networks. At the beginning of the operation, the sensor node connects to a WiFi network and asks the Edge device to send initial configuration instructions, which include the PPG LED power, recording duration, and hibernate state duration. Then, the sensing device begins to record PPG signal on the Flash memory and then goes to the hibernation mode for the duration defined in the configuration instructions. After the hibernation period, it connects again to the Edge device via the WiFi connection, sends the recorded data, waits for the Edge device to process the transferred data, receives the results as a new set of configuration instructions, and initiates a new recording cycle. Note that the sensor turns the radio communication circuitry off during the record and hibernation to save the

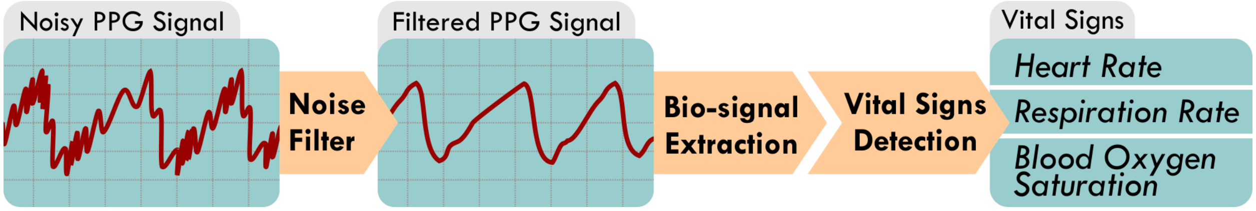


Figure 4.5: Local data analysis to extract vital signs from raw PPG signals.

energy. Table 4.1 summarizes the system's power consumption specifications.

Table 4.1: System states with sensing power consumption

State	Power consumption
Recording mode: LEDs setting: 0.8mA	69.30mW
Recording mode: LEDs setting: 3.5mA	73.26mW
Recording mode: LEDs setting: 6.2mA	79.86mW
Recording mode: LEDs setting: 9.3mA	84.15mW
Recording mode: LEDs setting: 12mA	89.43mW

The Edge device is a Linux-based computer running an Apache web server. It receives each set of the recorded data from the sensing device via several HTTP Upload POST requests, merges data chunks into one file, analyzes the data, and returns the new configuration in reply to the latest HTTP request. As shown in Figure 4.5, local data analysis on the Edge device consists of two sequential phases: *1) bio-signal extraction* and *2) vital signs detection*.

1) Bio-signal extraction: Respiratory and heartbeat signals are extracted from the raw PPG signal. Different studies proposed various techniques which can be categorized into two major classes: feature-based extraction and filter-based extraction [14, 50]. The former class first derives features from the waveform to extract respiratory and heartbeat signals [31]. These methods might be, nevertheless, inappropriate in real-world applications since the features quality is highly susceptible to motion and ambient noise. Such artifacts are inevitable in wearable devices as patients engage in different activities in non-clinical settings. On the contrary, filter-based extraction techniques remove the artifacts before deriving the bio-signals, filtering the raw PPG signal [22].

We adopt a filter-based technique by designing two band-pass filters for heartbeat and respiratory signals extraction. In general, the bio-signals have different frequency ranges: $0.1 - 1$ Hz ($6 - 60$ breath/minute) and $0.5 - 3.3$ Hz ($30 - 180$ heartbeat/minute). The boundaries could be selected as the filters' cutoff frequency. However, such a naive approach leads to high error rates as the pass range is too broad. Then, we select the cutoff frequency using the peak values in the power spectral density (PSD) of the PPG signals [39]. Figure 4.6 shows the PSD as a function of frequency for a one-minute PPG signal while the user is sleeping.

The peak in the heart rate (HR) frequency range reflects the heartbeat signal frequency. Similarly, the respiratory signal frequency is selected using the peak in the respiration rate (RR) frequency range. As the RR frequency range might contain the HR frequency peak, we first extract the heartbeat signal and remove the HR peaks; then we acquire the respiratory signal. The filter's cutoff frequencies are dynamically selected during the monitoring with respect to the incoming signals. Note that an acceptable SNR is necessary as an excessive distortion of the signal would impair the process.

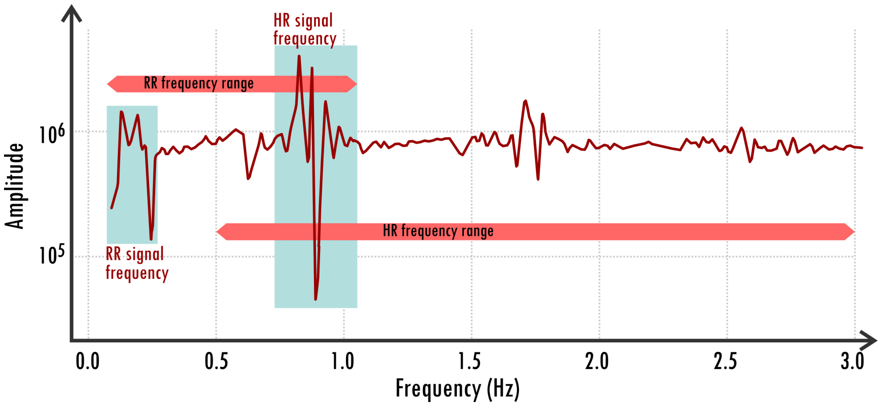


Figure 4.6: Power Spectral Density (PSD) of one-minute PPG signal while the user is sleeping

2) Vital signs detection:

Several time-domain and frequency-domain techniques can be exploited to derive respiration rate and heart rate from the bio-signals [14]. In our setup, we use a peak detection algorithm where the peaks are obtained by detecting local maximum points in the derivative of the bio-signals. The period between two consecutive peaks reflects the respiration rate and heart rate values.

In contrast to the respiration rate and heart rate, SpO$_2$ (blood oxygen saturation) is calculated using a feature-based technique. As indicated in Figure 4.7, four features – AC_{RED}, DC_{RED}, DC_{IR}, and AC_{IR} – are acquired from the infra-red (IR) and red signals. Such features are obtained by mapping consecutive local extrema of the heartbeat signals into the raw PPG signals. The SpO$_2$ is calculated using the following equations:

$$R = \frac{AC_{RED}.DC_{IR}}{AC_{IR}.DC_{RED}}, \qquad SpO_2 = \alpha R^2 + \beta R + \gamma \tag{4.23}$$

where α, β and γ are constants determined by the sensor's specification [41].

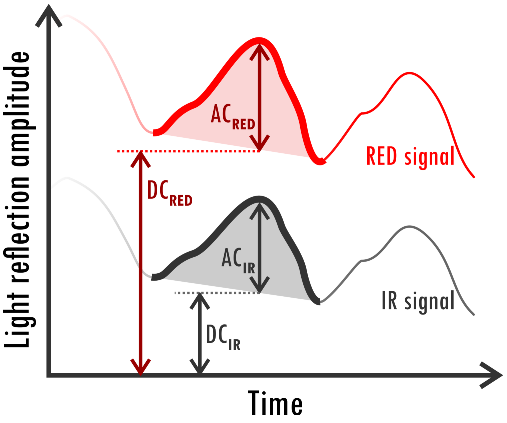


Figure 4.7: Features in the PPG signals for SpO$_2$ Calculation

4.2.2 Data Collection

In order to evaluate the accuracy model of PPG sensor's measurements, we first collect data for PPG signal during different activities. We conduct our 5 hour experiment with 25 different combinations, in which a healthy subject is monitored during "Sleeping", "Sitting", “Walking”, “Jogging” and “Running” and the sensor’s current level is varied with different values (e.g., 0.8 mA, 3.5 mA, 6.3 mA, 9.2 mA and 12 mA). Each experiment lasted for 12 minutes. Windows of 30 seconds were used to calculate the vital signs from the PPG signal. The PPG signal from the sensor is preprocessed, filtered and the features SpO_2, heart rate, and respiration rate are evaluated for each combination of current level and activity.

Simultaneously, we use three different reference sensors, to compare the calculated vital signs from our proposed PPG sensor with a ground-truth. We use a chest strap ECG sensor to capture the reference heart rate, an airflow sensor for respiration reference, and another PPG sensor with higher signal quality as a reference for SpO_2. We then calculated the error between calculated vital signs and the reference signal to model the accuracy of measurements.

In order to model and test our proposed algorithms, physical data is recorded. The physical activity data is part of a longitudinal study conducted to investigate the maternal body changes on pregnant women for seven months (i.e., six months of pregnancy and one month postpartum). The study was performed in collaboration with the Department of Nursing Science, University of Turku (UTU) and Turku University Hospital (TYKS), and in accordance with the code of ethics of the World Medical Association (Declaration of Helsinki). In addition, it was approved by the joint ethics committee of the hospital district of Southwest Finland (35/1801/2016) and Turku University Hospital (TYKS).

Between May 2016 and June 2017, twenty pregnant women were selected and recruited in Southern Finland, considering different criteria such as “singleton pregnancy”, “18 as the lower age limit” and “Gestational age less than 15 weeks.” Data collection was performed 24/7 via a lightweight fitness tracker (i.e., Garmin Vivosmart® HR) [23] consisting of a PPG sensor and an Inertial Measurement Unit (IMU). The data were regularly transferred to remote computers for storage and post-processing. The physical activity data utilized in this chapter was abstracted from the user’s steps and hand movements in every 15-minute interval [65, 64]. Data from 6 subjects were rejected due to discontinuity in data recording, with more than 50% of missing data. The output of the study allowed us to build the transition probabilities capturing the dynamics of the activity engaged by the monitored person. Note that the model evolves over time, and is specific to a person. Therefore, the cloud layer continuously updates the model and periodically sends it to the edge server.

4.3 Numerical Results

In this section, we provide extensive results assessing the performance of the proposed system and optimization framework. To assess the accuracy, the features extracted through the PPG sensor are compared to a ground truth. We first calculate the total variance of $RMSE$ to determine errors in the sensor with different activities and different PPG sensor's current levels. Figure 4.8 shows that the variance decreases when the current level is increased. The variance increases in strenuous activities, such as "Jogging" or "Running" due to noise caused by motion artifacts. Using the total variance of $RMSE$, we model the probability of misdetection in abnormality to follow a Gaussian distribution. PDF of two activities is shown in Figure 4.9. Higher values of variance in lower current levels resulted in higher values of probability of error (e.g., $\mathcal{P}_{\text{error}}$ in Eq. 4.1). Figure 4.9 shows the PDF of "Sitting" and "Running" with the calculated error variance. Higher values of variance in lower current

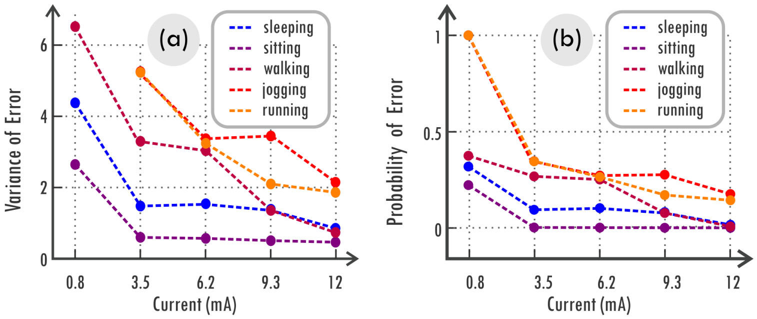


Figure 4.8: (a) Total Error variance for activity X and current level U. (b) Error probability for activity X and current level U with maximum $RMSE = 2$

levels lead to higher values of the error probability $\mathcal{P}_{\text{error}}$. The shaded regions in Figures 4.9 (a) and 4.9 (b) correspond to $\mathcal{P}_{\text{error}}$ with a threshold of $RMSE$, $T = 2$. Decrease in probability of error with higher current level during vigorous decreases is more evident. In addition, $\mathcal{P}_{\text{error}}$ is one during "Running" or "Jogging" with the minimum current level of $0.8mA$. Thus the heart rate, respiration rate or SpO_2 cannot be extracted from the acquired signal due to the vigorous movements of the subject. $3.5mA$ is the lowest acceptable current level for activities such as "Running" or "Jogging".

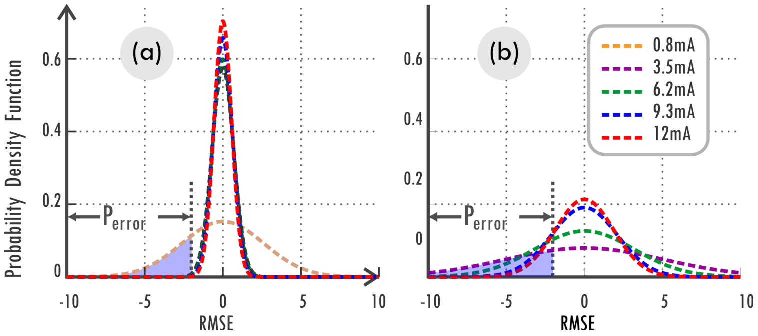


Figure 4.9: Error probability when $RMSE = 2$ for different current levels during two states: (a) sitting, (b) running

***Myopic* Strategy:** First, We evaluate the *myopic* strategy presented in Section 4.1.2. We first monitor a healthy individual for 24-hour. 3D acceleration signal is used to continuously estimate the user's physical activity. The accelerometer sensor is placed at the user's hand and hand movements are tracked to extract user's steps. The acceleration signal is filtered, mitigating ambient noise, and steps are counted in each time interval. Moreover, when the user is still (i.e., no step is detected), the orientation of the user is leveraged to differentiate sitting and sleeping.

First, we set $T = 2$ to be the predefined threshold of the $RMSE$. We then calculate the error probability for all possible power levels. The *myopic* algorithm chooses the lowest power level that satisfies the maximum misdetection probability $\mathcal{P}_{\text{UB}}$ calculated in Equation 4.14. Vigorous activities necessitate accurate monitoring which leads to choosing higher power levels. In contrast, lower power levels can satisfy the same constraint on the probability of error in activities such as, sleeping or sitting. In the results, we set the maximum probability of error to $\zeta = 0.17$ and set the weights as $\gamma_1 = 0.25, \gamma_2 = 0.35, \gamma_3 = 0.4$. Using these weights, we determine the mean and variance of abnormal vital signs for each activity. In the considered case and chosen parameters, the aggregate of normal vital signs y follows the distribution $\mathcal{N}(\mu_n, \sigma_n)$ with $\mu_n = \{57.72, 60.43, 74.29, 83.97, 91.53\}$ and $\sigma_n = \{0.25, 0.73, 1.17, 1.45, 0.62\}$ in the order of "Sleeping", "Sitting", "Walking", "Jogging" and "Running", respectively. Abnormal vital signs follow the distribution $\mathcal{N}(\mu_a, \sigma_a)$, where we set $\mu_a = \{56.83, 61.07, 74.15, 81.50, 97.00\}$ and $\sigma_a = \{2.02, 0.71, 2.29, 1.67, 2.88\}$, respectively, and each element of the vectors corresponds to a different activity as indicated earlier. The threshold τ defined in Equation 4.5 is the intersection of two Gaussian distributions corresponding to normal and abnormal vital sings. The Cumulative Distribution Function (CDF) for abnormal vital signs is calculated with $\tau = \{57.20, 60.73, 72.76, 82.68, 89.58\}$.

Based on these parameters, the probability $\mathcal{P}(\beta \mid X = x) = \{0.57, 0.68, 0.72, 0.76, 0.99\}$ is then derived according to Equation 4.5. Assuming the predefined threshold $\zeta = 0.17$ in error probability of sensor and $\mathcal{P}(\beta \mid X = x)$, the new threshold is set for the probability of misdetection to fulfill the upper bound $\theta = \zeta \times \mathcal{P}(\beta \mid X = x)$ (see Equation 4.14). The threshold $\theta = \{0.09, 0.11, 0.12, 0.13, 0.17\}$ is the final vector corresponding to the desired upper bound of misdetection probability $\mathcal{P}_{\text{UB}}$ in the order of "Sleeping", "Sitting", "Walking", "Jogging" and "Running", respectively.

We first evaluate *myopic* method to monitor a subject for 24 hour. We monitor their activity and compare constant power consumption with *myopic* method. The user's physical activity is estimated using the 3D acceleration signal. Placing the accelerometer sensor at the user's hand, we use hand movements to extract user's steps. The acceleration data is filtered, and steps are counted in each time interval. In addition, when no steps are detected, the orientation of the user is leveraged to differentiate between sitting and sleeping activities. Figure 4.10 (a) shows the activity level labeled "Sleeping", "Sitting", "Walking", "Jogging"

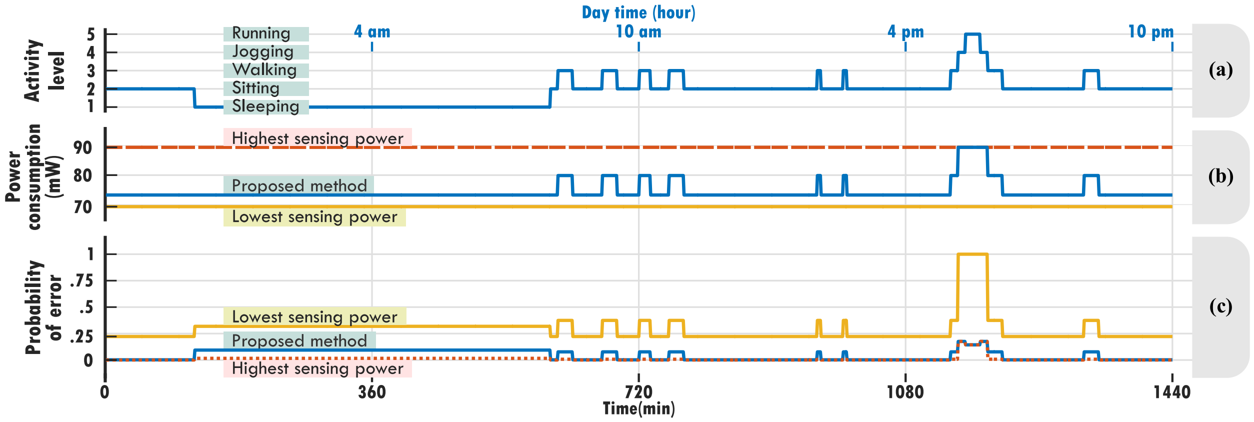


Figure 4.10: 24-hour health monitoring of a healthy person. (a) user's activity level. (b) sensor's sensing power consumption. (c) Error probability in abnormality detection expected regarding the user's activity. The red line, yellow line and blue line indicate the baseline of highest, lowest and our proposed platform sensing power consumption, respectively.

and "Running" as $\{1, 2, 3, 4, 5\}$ respectively.

Considering $T = 2$ as the predefined threshold of $RMSE$, we calculated the error probability for each current level. At each given time, based on the activity level, we are able to choose the lowest current level that satisfies the misdetection probability $\mathcal{P}_{\text{UB}}$ shown in Equation 4.14. High activity levels necessitate accurate monitoring that require choosing high current levels.

In contrast, low activity levels, can fulfill the same threshold of probability of error with lower current levels. In the experiment shown in Figure 4.10 (a), we set the maximum probability of error to $\zeta = 0.17$ and set the weights as $\gamma_1 = 0.25, \gamma_2 = 0.35, \gamma_3 = 0.4$. In the results, we define the mean and variance of abnormal vital signs for each activity. In particular, the aggregate of normal vital signs y follows the distribution $\mathcal{N}(\mu_n, \sigma_n)$ with $\mu_n = \{57.72, 60.43, 74.29, 83.97, 91.53\}$ and $\sigma_n = \{0.25, 0.73, 1.17, 1.45, 0.62\}$ in the order of "Sleeping", "Sitting", "Walking", "Jogging" and "Running", respectively.

In case of abnormal vital signs, y follows $\mathcal{N}(\mu_a, \sigma_a)$ with $\mu_a = \{56.83, 61.07, 74.15, 81.50, 97.00\}$ and $\sigma_a = \{2.02, 0.71, 2.29, 1.67, 2.88\}$, respectively. The threshold τ in Equation 4.5 is the intersection of two Gaussian distributions for each activity. The probability of abnormal vital signs is calculated using the Cumulative Distribution Function (CDF) with $\tau = \{57.20, 60.73, 72.76, 82.68, 89.58\}$. Based on this parameters, the probability $\mathcal{P}(\beta \mid X = x) = \{0.57, 0.68, 0.72, 0.76, 0.99\}$ is then calculated according to Equation 4.5. Based the predefined threshold $\zeta = 0.17$ in error probability of sensor and $\mathcal{P}(\beta \mid X = x)$, we calculate the new threshold for the probability of misdetection to fulfill the upper bound $\theta = \zeta \times \mathcal{P}(\beta \mid X = x)$ (see Equation 4.14). The threshold $\theta = \{0.09, 0.11, 0.12, 0.13, 0.17\}$ corresponds to the desired upper bound of misdetection probability $\mathcal{P}_{\text{UB}}$ for each activity.

Figure 4.10 (b) illustrates the comparison between our proposed methodology, and static current levels corresponding to the lowest and the highest sensing power. The lowest sensing power consumption in the sensor leads to unacceptably high error probability, while the highest current level leads to an excessively high energy consumption. During the 24-hour experiment, we measured 5983.4J and 7721.4J consumed by the lowest and highest sensing power, respectively. Our energy efficient algorithm reduced the energy consumption to 6417.5J. Average sensing power consumption of 74.32mW was observed which amounts to a 16.9% reduction compared to 89.43mW of the highest sensing power level. Figure 4.10 (c) shows the error probability associated with each current level. Note that using the lowest current level leads to an unacceptable $P_{error} = 1$ during “Jogging” and “Running”.

Figure 4.11 illustrates the trade-off between the constraint on the maximum error probability and overall energy consumption in a 24-hour temporal period. The *myopic* algorithm can detect abnormalities with maximum tolerance threshold of $RMSE = 50$ by choosing the lowest current level. This setting leads to an energy expense equal to 5983.4J. Setting the maximum error tolerance to $RMSE = 0.1$ or less, results in choosing the highest current level with the lowest probability of error inevitably. In this case, the total sensing energy consumption during a 24-hour period is equal to 7721.4J. Values between these two extreme cases influence the maximum probability of error and the sensing energy consumption in

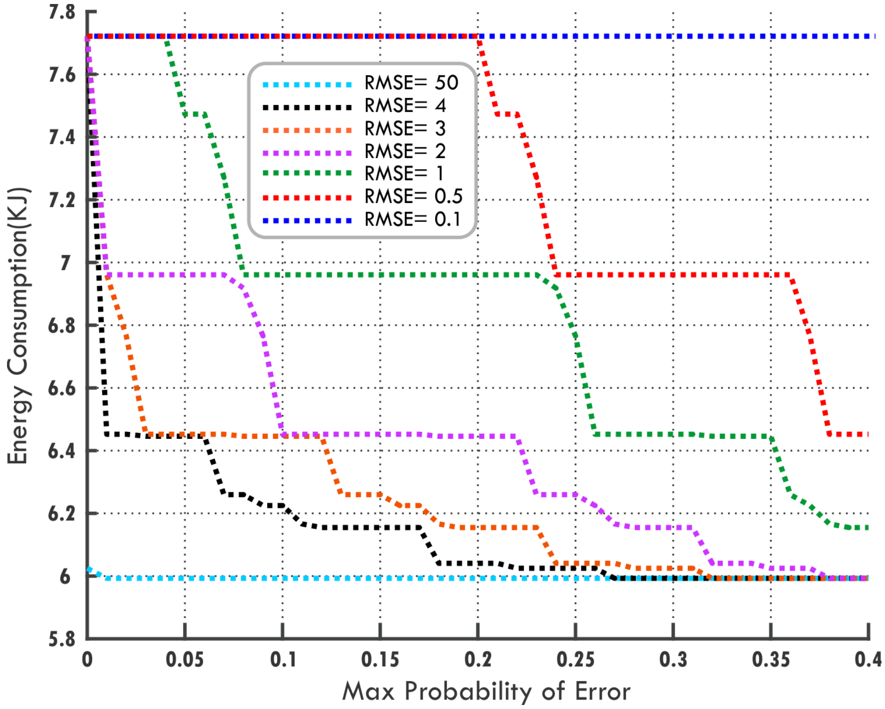
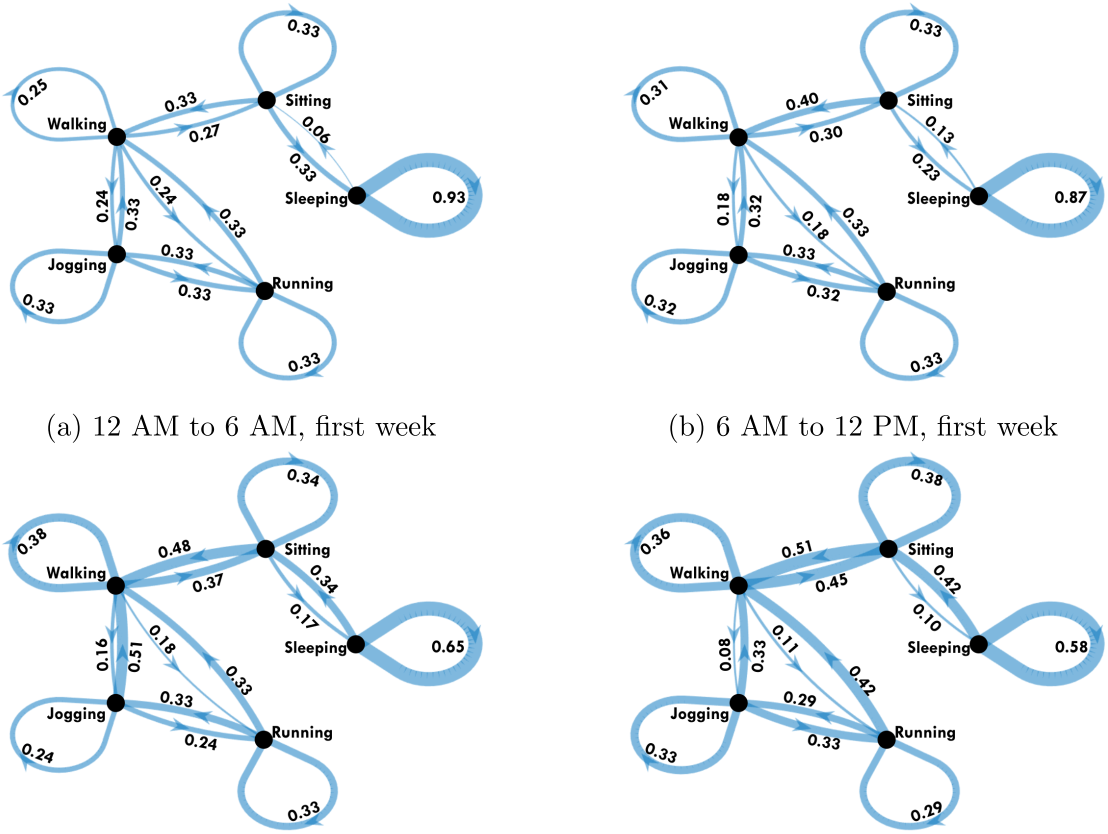


Figure 4.11: Sensing energy consumption as a function of maximum probability of error with different RMSE levels.

the sensor layer. For a specific value of the $RMSE$ threshold, increasing the maximum probability of error will let the system monitor the subject less accurately using a lower current level. Therefore, the sensing energy consumption in the sensor will decrease.

MDP Strategy: The myopic strategy only considers the instantaneous state of the system. Intuitively, a strategy incorporating a longer-term evaluation of the outcome of actions can control the current level to avoid outage, which inevitably reduces performance, saving energy until the subject is more likely to charge. We emphasize that this planning is performed on a non-homogeneous model, where transition probabilities are a function of the time of the day. This further emphasizes the need for a wise planning aligning energy availability with periods of the day most likely presenting vigorous activities. We further remark that such planning is informed by highly personalized models, capturing the dynamics of the specific person wearing the sensor.

Figure 4.12 (a-d) shows the transition probabilities over the 24% for one of the monitored subjects obtained training the model over one week. This specific subject's activities mostly included sleeping between 12 AM to 6 AM, with a probability of 0.937 to remain in the sleeping state and 0.063 of transitioning to the sitting state. The probability of staying in sleeping state reduces to 0.58 and transitioning to sitting state increases to 0.42 in the temporal period between 6 PM to 12 AM because of decrease in sleeping state. In addition, Markov chain for each period can be updated after each week. Figure 4.13 shows the transition probabilities extracted between 12 PM to 6 PM trained over a one month period. The individual has a decrease in probability of staying in sleeping state from 0.65 to 0.50 in the span of four weeks. Meanwhile, the probability of remaining in the sitting increased from 0.34 to 0.44.



(a) 12 AM to 6 AM, first week

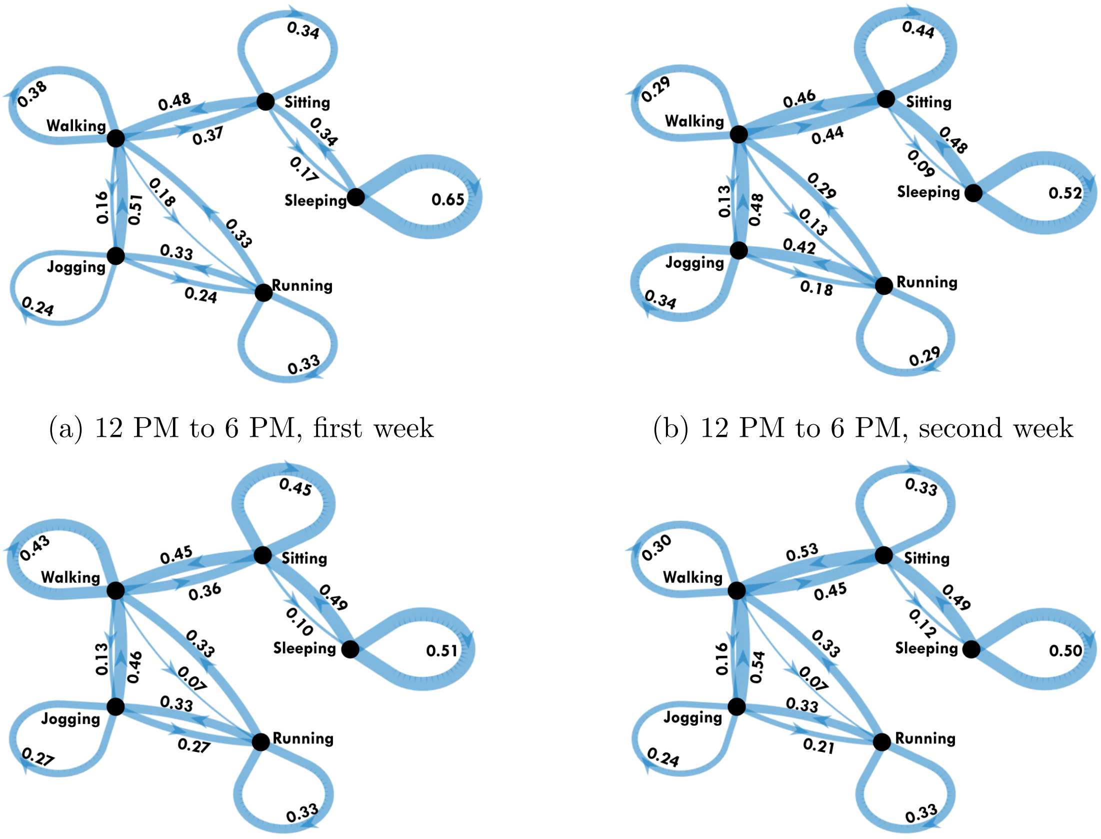
(b) 6 AM to 12 PM, first week

(c) 12 PM to 6 PM, first week

(d) 6 PM to 12 AM, first week

Figure 4.12: Markov chain of activities of a subject in a 24 hour period with average transition probabilities calculated from first week in one month

Figure 4.14 illustrates how the activity of this specific individual evolves over one month.



(a) 12 PM to 6 PM, first week

(b) 12 PM to 6 PM, second week

(c) 12 PM to 6 PM, third week

(d) 12 PM to 6 PM, fourth week

Figure 4.13: Markov chain of activities of a subject between 12 PM to 6 PM with average transition probabilities calculated from weekly activities in the span of four weeks

For instance, the subject had a decline in walking time from the first week to the third week, but then an increase in her fourth week. In contrast, the sleeping state had an increase of 8% from week one to week three and dropping by 4% on week four. Clearly, these changes should be reflected in the control strategy and planning.

Figure 4.15 compares the *myopic* and MDP strategies for the aforementioned subject over a 48 hours monitoring period. We set the threshold for the probability of error (e.g. ζ in Eq. 4.14) to 0.002. The MDP cost function, which captures a trade-off between the upper-bound misdetection probability and energy consumption, is a function of the weight $0 \leq \omega \leq 1$. If the value of ω is close to zero, the control strategy tends to privilege lower current levels and higher probability of misdetection. In contrast, a value of ω close to one pushes the actions

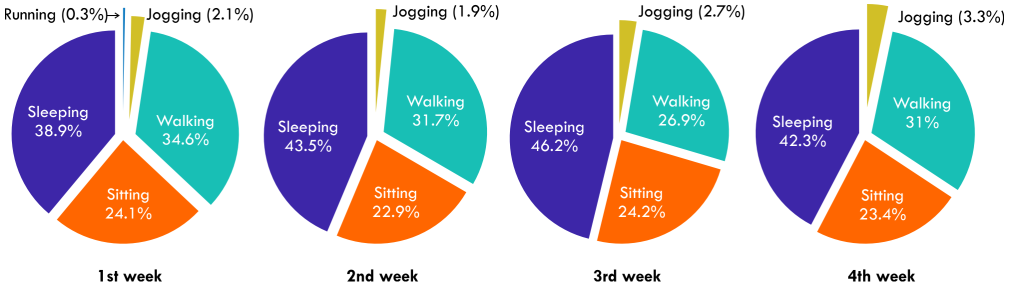


Figure 4.14: Weekly activity percentage for one month

to compromise energy saving to achieve a lower error probability. In this experiment, we set $\omega = 0.176$ to have the closest probability of misdetection to that of the *myopic* strategy in the previous results.

Figure 4.15 (b) shows the optimal current level selected by the *myopic* and MDP methods. Figure 4.15 (b) and (c) show the actions and dynamics of the battery state induced by the two methods actions, respectively. The system tracks the battery state for both methods with 30 being the highest charging state. Based on the optimal action taken, the battery state will be updated at each given time during sensing. During each time the user charges the battery the sensing stops and the state of battery will be updated.

Figure 4.15 (d) shows the probability of error corresponding to current level chosen in optimal current level chosen based on Figure 4.15 (b). At each given time, the system chooses an optimal action based on the user's activity. *Myopic* strategy chooses the optimal action by considering the current activity of an individual. However, MDP strategy plans based on the current state of the battery and the activity of a user. Therefore, the battery is drained after a longer time in MDP method compared to *myopic*. *Myopic* method chooses the optimal action without taking into account that the action chosen will drain the battery faster and this will increase the possibility that the battery will be drained during a strenuous activity such as, walking, jogging and running. the battery will not be charged till the user is sleeping or sitting. *Myopic* method chooses a higher current level that will have lower probability of misdetection compared to MDP in a short term. However, the frequency of necessity to

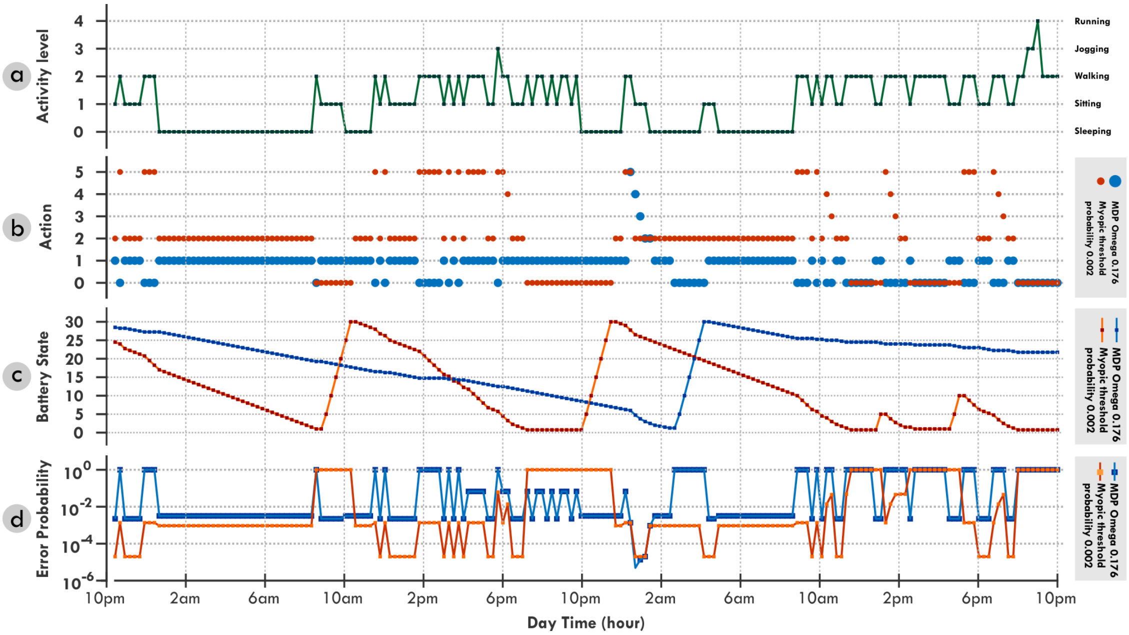


Figure 4.15: 48-hour health monitoring of a healthy person during first week of monitoring. (a) user's activity level. (b) sensor's sensing power consumption. (c) battery state tracking based on sensor's power consumption. (d) Probability of error expected regarding the user's activity. The red line, and blue line indicate the *myopic* and MDP methods, respectively.

charge the battery will be decreased in MDP resulting in using the resource of the battery more wisely. For instance, between 6 PM to 10 PM of the first 24 hour monitoring, the user was walking but due to drainage of battery, sensor was in sleeping mode. However, MDP strategy chose the current level U_1 and monitored the user for longer time.

To update the activity model, the system retrains the MDP model weekly. MDP model based on activity history from the first week is used to find the optimal solution in Figure 4.16. Respectively, Figures 4.17 and 4.18 show the results of 48 hours of monitoring during third and week of results starting at 10 PM and ending 48 hours later.

Figure 4.19 shows the probability of error over the entire day as a function of the average energy consumption averaged over a month. We compare the three methods – *myopic*, MDP and static power allocation.

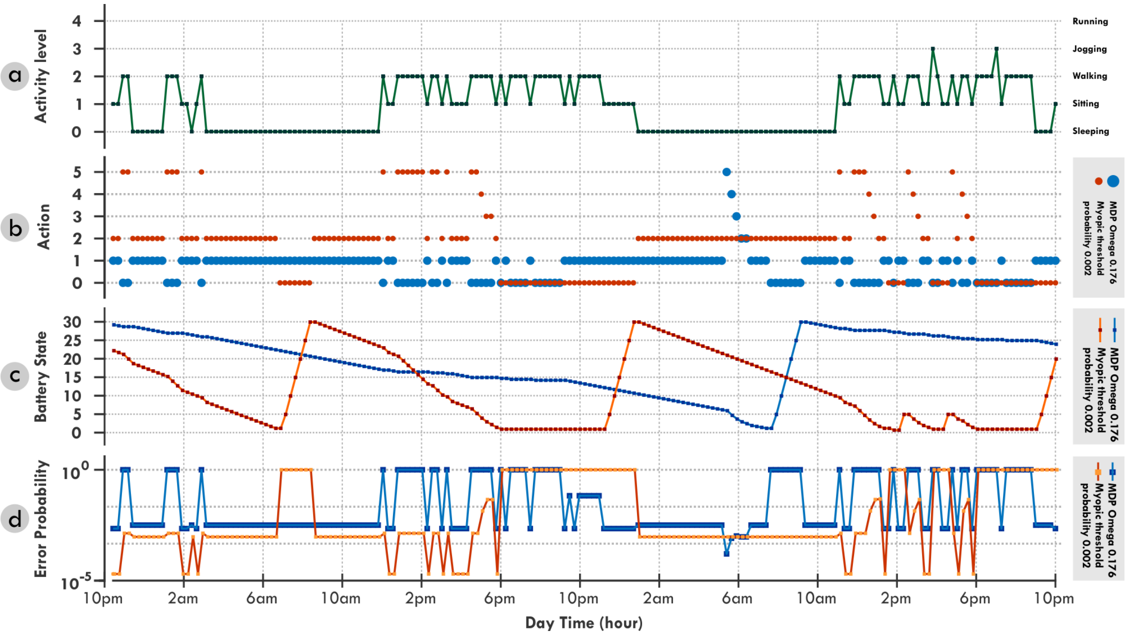


Figure 4.16: 48-hour health monitoring of a healthy person during second week of monitoring. (a) user's activity level. (b) sensor's sensing power consumption. (c) battery state tracking based on sensor's power consumption. (d) Probability of error expected regarding the user's activity. The red line, and blue line indicate the *myopic* and MDP methods, respectively.

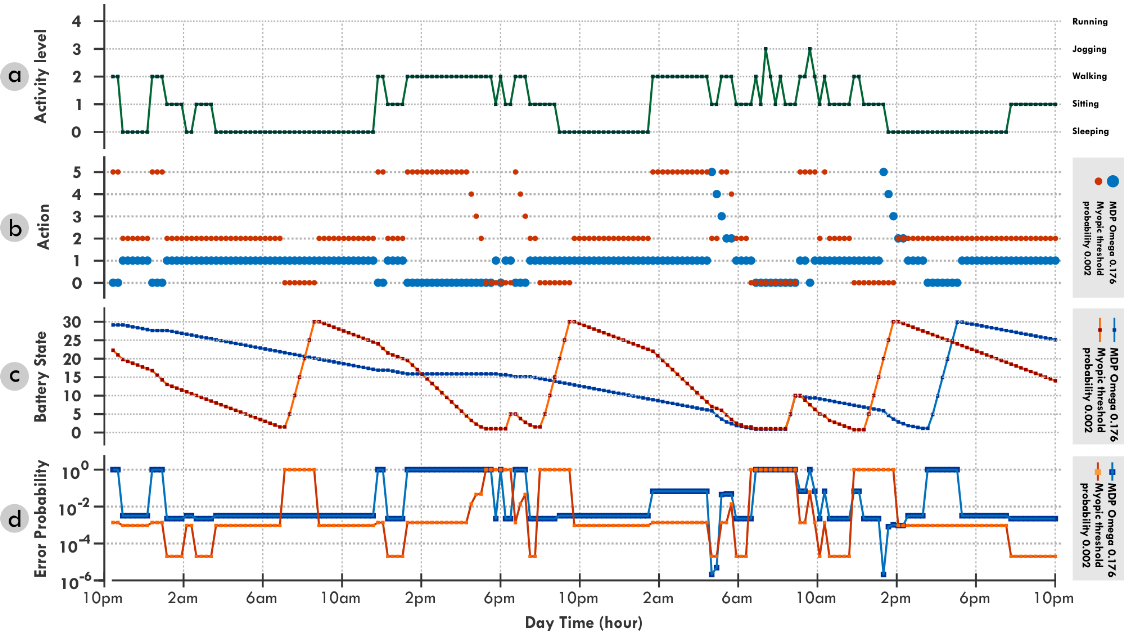


Figure 4.17: 48-hour health monitoring of a healthy person during third week of monitoring. (a) user's activity level. (b) sensor's sensing power consumption. (c) battery state tracking based on sensor's power consumption. (d) Probability of error expected regarding the user's activity. The red line, and blue line indicate the *myopic* and MDP methods, respectively.

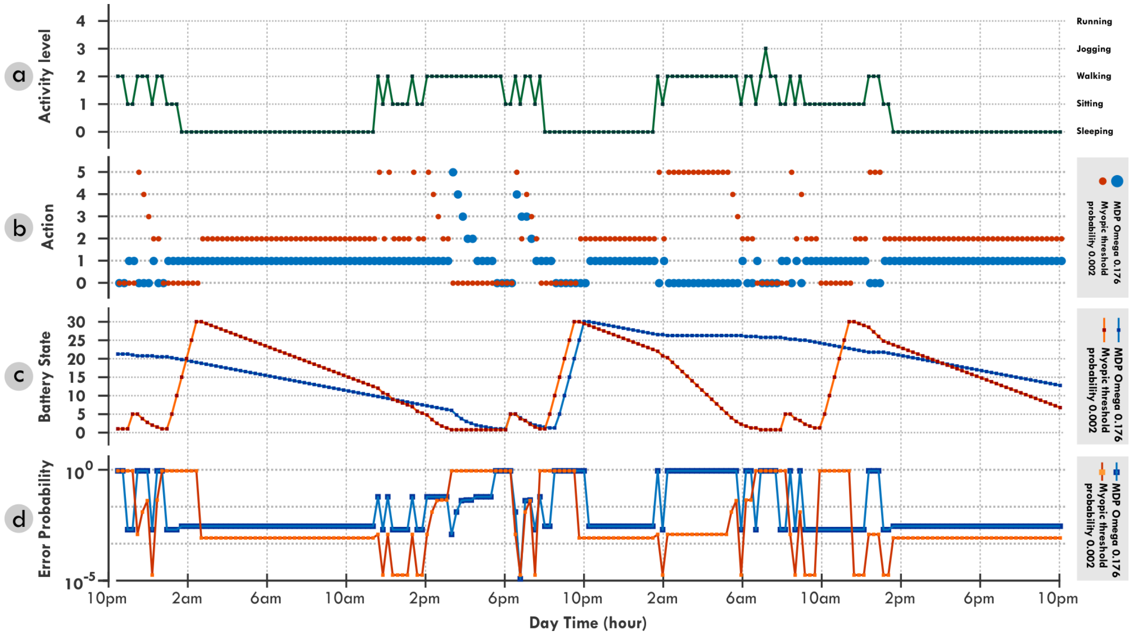


Figure 4.18: 48-hour health monitoring of a healthy person during fourth week of monitoring. (a) user's activity level. (b) sensor's sensing power consumption. (c) battery state tracking based on sensor's power consumption. (d) Error probability in abnormality detection expected regarding the user's activity. The red line, and blue line indicate the *myopic* and MDP methods, respectively.

Note that the sensor has 5 current levels, that is,

$$U \in \{0.8 \text{ mA}, 3.5 \text{ mA}, 6.2 \text{ mA}, 9.2 \text{ mA}, 12 \text{ mA}\}$$

.

We take into account the time slots in which the sensor is off due to battery outage, impacting the average energy consumption and the probability of misdetection in abnormality. Note that the highest current level (e.g., 12 mA) consumes the highest amount of energy, but quickly drains the battery, thus inducing outage approximately every 6 hours. Note that due to the frequent outages, and thus the frequent 1.5 hours charging periods, the average energy consumption over the 24 hours is smaller than in other current levels. However, during the charging mode, the probability of abnormality misdetection is equal to one, resulting to a large misdetection probability. Using a constant current level of 0.8 mA increases the

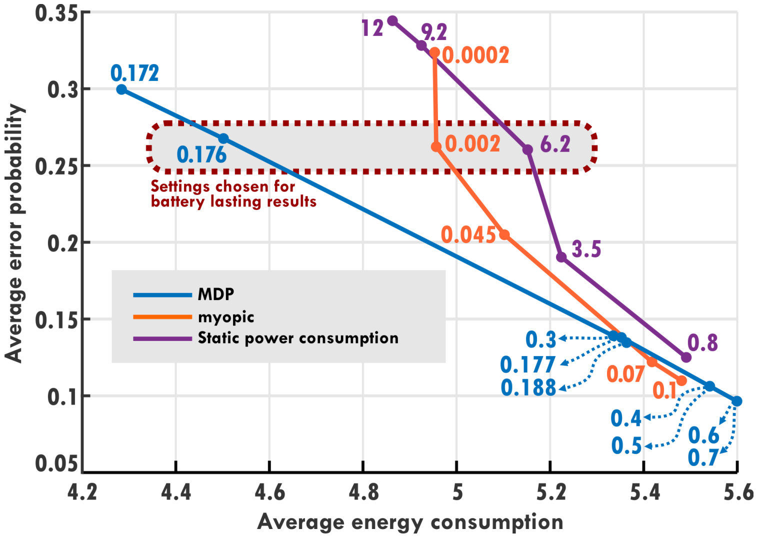


Figure 4.19: Average of probability of error as a function of energy consumption (KJ) in one month for one subject. Comparison between three methods of MDP, *myopic* and static power consumption.Myopic method is evaluated based on $\zeta \in \{0.0002, 0.002, 0.045, 0.07, 0.1\}$ in Eq. 4.14. MDP is evaluated based on, $\omega \in \{0.172, 0.176, 0.177, 0.188, 0.3, 0.4, 0.5, 0.6, 0.7\}$ in Eq. 4.20. Static power consumption is evaluated based on current levels $U \in \{0.8, 3.5, 6.2, 9.2, 12\}$ mA

average 24 hour energy consumption, but leads to a low probability of detecting anomalies in difficult states. The dynamic context-aware selection operated by the *myopic* and MDP frameworks is therefore instrumental to achieve good performance.

To derive the MDP-based strategy we set the parameter of cost function (e.g. ω in Eq. 4.20 to $\{0.172, 0.176, 0.177, 0.188, 0.3, 0.4, 0.5, 0.6, 0.7\}$ to determine the importance of energy consumption over error probability. Comparing MDP with *myopic* and constant current level, we see a great reduction in energy consumption when the probability of error is around 0.25. In order to compare the battery lasting for the three methods, we chose the parameters to ssatisfy the fixed probability of error 0.26. In this case, the parameter for MDP cost function ($\omega = 0.176$), parameter for upper bound probability of error in *myopic* method ($\zeta = 0.1$) and static current level is chosen as 6.2 mA.

Using the settings mentioned above, we calculated the average time between consecutive battery outages and compared the methods in Figure 4.20. We also included the battery lasting

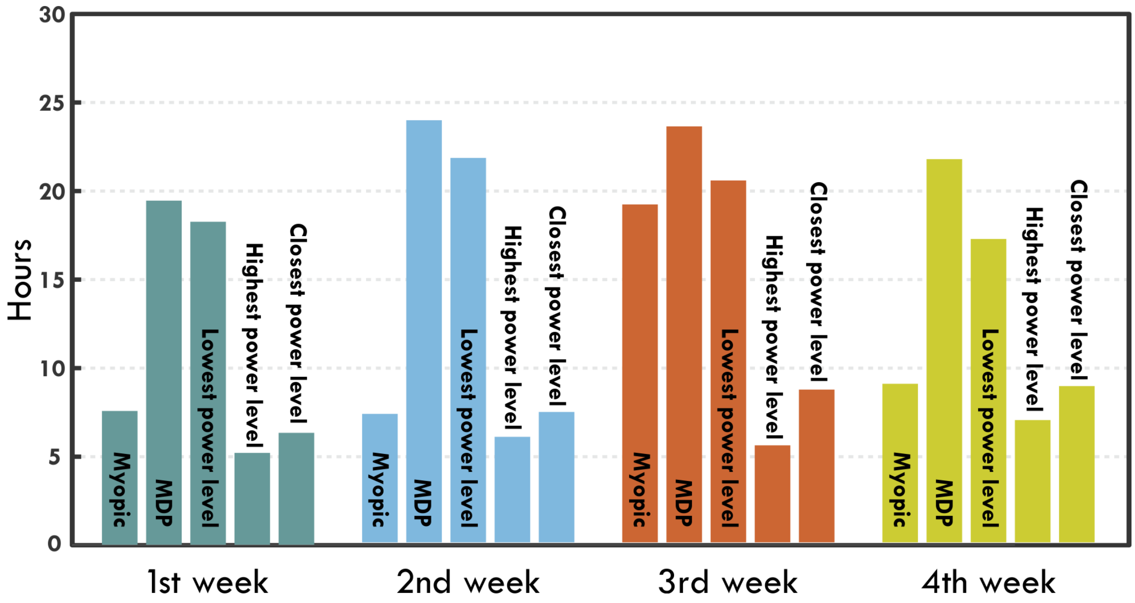


Figure 4.20: Average Battery lasting during one month

using highest and lowest current level in the sensor (e.g., 12 mA and 0.8 mA accordingly).

Results indicate that MDP achieves a high efficiency than using the lowest current level. This is because MDP policy chooses the sleeping mode in the sensor when the battery has very low charge leading to longer periods between battery outages. Results show that the MDP doubles the battery duration compared to the *myopic* strategy.

We observe a considerable increase of battery lasting in the third week granted by the *myopic* approach compared to the other weeks for this specific user. The increase is motivated by the evolution of the statistics of the activity pattern, illustrated in Figure4.14. The user has an increase in the fraction of time spent sleeping and sitting, which grow to 46% and 24%. Therefore, based on our model, this user had a higher probability to charge her wearable. Consequently, the battery was fully charged more often compared to other weeks. Furthermore, during these activities the sensor can use a lower power level, resulting in a longer battery duration even when using a simple approach.

Figure 4.21 shows the comparison between MDP, *myopic* and static power allocation strategies averaged over 14 subjects. We set a fixed value of average probability of error to 0.32.

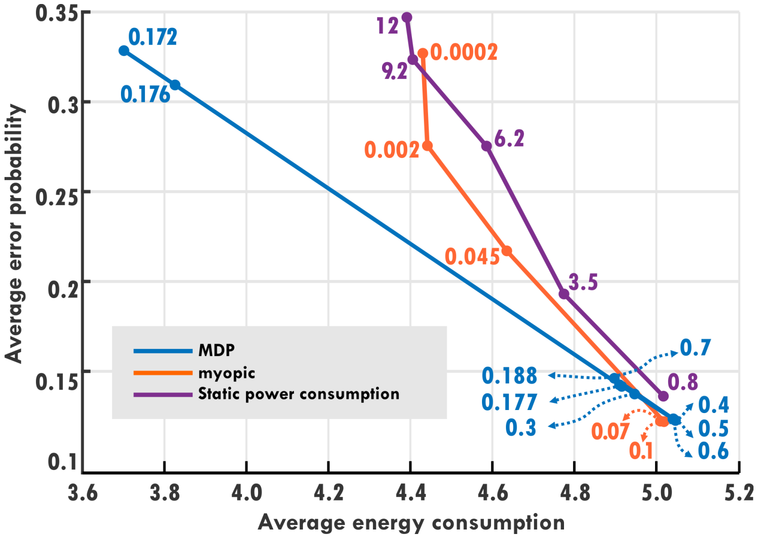


Figure 4.21: Average probability of error as a function of energy consumption (KJ) in one month averaged over 14 subjects. Comparison between three methods of MDP, *myopic* and static power consumption. Myopic method is evaluated based on $\zeta \in \{0.0002, 0.002, 0.045, 0.07, 0.1\}$ in Eq. 4.14. MDP is evaluated based on $\omega \in \{0.172, 0.176, 0.177, 0.188, 0.3, 0.4, 0.5, 0.6, 0.7\}$ in Eq. 4.20. Static power consumption is evaluated based on current levels $U \in \{0.8, 3.5, 6.2, 9.2, 12\}$ mA

The MDP strategy consumes 3.7 KJ over one month, while *myopic* method consumes 4.42 KJ and the static power allocation strategy consumes 4.4 when using current level 9.2 mA. Therefore, MDP has an average of 12% reduction in energy consumption fulfilling the same probability of error compared to *myopic* and static power consumption.

Chapter 5

Bandwidth Efficiency in Edge Computing

One of the many challenges in developing clinical-level monitoring systems, is that such systems often require the execution of complex algorithms analyzing the morphology of biosignals, rather than tracking the simple features that can be extracted by resource-constrained sensors [52, 35]. However, while wearable devices do not possess the needed computing capacity to analyze these signals, transmitting the raw signal to compute-capable devices – as in edge computing frameworks – requires a stable and high capacity channel, which may not be available in real-world settings.

In this chapter, we focus our attention on electrocardiogram (ECG) signals used to detect early signs of heart arrhythmia to prevent heart failures. Specifically, the 4 different classes of heart arrhythmia we aim to detect in this work are Ventricular Ectopic Beat, Supraventricular Ectopic Beat, Fusion Beat, and Unknown Beat. Importantly, this difficult detection task necessitates extensive computing capacity due to the high complexity of the classification algorithms.

Recent work in this area and application focuses on offline classification [34, 2, 10], where the acquired signal is assumed to be available at the device executing the algorithm. Due to the limited computational capacity of wearable sensors, most prior work has assumed local storage of the signal for later processing on fixed machines. In contrast, in this chapter we focus on real-time monitoring [47], where classification is necessarily performed at a remote device, which is interconnected to the sensor by means of a communication channel - often composed of wireless link and, possibly, a series of wireline links. Herein, we specifically consider an edge computing system, where a compute-capable device located at the network edge server, assists the sensor in performing the classification task. In such a setting, the volatility of the wireless channel capacity may impair the ability of the sensor to timely deliver ECG segments to the edge server for analysis.

To facilitate this transfer, we propose a form of adaptive compression. Unlike prior contributions on ECG compression [30, 19], the methodology we propose is *semantic*, meaning that the representation is tuned to the classification goal, rather than aiming at the accurate reconstruction of the signal. Specifically, we propose to use Sequentially Hypothesis Testing [69] to enable the collaborative selection of an optimal set of wavelet coefficients associated with ECG cycles. Guided by the output of the classifier – which is repeatedly executed as new coefficients are received – the Sequential Hypothesis Testing framework guides the selection of the coefficients to be requested by the edge server. To the best of our knowledge, this chapter is the first contribution proposing the use of this family of algorithms in sensor-edge server systems for healthcare.

Notably, one of the main benefits of the technique we propose is to make the representation of ECG segments specific to the characteristics of the signal in the interval. For instance, the set of coefficients that are sent by the end of the iterative process might be different for abnormal and abnormal cycles, as well as, for different subjects.

In summary, this chapter makes the following contributions:

- We propose a an iterative and “semantic” form of compression guided by a Sequential Hypothesis Testing core and based on a wavelet representation of an ECG signal.

- We develop a Convolutional Neural Network model to detect 4 types of arrhythmia from the ECG signal harvested by a sensor. The CNN classifier achieves 4% of False Positive Rate in the detection of abnormal heart cycles when using the original ECG signal.

- We present and discuss extensive results based on real-world ECG trials.Two sets of data, the MIT-BIH arrhythmia database and MIT-BIH Supraventricular arrhythmia database(MIT-SVDB), are used to evaluate the system we propose. These data sets involve 63 subjects to train the Sequential Testing model and 14 subjects to test the proposed system. Numerical results show that the sequential hypothesis testing selects different sets of coefficients for normal and abnormal ECG cycles. In addition, we show that the signal representation is personalized, as the system selects different sets of coefficients for different patients. Results over 14 subjects indicate that average channel usage is reduced by up to 43% to non-adaptive optimization to achieve the same accuracy of classification.

The rest of the article is organized as follows. Section 5.1 describes the problem formulation and high-level system architecture. Proposed solution using Sequential Hypothesis Testing is presented in Section 5.2. Details of ECG arrhythmia classification proposed in solution is described in Section 5.3. Section 5.4 presents and discusses numerical results.

5.1 System Overview

We consider an edge computing-based system, where a wearable sensor collecting an ECG signal is used to detect abnormal heart cycles. The sensor records and transmits the ECG signal to the edge server, which executes the classification algorithm. We remark that while edge computing overcomes the energy and computing capacity limitations of wearable sensors, its use necessitates the transmission of the acquired signal over a wireless channel. Due to capacity constraints and fluctuations that affect wireless channels, the transfer of the raw signal might be unfeasible. Thus, we first need to identify a suitable representation of the signal. We propose to use a wavelet based representation, where wavelet coefficients are transmitted in lieu of the signal time series for each heart cycle. As transmitting the coefficients over the channel might exceeds its capacity, we use Sequential Hypothesis Testing to build a collaborative and sequential compression strategy between the sensor and the edge server to control the subset of coefficients sent by the sensor. Then, the edge server reconstructs the signal based on the received coefficients and executes the classifier. We use a Convolutional Neural Network (CNN) to classify abnormal ECG cycles from normal cycles using the received coefficients. Figure 5.1 depicts the schematics of the proposed system.

Intuitively, the set of coefficients influences the accuracy of the reconstruction at the edge server and, thus, has an impact on the accuracy of the classification's output for each cycle. The core of our framework is an optimization problem whose associated cost functions reflect channel usage and classification accuracy. To describe the former measure, we estimate the channel usage as a function of the subsets of coefficients.

The sensor-edge server system uses an iterative procedure, which we realize by means of a Sequential Hypothesis Testing engine, to determine how many and which coefficients are transmitted for each ECG cycle.

Let's focus on an individual ECG cycle, which is represented using a series of wavelets coefficients' sets $C_1, C2, \ldots, C_n$, and suppose that the sensor transmits the set C_1 first. The edge server reconstructs the signal, applies the classification algorithm, and based on its output uncertainty decides to request more coefficients. Then, the sensor transmits the set C_2, and this process continues till the desired accuracy of classification is fulfilled. At time slot j, if the sequence of coefficients $C = \{C_1, C_2, \ldots, C_j\}$ achieves the desired accuracy of classification $\gamma(C)$ then the process is terminated.

Clearly, while transmitting more coefficients reduces the error in the reconstructed signal, it also increases channel usage. Here, channel usage is computed considering that transferring coefficient $C_j : \forall j \in \{1, \ldots, k\}$ requires B_j number of bits. As the full representation of the signal requires B bits using all the coefficients.

The general optimization problem can be defined as the following,

$$\begin{aligned} k = \ & \underset{j}{\text{maximize}} \ \gamma(C_1, \ldots, C_j) \\ & \text{subject to} \frac{B_j}{B} \leq \theta \end{aligned} \tag{5.1}$$

Where k determines the optimal number of coefficients to reach a desired accuracy of classification. In this case, channel access is the ratio of bandwidth used by k number of coefficients over the number of bits to represent the total number of possible coefficients as B.

The problem, then, is to find for each ECG cycle the sequence that allows accurate classification while satisfying the requirements for channel usage. Notably, the sequence is not the same for each cycle, as it depends on its features, and cannot be predetermined. The objective of the Sequential Hypothesis Testing framework is to determine that sequence in real-time based on the outcomes of the classifier in each cycle. For instance, the most efficient sequence might be different for normal vs abnormal cycles, and for different subjects.

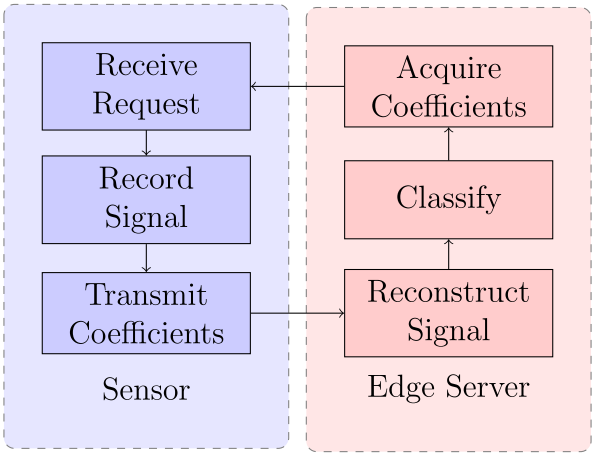


Figure 5.1: High-level schematics of the proposed System

5.2 Optimal Solution

In this section we provide the optimal solution to the problem we presented in Section 5.1. In Section 5.2.1, we propose Sequential Hypothesis Testing as an approach to solve the problem of determining the optimal stopping time, intended as the iteration at which the edge server does not request more coefficients to the sensor. Specifically, we adopt the optimal stopping time based on calculating the log-likelihood ratio of the coefficients, and prove that an optimal stopping time can be reached having statistical information of the signal. Section 5.2.2 discusses wavelet transform as a way to represent ECG based on number of coefficients.

5.2.1 Sequential Hypothesis Testing

Let $C_1, C_2, ..., C_j$ be a finite sequence of j coefficients to represent the signal at the edge server. As we explained in Section 5.1, the goal is to achieve the desired accuracy while saving communication bandwidth. To achieve this, the edge server and the sensor determine the optimal number of coefficients to represent a signal with a desired threshold of accuracy of classification. In the case of not fulfilling a threshold of accuracy, the edge server will request the sensor to transmit additional coefficients to reconstruct the signal accurately, since if too few coefficients have been transmitted the edge device is not able to reconstruct the signal adequately.

Typically, coefficients leading to high signal reconstruction accuracy require a larger amount of channel usage. Hence it is critical to choose the coefficients that minimize the channel access while still being able to provide the desired accuracy. Figure 5.2 shows how the system works. Suppose at each given time that the sensor transmits the coefficient C_j where $j \in \{1, 2, \ldots, n\}$ with n being the maximum number of coefficients, we observe an accuracy of γ_j with channel access of $\frac{B_j}{B}$. During the time slot $j+1$, the sensor receives feedback from the edge server. If more coefficients are requested, then, the sensor chooses the combination of coefficients (C_j, C_{j+1}) such that the accuracy of classification will satisfy a threshold while channel access is minimized. If the desired accuracy is not met, the edge device asks for more coefficients. The Sequential Hypothesis Testing stops at iteration k, corresponding to k coefficients transmitted, when the threshold of accuracy is fulfilled. The accuracy of classification is the performance of the classifier in terms of abnormality detection using the

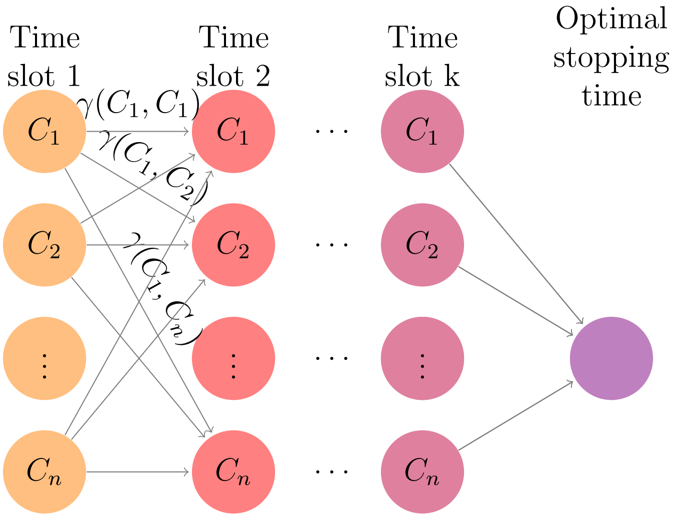


Figure 5.2: Proposed Sequential Hypothesis Testing method

reconstructed signal as a function of a wavelet transform with k coefficients, more formally:

$$p_{0k}(C_1, \dots, C_k) = P\{H(k) = H_0 \mid C_1, \dots, C_k\} \tag{5.2}$$

We now formally define the Sequential Hypothesis Test. Suppose that before the coefficients are chosen, there exists a-priori probability that an abnormality is detected under hypothesis H_0 and this probability is determined. We define this a-priori probability as p_j. In addition, we also define another a-priori probability of the alternative hypothesis H_1, which is defined as $1 - p_j$. After observing k time slots, the probability of detecting an abnormality in heart cycles p_j will be updated. According to Bayes formula the a-posteriori probability that H_0 is true can be calculated as,

$$g_{0j} = \frac{p_j p_{0j}(C_1, \dots, C_j)}{p_j p_{0j}(C_1, \dots, C_j) + (1 - p_j) p_{1j}(C_1, \dots, C_j)} \tag{5.3}$$

In this notation, $p_j(C_1, \dots, C_j)$ shows the probability density of observations from 1 to j under the hypothesis H_0. The sequence of coefficients $\{C_1, \dots, C_j\}$ is denoted as C for readability. The a posteriori probability that alternative hypothesis H_1 is true can be calculated as follows

$$g_{1j} = \frac{(1 - p_i) p_{1j}(C)}{p_j p_{0j}(C) + (1 - p_j) p_{1k}(C)} \tag{5.4}$$

Suppose that at each given time, we want to form a sequential test such that the conditional probability $g_{0k} \geq \zeta$.

$$\frac{p_j p_{0j}(C)}{p_j p_{0j}(C) + (1 - p_j) p_{1j}(C)} \geq \zeta \tag{5.5}$$

According to Equation 5.3, the fraction of a priori distribution can be written as,

$$\frac{p_{0j}(C)}{p_{1j}(C)} \geq \frac{\zeta(1 - p_j)}{(1 - \zeta) p_j} \tag{5.6}$$

Note that probability of occurrence of abnormal ECG cycles is independent with respect to the sequence of coefficients $C = \{C_1, \ldots, C_j\}$. Therefore, we can write the joint the probability in Equation 5.6 as the likelihood ratio,

$$\Lambda(C) = \frac{p_{01}(C_1) p_{02}(C_2) \ldots p_{0j}(C_j)}{p_{11}(C_1) p_{12}(C_2) \ldots p_{1j}(C_j)} \geq \frac{\zeta(1 - p_j)}{(1 - \zeta) p_j} \tag{5.7}$$

We model the distribution of the abnormal ECG cycle with a Gaussian Probability Density Function (PDF) f_0 with zero mean and variance σ_0^2, and normal ECG cycles with a Gaussian distribution f_1 with zero mean and variance σ_1^2, we calculate the log-likelihood ratio function for one sample x as follows:

$$\log \Lambda(C) = \log \left\{ \frac{f_0}{f_1} \right\} \tag{5.8}$$

Substituting the Gaussian distribution in the log-likelihood ratio,

$$\begin{aligned} \log \Lambda(C) &= \log \left\{ \sqrt{\frac{2\pi\sigma_1^2}{2\pi\sigma_0^2}} e^{-\frac{C^2}{2{\sigma_0}^2} + \frac{C^2}{2{\sigma_1}^2}} \right\} && (5.9) \\ &= \log \left\{ \sqrt{\frac{\sigma_1^2}{\sigma_0^2}} \right\} - \frac{C^2}{2{\sigma_0}^2} + \frac{C^2}{2{\sigma_1}^2} && (5.10) \end{aligned}$$

The cumulative sum of log-likelihood ratio functions over the sequence of coefficients is computed as

$$\begin{aligned} S_j &= \sum_{i=1}^{j} \log \Lambda(C_i) && (5.11) \\ &= j \log \left\{ \sqrt{\frac{\sigma_1^2}{\sigma_0^2}} \right\} - \left(\frac{1}{2{\sigma_0}^2} - \frac{1}{2{\sigma_1}^2} \right) \sum_{i=1}^{j} {C_i}^2 && (5.12) \end{aligned}$$

Leveraging the Sequential probability Ratio Test (SPRT) developed by Wald [69], the optimal stopping is achieved when

$$\begin{aligned} A < S_j < B : &\quad \text{Continue sampling} \\ S_j \leq A : &\quad \text{Accept } \mathcal{H}_1 \\ S_j \geq B : &\quad \text{Accept } \mathcal{H}_0, \end{aligned} \qquad (5.13)$$

where $\mathcal{H}_1$ represents a normal ECG cycle detection and $\mathcal{H}_0$ represents the detection of an abnormal cycle. The thresholds can be determined based on the desired probability of detection P_D and false alarm P_{FA}. The probability of abnormality detection given the sequence of coefficients, $C = \{C_1, \ldots, C_j\}$, is $R = \{C : \Lambda(C) > \exp(B)\}$,

$$P_D = \int_R p_{0j}(C)dC = \int_R \frac{p_{0k}(C)}{p_{1j}(C)} p_{1j}(C)dC \tag{5.14}$$

$$= \Lambda(C)p_{0j}(C) \geq \exp(B) \int_R p_{0j}(C) = \exp(B)\, P_{FA} \tag{5.15}$$

Similarly with $R_0 = \{C : \Lambda(C) > \exp(A)\}$,

$$\begin{aligned} 1 - P_{FA} &= 1 - \int_R p_{1j}(C)dC = \int_{R'} p_{1j}(C)dC \\ &= \int_{R'} \frac{p_{1j}(C)}{p_{0j}(C)} p_{0j}(C)dC = \int_{R'} \Lambda^{-1}(C)p_{0j}(C)dC \\ &\geq \exp^{-1}(A) \int_{R'} p_{0j}(C)dC = \frac{1}{\exp(A)}(1 - P_D) \end{aligned} \tag{5.16}$$

Therefore, A and B can be calculated as,

$$A \approx log \frac{1 - P_D}{1 - P_{FA}} \text{ and, } B \approx log \frac{P_D}{P_{FA}} \tag{5.17}$$

Algorithm 3 Sequential Hypothesis Algorithm: Edge Server

```
procedure SOLUTION($C_1, C_2, ..., C_j$)
    Calculate $S_j$ according to Equation 5.12
    if $A < S_j < B$ then
        Send a Request to the sensor for more coefficients
    if $S_j \leq A$ then
        Return $C_1, \ldots, C_j$, normal ECG cycle
    if $S_j \geq B$ then
        Return $C_1, \ldots, C_j$, abnormal ECG cycle
    until the system is terminated
```

Algorithm 3 shows the sequential hypothesis testing on the edge server to determine the optimal stopping time. The sensor then determines the coefficients such that it satisfies the constrained bandwidth requirement and achieves the desired accuracy of classification.

5.2.2 Wavelet Transform

We propose Wavelet Transform as a methodology to transform the raw signal into a sequence of coefficients. In this section, we describe how to reconstruct a signal based on accumulated coefficients and introduce a metric to measure the difference between the reconstructed signal and the original signal. The wavelet transform uses scalable modulated windows to convolve the wavelet functions with the original signal. In other words, the output signal is the result of a convolution of the raw signal with a cutting window, and the window is shifted with every position on which the spectrum is calculated [67]. The reconstructed signal with k optimal coefficients on orthonormal basis functions $\{\psi_{i,j} : i, j \in \mathbb{Z}\}$ can be written as:

$$\hat{x}_k = \sum_{i \in k} \sum_{j} \langle x[m], \psi_{i,j}[m] \rangle \psi_{i,j}[m] \tag{5.18}$$

The mother wavelet $\psi_{i,j}$ is defined as,

$$\psi_{i,j}[m] = \frac{1}{\sqrt{2^i}} \psi \left[\frac{m - j2^i}{2^i} \right] \tag{5.19}$$

The wavelet coefficient $\phi_{k,j}$ of signal x, then, is the projection of x onto wavelet $\psi_{i,j}$,

$$\phi_{ij} = \sum_{m} x[m] \psi_{k,j}[m] \tag{5.20}$$

Where i and j are the scale and shift parameters respectively. The Percent Root Difference (PRD) is a metric to determine the accuracy of the reconstructed signal compared to the original signal, defined as follows

$$\alpha_k = \sqrt{\left(\frac{\sum_{m=1}^{M}[x[m] - \hat{x}_k[m]]^2}{\sum_{m=1}^{M} x^2[m]}\right)} \tag{5.21}$$

We summarize again the information flow in Figure 5.3. In this system, the ECG sensor collects the raw data x and transmits the coefficient C_j to the edge server. The edge server reconstructs the signal $\hat{x}_j$, using the accumulated coefficients $C_1, \ldots, C_j$. The convolutional neural network classifier uses the reconstructed signal and generates the Probability Density Function π_j for arrhythmia and normal ECG signal. The output of the PDF classifier will be used in Sequential Hypothesis Testing using Algorithm 3 to determine the optimal stopping time. The sequential test performs the Wald test introduced in Equation 5.13 and decides whether to send a request for more coefficients or detect abnormal vs arrhythmia ECG cycle. Meanwhile, the sensor receives the request from the edge server to determine the specific coefficient to be transmitted. The sensor chooses the coefficient such that the accuracy of classification in the edge is maximized while the bandwidth consumption is less than a desired threshold.

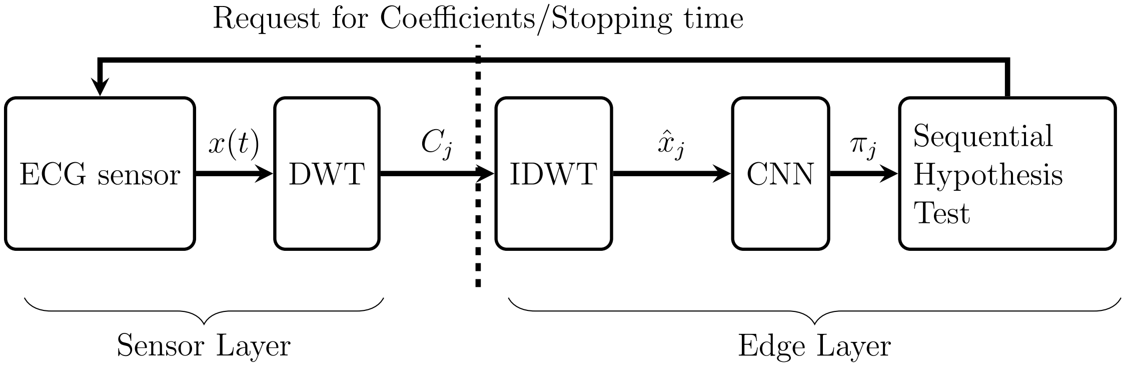


Figure 5.3: Block Diagrams of the System

5.3 Arrhythmia Classification

In the previous sections we assumed a classifier exists at the edge server, supporting both the main task of the system (cycle classification), and the selection of the coefficients. Herein, we specify the architecture of the classifier we used in our experiments.

We use a one-dimensional convolutional neural network (CNN) to classify ECG cycles into five different classes, presented by Takalo-Mattila *et al.* [60]. With CNN based classification we are able to minimize the need of feature engineering and utilize filtered ECG as an input signal for the classifier. In our approach, the labeled ECG heart cycles are fed to the classification model together with heart cycles before and after the labeled heart cycle. This is how our classifier can automatically utilize both morphological features of the heart cycle and temporal relation between its neighboring heart cycles.

5.3.1 Data

Two different clinical ECG signal databases are used in this study, namely, MIT-BIH arrhythmia database (MIT-BIH) and MIT-BIH Supraventricular Arrhythmia database (MIT-SVDB) [45, 25, 24]. All datasets used in this study are shown in table 5.1. The MIT-BIH database contains 48 samples of thirty minutes obtained from two-channel ECG recordings from a total of 47 patients, with a sampling rate of 360 Hz. The MIT-SVDB dataset contains 78 samples of thirty minutes ECG recordings sampled at 128 Hz. With both databases we are using the recommendation of the Association for the Advancement of Medical Instrumentation (AAMI) to arrange MIT-BIH and MIT-SVDB arrhythmia labels into five classes, which are Normal (N), Supraventricular Ectopic beat (SVEB), Ventricular Ectopic beat (VEB), Fusion Beat (F) and Unknown beat (Q) [18]. Figure 5.4 shows the raw signal of an ECG cycle with Ventricular Ectopic Beat.

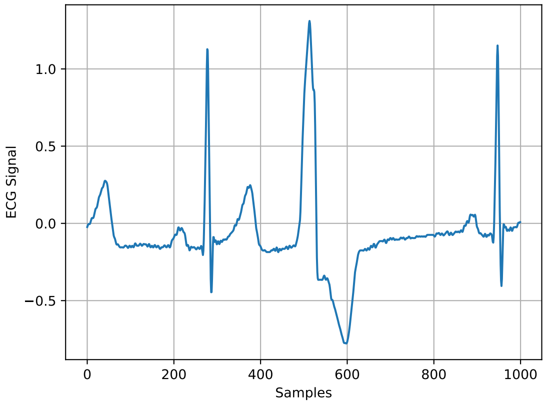


Figure 5.4: Raw ECG signal with Ventricular Ectopic Beat cycle

We remove 4 ECG records in the MIT-BIH database as they contain paced beats. The MIT-BIH database is divided into training and test dataset using the convention presented by De Chazal *et al.* [17]. In this dataset we divide patients into training and test sets, so that no patient is present in both sets. Hence, we can also estimate the generalization capability of the classifier more realistically.

In addition to MIT-BIH arrhythmia database, MIT-SVDB is used for evaluating the classification performance. With this dataset, we can test how our classification approach would work in practical clinical work. The classifier is trained using dataset DS1, whose subset DS11 is used in actual training and DS12 is used in the validation phase. Finally, the classifier is evaluated using datasets DS2 and SVDB. Each of these datasets contains ECG data from different patients. Specific patient numbers in MIT-BIH based datasets are the same that were used by Takalo-Mattila *et al.* [60]. In SVDB dataset, all the patients from SVDB database are used.

Table 5.1: Databases and datasets in the study

Database	Dataset	Number of patient files
MIT-BIH [43]	DS1	22
	DS11	11
	DS12	11
	DS2	22
MIT-SVDB[44]	SVDB	78

5.3.2 Classification approach

There are two main steps in our arrhythmia classification approach. At first, the quality of ECG signal is improved using a lightweight preprocessing. Secondly, the classification model is trained using preprocessed ECG samples.

In the preprocessing phase the time series ECG signal is filtered using:

- 0.5 Hz high pass filter for removing baseline wander
- 100 Hz low pass filter for removing electromyographic (EMG) noise
- 60 Hz band stop filter for removing power line noise

Eventually the ECG signal is normalized to range [-1,1] after filtering. Filtered and normalized time series ECG signals are segmented into a set of heart cycles, where one segment contains a set of heart cycles with 500 samples before and 500 samples after R-peak that represents typically the largest wave in QRS complex. Using the datasets presented in Section 5.3.1 one segment contains three to seven separate heart cycles. Figure 5.5 shows the layers of the CNN architecture used in the system.

The CNN architecture in this thesis contains four convolutional layers and a fully connected layer. The set of heart cycles is used as input data for a one dimensional convolutional

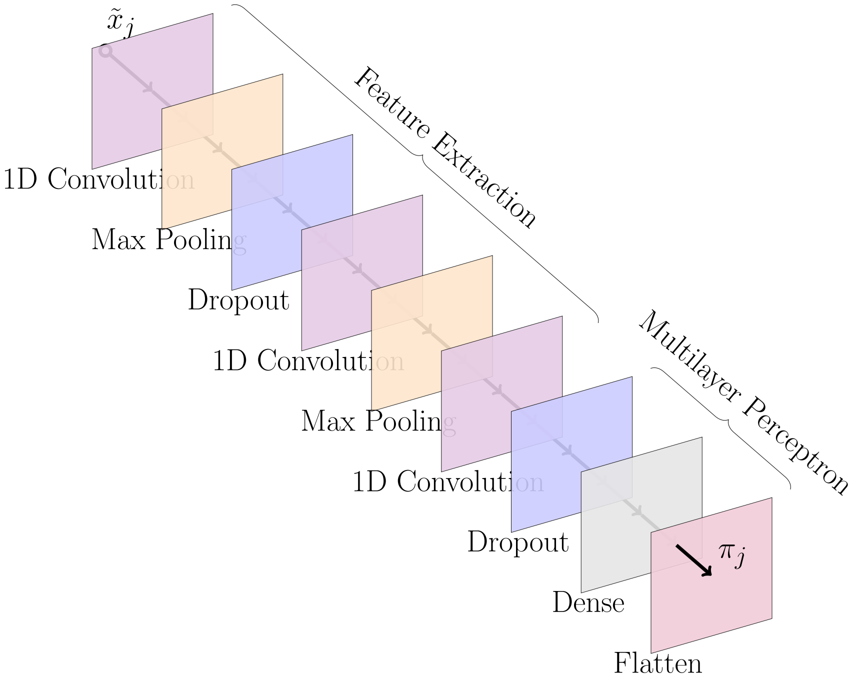


Figure 5.5: Architecture of CNN

layer. Convolutional neural networks are often used in image analysis, but can be utilized also to learn features from other types of signals, such as ECG data. In a convolutional layer, the input is modified by the set of filters that slides through the whole input of layer. Max pooling and dropout layers are used between convolutional layers, as their role is to prevent overfitting and computational cost. Max pooling reduces the size of the input by taking the maximum value from the filter region and creates a new output smaller than the original input. In dropout layers some neurons are randomly dropped in the training phase, to prevent overfitting. The dense layer finally performs the actual classification based on the features learned in previous convolutional layers. The fully connected layer has 512 neurons in our architecture, and the probability distribution of the heart cycle class is calculated using the softmax function and the predicted class is calculated using the argmax function.

5.4 Experimental Results

In this section, we illustrate the performance of the proposed system. Figure 5.6 evaluates the PRD of DWT with different number of the coefficients defined in Equation (5.21), where we used different combinations of DWT coefficients and reconstructed the signal based on the information those coefficients provide. We, then, calculate the PRD according to Equation (5.21), where the reconstructed signal degradation is compared to the original raw signal. Intuitively, increasing the number of coefficients helps in reconstructing the signal more accurately, which results in a decreased PRD. Since different combinations of coefficients create different error values, we used bar plots to show the range of PRD changes compared to the average error as a function of the number of coefficients. As it can be seen, the maximum number of DWT coefficients reconstructs the signal with an error close to zero. Thus the transmission of the full set enables accurate reconstruction, which leads to performance analogous to those obtained with the original signal. However, we are interested in evaluating the impact of transmitting a lower number of coefficients on signal distortion. We remark that the DWT of a signal is calculated by using a series of filters as shown by Lee *et al.* [36]. The signal is decomposed into high and low frequency ranges and then down-sampled in a cascading fashion. As an example, results show that with $k = \{1, 2, ..., 7\}$ coefficients, we can achieve a mean PRD percentage of $\{90\%, 79\%, 68.3\%, 56\%, 42\%, 25\%, 8.4e - 14\%\}$.

We divide our experiments into two phases. The first one is the *training* phase and the second one is the *testing* phase. During the *training* phase, the CNN classifier is evaluated with different coefficients. Eventually, the *Testing* phase evaluates Sequential Hypothesis Testing using the results obtained through the *training* phase. In the following we describe better each phase.

Training Phase: in this phase, we use the SVDB data set to train the algorithm. In total 63 patients were used to calculate the positive predictivity of normal ECG cycles, SVEB and

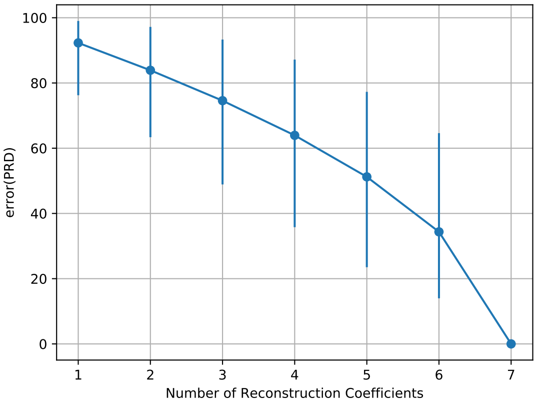


Figure 5.6: PRD of the reconstructed signal compared to the original signal

VEB. We recall that the positive predictivity is defined as *rate of true positives by total true positives and false positives*, more formally

$$Pp = \frac{TP}{TP + FP}. \tag{5.22}$$

This metric shows the performance of the model to predict the positive class. Values close to 1 indicates an accurate model, while lower values show an increasing number of False Positives classifications. As we increase the number of coefficients, the average positive predictivity for normal ECG, and abnormal cycles with SVEB and VEB diseases increases. Results from the *Training* phase on the SVDB dataset are shown for the following number of DWT coefficients $k = \{1, 2, 3, 4, 5, 6, 7\}$.

In this case, Pp ranges in $\{0.886, 0.887, 0.89, 0.892, 0.895, 0.898, 0.899\}$. In addition, SVEB positive predictivity with mean,

$$\{0.14, 0.21, 0.25, 0.27, 0.29, 0.3, 0.29\}$$

,

and VEB is $\{0.38, 0.49, 0.56, 0.64, 0.69, 0.7, 0.7\}$.

The goal of the optimization problem in Equation 5.1 is to minimize the bandwidth usage while targeting for a threshold of accuracy classification. Accuracy targets trained for the training data set with coefficients varied from 1 to 7 while ranging in the set of $\{0.872, 0.873, 0.874, 0.876, 0.879, 0.88, 0.88\}$.

The false positive rate (FPR) is defined as

$$FPR = \frac{FP}{TN + FP},$$

which is the *ratio of false positives divided by total true negatives and false positives(total actual negative events).* The FPR normal metric is

$$\{0.93, 0.914, 0.891, 0.866, 0.84, 0.816, 0.8\},$$

and FPR Sup is

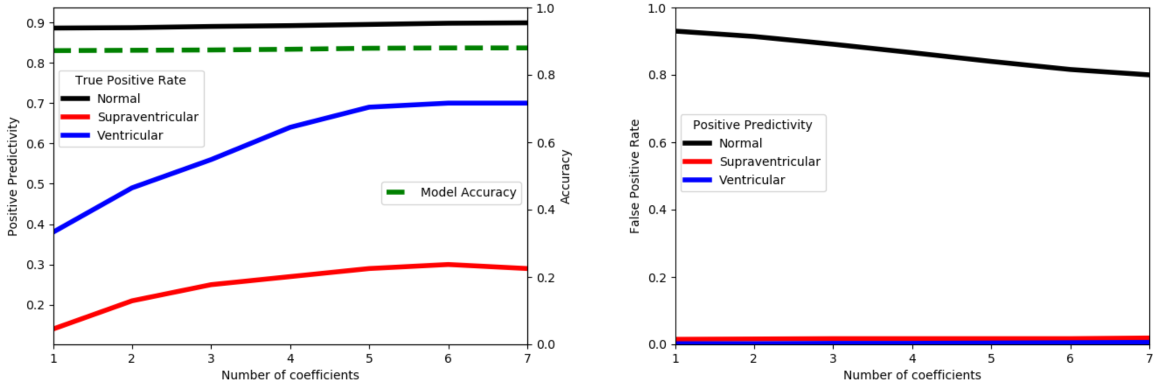
$$\{0.015, 0.016, 0.017, 0.017, 0.017, 0.017, 0.019\},$$

and FPR Vent is

$$\{0.002, 0.002, 0.003, 0.003, 0.004, 0.005, 0.006\}.$$

Figure 5.7 shows the performance of CNN classifier using different combinations of DWT coefficients. What is interesting to see is that although a larger number of coefficients yields better results, even with a reduced number of coefficients, the results are reasonably high. Moreover, FPR normal achieves almost identical results with respect to the positive predictivity, while FPR Sup and FPR Vent have almost zero false positive rate regardless of the number of coefficients.

Testing Phase: We used the remaining 14 subjects in the SVDB dataset to evaluate the



(a) (b)

Figure 5.7: Results from the training phase. In Figure 5.7a We show the Positive Predictivity and the model accuracy. It is clear how the normal ECG can be recognized well regardless of the number of coefficients, while Supraventricular and Ventricular ECG states benefits from more coefficients to reconstruct the signal. Figure 5.7b shows instead the FPR.

Testing phase. In this case, 2/3 of each patient's data is used to calculate the variance of probability in the CNN classifier. The Variance of probability distribution over the different classes of abnormal and normal ECG cycles can be used to derive the Gaussian distribution defined in Equation 5.10. Using all combinations of the coefficients, the variance of probability for normal cycles is compared to 4 arrhythmia classes, including Ventricular Ectopic Beats(VEB), Supraventricular Ectopic Beats(SVEB), Q beats and F beats. We plot in Figure 5.8 the variance of corresponding cycles for a single user. To determine the variance of probability in CNN classifier, we use the personalized statistical information to test the Sequential testing for the rest of 1/3 of data.

Using the data from the user in Figure 5.8, we calculate the optimal stopping time for a single abnormal ECG cycle, which we show as an example in Figure 5.4. This raw signal includes an ECG cycle with Ventricular Ectopic Beat. During the first time slot, the sensor chooses the first coefficient that maximizes the accuracy of classification in the edge server while satisfying a threshold of bandwidth usage. Figure 5.10a shows the reconstructed signal in the edge server using the first coefficient that was transmitted from the sensor. While receiving the

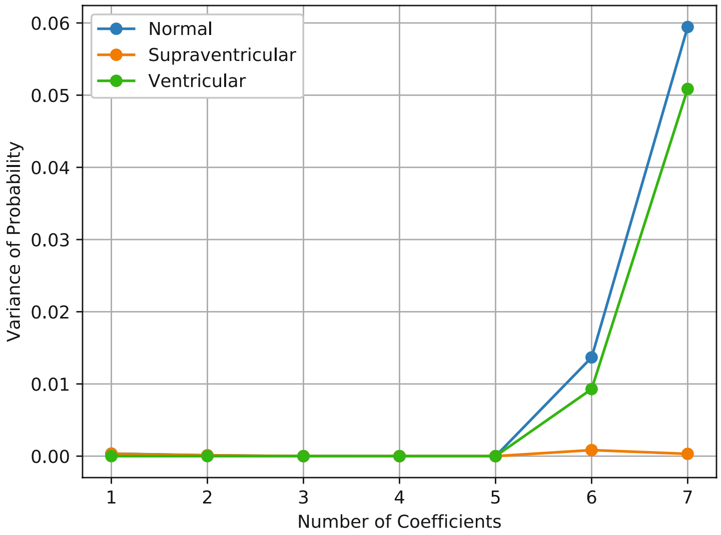


Figure 5.8: Convolutional Neural Network performance as a function of DWT coefficients

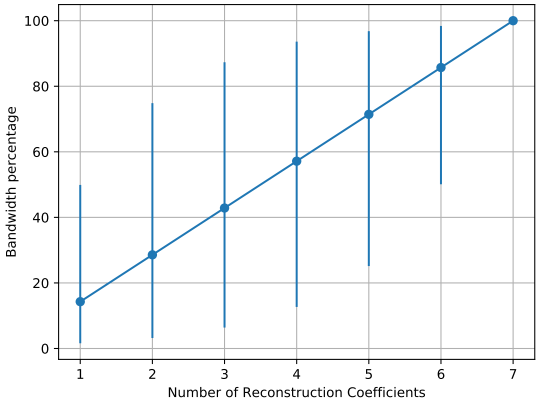


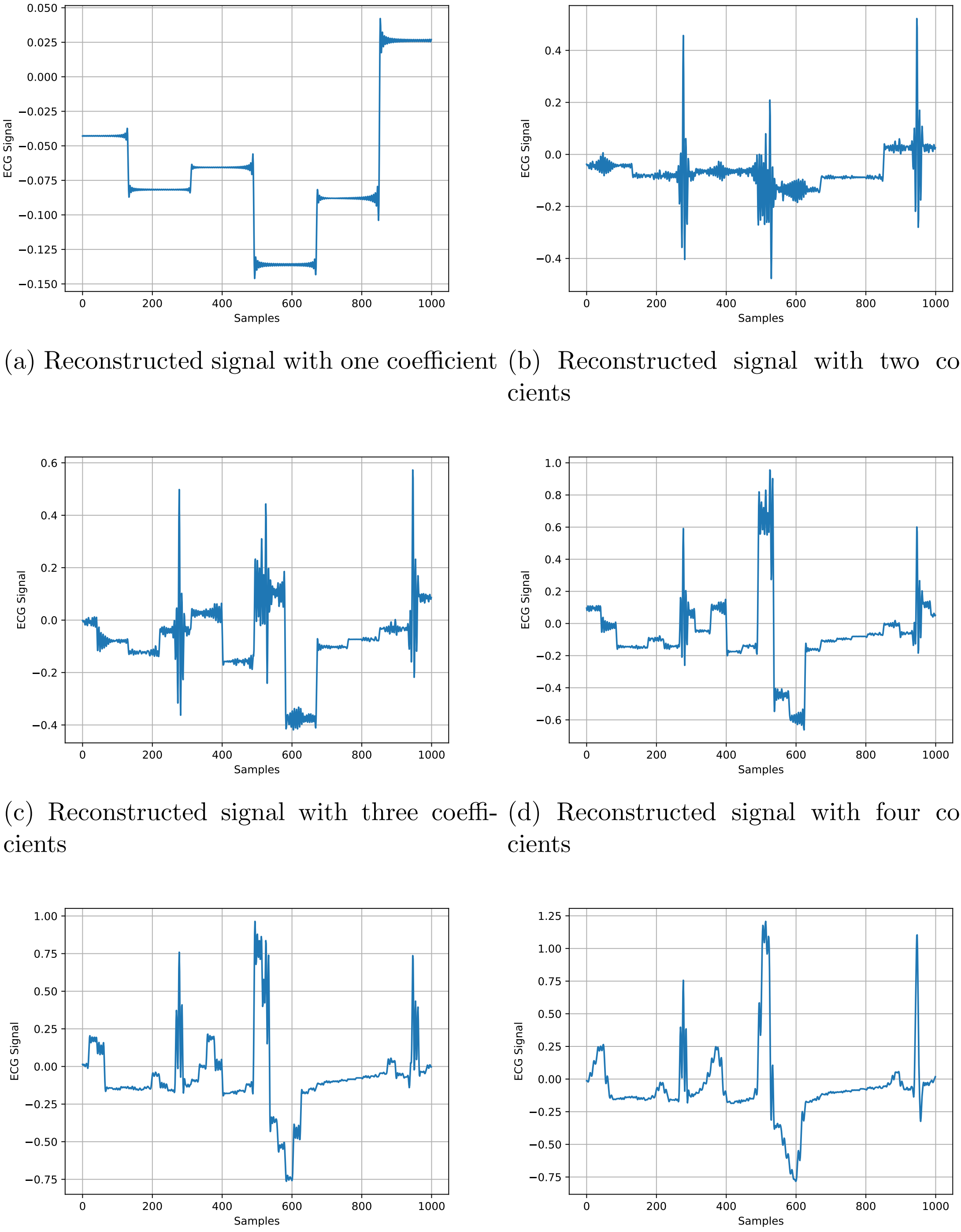
Figure 5.9: Bandwidth percentage as a function of DWT coefficients

first coefficient, edge server classifies the reconstructed signal using CNN trained during the *training* phase. The classifier outputs a model's prediction probability, following which we calculate the log likelihood of the corresponding output of the CNN classification. Sequential Hypothesis Testing evaluates classifier's probability using Wald stopping time, as we already discussed in Section 5.2.1. It determines whether more coefficients are needed in order to correctly classify the data, or it detects abnormal/normal ECG cycle based on the collected coefficients. In this case, we set an accuracy of CNN classification for abnormality detection as 88 percent. Clearly, higher thresholds would increase the number of coefficients needed, while lower thresholds may be satisfied with a lower number of coefficients. Figure 5.10 shows Sequential Hypothesis Testing required 6 coefficients to achieve this target. Requiring more coefficients helps to reconstruct the signal more accurately. Therefore, the reconstructed signal in Figure 5.10 f resembles to the original raw signal recorded by the sensor in Figure 5.4, at the cost of an increased bandwidth usage.

Figure 5.11 shows the Sequential of DWT coefficients that the sensor chooses for the mentioned abnormal cycles discussed earlier. At each time slot, the corresponding coefficient which satisfies the overall threshold of bandwidth consumption is chosen.

The Percent Root mean square Difference (PRD) discussed in Equation 5.21 between the reconstructed signal and the original signal is calculated for each time slot. This percentage value determines the distortion between the reconstructed signal and the original signal. Higher values of PRD demonstrate higher distortion compared to the original signal.

In this experiment, we set the goal of reaching 88% of accuracy of classification in the edge device with the threshold of bandwidth consumption of 80%. We select 88% as target accuracy as it can show better than others the difference between normal and abnormal classifications, but any accuracy can be selected according to the applications' requirements. Bandwidth consumption is assumed to be the percentage of set of coefficients compared to a full representation of signal using all the coefficients. For instance, the first coefficient that is transmitted, which we show in Figure 5.10a, consumes only 1 percentage of bandwidth with PRD 96.01%. VEB disease is misdetected as a normal cycle and the edge server sends a request for more coefficients to reach the goal of 88% accuracy of detection. The sensor transmits the second coefficient and the edge server reconstructs the signal using the first and second coefficients. The output of CNN classifier is the probability distribution of arrhythmia versus normal cycle and the Sequential Hypothesis Testing method calculates the log likelihood of probability for collected coefficients and based on predefined thresholds in Equation 5.17. It is worth mentioning that the CNN classifier misdetects an abnormal cycle



(a) Reconstructed signal with one coefficient

(b) Reconstructed signal with two coefficients

(c) Reconstructed signal with three coefficients

(d) Reconstructed signal with four coefficients

(e) Reconstructed signal with five coefficients

(f) Reconstructed signal with six coefficients

Figure 5.10: ECG signal with Ventricular Ectopic Beat cycle for Patient “One”

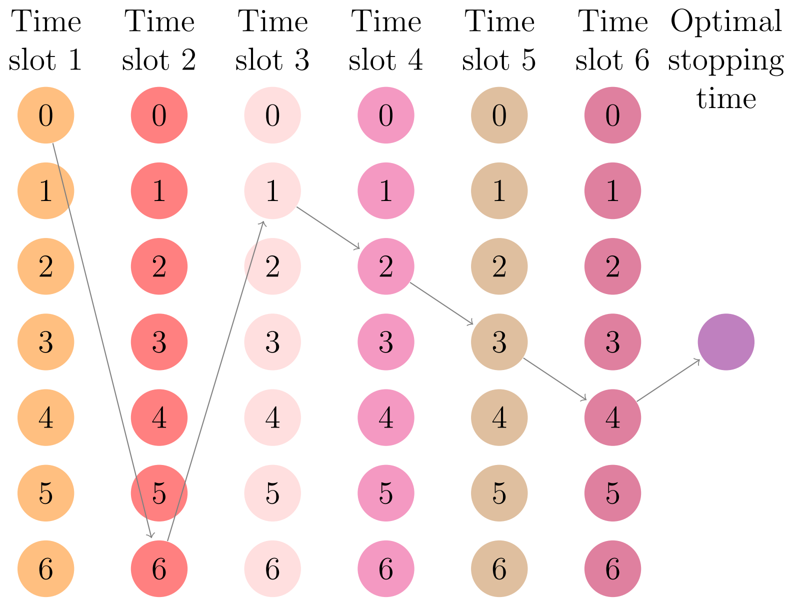


Figure 5.11: Sequential Testing for an abnormal cycle for Patient “One”

as normal. Therefore, more coefficients are needed to detect the abnormality with higher accuracy of classification. We used data for aforementioned user and tagged them as patient “One”. Table 5.2(a) shows PRD, bandwidth consumption, sequence of coefficients chosen, the type of arrhythmia, and output of CNN classifier during the sequential hypothesis are shown. The percentage of bandwidth consumption is calculated according to the maximum bandwidth, which is consumed when all the coefficients are transmitted. In addition, Table 5.2(b) shows the performance of sequential hypothesis for a single normal ECG cycle of patient “One”. Results show that optimal stopping time for an abnormal cycle is higher than a normal cycle. This can be useful since the classifier can detect an abnormality with a higher confidence since the higher number of coefficients provide a more accurate representation of the ECG signal.

Table 5.3 shows performance of Sequential Hypothesis Testing for another patient tagged as patient “Two”. In this case, the sequence of coefficients are different from patient “One”. This concludes that the proposed system is working as a personalized model since the Training phase on finding the variance of probability using CNN classifier is trained on each patient

PRD	Bandwidth	Coefficients	Heart Condition	Detection
96.01	0.015	$\{0\}$	VEB	Normal
94.09	0.51	$\{0, 6\}$	VEB	Normal
85.50	0.53	$\{0, 6, 1\}$	VEB	Normal
58.05	0.56	$\{0, 6, 1, 2\}$	VEB	VEB
51.13	0.62	$\{0, 6, 1, 2, 3\}$	VEB	VEB
30.52	0.75	$\{0, 6, 1, 2, 3, 4\}$	VEB	VEB

(a) One abnormal ECG cycle

PRD	Bandwidth	Coefficients	Heart Condition	Detection
93.8	0.015	$\{0\}$	Normal	Normal
88.74	0.51	$\{0, 6\}$	Normal	Normal
87.57	0.53	$\{0, 6, 1\}$	Normal	Normal
80.99	0.56	$\{0, 6, 1, 2\}$	Normal	Normal

(b) One normal ECG cycle

Table 5.2: Performance of Sequential Hypothesis Testing for Patient “One”

individually. Therefore, the sequence of coefficients to reach the same accuracy as patient “One” is different. We can conclude that bandwidth and sequence of coefficients are varied among people and between normal and abnormal ECG cycles. Table 5.3 (a) demonstrates that sequential testing for an abnormal cycle requires more coefficients than a normal cycles in Table 5.3 for the same patient.

The performance of Sequential testing is evaluated based on achieving different goals of accuracy in classification and are calculated on the average bandwidth consumption for normal and abnormal ECG cycles for the aforementioned user. Figure 5.12 shows the bandwidth consumption as a function of threshold in achieving accuracy of classification. In fact, increasing the goal of accuracy eventually results in requesting more coefficients in the edge server, hence consuming more bandwidth. With accuracies beyond 90% results in maximum request for the coefficients for both abnormal and normal cycles with the average of 50% bandwidth consumption. It is important to note that the abnormal cycles and normal cycles require different number of coefficients to reach the same accuracy. In fact, between accuracy

PRD	Bandwidth	Coefficients	Heart Condition	Detection
87.64	0.25	$\{5\}$	VEB	Normal
86.14	0.31	$\{5,3\}$	VEB	Normal
72.29	0.34	$\{5,3,2\}$	VEB	VEB
69.82	0.36	$\{5,3,2,0\}$	VEB	VEB
51.23	0.37	$\{5,3,2,0,1\}$	VEB	VEB
30.81	0.5	$\{5,3,2,0,1,4\}$	VEB	VEB

(a) One abnormal ECG cycle

PRD	Bandwidth	Coefficients	Heart Condition	Detection
95.13	0.25	$\{5\}$	Normal	Normal
87.46	0.31	$\{5,3\}$	Normal	Normal
83.38	0.34	$\{5,3,2\}$	Normal	Normal

(b) One normal ECG cycle

Table 5.3: Performance of Sequential Hypothesis Testing for Patient "Two"

of 80% to 87% abnormal cycles require 19% more bandwidth than the normal cycles. This concludes that sequential testing chooses adaptive coefficients between normal and abnormal cycles to reconstruct the abnormal cycles with more coefficients to detect abnormality more accurately.

It is straightforward to understand that if high accuracies are needed, the reconstructed signal should be as close as possible to the original raw signal. This would yield better results for the CNN classifier, achieving the desired high accuracy. However, there may be scenarios in which a lower accuracy would suffice, particularly for normal cycles. In Figure 5.12 we plot the bandwidth needed to transmit the required as a function of the desired accuracy for normal and abnormal cycles. For accuracies beyond 0.86, both normal and abnormal cycles require the use of the same amount of bandwidth, since the signal has to be classified with higher accuracy in order to meet the application requirement. However, for lower accuracies, the normal cycle is easier to be recognized, as it requires less bandwidth compared to the abnormal cycles. This shows that sequential testing chooses adaptive coefficients between

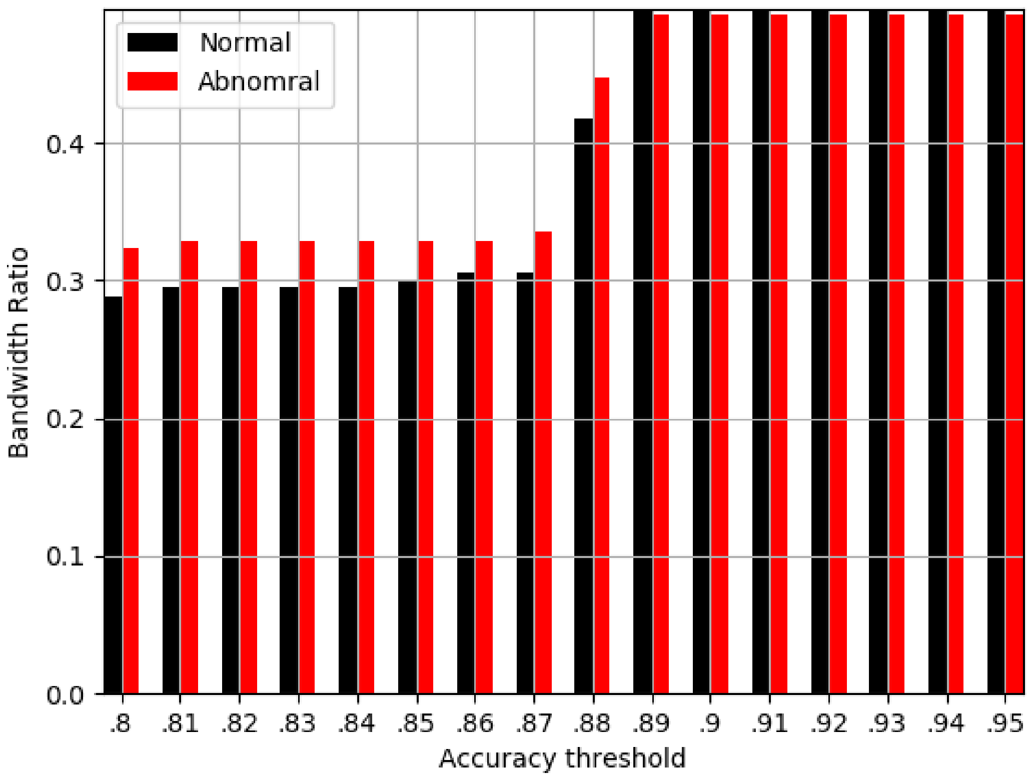
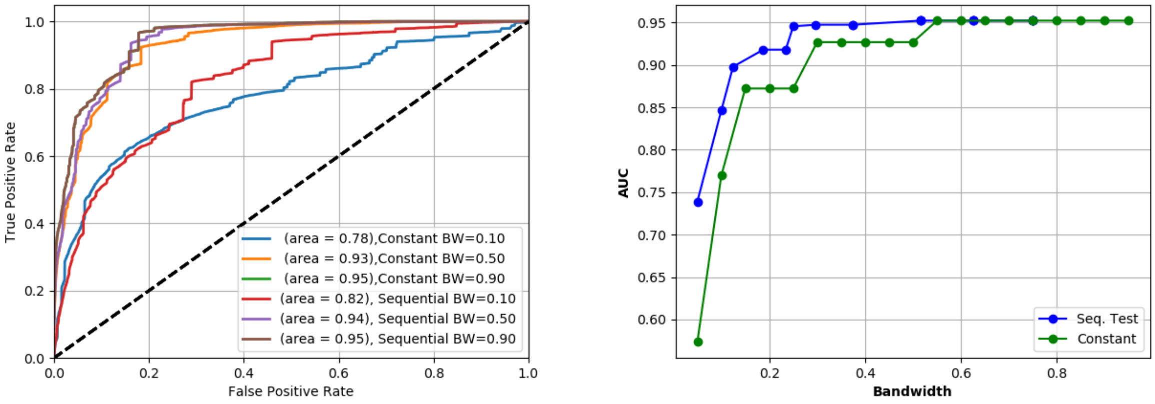


Figure 5.12: Overall bandwidth consumption as a function of accuracy classification.

normal and abnormal cycles to reconstruct the abnormal cycles with more coefficients to detect abnormality more accurately, while it sends a lower number of coefficients for lower accuracies, in case a normal cycle has to be detected.

We evaluated the system performance based on the constraint of bandwidth. In this case, we considered that the bandwidth consumption using Sequential method should satisfy a specific threshold. The constant coefficients are chosen in cascading fashion from the highest to lowest frequency such that the coefficients satisfy the bandwidth constraint. We plot in Figure 5.13a the True Positive Rate as a function of False Positive rate known, also known as the Receiver Operation Characteristic (ROC) curve. We compare the Sequential and constant methods with three different values of bandwidth constraints (e.g. $\{10\%, 50\%, 90\%\}$). Results show that Sequential testing outperforms the constant method in Area Under the

Curve (AUC) in ROC with lower bandwidth constraints. We also show in Figure 5.13b the comparison of Sequential testing with constant coefficient method as a function of actual bandwidth consumption in the sensor. The results show that for the same percentage of bandwidth consumption, Sequential testing outperforms the constant policy, particularly for lower bandwidth constraints. Bandwidth consumption of 0.5 and more shows the same AUC for constant and Sequential testing methods.



(a)

(b)

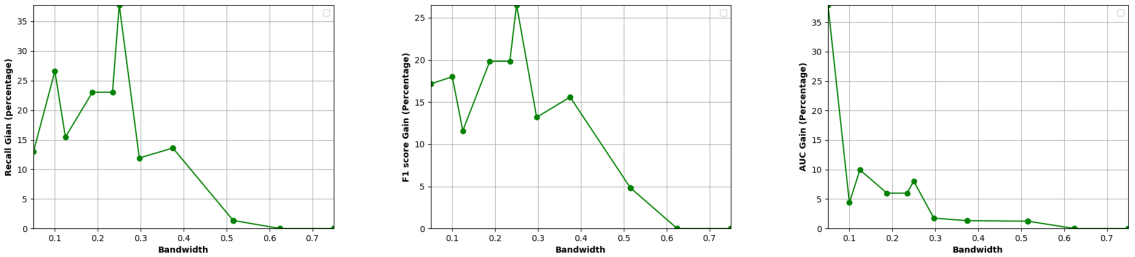
Figure 5.13: Classification results. Figure 5.13a shows the Receiver Operating Characteristic (ROC) curve, while Figure 5.13b shows instead the Area Under Curve (AUC) as a function of bandwidth consumption.

Table 5.4 shows the minimum bandwidth required to achieve accuracy ranging between .75 to .95. For instance, Sequential hypothesis requires minimum of 24 bandwidth percentage to achieve accuracy of .95 while constant policy requires minimum of 55 bandwidth percentage which shows 56% bandwidth reduction in Sequential testing compared to constant coefficients. Results show average of 43% of bandwidth reduction using Sequential testing.

Figures 5.14a,5.14b,5.14c show the gain of Sequential testing compared to policy of constant coefficients as a function of the available bandwidth. On average, the gain for the Sequential method of the recall score is 15.86% over the constant method, as Figure 5.14a shows. Similarly the F1 score gain averages at 14.08%, and the AUG gain averages at 7.98%. Overall,

AUC	Constant Min Bandwidth	Sequential Min Bandwidth	Bandwidth Reduction
.95	.55	.24	56%
.92	.28	.19	32%
.90	.27	.12	55%
.86	.14	.10	28%
.75	.09	.05	44%

Table 5.4: Minimum Bandwidth required for a given accuracy and Bandwidth Reduction Percentage of Sequential testing compared to constant policy



(a) Recall Gain

(b) F1 score Gain

(c) AUC Gain

Figure 5.14: Gain achieved using Sequential Testing compared to Constant coefficients as a function of bandwidth consumption

these 3 figures show that the Sequential Hypothesis Testing always performs better than the constant method, and the lower the bandwidth, the higher the gain, as the Sequential method is able to better select the appropriate coefficients.

Chapter 6

Discussions and Conclusions

The IoT paradigm, through a high-level network of connected devices, enables ubiquitous health monitoring for patients at risk in everyday settings. Such a remote IoT-based health monitoring system typically includes battery-powered sensors which require a satisfactory energy-efficient control approach. Contemporary published approaches mostly focus on optimizing the power consumption of the sensor layer that may compromise the accuracy of measurements.

In this thesis we proposed two frameworks, *myopic* and Markov Decision Processes (MDP) to control the energy-constrained sensor layer. We used the key low-latency characteristics of edge computing to build a context-aware control system whose objective is to adaptively determine the sensing accuracy, and thus the energy dissipation, to maximize the sensor lifetime. We modeled the accuracy of sensor measurement in capturing abnormality as a Gaussian distribution. The optimization problem minimizes the energy consumption during different contexts (e.g., activity), to achieve a satisfactory Probability of misdetection in abnormality. We extend our implementations to monitor 14 healthy subjects for one month.

We demonstrated that the *myopic* method saves on average 16.9% compared to the non-adaptive case in a 24-hour health monitoring period by adapting to an activity while assuring a minimum level of Probability of misdetection in abnormality. Moreover, our results show that MDP can increase the battery lasting with average of over $2x$ compared to *myopic* and non-adaptive power consumption fulfilling the same probability of mis-detection during one month. As the future work, we will investigate calculating the distance of adjacent policies among different subjects.

In addition, we studied the bandwidth efficiency of wearable devices. Monitoring patients in everyday settings necessitate the use of a medical sensor coupled with an edge device where the sensor transmits the signal from the bandwidth constrained sensor to the edge server, though transmitting raw signal over a constrained wireless channel is expensive. Therefore, transmitting a representation of the signal instead of the raw signal opens new perspectives of optimizing the bandwidth consumption between sensor and the edge device. To address these issues, we proposed a framework to determine the optimal number of coefficients using Sequential Hypothesis Testing to optimize the accuracy of detecting abnormal ECG cycles while satisfying requirements for channel access between the sensor and the edge device. Results show that an average of 43% of bandwidth is saved while satisfying the same accuracy of arrhythmia detection on the edge server over 14 subjects compared to the constant method. We have also shown that the proposed model is personalized on each user, as ECG cycles are different hence require a different number of coefficients to be sent.

Bibliography

[1] S. F. Abedin, M. G. R. Alam, R. Haw, and C. S. Hong. A system model for energy efficient green-iot network. In *Information Networking (ICOIN), 2015 International Conference on*, pages 177–182. IEEE, 2015.

[2] N. Alajlan, Y. Bazi, F. Melgani, S. Malek, and M. A. Bencherif. Detection of premature ventricular contraction arrhythmias in electrocardiogram signals with kernel methods. *Signal, Image and Video Processing*, 8(5):931–942, 2014.

[3] D. Amiri, A. Anzanpour, I. Azimi, M. Levorato, P. Liljeberg, N. Dutt, and A. M. Rahmani. Context-aware sensing via dynamic programming for edge-assisted wearable systems. *ACM Trans. Comput. Healthcare*, 1(2), Mar. 2020. `https://doi.org/10.1145/3351286`.

[4] D. Amiri, A. Anzanpour, I. Azimi, M. Levorato, A. M. Rahmani, P. Liljeberg, and N. Dutt. Edge-assisted sensor control in healthcare iot. In *2018 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6. IEEE, 2018.

[5] D. Amiri, A. Anzanpour, I. Azimi, A. M. Rahmani, P. Liljeberg, N. Dutt, and M. Levorato. Optimizing energy efficiency of wearable sensors using fog-assisted control. In *Fog Computing: Theory and Practice*. Wiley, 2020. `https://doi.org/10.1002/9781119551713.ch9`.

[6] D. Amiri, J. Takalo-Mattila, L. Bedogni, M. Levorato, and N. Dutt. Optimal accuracy bandwidth tradeoff for ecg classification via sequential hypothesis testing. *ACM Trans. Comput. Healthcare*, 2020.

[7] A. Anzanpour, D. Amiri, I. Azimi, M. Levorato, P. Dutt, Nikil andLiljeberg, and A. M. Rahmani. Edge-assisted control for healthcare internet-of-things: A case study on ppg-based early warning score. *ACM Transactions on Internet of Things*, 2020.

[8] I. Azimi, A. Anzanpour, A. M. Rahmani, T. Pahikkala, M. Levorato, P. Liljeberg, and N. Dutt. Hich: Hierarchical fog-assisted computing architecture for healthcare iot. *ACM Transactions on Embedded Computing Systems (TECS)*, 16(5s):174, 2017.

[9] P. Bellavista, J. Berrocal, A. Corradi, S. K. Das, L. Foschini, and A. Zanni. A survey on fog computing for the internet of things. *Pervasive and Mobile Computing*, 2018.

[10] S. K. Berkaya, A. K. Uysal, E. S. Gunal, S. Ergin, S. Gunal, and M. B. Gulmezoglu. A survey on ecg analysis. *Biomedical Signal Processing and Control*, 43:216–235, 2018.

[11] F. Bonomi et al. Fog computing and its role in the internet of things. In *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, 2012.

[12] I. Chadès, G. Chapron, M.-J. Cros, F. Garcia, and R. Sabbadin. Mdptoolbox: a multi-platform toolbox to solve stochastic dynamic programming problems. *Ecography*, 37(9):916–920, 2014.

[13] L.-H. Chang, T.-H. Lee, S.-J. Chen, and C.-Y. Liao. Energy-efficient oriented routing algorithm in wireless sensor networks. In *Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on*, pages 3813–3818. IEEE, 2013.

[14] P. H. Charlton, T. Bonnici, L. Tarassenko, D. A. Clifton, R. Beale, and P. J. Watkinson. An assessment of algorithms to estimate respiratory rate from the electrocardiogram and photoplethysmogram. *Physiological measurement*, 37(4):610, 2016.

[15] R. Cortésa, X. Bonnaireb, O. Marina, and P. Sensa. Stream processing of healthcare sensor data: Studying user traces to identify challenges from a big data perspective. *Procedia Computer Science*, 52:1004–1009, 2015.

[16] A. V. Dastjerdi and R. Buyya. Fog computing: Helping the internet of things realize its potential. *Computer*, 49(8):112–116, 2016.

[17] P. De Chazal, M. O'Dwyer, and R. B. Reilly. Automatic classification of heartbeats using ecg morphology and heartbeat interval features. *IEEE transactions on biomedical engineering*, 51(7):1196–1206, 2004.

[18] A. ECAR. Recommended practice for testing and reporting performance results of ventricular arrhythmia detection algorithms. *Association for the Advancement of Medical Instrumentation*, page 69, 1987.

[19] M. Elgendi, A. Mohamed, and R. Ward. Efficient ecg compression and qrs detection for e-health applications. *Scientific reports*, 7(1):1–16, 2017.

[20] F. Firouzi, A. M. Rahmani, K. Mankodiya, M. Badaroglu, G. V. Merrett, P. Wong, and B. Farahani. Internet-of-things and big data for smarter healthcare: from device to architecture, applications and analytics, 2018.

[21] S. G., K. P. T., and K. K. V. Classification of ecg beats using deep belief network and active learning. *Medical & Biological Engineering & Computing*, 56(10):1887–1898, Oct 2018.

[22] A. Garde, W. Karlen, J. M. Ansermino, and G. A. Dumont. Estimating respiratory and heart rates from the correntropy spectral density of the photoplethysmogram. *PloS one*, 9(1):e86427, 2014.

[23] Garmin. Vivosmart HR | Activity Tracker, Retrieved on November 2018. `https://buy.garmin.com/en-US/US/p/531166`.

[24] A. L. Goldberger, L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley. Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220, 2000.

[25] S. D. Greenwald, R. S. Patil, and R. G. Mark. *Improved detection and classification of arrhythmias in noise-corrupted electrocardiograms using contextual information*. IEEE, 1990.

[26] H.-J. Hong. From cloud computing to fog computing: Unleash the power of edge and end devices. In *2017 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, pages 331–334, 2017.

[27] M. Hooshmand, D. Zordan, T. Melodia, and M. Rossi. Surf: Subject-adaptive unsupervised ecg signal compression for wearable fitness monitors. *IEEE Access*, 5:19517–19535, 2017.

[28] S. R. Islam, D. Kwak, M. H. Kabir, M. Hossain, and K.-S. Kwak. The internet of things for health care: a comprehensive survey. *IEEE Access*, 3:678–708, 2015.

[29] J.-H. Jeon, H.-J. Byun, and J.-T. Lim. Joint contention and sleep control for lifetime maximization in wireless sensor networks. *IEEE Communications Letters*, 17(2):269–272, 2013.

[30] S. B. Kale and D. H. Gawali. Review of ecg compression techniques and implementations. In *2016 International Conference on Global Trends in Signal Processing, Information Computing and Communication (ICGTSPICC)*, pages 623–627. IEEE, 2016.

[31] W. Karlen, S. Raman, J. M. Ansermino, and G. A. Dumont. Multiparameter respiratory rate estimation from the photoplethysmogram. *IEEE Transactions on Biomedical Engineering*, 60(7):1946–1953, 2013.

[32] D. Kartik, A. Nayyar, and U. Mitra. Active hypothesis testing: Beyond chernoff-stein. *arXiv preprint arXiv:1901.06795*, 2019.

[33] N. Kaur and S. K. Sood. An energy-efficient architecture for the internet of things (iot). *IEEE Systems Journal*, 11(2):796–805, 2017.

[34] A. Khazaee and A. Ebrahimzadeh. Classification of electrocardiogram signals with support vector machines and genetic algorithms using power spectral features. *Biomedical Signal Processing and Control*, 5(4):252–263, 2010.

[35] J. Laitala, M. Jiang, E. Syrjälä, E. K. Naeini, A. Airola, A. M. Rahmani, N. D. Dutt, and P. Liljeberg. Robust ecg r-peak detection using lstm. In *Proceedings of the 35th Annual ACM Symposium on Applied Computing*, pages 1104–1111, 2020.

[36] G. Lee, R. Gommers, F. Waselewski, K. Wohlfahrt, and A. OLeary. Pywavelets: A python package for wavelet analysis. *Journal of Open Source Software*, 4(36):1237, 4 2019.

[37] S. Li, X. Li, X. Wang, and J. Liu. Sequential hypothesis test with online usage-constrained sensor selection. *IEEE Transactions on Information Theory*, 2019.

[38] S. Li and X. Wang. Fully distributed sequential hypothesis testing: Algorithms and asymptotic analyses. *IEEE Transactions on Information Theory*, 64(4):2742–2758, 2018.

[39] L.-G. Lindberg, H. Ugnell, and P. Öberg. Monitoring of respiratory and heart rates using a fibre-optic sensor. *Medical and Biological Engineering and Computing*, 30(5):533–537, 1992.

[40] Y. Liu, F. R. Yu, X. Li, H. Ji, and V. C. M. Leung. Distributed resource allocation and computation offloading in fog and cloud networks with non-orthogonal multiple access. *IEEE Transactions on Vehicular Technology*, 67(12):12137–12151, 2018.

[41] Maxim Integrated. , (accessed 2018-11-01). `https://www.maximintegrated.com/en/products/sensors/MAX30102.html`.

[42] R. Mieronkoski, I. Azimi, A. M. Rahmani, R. Aantaa, V. Terävä, P. Liljeberg, and S. Salanterä. The internet of things for basic nursing care—a scoping review. *International journal of nursing studies*, 69:78–90, 2017.

[43] MIT-BIH. MIT-BIH Arrhythmia Database, 2000.

[44] MIT-BIH. The MIT-BIH Supraventricular Arrhythmia Database, 2000.

[45] G. B. Moody and R. G. Mark. The impact of the mit-bih arrhythmia database. *IEEE Engineering in Medicine and Biology Magazine*, 20(3):45–50, 2001.

[46] E. K. Naeini, I. Azimi, A. M. Rahmani, P. Liljeberg, and N. Dutt. A real-time ppg quality assessment approach for healthcare internet-of-things. *Procedia Computer Science*, 151:551–558, 2019.

[47] E. K. Naeini, S. Shahhosseini, A. Subramanian, T. Yin, A. M. Rahmani, and N. Dutt. An edge-assisted and smart system for real-time pain monitoring. In *2019 IEEE/ACM International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE)*, pages 47–52. IEEE, 2019.

[48] M. Naghshvar and T. Javidi. Active m-ary sequential hypothesis testing. In *2010 IEEE International Symposium on Information Theory*, pages 1623–1627. IEEE, 2010.

[49] M. Naghshvar, T. Javidi, et al. Active sequential hypothesis testing. *The Annals of Statistics*, 41(6):2703–2738, 2013.

[50] M. A. Pimentel, P. H. Charlton, and D. A. Clifton. *Probabilistic estimation of respiratory rate from wearable sensors*, pages 241–262. Springer, 2015.

[51] G. Pradhan, R. Gupta, and S. Biswasz. Study and simulation of wban mac protocols for emergency data traffic in healthcare. In *2018 Fifth International Conference on Emerging Applications of Information Technology (EAIT)*, pages 1–4. IEEE, 2018.

[52] A. M. Rahmani et al. *Fog Computing in the Internet of Things - Intelligence at the Edge*. Springer, 2017.

[53] A. M. Rahmani, T. N. Gia, B. Negash, A. Anzanpour, I. Azimi, M. Jiang, and P. Liljeberg. Exploiting smart e-health gateways at the edge of healthcare internet-of-things: A fog computing approach. *Future Generation Computer Systems*, 78:641–658, 2018.

[54] R. Raju, M. Moh, and T.-S. Moh. Compression of wearable body sensor network data using improved two-threshold-two-divisor data chunking algorithms. In *2018 International Conference on High Performance Computing & Simulation (HPCS)*, pages 949–956. IEEE, 2018.

[55] S. A. Razavi, E. Bozorgzadeh, and S. S. Kia. Communication-computation co-design of decentralized task chain in cps applications. In *2019 Design, Automation Test in Europe Conference Exhibition (DATE)*, pages 1082–1087, 2019.

[56] S. A. Razavi, E. Bozorgzadeh, K. Kim, and S. Kia. Resource-aware decentralization of a ukf-based cooperative localization for networked mobile robots. In *2018 21st Euromicro Conference on Digital System Design (DSD)*, pages 296–303. IEEE, 2018.

[57] S. Sarkar and S. Misra. Theoretical modelling of fog computing: a green computing paradigm to support iot applications. *IET Networks*, 5(2):23–29, 2016.

[58] S. N. Shirazi, A. Gouglidis, A. Farshad, and D. Hutchison. The extended cloud: Review and analysis of mobile edge computing and fog from a security and resilience perspective. *IEEE Journal on Selected Areas in Communications*, 35(11):2586–2595, 2017.

[59] C. Systems. Fog computing and the internet of things: Extend the cloud to where the things are. *Available online: www.cisco.com (accessed on 21 March 2019)*, 2015.

[60] J. Takalo-Mattila, J. Kiljander, and J.-P. Soininen. Inter-Patient ECG Classification Using Deep Convolutional Neural Networks. In *2018 21st Euromicro Conference on Digital System Design (DSD)*, pages 421–425. IEEE, aug 2018.

[61] T. Tamura, Y. Maeda, M. Sekine, and M. Yoshida. Wearable photoplethysmographic sensors—past and present. *Electronics*, 3(2):282–302, 2014.

[62] D.-S. Zois, M. Levorato, Marco and U. Mitra. Active classification for pomdps: A kalman-like state estimator. *IEEE Transactions on Signal Processing*, 62(23):6209–6224, 2014.

[63] N. V. Thakor, Y.-S. Zhu, and K.-Y. Pan. Ventricular tachycardia and fibrillation detection by a sequential hypothesis testing algorithm. *IEEE Transactions on Biomedical Engineering*, 37(9):837–843, 1990.

[64] C. Tudor-Locke, S. M. Camhi, C. Leonardi, W. D. Johnson, P. T. Katzmarzyk, C. P. Earnest, and T. S. Church. Patterns of adult stepping cadence in the 2005–2006 nhanes. *Preventive medicine*, 53(3):178–181, 2011.

[65] C. Tudor-Locke et al. Pedometer-determined step count guidelines for classifying walking intensity in a young ostensibly healthy population. *Can J Appl Physiol.*, 30(6):666–76, 2005.

[66] C. Tunc and N. Akar. Markov fluid queue model of an energy harvesting iot device with adaptive sensing. *Performance Evaluation*, 111:1–16, 2017.

[67] C. Valens. A really friendly guide to wavelets. *ed. Clemens Valens*, 1999.

[68] O. Vermesan, P. Friess, P. Guillemin, S. Gusmeroli, H. Sundmaeker, A. Bassi, I. S. Jubert, M. Mazura, M. Harrison, M. Eisenhauer, et al. Internet of things strategic research roadmap. *Internet of Things-Global Technological and Societal Trends*, 1(2011):9–52, 2011.

[69] A. Wald. Sequential tests of statistical hypotheses. *The annals of mathematical statistics*, 16(2):117–186, 1945.

[70] K. Wang, Y. Wang, Y. Sun, S. Guo, and J. Wu. Green industrial internet of things architecture: An energy-efficient perspective. *IEEE Communications Magazine*, 54(12):48–54, 2016.

[71] Y. Wang, H. Chen, X. Wu, and L. Shu. An energy-efficient sdn based sleep scheduling algorithm for wsns. *Journal of Network and Computer Applications*, 59:39–45, 2016.

[72] W. Wei, K. Suh, B. Wang, Y. Gu, J. Kurose, and D. Towsley. Passive online rogue access point detection using sequential hypothesis testing with tcp ack-pairs. In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, pages 365–378. ACM, 2007.

[73] S. S. Xu, M. Mak, and C. Cheung. Towards end-to-end ecg classification with raw signal extraction and deep neural networks. *IEEE Journal of Biomedical and Health Informatics*, 23(4):1574–1584, July 2019.

[74] D. Ye et al. A self-adaptive sleep/wake-up scheduling approach for wireless sensor networks. *IEEE transactions on cybernetics*, 48(3):979–992, 2018.

[75] C. Zhu, V. C. Leung, L. T. Yang, and L. Shu. Collaborative location-based sleep scheduling for wireless sensor networks integratedwith mobile cloud computing. *IEEE Transactions on Computers*, 64(7):1844–1856, 2015.

[76] D.-S. Zois, M. Levorato, and U. Mitra. A pomdp framework for heterogeneous sensor selection in wireless body area networks. In *2012 Proceedings IEEE INFOCOM*, pages 2611–2615. IEEE, 2012.

[77] D.-S. Zois, M. Levorato, and U. Mitra. Energy-efficient, heterogeneous sensor selection for physical activity detection in wireless body area networks. *IEEE Transactions on signal processing*, 61(7):1581–1594, 2013.