University of California
Los Angeles

Non-Invasive Evaluation of Diet: Devices and Algorithms

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Computer Science

by

Haik Kalantarian

2016

Abstract of the Dissertation

Non-Invasive Evaluation of Diet: Devices and Algorithms

by

Haik Kalantarian

Doctor of Philosophy in Computer Science

University of California, Los Angeles, 2016

Professor Majid Sarrafzadeh, Chair

In 2008, medical costs associated with obesity were estimated to be over $147 billion [cdd14], and over one-third of adults in the United States are considered obese. The average BMI (body mass index) has consistently increased over the last two decades, which has been shown to be a contributor to risk of stroke, diabetes, certain cancers, heart disease, and other conditions [cdd14]. Though many activity-monitoring systems have been proposed [FMS98, FLK11, PPB12], little research has been conducted on quantifying the volume of food consumption, which has been shown to correlate with weight gain [SS85]. Though countless manual data collections have been proposed such as food records and 24-hour recall, these approaches suffer from poor accuracy, high user burden, and low compliance. Wireless health-monitoring technologies have the potential to promote healthy lifestyle behavior and address the ultimate goal of enabling better lifestyle choices. This thesis explores the application of hardware, software, and algorithms to the domain of food intake monitoring and medication adherence. Moreover, we propose several new methods to improve the processing and segmentation of time-series data based on our nutrition monitoring dataset.

The dissertation of Haik Kalantarian is approved.

Janet Mentes

Mario Gerla

Douglas Stott Parker

Majid Sarrafzadeh, Committee Chair

University of California, Los Angeles

2016

Table of Contents

1 A Survey of Nutrition Monitoring Technologies 1

1.1 Introduction 1

1.2 Manual Record Keeping 3

1.2.1 24-Hour Recall 3

1.2.2 Food Records 3

1.2.3 Food-Frequency Questionnaire 4

1.2.4 Summary 4

1.3 Acoustic Methods 5

1.3.1 Headset Microphone 6

1.3.2 Integrated Ear-Canal Microphone 6

1.3.3 Throat Microphone 7

1.3.4 Summary 7

1.4 Gesture Recognition 8

1.4.1 Custom Sensor Nodes 8

1.4.2 Smartwatch Platform 9

1.4.3 Smart Utensils 9

1.4.4 Summary 9

1.5 Other Notable Approaches 10

1.5.1 Smart Tablecloth 10

1.5.2 E-Button 11

1.6 Chewing and Swallowing Motion 12

1.6.1 WearSens 13

1.6.2 Jaw-Mounted Strain Sensor 14

1.6.3 Summary 14

1.7 A Survey of Individual Preferences: Methodology 14
1.8 Survey Results 16
1.8.1 Demographic Information 16
1.8.2 Detailed Results 16
1.8.3 Analysis 17
1.9 Conclusion 19

2 Necklace 20
2.1 Introduction 20
2.2 Hardware Architecture 21
2.2.1 Audio-based Approach 21
2.2.2 Piezo-Based Approach 22
2.3 Algorithms 23
2.3.1 Audio Feature Extraction and Classification 23
2.3.2 Piezoelectric Feature Extraction and Classification 25
2.4 Experimental Procedure 27
2.4.1 Piezoelectric Sensor Data Collection 27
2.4.2 Audio Data Collection 27
2.5 Evaluation 29
2.5.1 Audio-Based Classification Results 30
2.5.2 Piezoelectric-Based Classification Results 31
2.6 Energy Modeling Methods 32
2.7 Power Evaluation 33
2.7.1 Sample Rate, Window Size, and Power 34
2.7.2 Power vs. Connection Interval 35
2.7.3 Energy Evaluation of Fourier-Based Algorithms 35
2.8 Conclusion 37

3 Smartwatch **40**

3.1 Introduction 40

3.2 System Architecture 43

3.3 Algorithm Design 44

3.3.1 Frequency-Domain Evaluation: Liquids 44

3.3.2 Generalizable Feature Extraction 46

3.4 Experimental Procedure 48

3.4.1 Data Collection for Recognition 48

3.4.2 Smartwatch Feedback: A Survey 49

3.5 Results and Discussion 50

3.5.1 Audio Classification 50

3.5.2 Feature Extraction 51

3.5.3 Battery Life Implications 53

3.5.4 Smartwatch Feedback: A Survey 56

3.6 Conclusion 58

4 Gesture Recognition **59**

4.1 Introduction 59

4.2 Related Work 62

4.2.1 Mobile-Phone Solutions 62

4.2.2 Hardware Approaches 62

4.3 System Architecture 63

4.3.1 Hardware Description 63

4.3.2 Android Application 64

4.4 Algorithm Design 64

4.4.1 Bottle Opening: Data Transformation 65

4.4.2 Bottle Opening: Detection 68

4.4.3 Medicine Removal: Data Transformation 69
4.4.4 Removing the Medicine: Detection 71
4.5 Secondary Motions and Other Future Works 71
4.6 Experimental Procedure 72
4.6.1 Gesture Recognition 73
4.6.2 Online Survey of Habits 73
4.6.3 Observational Survey 73
4.7 Results 74
4.7.1 Online Survey 74
4.7.2 Observational Survey 75
4.7.3 Motion Classification Results 76
4.8 Conclusion 77

5 Classification and Segmentation for Wearable Activity-Monitoring Applications 79

5.1 Introduction 79
5.1.1 Overview of Real-Time Wearable Sensor Systems 80
5.1.2 Classifiers 81
5.1.3 Time-Series Segmentation 82
5.1.4 A Novel Segmentation and Classifical Scheme 83
5.2 Related Work 84
5.3 Algorithms 85
5.3.1 Phase I - Window Selection 85
5.3.2 Phase II - Advanced Classification 86
5.3.3 Cost Analysis 87
5.3.4 Comparison to Baseline 90
5.3.5 Feature Selection 90

5.4 Experimental Methodology 91
5.4.1 Data Collection 91
5.4.2 Feature Extraction 92
5.4.3 Selected Features 92
5.5 Results and Discussion 92
5.5.1 Classifier performance vs. Input Size 93
5.5.2 Classifier Accuracy 94
5.5.3 Speed vs. Unoptimized Baseline 95
5.5.4 Boundary Selection 96
5.6 Conclusion 97

6 Probabilistic Time-Series Segmentation for Real-Time Classification Applications 103

6.1 Introduction 103
6.2 Motivation 105
6.3 System Challenges and Considerations 106
6.4 Related Work 108
6.5 Posterior Probabilities for SVM 110
6.5.1 An Introduction to Support Vector Machines 110
6.5.2 Motivation 110
6.5.3 Probability Model 112
6.6 Window Size Selection 113
6.6.1 Naive Window Selection 113
6.6.2 Exhaustive Search 114
6.6.3 Optimized Search 115
6.7 Experimental Setup 119
6.7.1 Feature Extraction 120

6.7.2 Model Development 121
6.8 Results 121
6.8.1 Classification Results 121
6.8.2 Probability Model Validation 122
6.9 Future Works 123
6.10 Conclusion 124

7 Dynamic Computation Offloading for Low Power Wearable Health Monitoring Systems 125
7.1 Introduction 125
7.1.1 Introduction to Wearables 125
7.1.2 Energy Usage 127
7.1.3 Computation Offloading 128
7.2 Classification Flow 129
7.2.1 Acquisition 129
7.2.2 Segmentation 130
7.2.3 Extraction 130
7.2.4 Classification 132
7.3 Energy Modeling 132
7.3.1 A General Model 133
7.3.2 Connection Interval 134
7.4 Algorithm 135
7.4.1 Classifier Accuracy Adjustment 135
7.4.2 Dynamic Offloading 136
7.5 Experimental Methodology 138
7.5.1 Dataset 138
7.5.2 Feature Extraction 139

7.5.3 Bluetooth Modeling 140
7.5.4 Computation Modeling 140
7.6 Results and Discussion 140
7.6.1 Classifier Performance 140
7.6.2 Comparison of Local and Remote Processing 142
7.6.3 Variations in Sample Rate 142
7.6.4 Variations in Sample Rate 143
7.7 Conclusion 143

8 Concluding Remarks 147

References 148

List of Figures

1.1 Overview of techniques to monitor eating behavior. 2

1.2 BodyScope: detecting swallow sounds using a headset. 5

1.3 Sensor nodes used to identify gestures associated with eating behavior. . 8

1.4 eButton: Using a camera to take snapshots of food and estimate composition, mass, and calories. 11

1.5 Wearsens necklace 13

1.6 The age ranges of survey respondents. 16

1.7 Detailed survey results. 18

2.1 Systems architecture of the piezoelectric sensor-based necklace. 22

2.2 Signal processing flow used by the WearSens necklace to identify swallows. 25

2.3 Comparison of the accuracies of various classifiers for the throat microphone. 29

2.4 Effect of window size on classification accuracy (audio). 34

2.5 Percentage of time the MSP430 spent in various modes, as a result of 16-point FFT operations performed at different rates. 35

2.6 Power dissipation of the microcontroller when 16-point FFT operations are performed at various rates. 36

2.7 Major features for audio-based classification using the openSMILE toolkit, for audio-based classification. 38

2.8 State transitions of the MSP430 microcontroller, which occur on an interrupt callback. in LPM3, all clocks besides ACLK (necessary for the timer) are disabled. 38

2.9 Relationship between window size, sample rate, and mean current drawn on MSP430 mcu with on-board swallow detection. 38

2.10 Relationship between window size, sample rate, and mean current drawn on MSP430 mcu when swallow detection is offloaded to the mobile phone. 39

3.1 High level architecture of the smartwatch system. 42
3.2 Samsung Galaxy Gear phone with Android 4.2.1. 43
3.3 Spectrogram for five swallows using a Hamming window. 44
3.4 Frequency distribution of a water swallow vs. silence (noise). 45
3.5 Post-processing of the audio signal corresponding with water can dramatically improve signal-to-noise ratio. 45
3.6 Precision, recall, and F-measure are common measures of classification accuracy. 51
3.7 A graph of battery life for three different use cases is shown above. 54
3.8 Partial survey results are shown above. 56
3.9 Survey results for the smartwatch-based scheme. 57

4.1 Illustration of methods in which a SmartWatch or similar wrist-worn device can be employed to detect medication intake. 60
4.2 (A) The wrist motion necessary to twist the bottle cap open is detected using a tri-axial accelerometer. (B) The act of turning the palm upward to pour medicine from the bottle is detected using a gyroscope. 62
4.3 Accelerometer data from opening the pill bottle nine times. 65
4.4 Data converted to a sliding window representation. 66
4.5 Phase (3) waveforms for processing smartwatch signals. 67
4.6 Phase (4) waveforms for processing smartwatch signals. 67
4.7 Output pulses from Phase (4). 68
4.8 An analysis of the clustering patterns for different features. 69
4.9 Different motions observed during data collection and reported by users in the survey. 72
4.10 Partial online survey results are shown above. 78

5.1 System architecture of the proposed two-stage segmentation and classification scheme. 80

5.2 Potential problems when attempting to identify an action from a large discrete signal. 81

5.3 First stage of the algorithm, which adjusts the window boundaries and assigns labels of *Relevant* or *Not Relevant* to each window. 85

5.4 This figure shows how an individual window is resized several times. After each resize, it is input into a classifier to determine its relevance. 86

5.5 Phase II: discarding irrelevant segments of the window are discarded. The remaining windows are binned according to their length, and a more robust (and expensive) classifier produces the final class label. 88

5.6 SMO [Pla99a], Bayesian Networks [FGG97], and LogicalRegression [HL04] classifiers are evaluated for speed as a function of feature set size. 93

5.7 The ability of three classifiers to distinguish noise from relevant data (Phase I) as a function of number of features. 94

5.8 The ability of three classifiers to identify from three different classes (Phase II) as a function of number of features. 94

5.9 Comparison between the baseline and proposed scheme as we sweep across β values and vary the α parameter. 96

5.10 Classification accuracy vs. α parameter. 97

6.1 Typical system flow for classifying sensor data. 104

6.2 Incorrect segmentation scenarios. 107

6.3 Window overlapping to avoid boundary issues. 108

6.4 SVM hyperplane-based separation of instances. 109

6.5 Proposed scheme in which instances close to the hyperplane are not assigned class labels. 110

6.6 Relabeling instances original in the ambiguous region through parameter adjustment. 119

6.7 Percentage of correctly classified instances using a baseline with no optimizations, an exhaustive search, and optimized search. 122

6.8 Validation of the trained probability model. 123

7.1 Wearable devices must evaluate the performance penalty of RF transmission when deciding to perform data computation locally vs. remotely. . . 126

7.2 System flow of proposed computation offloading scheme. 126

7.3 The overall classification model. 131

7.4 Tradeoff between local and remote processing. 144

7.5 Variations in accuracy as a function of classifier, and feature set size. . . 144

7.6 Variations in runtime speed as a function of classifier, and feature set size. 144

7.7 Ratio of power in the remote and local processing schemes at various values of α. 145

7.8 Ratio of power in the remote and local processing schemes at various values of α. 145

7.9 Nonlinearity between sample rate and Bluetooth transmission power. . . 145

7.10 Relationship between local computation power and sample rate for three different accuracies. 146

7.11 Comparison between local, remote, and adaptive schemes in a system which alternates between low and high accuracy requirements. 146

List of Tables

2.1 Partial List of openSMILE Speech Features. 24

2.2 Partial List of openSMILE Statistical Features. 24

2.3 Selected features extracted from the audio spectrogram. 25

2.4 Feature Table for Piezoelectric Sensor Feature Extraction 26

2.5 Audio: Confusion Matrix (Random Forest) using a 1-second window . . . 29

2.6 Audio: Confusion Matrix (Random Forest) using a 1-second window. . . 29

2.7 Piezoelectric Sensor: Confusion Matrix (Random Forest). 30

2.8 Piezoelectric Sensor: Confusion Matrix (Random Forest). 30

2.9 A List of System Components Used in Evaluation 32

2.10 Average Transmit Power for a Fixed Bandwidth (20 Hz) at 3.7V 39

3.1 Partial List of openSMILE Speech Features from [smi] 52

3.2 Partial List of openSMILE Statistical Features from [smi] 52

3.3 Audio: Confusion Matrix (Random Forest) 55

3.4 Partial List of Selected Features 55

4.1 Observational Survey of medication ingestion habits 75

4.2 Confusion Matrix using Accelerometer Data 76

4.3 Confusion Matrix using Gyroscope Data 77

5.1 Definition of Terms 101

5.2 Top 20 Features Selected by the Feature Reduction Algorithm 102

6.1 Partial List of Selected Features 120

7.1 Definition of Terms 132

7.2 Top 10 Selected Features 139

7.3 Best Classifier and Feature Set Combinations For a Given Accuracy . . . 141

Acknowledgments

I would like to thank my advisor and committee members for their valuable guidance. Specifically, I would like to acknowledge the following contributions:

Dr. Majid Sarrafzadeh for providing research direction for all chapters. Nabil Alshurafa for his contributions to the literature survey and experiments in Chapters 1 and 2, and general development of the Wearsens classification algorithms. Mohammad Pourhomayoun, for assistance with spectrogram analysis algorithms in Chapter 2. Costas Sideris, for feedback and general direction in Chapters 5, 6, and 7. Bobak Mortazavi, for editing and feedback in Chapter 2. Tuan Le for editing, discussion, and assistance with figures in Chapters 2, 5, 6, and 7. Krithika Chandramouli, Shruti Sarin, Mahir Eusufzai, Tanay Agarwal, for their assistance with data collection and app design (Chapter 2).

Portions of Chapter 2 was published in Elsevier Obesity Medicine under the title "A comparison of piezoelectric-based inertial sensing and audio-based detection of swallows". Chapter 3 was published in Elsevier Computers in Biology and Medicine under the title "Audio-Based Detection and Evaluation of Eating Behavior using the Smartwatch Platform." Chapter 4 was published in Elsevier Computers in Biology and Medicine under the title "Detection of Gestures Associated with Medication Adherence using Smartwatch-based Inertial Sensor."

Vita

2008–2011 BS (Electrical Engineering), University of Washington.

Publications

Detection of Gestures Associated with Medication Adherence using Smartwatch-based Inertial Sensors. H. Kalantarian et al. IEEE Sensors Journal. Volume 16, Issue 4, November 2015. Pages 1054-1061.

Characterization of Eating Habits using a Piezoelectric-Based Necklace. H. Kalantarian et al. Elsevier Computers in Biology and Medicine. March 1, 2015. Volume 58, Pages 46 - 55.

Audio-Based Detection and Evaluation of Eating Behavior using the Smartwatch Platform. H. Kalantarian et al. Elsevier Computers in Biology and Medicine. July 24, 2015. Volume 65, Pages 1 - 9.

A Wearable Piezoelectric Sensor for Recognition of Nutrition-Intake using Spectrogram Analysis. N. Alshurafa et al. IEEE Sensors Journal. February 11,2015. Volume 15, Issue 7, Pages 3909 - 3916.

A comparison of piezoelectric-based inertial sensing and audio-based detection of swallows. H. Kalantarian et al. Elsevier Obesity Medicine. March 2015. Volume 1, Pages 6-14.

CHAPTER 1

A Survey of Nutrition Monitoring Technologies

In this chapter, we present a survey of different techniques for evaluating eating habits, with applications to weight loss and preventative healthcare. We emphasize recent sensor-based approaches to monitoring diet using techniques such as audio signal processing, inertial sensing, image processing, and gesture recognition, while describing the major advantages and disadvantages of each. We primarily emphasize non-invasive technologies that could be developed into real-time wearable devices, rather than techniques whose use is limited to laboratory settings. We then present the results of an online survey, in which respondents rate and describe their impressions of various approaches.

1.1 Introduction

Billions of dollars have been invested worldwide to improve medical treatments, develop pharmaceutical drugs, and mitigate the effects of various diseases. By comparison, relatively little emphasis is placed on *preventative healthcare*. By ensuring that individuals exercise regularly and eat a balanced diet, we can reduce their risk of various health risks such as cancer, diabetes, and heart disease. Though public health measures have been taken to encourage individuals to lead healthy lifestyles, quantifying diet and exercise has generally remained an unaddressed challenge. Only very recently has accurate evaluation of physical activity become a reality, as numerous non-invasive activity monitoring systems have entering the market such as FitBit, MisFit, and Jawbone. However, resources for evaluating diet are trivial by comparison, as most techniques outside of academic literature focus on manual record keeping. These methods are notoriously inaccurate, inefficient, and present high levels of burden to individuals who simply fail to adhere to the record keeping procedure for extended periods of time.

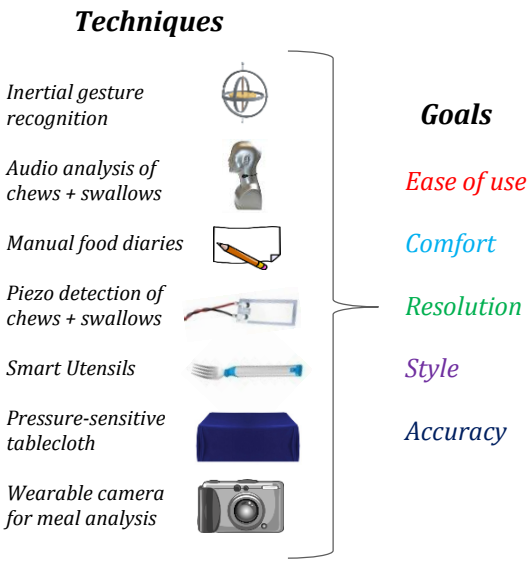


Figure 1.1: Overview of techniques to monitor eating behavior.

In this chapter, we evaluate the potential of various approaches to dietary monitoring, with respect to convenience, accuracy, and applicability to real-world environments. We emphasize the application of technology and sensor-based solutions to the domain of health monitoring, and evaluate various form factors including necklaces, watches, and cameras, to provide a comprehensive survey of the prior art in the field. Though it is not possible to cover every approach, we review most major techniques with an emphasis on works that can be applied to real-world environments rather than constrained to laboratory settings. Several of these approaches are illustrated briefly in Figure 1.1. Furthermore, we present the results of an online survey which evaluates individual perceptions of some of the most promising techniques for nutrition monitoring in non-clinical settings, including histograms of interest in various devices, and comments from individuals with respect to the proposed ideas. Lastly, we summarize the different approaches and directly compare them on various criteria.

The structure of this chapter is as follows. In Section II, we describe manual record-keeping techniques for evaluation of eating behavior. In Section III, we provide an overview of several promising audio-based approaches in which microphones are used to capture eating sounds. In Section IV, we describe gesture-recognition approaches including sensor-nodes and smartwatches. Section V presents some alternative approaches using custom devices. In Section VI, we describe two approaches that monitor chew-

ing and swallowing behavior using piezoelectric sensors. In Section VI, we describe the methodology of our online survey, presenting results in Section VII followed by concluding remarks in Section VIII.

1.2 Manual Record Keeping

Wearable nutrition monitoring devices are becoming a subject of interest in the research community. However, the most pervasive techniques today are still manual record-keeping approaches. While these techniques are simple and affordable, they are associated with user burden and poor accuracy. We describe the major techniques below.

1.2.1 24-Hour Recall

One of the most simple and yet pervasive methods of monitoring dietary intake is the multi-pass 24-hour dietary recall method, which is based on the data patients provide at the end of a randomly selected day. Each individual gives an oral or written report including the amount and type of food they have eaten during the day, which is used to calculate total food intake. This approach measures food intake in a reasonably quantitative manner but with significant error because people don't recall the exact amount of food they have eaten [HOK88]. Experimental data shows that food intake is usually reported with error and measurement variance also depends on the patient's experience with the system [BWS05].

1.2.2 Food Records

Food records generally are not impacted by the accuracy of a subject's memory; they typically require individual to make note of their eating habits during or immediately after a meal. However, there are several problems with this approach. In cases where assessment of a typical diet is the goal, this technique is not feasible because it has been found that the necessity of completing a food record affects dietary choices. Other concerns include patient compliance, and the difficulty that untrained individuals face when accurately assessing portion size. For example, caloric density may vary based

on cooking method and other factors that are not necessarily visually apparent [CB08]. Moreover, recording each meal manually can be tedious, and many individuals will be unwilling to complete food records for extended periods of time.

1.2.3 Food-Frequency Questionnaire

A third method for manually assessing dietary intake is to use a food frequency questionnaire (FFQ), in which individuals specify their rate of consumption for various food items over the long-term. Nutritional intake can subsequently be assessed by summing various food types provided within the list [CB08]. This technique is inexpensive to administer and insensitive to recent changes in diet. Furthermore, it is less time-consuming than food records because it is not intended to be completed on a daily basis. However, FFQs are typically inaccurate in comparison with other techniques. This is often a result of several factors including incomplete lists of food, poor user compliance, errors in frequency, and errors in serving size [PMH11].

1.2.4 Summary

Despite the pervasiveness of the previously outlined techniques, as well as the relative simplicity, affordability, and ease of use, manual methods for assessing dietary intake suffer from several drawbacks. These include extensive user burden, poor compliance, low precision, and a lack of real-time user guidance [SSL09]. Below, we summarize the advantages and disadvantages of these techniques.

Advantages:

- **Simplicity:** Users do not require training on the operation of custom hardware.
- **Initial Cost:** Manual record keeping methods do not require the purchase of any expensive devices.

Disadvantages:

- **Burden:** Manual record-keeping can be tedious, and long-term compliance rates are low.

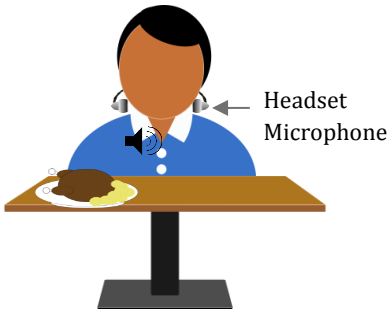


Figure 1.2: BodyScope: detecting swallow sounds using a headset.

- **Accuracy:** Individuals often do not accurately remember what they have eaten. Moreover, they may deliberately misreport their eating habits.

1.3 Acoustic Methods

Audio-based methods are perhaps the most popular approaches for monitoring eating habits. They general rely on the placement of microphones near the throat, which can record chewing and swallowing sounds. Classifiers such as Support Vector Machines are then used to distinguish between eating related activities such as chewing, drinking, and swallowing food. Generally speaking, there are two primary challenges associated with this approach. First, relatively robust classification techniques are necessary for distinguishing between eating sounds and unrelated background sounds. Second, most swallow sounds are contained within 400 Hz and 1000 Hz frequency bands [TK12]. This frequency range was shown by Dr. Taniwaki et al. to be essential for distinguishing between liquids, semiliquids, and solid foods citetaniwaki2012fast. The work by Dr. Santamato et al. in [SPS09] analyzed the acoustic properties of swallow sounds for semi-solids, semi-liquids, solids, and liquids, and found that the maximum frequencies were associated with liquids, which ranged from 2281.3 Hz to 4244.0 Hz. Thus, effective characterization of swallow sounds may require sample rates of at least 8.4 kHz based on Shannon-Nyquist sampling theory. Very high speed data acquisition and transmission is associated with significant power overhead, as shown by our prior work in [KAP15]. In this section, we describe several of the most promising techniques.

1.3.1 Headset Microphone

In 2011, Professor Koji Yatani described a system called BodyScope: a wearable solution for activity recognition using audio signals [YT12]. This device consists of a Bluetooth headset with an embedded microphone for recording throat noises, and the chestpiece of a stethoscope for amplifying body signals to improve classification accuracy. The device is worn such that it rests on the lower neck, with an earpiece pointed inward towards the skin. A simplified visual of the scheme is shown in Figure 1.2. The microphone was attached to a stethoscope chestpiece to filter external sounds that do not originate from the neck. The system is capable of analyzing eating, drinking, speaking, coughing, and eight other activities in a laboratory environment with 78.5% classification accuracy. In a real-time evaluation, a subset of four activities were recognized with 71.5% accuracy. Classification was achieved using Support Vector Machines (SVM), using a Radial Basis Function (RBF) kernel. Features were extracted from audio spectrograms, which consist of a series of overlapping short-time Fourier transforms. These features included Zero Crossing Rate, Total Spectrum Power, Subband Power, Brightness, Spectral Rolloff, Spectral Flux, and Mel-Frequency Cepstral Coefficients (MFCCs). Results for leave-one-sample-out cross validation are promising, given the large number of classes of data. However, Leave-one-subject-out cross validations significantly reduces classification accuracy, suggesting then need for individual calibration.

1.3.2 Integrated Ear-Canal Microphone

Several works have analyzed eating sounds using in-ear microphones [NK08][P12]. In 2012, Dr. Pabler et al. presented an audio-based approach for audio signals analysis in order to detect swallowing sounds [P12]. The hardware device presents a hearing-aid package with two integrated microphones acquiring data at approximately 11 kHz, along with associated amplification and filtering. The operation is not purely real-time, but the wearable device was used for data collection and a MATLAB model was used to perform the classification among several different food types. Classification results were quite promising; 8 classes of food were detected with an overall accuracy of 79% using Hidden Markov Models, which were used to analyze both chewing and swallowing sounds. A

significant novelty of this system was their use of a reference microphone placed outside of the ear, in addition to the in-ear microphone. The ratio of signal energy from both microphones is used to differentiate between external environmental sounds and those related to eating.

1.3.3 Throat Microphone

In 2010, Professor Sazonov et al. proposed a technique for predicting food intake based on audio signals acquired from a throat microphone with a frequency range of 20 Hz - 8000 Hz [SMS10]. The dataset used in experimental evaluation was substantial, containing a total of 9966 swallows from 20 subjects. The IASUS throat microphone employed in the study was placed over the laryngopharynx, in order to minimize the distance between the microphone and the source of the swallow sound compared to microphones placed in the ear. Using techniques based on the Mel Scale Fourier Spectrum, Wavelet-Packet Decomposition, and Support Vector Machines, swallow events were identified with 84.7% weighted accuracy despite the presence of various artifacts including respiration, speech, and head movements.

1.3.4 Summary

Advantages:

- **Scope:** Audio-based techniques are often able to identify specific foods, rather than eating events, with relatively high accuracy.
- **Versatility:** Many activities can be detected by analyzing audio signals, which renders this system useful for a variety of applications such as exercise monitoring.

Disadvantages:

- **Comfort:** Many individuals would object to wearing a headset or custom earpiece throughout the day.
- **Battery Life:** Higher sample rates are required for audio signal processing, compared to inertial sensing.

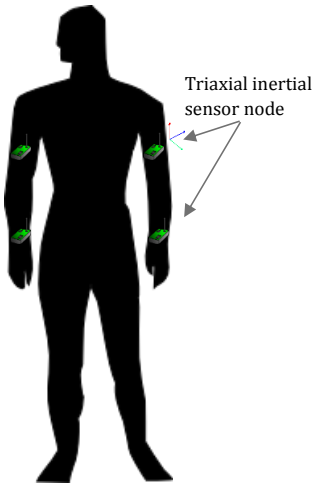


Figure 1.3: Sensor nodes used to identify gestures associated with eating behavior.

1.4 Gesture Recognition

Individuals generally make distinct and recognizable gestures with their arms as they eat. Examples include: picking up a sandwich, raising a glass of water to their mouth, scooping up food with a fork, or cutting a steak with a knife. Hardware devices mounted on the arm or wrist with embedded inertial sensors such as accelerometers and gyroscopes can recognize these gestures and infer eating activity. Unlike audio signals that must be acquired using sample rates of up to 8 kHz, activity-recognition techniques use much lower sample rates such as 100 Hz or below [AT08]. The power savings of this approach can be significant, as transparency in wearables necessitates long-intervals between recharges, as well as miniature designs with small batteries.

1.4.1 Custom Sensor Nodes

The work by Professor Amft et al. in [AT08] explored different methods for monitoring food intake, among which was gesture recognition. In their experiments, they used sensor modules including tri-axial gyroscopes, accelerometers, and compass-sensors to detect four food-related gestures: cutting lasagna with a fork and knife, drinking from a glass, eating soup with a spoon, and eating sliced bread with one hand. The sensors were attached on a jacket in the lower and upper arm, as shown in Figure 1.3. In order to present a realistic experimental procedure, subjects periodically performed other motions such as reading a newspaper and making telephone calls. Most activities, with the exception

of eating bread, were detected with high accuracy: 94% classification accuracy was obtained based on 1020 recorded intake gestures. Despite the high results, the multitude of sensors may be somewhat unwieldy for real-world situations; more practical realizations of these schemes would probably have sensors in a single location. An example of such an approach was presented by Dong et al. in [DHM09], in which the authors present a bite-counter device shown to have a sensitivity of 91% by analyzing the rolling movement of the wrist to detect biting behavior.

1.4.2 Smartwatch Platform

In [KS15b], Kalantarian et al. described a technique in which the smartwatch platform is used for detection of medication intake, using the tri-axial accelerometers and gyroscopes embedded in the Samsung Galaxy Gear device. However, the proposed technique addressed the issue of gesture recognition applied to the domain of medication adherence. Nevertheless, it likely that many activity monitoring and gesture-recognition schemes will use smartwatches in the coming years, rather than sensor nodes.

1.4.3 Smart Utensils

A third inertial-sensing approach for nutrition monitoring is the HapiFork [Lep15]: a commercial product that incorporates inertial sensors into a "smart-fork" package. The device can detect the movement of the hand to the mouth, and vibrates after detecting that eating pace is too fast. Eating at the wrong pace has been associated with weight gain, digestive problems, and gastric reflux. However, it is clear that the scope of this technique is limited, because appropriate eating pace is dependent on the texture of the food. Moreover, many foods are not eaten with a fork such as a sandwich or chips. However, clearly the device is simple, unobtrusive, and is reported to have a battery life of up to two weeks.

1.4.4 Summary

Advantages:

- **Comfort:** A smart utensil or lightweight sensor mounted on the arm may be sufficient for detection of many eating gestures, which is quite practical for day-to-day use. This is particularly true of algorithms that operate on a smartwatch device.

- **Battery Life:** Activity-recognition techniques do not require the high sample rates that audio-based solutions require; they can potentially be powered for days using a small coincell battery.

Disadvantages:

- **Scope:** Gesture recognition can provide only high-level information about eating habits, with respect to general meal consumption and timing. Furthermore, it is unclear if snacking with one hand, and eating "on the go" can be accurate detected with a single sensor node.

1.5 Other Notable Approaches

1.5.1 Smart Tablecloth

A "smart tablecloth" was presented in 2015 by Bo Zhou et al. in [ZCS15]. The system detects eating behavior on solid surfaces (such as tables), based on changes in the pressure distribution of these tables during the eating process. The tablecloth was a matrix of pressure sensors based on a carbon polymer sheet, which changes its electrical resistance in response to electrical force. At the corners of the table cloth, force-sensitive resistors (FSRs) are installed with the primary purpose of determining weight, rather than spatial density. Features extracted from the FSRs, as well as the pressure-sensitive tablecloth, are analyzed using classifiers such as decision trees to distinguish between various eating-related actions such as stirring, scooping, and cutting.

Based on the ratio of different actions performed, the authors were able to distinguish between four different meal types with high accuracy. Furthermore, changes in the average pressure values from the data stream were associated with a decrease in the remaining amount of food on the table, which was used to estimate food weight with an error of

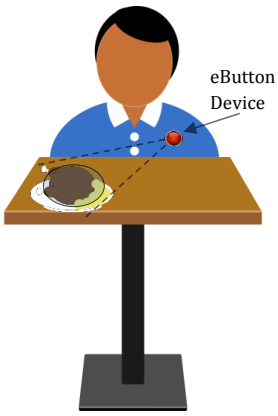


Figure 1.4: eButton: Using a camera to take snapshots of food and estimate composition, mass, and calories.

approximately 16.62%.

Advantages:

- **Convenience:** The proposed technique can estimate food composition and weight without requiring a cumbersome wearable device.

Disadvantages:

- **Scope:** Most individuals do not eat at the same surface throughout the day. Moreover, intermittent snacking and hand-held foods may not be detected using the proposed technique as they are often not consumed while seated at a table. Therefore, only a limited subset of eating activities can be detected using this approach.

1.5.2 E-Button

The E-Button, shown in Figure 1.4, was presented in 2014 by Professor Mingui Sun at the University of Pittsburgh [SBM14]. In this work, Sun et al. propose a chest-mounted button with an embedded camera that among other applications, can be applied to the domain of dietary monitoring. The button is attached to a shirt using a pin or pair of disk magnets, and contains an ARM Cortex processor, two wide-angle cameras, a UV sensor for distinguishing between indoor and outdoor environments, inertial sensors, proximity sensors, a barometer, and a GPS. The acquired data is transmitted to a smartphone using Bluetooth or WiFi.

The E-Button operates by taking photos at a preset rate, thereby recording the entire eating process. Using image processing techniques, the utensils (such as a plate or bowl) are detected. Subsequently, the food items are identified based on color, texture, and other heuristics. Using this information and additional DSP techniques, volume figures can be calculated for each food, which are converted to a Calorie count using a public domain database that equates volume and food type to Calories. Evaluation of 100 foods was conducted, and the error was approximately 30% for 85% of the foods, which were regularly shaped. However, irregularly shaped food was not detected with high accuracy.

Advantages:

- **Comfort:** The E-Button is a small pin worn on a shirt, whose dimensions are limited by the battery size.
- **Scope:** The E-Button can identify specific foods, rather than report meal events.
- **Versatility:** The E-Button can do much more than monitor meals. The device can assist the visually impaired, monitor adults with dementia, and study sedentary behavior using a multitude of sensors.

Disadvantages:

- **Battery Life:** The battery life is not described in the work, though it is quite likely to be poor given the high rate of data capture and frequent wireless transmission. Constant recharging may be an impediment to the user.
- **Accuracy:** 30% error for the majority of foods is not particularly impressive, as the standard deviation of an individual's daily caloric intake is generally not that high.

1.6 Chewing and Swallowing Motion

Several works attempt to characterize eating habits based on the motion of the skin during chewing and swallowing events, which we describe in this section. The primary challenge in the implementation of these techniques is comfort, as an inertial sensor (usually a piezoelectric sensor) is required to be contact with the skin during the meal.

Figure 1.5: Wearsens necklace

1.6.1 WearSens

The WearSens necklace, shown in Figure 1.5, was first presented by Kalantarian et al. in [KAS14a], and later by Alshurafa et al. in [AKP15]. The device consists of a pendant-style necklace, with the pendant resting in the lower part of the neck. While an individual eats various foods, chews and swallows produce vibrations in the skin of the lower neck. Different foods can produce different vibrations, based on the unique properties of each food. A piezoelectric sensor, mounted upon the pendant, is attached such that it is in contact with the skin of the neck while the subject eats. This sensor produces a voltage in response to mechanical stress, and is thus able to represent the food being consumed in the output signal, which is sampled at a rate of 20 Hz by a Bluetooth LE-enabled microcontroller. A mobile aggregator extracts various mathematical and statistical features from these signals, and applies classifiers such as Support Vector Machines and RandomForest to distinguish between a variety of different foods. The system is able to distinguish between water, a sandwich, and chips with an F-Measure of over 90%, while running for several days using a simple CR2032 coin-cell battery. The system includes an Android application that aggregates sensor data and informs the user of deficiencies in their diet based on quantity consumed, timing, eating pace, and skipped meals. The limitations of this proposed scheme are user acceptance: impressions of device comfort and social acceptance have not been reported. This is particularly important as the necklace must be sufficiently tight to maintain skin contact during meals.

1.6.2 Jaw-Mounted Strain Sensor

In 2012, Professor Sazonov et al. proposed a method for monitoring chewing using a piezoelectric strain gauge placed on the lower jaw, right below the ear [SF12]. The strain gauge is similar to that of the previously described WearSens device [KAS14a], but the placement is different; the sensor is used to detect chewing rather than swallows. As the user chews, vibrations cause voltage fluctuations at the terminals of the strain gauge, which are then processed and amplified using a custom circuit. Support vector machines are subsequently used to identify meal events, successfully distinguishing them from rest events and talking using a linear kernel function. However, as in the case of other devices that use inertial sensors to detect chewing and swallowing, the comfort and social acceptance aspects of this approach must be more thoroughly investigated.

1.6.3 Summary

Advantages:

- **Battery Life:** The piezoelectric sensor is passive and does not need to be powered. It can be sampled at rates of 20 Hz or less, which is a small fraction of the rates required by audio-based techniques.

Disadvantages:

- **Comfort:** The placement of the device must ensure that the sensor is in contact with the skin at all times.
- **Practicality:** Further work is needed to validate the accuracy, social acceptance, and comfort in real-world conditions; experiments were conducted in a laboratory setting.

1.7 A Survey of Individual Preferences: Methodology

What makes a device or technique successful in real-world applications, rather than an academic exercise? There are many factors, ranging from objective to subjective. An

example of an objective factor is accuracy: the ability of the device to measure or characterize eating behavior. Scope is can also be significant: can the device detect individual chews, classify between individual foods, or simply reports meal timing or eating pace? Issues such as reliability and battery life are also significant for long-term user adherence. However, even the most reliable and accurate techniques will fail to produce meaningful health outcomes if *user adherence* is low. Thus, effective wearables must be sleek, comfortable, and attractive without drawing unwanted attention. However, quantifying these factors is challenging, particularly because many of the proposed devices do not have publically available working prototypes. Therefore, in an attempt to analyze subjective factors such as comfort, convenience, and real-world feasibility, we conducted an online survey to evaluate user interest in various devices based on a brief description of major techniques.

Nine devices were presented to the survey subjects, which represented the most promising techniques across all the different categories described in this survey. Each device was accompanied by an image and a brief description of how it is worn and used. The system outcomes were also clearly defined in each case (ie. what the device was specifically reporting to the user). After the description and photo of each device, subjects were asked the following questions:

- Describe your overall impression of the device.
  Where "1" represents "I would never wear this device" and "5" represents "I would be eager to use this device."

- Describe your willingness to use this device on public.
  Where "1" represents "I would never wear this or use this around others." and "5" represents "I would have no problem wearing / using this around others."

- How useful are the device outputs?
  Where "1" represents "Not useful at all" and "5" represents "Very useful."

Finally, users were given an opportunity to provide comments on their overall impressions of these devices, though this field was not required to complete the survey. Lastly, basic demographic information was provided. Subjects provided their age range, gender,

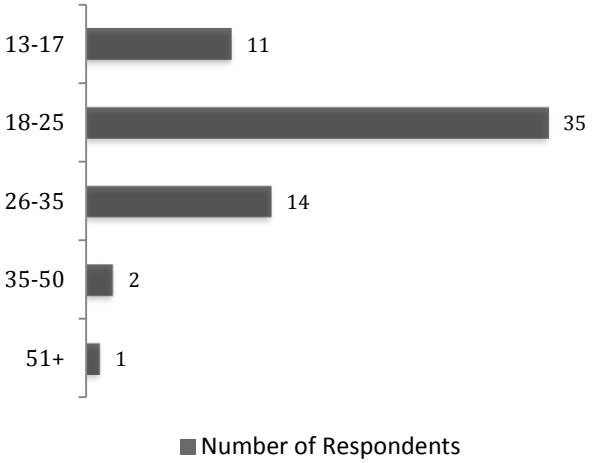


Figure 1.6: The age ranges of survey respondents.

and the approximate location that they live in, as these factors could influence their preferences.

1.8 Survey Results

1.8.1 Demographic Information

A total of 63 subjects participated in the online survey. 57.1% of respondents identified as female, 39.7% identified as male, and the remainder did not disclose. 17.5% of subjects were under the age of 18, with the majority (55.6%) between the ages of 18 and 25. An additional 22.2% of subjects were between 26-35, with only 4.8% of participants reporting their age as over 35. The age ranges of the respondents are shown in Figure 1.6. Overall, 73% of respondents claimed to be residents of the United States or Canada, with the next predominant demographic being Europe (20.6%).

1.8.2 Detailed Results

Detailed survey results are shown in Figure 1.7. 9 devices are compared on the basis of the respondent's overall impressions, their comfort levels wearing or using the device in public, and their assessment of the usefulness of the system outputs. These attributes were rated on a scale of 1 to 5. The highest rated overall device was the smartwatch, which received a rating of 3.13 out of 5. The next highest-scoring devices were the smart

table and smart utensil, at 2.67 and 2.27, respectively. On the other end of the spectrum, the audio headset received the lowest overall score at 1.75, with the smart necklace and jaw-mounted sensor receiving ratings of 1.83 and 1.87, respectively.

The social acceptance of each device appeared to be strongly correlated with the respondent's overall impressions, as the rankings of both criteria produced the same order of devices. The smartwatch was the device respondents were most comfortable wearing by a considerable margin, receiving a score of 3.86 out of 5, with the smart-utensil receiving the next highest rating at 2.65 out of 5. The least accepted devices were the audio headset and smart necklace, at 1.46 and 1.52, respectively.

The last evaluation criteria was system outputs, which were described in the survey alongside photos of each device. The motivation behind this rating is that some devices claim to count calories, others aim to identify foods, and others simply recognize eating behavior. Though it possible that some survey respondents may not assess the utility of system outputs independently of their overall impressions of the device, the results were nevertheless interesting. In this category, the smartwatch only received the 4th highest score (2.46 out of 5) due to the described limitations of this platform. The highest-rated devices were the wearable camera, smart utensil, and audio headset, with ratings of 2.94, 2.65, and 2.49, respectively. The lowest ratings were received by the jaw-mounted sensor and audio earpiece, at 2.21 and 2.27 respectively.

1.8.3 Analysis

A few general conclusions can be drawn from the survey results. First, several survey comments expressed privacy concerns with respect to the mounted camera. One respondent stated, *"Privacy and being self-conscious could be an issue"* while another said *"I have zero desire to wear a camera due to overall greater personal privacy concerns"*. With respect to the audio earpiece, one respondent stated, *" Could easily be hacked to use as recording devices for surveillance purposes"*.

Furthermore, it seems custom devices are generally less preferred to established hardware solutions, as the smartwatch was the overwhelmingly favorite choice in most categories. Though custom devices have a novelty factor that may appeal to some, it appears

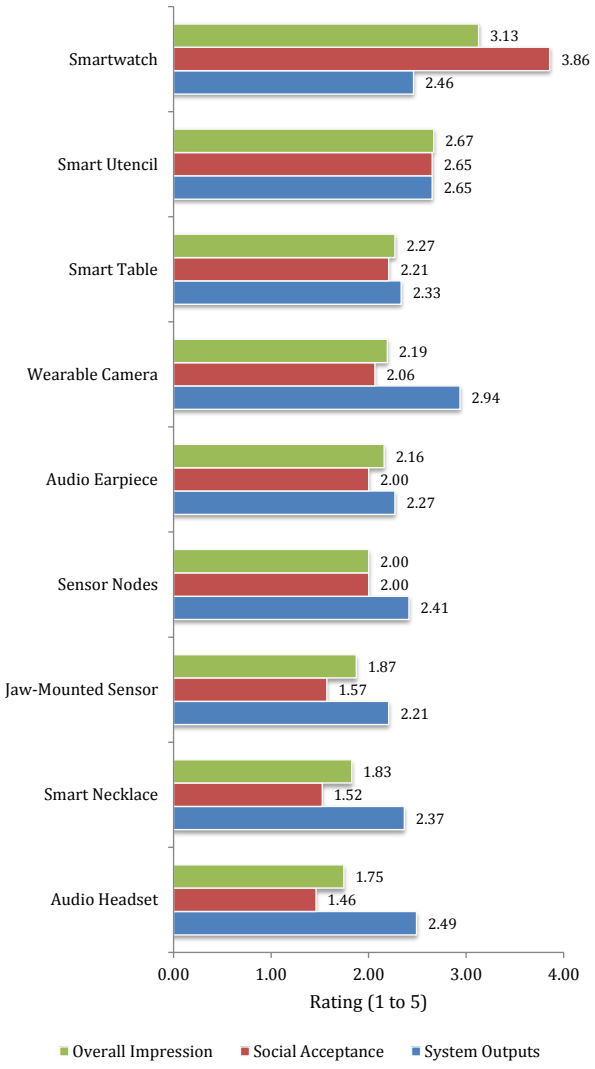


Figure 1.7: Detailed survey results.

that inconspicuous approaches were preferred by most respondents. The smartwatch platform has both inertial sensors and a microphone, and may therefore be a viable platform for recognition of eating behavior as a sensor fusion model can be applied to gestures and swallow sounds for improved accuracy. However, very few works employ the use of smartwatches for dietary-monitoring.

Another observation is that user interest in these devices appears to be more closely associated with the comfort and appearance of the devices, rather than the nutrition-monitoring capabilities.

1.9 Conclusion

In this chapter, we provide a survey of the most promising technologies for non-invasive detection of eating behavior. The devices cover a broad spectrum of techniques ranging from smart table surfaces, wearable cameras, jaw motion sensors, and wearable microphones. We then present results from an online survey, which suggest that a smartwatch or smart-utensil are the preferred platforms for monitoring diet, while the smart necklace and audio-headset are the least-preferred approaches.

CHAPTER 2

Necklace

Prior research has shown a correlation between poor dietary habits and countless negative health outcomes such as heart disease, diabetes, and certain cancers. Automatic monitoring of food intake in an unobtrusive, wearable form-factor can encourage healthy dietary choices by enabling individuals to regulate their eating habits. This chapter presents an objective comparison of two methods for digital dietary intake monitoring: piezoelectric swallow sensing by means of a smart necklace which monitors vibrations in the neck, and audio-based detection using a throat microphone.

2.1 Introduction

Healthy eating can reduce the risk of heart disease, stroke, diabetes, and several cancers. In 2008, medical costs associated with obesity were estimated at $147 billion, and the Centers for Disease Control (CDC) believes that the best areas for treatment and prevention are monitoring behavior and environment settings [cdd14]. Wireless technologies and health-related wearable devices have the potential to enable healthier lifestyle choices. These devices and systems are designed to encourage behavior modifications needed to reduce the risk of obesity and obesity-related diseases [DYN10].

Studies have shown that the number of swallows recorded during a day strongly correlate with weight gain on the following day [SS85]. This provides motivation for the analysis of food intake patterns based on volume. Though many wearable devices have been designed for monitoring activity [FMS98][FLK11][PPB12], automatically and accurately inferring eating durations and patterns in a non-intrusive manner has been for the most part an unaddressed challenge.

Prior works have attempted to characterize eating habits through various means.

Though many methods have been proposed, two of the more promising techniques include inertial-systems using piezoelectric sensors, as well as audio-based detection using throat microphones. In piezoelectric-based techniques, piezoelectric sensors, which produce a voltage in response to mechanical stress, can be used to detect movement in the skin on the lower-neck associated with swallowing. This approach differs from microphones based on piezoelectric technology: our system does not detect sound waves, instead assessing motion in the skin that results from swallows and chewing. Alternatively, audio-based techniques typically place a small microphone near the jaw or neck, and record eating noises such as chewing and swallowing. These sounds can be disambiguated from other background noises using classifiers and other signal-processing techniques. These approaches differ significantly from a perspective of comfort, practicality, convenience, power usage, and detection accuracy.

The primary novelties of our work are the description of a system in which a piezoelectric sensor is placed in the lower part of the neck for detecting swallow motions, and a comparison of this technique with audio-based monitoring using datasets derived from the same experiments. This provides a much more objective comparison of these two technologies than otherwise possible by comparing results from separate works using different datasets and methodologies. Furthermore, we provide an evaluation of the power overhead of these techniques as a function of sample rate, computational overhead, and Bluetooth connection interval.

2.2 Hardware Architecture

2.2.1 Audio-based Approach

In order to analyze the volume and consistency of a meal, audio samples are acquired while an individual is eating, using a commercial throat microphone placed freely in the lower part of the neck. A primary advantage of this method is comfort, as unlike the piezoelectric sensor, it is not necessary to have skin contact at all times using this approach. The microphone was resting loosely around the throat near the lower collarbone. The particular microphone used in this study is the Hyparion Flexible Throat Mic Microphone Covert Acoustic Tube Earpiece Headset, which is connected directly to the mobile

phone's audio input port using a 3.5mm male audio cable, and picks up audio signals from swallows, through the air. Commercially available audio-recording technology was used to acquire the audio recordings from the microphone.

2.2.2 Piezo-Based Approach

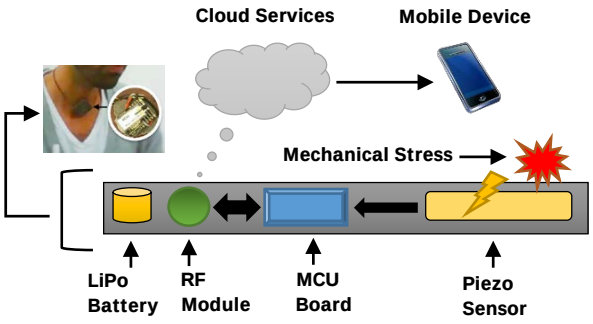


Figure 2.1: Systems architecture of the piezoelectric sensor-based necklace.

A piezoelectric sensor, sometimes known as a vibration sensor, produces a voltage when subjected to physical strain. By placing a piezoelectric sensor against the throat, the motion of the skin during a swallow is represented in the output of the sensor, when sampled at frequencies as low as 5 Hz. The NIMON necklace, presented by Kalantarian et al. in [KAS14b], describes a non-invasive, wearable device capable of detecting swallows by placement of a vibration sensor near the lower trachea. During a swallow event, muscular contractions cause skin motion, which pushes the vibration sensor away from the body and towards the fabric of the necklace, generating a unique output voltage pattern. The skin motion during a swallow is quite small and requires very high sensitivity to detect. Therefore, the long and thin design of a piezoelectric strip very well suited for this application, in which the necklace clamps the sensor to the skin and causes it to bend slightly during a swallow.

Raw data is sampled from the vibration sensor at a rate of 20 Hz, and a windowing algorithm computes the standard deviation of the values in each window (typically sized 20). Subsequently, the peaks are identified using on a thresholding technique, which typically correspond with swallows. The various steps in data processing are shown in Figure 2.2, with the raw value visible at the top. The noticeable dips in the waveform

generally correspond with swallows. After initial data acquisition, the data is smoothed using a moving-average low-pass filter with a span of 5, to reduce the impact of noise. Subsequently, a sliding window of length 9, corresponding with .45 seconds of data, is applied with a maximum overlap (shifted one point at a time). These numbers were experimentally determined to be optimal for preserving the critical features of the waveform based on simulations. The original implementation of the NIMON necklace used Bluetooth LE to transmit all raw data to an Android phone for processing; the algorithm does not run on the embedded hardware, which is powered by a small lithium-polymer battery.

The piezoelectric sensors can be used for more advanced classification activities, beyond counting swallows. Because the piezoelectric sensor is capable of detecting motions beyond swallows, the detection of consistent chewing between swallows is a reliable indicator that a solid food is being consumed, while several swallows, with no chewing between them, may indicate that a liquid is being consumed. Generally speaking, the foods with different textures produce different patterns of vibrations in the neck, as a result of the varying amounts of jaw strength necessary for chewing, as well as varying speeds at which foods are eaten, chewed, and swallowed. Analysis of the statistical features associated with the raw data sampled from the piezoelectric sensor can reveal the food being consumed, within a limited subset.

2.3 Algorithms

2.3.1 Audio Feature Extraction and Classification

The Munich open Speech and Music Interpretation by Large Space Extraction toolkit, known as openSMILE [EWS10], is a feature extraction tool intended for producing large audio feature sets. This tool is capable of various audio signal processing operations such as applying window functions, FFT, FIR filterbanks, autocorrelation, and cepstrum. In addition to these techniques, openSMILE is capable of extracting various speech related features and statistical features. A partial list of extracted features is shown in Table 3.1 and 3.2, respectively. Other audio-based features include frame energy, intensity, auditory spectra, zero crossing rate, and voice quality. After data is collected from a

Table 2.1: Partial List of openSMILE Speech Features.

Speech-Related Features		
Signal Energy	Loudness	Mel/Bark/Octave Spectra
MFCC	PLP-CC	Pitch
Voice Quality	Formants	LPC
Line Spectral Pairs	Spectral Shape	CENS and CHROMA

Table 2.2: Partial List of openSMILE Statistical Features.

Speech-Related Features		
Means	Extremes	Moments
Segments	Samples	Peaks
Zero Crossings	Quadratic Regression	Percentiles
Duration	Onset	DCT Coefficient

variety of subjects eating several foods, feature selection tools can be used to identify strong features that are accurate predictors of swallows and bites for various foods, while reducing the dimensionality by eliminating redundant or weakly correlated features.

A custom script concatenates all recorded audio clips from throat microphone experiments into one large audio file, and splits them into smaller clips of length n, based on the input parameters. This allows the extraction of eating clips without a-priori knowledge of when the swallows take place, for experimental variety. The feature extraction tool then extracts a large feature set of over 6500 attributes per clip, which provides the classifier with enough information to distinguish between the different categories. The InformationGain Attribute Evaluation tool then reduces the feature sets by selecting those which have minimum redundancy and maximum correlation to the defined classifier outcomes.

Extracted Feature	Description
$\frac{\sum_{x=1}^{m}\sum_{y=1}^{n} a_{xy}}{m*n}$	The average value of amplitude within a sample window.
$\sqrt{\frac{\sum_{x=1}^{m}\sum_{y=1}^{n}(a_{xy}-\mu)}{m*n}}$	The standard deviation of amplitude within a sample window.
$\frac{\sum_{x=1}^{n} a_{zx}}{n}$ $for\,\{z \mid 1 \le z \le m\}$	Average of the various frequency bins. Each frequency range is extracted separately as an independent feature.
$\sqrt{\frac{\sum_{x=1}^{n}(a_{zx}-\mu)}{n}}$ $for\,\{z \mid 1 \le z \le m\}$	The standard deviation of a frequency range over a period of time, for every frequency bin.

Table 2.3: Selected features extracted from the audio spectrogram.

2.3.2 Piezoelectric Feature Extraction and Classification

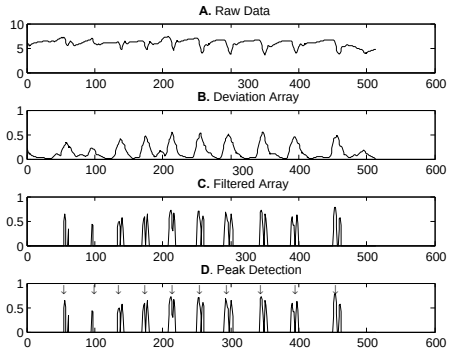


Figure 2.2: Signal processing flow used by the WearSens necklace to identify swallows.

A spectrogram is a visual representation of the frequency spectrum over time, and is an ideal representation for extracting distinguishing features in many classification problems. A spectrogram is typically generated using a short-time Fourier transform (STFT) with a fixed window size, the squared magnitude of which yields the spectrogram. Fundamentally, a spectrogram allows easy identification of changes in the frequency spectrum of a signal, over time. This is significant because eating certain foods produce vibrations in different frequency ranges, based on the texture of the food and the amount of chewing involved. A spectrogram can provide a relatively straightforward representation of changes in the frequency distribution over time as subjects eat, which can reveal the distinguishing characteristics of various foods.

Table 2.4: Feature Table for Piezoelectric Sensor Feature Extraction

Mean	Geometric Mean	Std. Dev.
Skewness	Mean of Standardized Z-Scores	IQR
Kurtosis	Harmonic Mean	Rank Corr.
Range	Median Absolute Deviation	Partial Corr.

The spectrogram is calculated from the time signal $x(t)$, as shown in Equation 2.1 using the short-time Fourier transform (STFT).

$$STFT\{x(t)\} \equiv X(n,\omega) = \sum_{t=-\infty}^{\infty} x[t]\omega[t-n]e^{-j\omega t}. \tag{2.1}$$

$x(t)$ is multiplied by a window function for a short period of time. The data is divided into frames F_i, which overlap. Each frame is Fourier transformed, and the result is added to a matrix that records the magnitude and phase of each point in time and frequency. A Hamming window was used of varying lengths of $w = 32, 64$, and 128, with an FFT length of $nfft = 32, 64$, and 128, and an of overlap of 25%, 50%, 75%, and no overlap. We set the dynamics range to 50dB. Each spectrogram is defined by a matrix $P \in R^{m\times k}$, where m is the number of bins in the time domain, and k is the number of bins in the frequency domain. P represents the power spectral density.

Once a spectrogram is generated for each swallow, we found an optimal division of the spectrogram images into 14 bins along the frequency domain and another 16 bins along the time domain, for a total of 30 bins. We then calculate statistical features on each bin, to generate a feature vector V_i for each swallow. Table 2.4 lists the main features that were calculated for each bin, which generates a total of $s = 360$ features per spectrogram swallow. The motivation for the use of these features is based on the work by Alshurafa et al. in [AKP15], which demonstrated the superiority of the spectrogram-based statistical approach over alternatives such as matching pursuit, and scalogram-based Gabor wavelets.

2.4 Experimental Procedure

2.4.1 Piezoelectric Sensor Data Collection

Two experiments were performed to validate the efficacy of our algorithm in accurately detecting swallows and recognizing eating patterns using statistical features collected from a spectrogram. To prevent bias in the classification results between each class label in the training set, we randomly select an equal number of swallows across categories. We also perform leave-one-out cross validation and report the results.

In the first experiment data was collected on ten subjects, two female and eight male with ages ranging between 20 and 40 years of age. We placed the necklace around their neck so that the sensor was loosely touching the skin. The necklace tightness was adjusted such that each subject was comfortable wearing the device. We placed the necklace centered between their right and left clavicle right above the sternum and asked the subject to eat the following foods, one at a time: a chicken salad or tuna salad sandwich, a small handful of Pringles potato chips, and a cup of 9oz water. No specific instructions were given about the manner in which the food was consumed, though subjects were under observation during the data collection process which may have increased the pace of eating beyond what would otherwise be typical.

In the second experiment we increased the number of subjects to twenty, eight female and twelve male, ages 20 to 40 years. The subjects each consumed a meat-like veggie patty, a handful of mixed nuts, and two small Snickers chocolate bars. We ensured that the portion sizes were identical from one subject to another. The subjects were asked to push a button every time they swallowed; this helped us further annotate the data in order to provide truth labels for the dataset.

2.4.2 Audio Data Collection

Our data collection includes data from 20 individuals using a throat microphone placed near the bottom of the neck. The moments at which food was swallowed were indicated by pressing a push button which added an annotation to the associated log file. In the original data collection, ten subjects were instructed to eat two identical sandwiches

(3-inch and 6-inch), and drink two cups of water (9 fl. oz and 18 fl. oz), and eat a small handful (approximately 3) of Pringles chips. They were also instructed not to eat, swallow, or speak for a brief period in order to acquire signals corresponding to silence, or background noise. Subsequently, 189 audio samples were extracted from the recordings. The next phase of experimentation employed twenty subjects, who were given a small portion of nuts, chocolate, a vegetarian meat-substitute patty. The foods were consumed sequentially, in that order. Over 50 additional samples were manually extracted from this experiment, though some data was corrupted. These recordings formed the basis of the algorithm design and experimental evaluation. For evaluation of classification results, leave-one-out cross validation was used.

The data collection took place in a lab environment; people can be faintly heard speaking in the background, and the microphone occasionally recorded doors closing and nearby footsteps. In most audio classification works, ambient noises can interfere with the signal and decrease classification accuracy. This issue is partially rectified by placing the throat microphone in the lower part of the neck. This microphone placement emphasizes swallow sounds, as they are in much closer proximity to the device than ambient noises. Furthermore, most commercial throat microphones contain active circuitry for filtering out these ambient signals. These factors make throat microphones particularly well-suited for the task of recognizing eating behavior from chew and swallow sounds. Our approach for detecting ingestion, despite the presence of background noises, is similar to the evaluation conducted by Kalantarian et al. in [KS15a]. The experiments presented in this chapter suggested that spectrogram-based feature extraction techniques are relatively resilient against the ambient noises, as the frequency bands associated with external sounds are typically not selected features by the classifier models and therefore do not significantly affect the classification results.

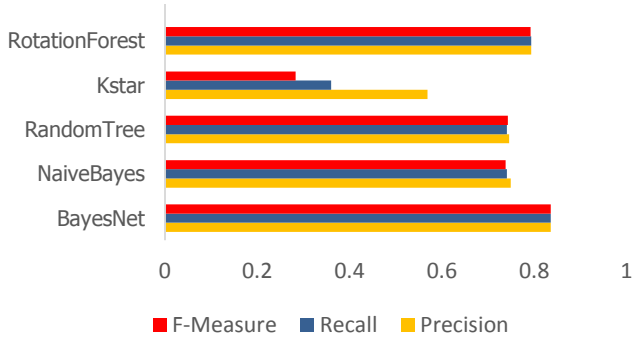


Figure 2.3: Comparison of the accuracies of various classifiers for the throat microphone.

2.5 Evaluation

Table 2.5: Audio: Confusion Matrix (Random Forest) using a 1-second window

	Predicted Outcome			
Swallow Type	Water	Sandwich	Chips	Recall
Water	43	7	0	86%
Sandwich	6	44	0	88%
Chips	0	0	50	100%
Precision	87.7%	86.3%	100%	

Table 2.6: Audio: Confusion Matrix (Random Forest) using a 1-second window.

	Predicted Outcome			
Swallow Type	Nuts	Chocolate	Patty	Recall
Nuts	39	6	5	78.0%
Chocolate	0	49	1	98.0%
Patty	4	1	45	90.0%
Precision	90.6%	87.5%	88.2%	

Table 2.7: Piezoelectric Sensor: Confusion Matrix (Random Forest).

	Predicted Outcome			
Swallow Type	Water	Sandwich	Chips	Recall
Water	43	3	4	86.0%
Sandwich	2	37	11	74.0%
Chips	5	12	33	66.0%
Precision	86.0%	71.1%	68.7%	

Table 2.8: Piezoelectric Sensor: Confusion Matrix (Random Forest).

	Predicted Outcome			
Swallow Type	Nuts	Chocolate	Patty	Recall
Nuts	35	11	4	70.0%
Chocolate	5	40	5	80.0%
Patty	2	4	44	88.0%
Precision	83.3%	72.7%	83.0%	

2.5.1 Audio-Based Classification Results

Table 2.5 shows the accuracy of acoustic swallow detection for three food types: water, sandwich, and chips using 1-second samples. Chips in particular had the highest classification accuracy, with a recall and precision of 100%, based on 50 samples. The precision and recall of water was lower, at 87.7% and 86%, respectively. This is partially because of the automated method of dividing the water sample clips. Because drinking water has no chewing, and very little information between swallows, it can be difficult to classify an audio clip that happens to be taken between swallows. Table 3.3 shows the classification accuracy of nuts, chocolate, and the veggie-patty. Of note are the poor results for nuts, with a recall of 78.0%.

Because classification was among four foods, rather than binary classification, results are quite promising. However, it is clear that the algorithm may not scale well if tested on a larger number of food types (ie. 50-100). Therefore, it is desirable for future systems to create very broad categories based on highly generalizable features. For example,

the work by Amft et al. in [Amf10] describes a scheme which generalizes food into three significant clusters of food: "wet and loud," "dry and loud," and "soft and quiet." Further research is necessary to determine typical nutritional qualities associated with these types of foods.

2.5.2 Piezoelectric-Based Classification Results

According to our classification results, using spectrogram-based features on a signal from a piezoelectric sensor can distinguish between liquid and solid swallows with high accuracy using the Random Forest Classifier (with n=100 trees), which yielded the optimal results for all three experiments. Best results were achieved using a window size of 32, an FFT length of 32, and an overlap of 50%. Generally, it was observed that the Random Forest Classifier consistently outperforms other well-known classifiers, even when distinguishing between solids, achieving an F-measure of 75.29% in the first experiment (water, sandwich, chips) and 79.44% in the second (nuts, chocolate, patty).

The RandomForest classifier [LW02] is an ensemble-learning decision tree classifier that, unlike most other tree-based classifiers which split each node based on the best subset of features, instead uses the best subset of predictors randomly chosen at that node. This unique property of the RandomForest classifier has been shown to make it relatively robust to overfitting, and has compared favorably against other classifiers such as support vector machines and neural-network classifiers. A more detailed investigation of the properties of this classifier can be found in the work by Leo Breiman in [Bre01].

The Bayesian Network and kNN classifier resulted in a 72.6% and 65.4% F-measure, respectively. Table 2.8 provides the confusion matrix for the Random Forest Classifier for the first experiment, while Table 2.7 shows results for the second. Though results using the piezoelectric sensor were generally strong, especially in light of the very low sample rate of 20Hz (compared to 44000 Hz for the microphone), audio-based classification has higher accuracy. However, approach this comes at the expense of computational and power overhead.

2.6 Energy Modeling Methods

Table 2.9: A List of System Components Used in Evaluation

Hardware Components (NIMON Necklace)	Description
Nordic nRF8002	Bluetooth 4.0 Module
LIS3DH Accelerometer	Accelerometer
MSP430G2744	Microcontroller
CR2032 Coin-Cell	Battery for powering device

To realize the intended goal of minimizing burden, it is desirable for wearable devices to remain powered for weeks, or months, without interruption. Required nightly charging or frequent coin-cell battery replacement can be considered an addition burden to the user, which is likely to lower long-term compliance rates. Therefore, modern wearable devices are carefully designed to minimize power usage, and run for days or even months. For example, the Misfit Shine activity monitor claims a battery life of four months [misa]. Other wearables devices such as the Jawbone UP24 claim their devices can sustain seven days of continuous use [Jaw]. Another important drawback of wearable devices with high power requirements are their reliance on large batteries, which may impact the comfort, convenience, and aesthetic appearance of the device. Therefore, it is necessary for wearable devices to carefully factor energy efficiency in their design. This is particularly true in the case of nutrition monitoring devices which are typically mounted in the throat or jaw area.

We now discuss a comparison of power consumption for both schemes. Table 2.9 presents the hardware components used for power evaluation of the devices. Note that the original microcontroller board, the Arduino-based RFDuino, has been substituted for an MSP430 variant for the superior simulation environment and embedded low-power architecture. The MSP430 has 5 different power modes: Active Mode, and Low Power mode (LP) 1-4. Active mode is used when executing algorithms, while the low power modes supply a clock signal to various peripherals. LP3 disables the CPU, MCLK, and SMCLK. Note that MCLK is the main system clock of the MSP430, used by the CPU. SMCLK is the sub-main clock used by other peripherals such as timers. To conserve

energy, the device is configured to alternate between Active Mode and LP3. This can be seen in Figure 2.8; the device briefly enters Active Mode to acquire, process, and transmit data, returning to LPM3 immediately thereafter to conserve energy and maximize battery life.

The selected Bluetooth transceiver is the Nordic nRF8002, commonly used in several commercial wearables including the Misfit Shine. Power evaluation of the Bluetooth Transceiver was provided by Nordic Semiconductor's nRFGo Studio software (version 1.17), which simulates power demands and battery lifetime based on many different parameters such as data size, encryption, connection intervals, advertising intervals, and transmit strength. Real-time power debugging of the MSP430 microcontroller was provided by the EnergyTrace++ technology in Texas Instruments Code Composer Studio (CCS) Version 6.0, which provides real-time information about processor state and power consumption. The data in the following sections corresponds with the piezoelectric-based designs unless otherwise specified. However, most of the results can be generalized to draw conclusions between these two schemes. For example, both techniques require sampling, buffering, and transmission. Specific to the audio-based approach is the FFT algorithm, for which we provide an analysis.

2.7 Power Evaluation

Equation 2.2 shows a representation of the energy needed to acquire a sample from the piezoelectric sensor via the on-chip ADC, process the data, and transmit via Bluetooth 4.0 to mobile phone. Equation 2.3 shows the power consumption of the system (P) in terms of the sample rate and the energy required to acquire, process, and transmit a sample, which are the major sources of power loss in both the audio-based and piezoelectric-based systems.

$$E_n = E_{adc} + E_{proc} + E_{tx} \tag{2.2}$$

$$P = \sum_{a=1}^{f} [E_{adc} + E_{proc} + E_{tx}] = \sum_{a=1}^{f} E_n \tag{2.3}$$

2.7.1 Sample Rate, Window Size, and Power

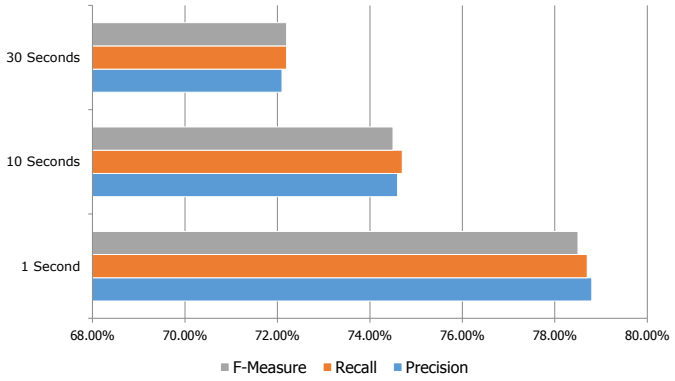


Figure 2.4: Effect of window size on classification accuracy (audio).

Table 2.9 shows the relationship between sample rate and power, at several different window sizes. Note that the window size refers to how many samples a given value acquired from the piezoelectric sensor is compared against, to calculate the variance of the data. Recall that the sample rate is associated with the frequency at which samples are acquired from the piezoelectric sensor, buffered locally, and processed. Therefore, a high sample rate not only incurs local processing overhead based on the implementation of the swallow detection algorithm, but also directly relates to the usage of the on-chip Analog-Digital converter of the MSP430. Table 2.10 shows similar numbers for the case when data processing is offloaded to the mobile phone. In this case, the device simply acquires raw data and transmits it immediately, and the contributions to the reported power are AD conversion and transmission. This table shows that increasing the sample rate between 1 Hz to 32 Hz has a quite limited effect on power usage, since the AD converter is in fact designed for significantly higher sample rates. In comparison with Table 2.9, we conclude that the high sample rate consumes more power largely by increasing the number of data points that must be processed.

Table 2.9 also shows the relationship between window size and power, at several different sample rates. It should be noted that the window size is associated with the piezoelectric-based algorithm only, in our evaluation. It refers to the size of the sliding average window used in the peak detection algorithm. A key observation is that increasing the window size does not have a substantial effect on power usage, for low sample rates

[KAP15]. The NIMON necklace has been tested at sample rates of 10 Hz and 20 Hz, with window sizes between 4 and 10 samples. The window size appears to be a significant contributor to low device lifetime only at very high sample rates. Also, note that in Table 2.10, offloading the processing from the necklace to the smart phone decouples the window size parameter from the current usage of the device.

2.7.2 Power vs. Connection Interval

Power results for various connection intervals are shown in Table 2.10. These numbers assume a fixed bandwidth based on data acquisition at 20 Hz. Therefore, higher connection intervals also have higher payloads. Results show that average power ranges from 1.2 mW at a connection interval of 50ms, to 0.026mW at a connection interval of 3 seconds. Compared to the extra .01 mW for performing swallow detection locally, it is clear this is more optimal to run detection algorithms locally for anything responsive enough to be considered real time user feedback (10 seconds or less). Clearly, the connection interval is a significant contributor to overall power dissipation, and this should be increased whenever possible.

2.7.3 Energy Evaluation of Fourier-Based Algorithms

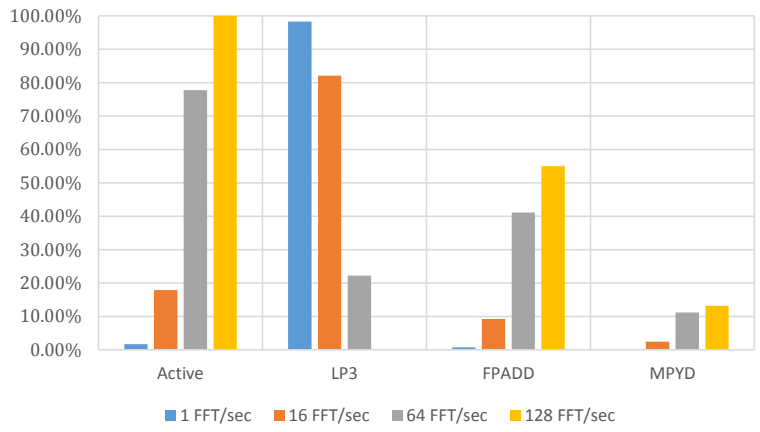


Figure 2.5: Percentage of time the MSP430 spent in various modes, as a result of 16-point FFT operations performed at different rates.

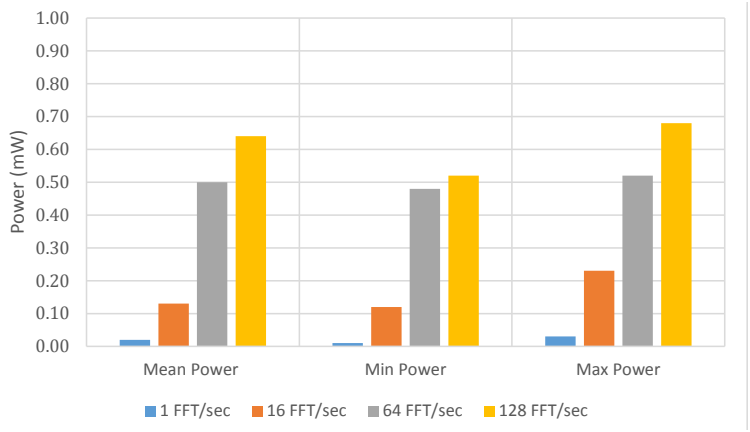


Figure 2.6: Power dissipation of the microcontroller when 16-point FFT operations are performed at various rates.

In this section, we briefly cover the device lifetime challenges using acoustic sensors. In [AKT09], Amft et al. were able to recognize chewing with high precision using a sample rate of 44kHz, which is quite high for an embedded application. This is especially the case when compared to the approach of the piezoelectric necklace in which sample rates as low as 10 Hz have been validated. Though clearly, a 44 kHz audio signal will have more information for classification than a much smaller sample rate, the implications of battery life will be substantial. In another work by Amft et al. in [ASL05], audio data was also acquired at 44 kHz and processed with a 512-point FFT. However, prior works [BPB92][HDG88] have shown spectral energy from potato chips to be primarily between 0-10kHz, with highest amplitude frequency ranges between 1 and 2 kHz. Based on Nyquist-Shannon sampling theory, this conservative estimate would require a sample rate of 4 kHz. Though this is well within the specifications of the MSP430 ADC unit, it far exceeds the sample rate of 10-20Hz, as required by the vibration sensor. The overhead of acquiring and transmitting up to 400x more data, not to mention the computational overhead of the FFT in comparison to the simple windowing algorithm described in this chapter, makes the vibration sensor a far better choice.

For evaluation, a 16-point FFT algorithm was implemented on the MSP430 microcontroller. The power debugging did not evaluate the impact of high-sample ADC units; we instead measured the power usage of the FFT algorithm itself. The sampling, buffering of results, and transmission would incur significant additional overhead. In a continuous

signal processing application such as this, the FFT algorithm must be run periodically to analyze frequency domain features of incoming data. Table 2.6 shows results for four different FFT rates (1, 16, 64, and 128). For example, an FFT rate of 16 represents an operating mode in which a 16-point FFT is evaluated 16 times per second (therefore processing 256 samples per second). The system was designed such that the MSP430 microcontroller would immediately enter LPM3 (low power mode 3) to conserve energy between FFT operations. The table shows that average power is quite high for high FFT rates such as 128 (corresponding to 2048 samples/second, or a sample rate of 2kHz); mean power approached 0.64 mW, without considering the additional overhead of sampling at 2 kHz, buffering the results, and transmitting to mobile phone. Table 2.5 shows that as the frequency of FFT operations increases, the MSP430 microcontroller spends a higher percentage of its time in Active Mode, in which power is not conserved. At the rate of 128 FFTs/second, the device spends all of its time in Active Mode, and 55% of its time performing a floating point add operation. This suggests that the microcontroller is unable to perform FFT operations at the requested rate, since it should enter LPM3 after completion. At this rate, the power dissipation is over ten times greater than that of a vibration-sensor based system with a sample rate of 8 Hz and a window size of 20, as shown in Table 2.9.

2.8 Conclusion

In this chapter, we provide an analysis of two emerging techniques for monitoring eating habits in consumer electronics applications: using piezoelectric sensors placed in the bottom of the neck, and throat microphones for analysis of audio signals associated with the ingestion of food. Results suggest that audio-based classification has somewhat higher accuracy, particularly for dry foods such as chips and nuts. However, we show that the audio-based approach incurs a power overhead approximately ten times greater than the vibration-sensor system, which may have broad implications with respect to device form factor, user acceptance and comfort.

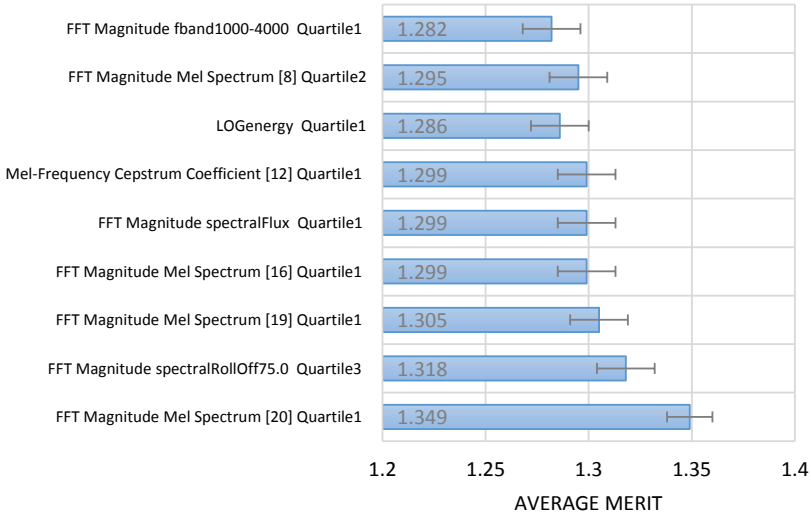


Figure 2.7: Major features for audio-based classification using the openSMILE toolkit, for audio-based classification.

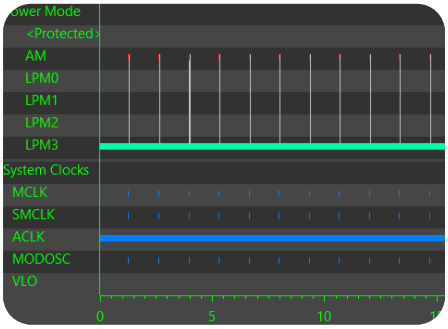


Figure 2.8: State transitions of the MSP430 microcontroller, which occur on an interrupt callback. in LPM3, all clocks besides ACLK (necessary for the timer) are disabled.

Window Size	Sample Rate				
	1 Hz	**4 Hz**	**8 Hz**	**16 Hz**	**32 Hz**
4	.05 mW	.06 mW	.06 mW	.07 mW	.07 mW
10	.05 mW	.06 mW	.06 mW	.07 mW	.08 mW
20	.05 mW	.06 mW	.06 mW	.08 mW	.10 mW
100	.06 mW	.08 mW	.10 mW	.16 mW	.26 mW

Figure 2.9: Relationship between window size, sample rate, and mean current drawn on MSP430 mcu with on-board swallow detection.

Window Size	Sample Rate				
	1 Hz	**4 Hz**	**8 Hz**	**16 Hz**	**32 Hz**
4	.05 mW	.05 mW	.05 mW	.06 mW	.06 mW
10	.05 mW	.05 mW	.05 mW	.06 mW	.06 mW
20	.05 mW	.05 mW	.05 mW	.06 mW	.06 mW
100	.05 mW	.05 mW	.05 mW	.06 mW	.06 mW

Figure 2.10: Relationship between window size, sample rate, and mean current drawn on MSP430 mcu when swallow detection is offloaded to the mobile phone.

Table 2.10: Average Transmit Power for a Fixed Bandwidth (20 Hz) at 3.7V

T_{conn} (ms)	Payload (bytes)	Avg Current (μA)	Avg Power (μW)
50	6	328.37	1214.97
100	12	260.70	964.59
200	24	195.66	723.94
300	36	99.98	369.92
500	60	41.98	155.32
1000	120	19.90	73.62
3000	360	7.18	26.56

CHAPTER 3

Smartwatch

In recent years, smartwatches have emerged as a viable platform for a variety of medical and health-related applications. In addition to the benefits of a stable hardware platform, these devices have a significant advantage over other wrist-worn devices, in that user acceptance of watches are higher than other custom hardware solutions. In this chapter, we describe signal-processing techniques for identification of chews and swallows using a smartwatch device's built-in microphone. Moreover, we conduct a survey to evaluate the potential of the smartwatch as a platform for monitoring nutrition. The focus of this chapter is to analyze the overall applicability of a smartwatch-based system for food-intake monitoring. Evaluation results confirm the efficacy of our technique; classification was performed between apple and potato chip bites, water swallows, talking, and ambient noise, with an F-Measure of 94.5% based on 250 collected samples.

3.1 Introduction

There is little doubt that obesity is associated with various negative health outcomes such as an increased risk for stroke, diabetes, various cancers, heart disease, and other conditions. In 2008, medical costs associated with obesity were estimated to exceed $147 billion, with over one-third of adults in the United States estimated to be obese [cdd14]. The two major contributors to weight gain are an inactive lifestyle and poor diet. Though the former has been addressed by many wearable devices in recent years both in research and the consumer electronics field, few works exist on automatic detection of dietary habits in an inconspicuous form-factor [FMS98][FLK11][PPB12]. Instead, characterization of an individual's eating habits is possible through manual record keeping such as food diaries, 24-hour recalls, and food frequency questionnaires. However, these approaches suffer from low accuracy, high user burden, and low rates of long-term com-

pliance. Wireless health-monitoring technologies have the potential to promote healthy behavior and address the ultimate goal of enabling better lifestyle choices.

In recent years, several electronic devices have been proposed for monitoring dietary habits. However, most works attempt to characterize eating from patterns in chewing and swallow counts, and very few proposed attempt to identify the nutritive properties of the foods themselves. Therefore, a fundamental question in the field of electronic food monitoring is the validity of chew and swallow counts as a heuristic for estimation of Caloric intake. A recent work by Fontana et al. [FHS14] addresses this issue by comparing several different techniques for estimation of Caloric intake: weighed food records (gold standard), diet diaries, and electronic sensor-based measurements of chews and swallows. Though the study was conducted under constrained conditions, the results suggest that chew and swallow counts may be a promising alternative to manual self-reporting techniques.

While many audio-based nutrition monitors are novel from a perspective of algorithmic techniques, they generally propose custom hardware solutions or bulky non-standard equipment which are of limited use outside of clinical environments. The primary challenge of monitoring a subject's eating habits is creating a system that provides passive monitoring of behavior, presenting a low level of user burden and providing no compromises on comfort and appearance: even the most accurate techniques have very limited scope if they do not encourage repeated use from users.

Many prior works address the problem of nutrition monitoring by processing audio signals associated with ingestion. Typically, these systems use a throat microphone for recognizing deglutition (swallows), or using time-frequency decomposition techniques, such as Wavelet Packet Decomposition (WPD) or Spectrogram Analysis to extract distinctive features, and either classify between different food groups or recognize anomalies in swallow patterns. While many of these works are novel from a perspective of algorithmic techniques, they generally propose custom hardware solutions or bulky non-standard equipment which are of limited use outside of clinical environments. The primary challenge of monitoring a subject's eating habits is creating a system that provides passive monitoring of behavior, presenting a low level of user burden and providing no compromises on comfort and appearance: even the most accurate techniques have very limited

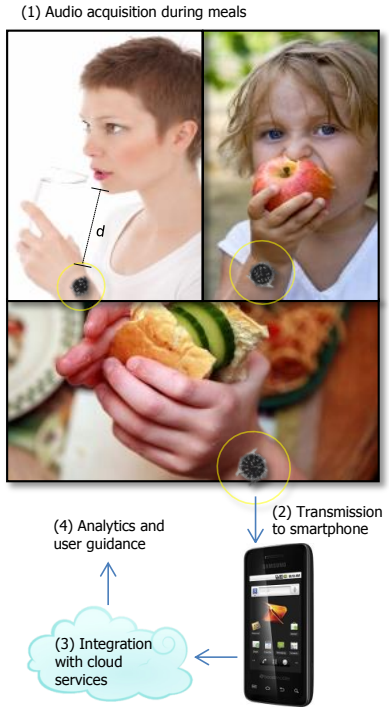


Figure 3.1: High level architecture of the smartwatch system.

scope if they do not encourage repeated use from users.

Recently, smartwatches have emerged as a new platform that provide several promising applications such as wrist-worn activity monitoring, heart rate tracking, and even stress measurement. Watch usage is well established and has a high level of social acceptance, as confirmed not only by our personal studies but by their ubiquity in day-to-day life. Furthermore, the smartwatch platform provides many useful services that can collectively improve user adherence rates, rather than specialized devices with just one application that may fail to sustain a user's interest. These devices contain a multitude of sensors including but not limited to: a microphone, camera, accelerometer, and gyroscope. Due to the ubiquity of watches, this technology can be used for various wireless health monitoring applications discretely, with low user burden. Furthermore, from a user-acceptance standpoint, these systems have a clear advantage over other proposed solutions based on custom hardware, which may require that these bulky and non-standard devices be worn in unconventional ways. Clearly, the multitude of sensors available on the smartwatch platform, wireless connectivity, as well as the comfort and social acceptance of the form-factor warrant further study into their potential applications in the medical

Figure 3.2: Samsung Galaxy Gear phone with Android 4.2.1.

and health-monitoring domain.

This chapter explores the idea of tracking eating habits using a custom Android application on the smartwatch platform. Though identifying eating-related gestures using wrist-worn devices is a viable application of the watch, the focus of our work is to explore the idea of using audio to detect eating behavior based on bites, rather than swallows as other works have done. A high-level system architecture is presented in Figure 3.1. The first step is audio-based acquisition of eating-related sounds such as bites, acquired from the microphone integrated within the smartwatch. After data acquisition, the audio is processed using various classifiers to identify the sound and infer the associated activity.

In addition, we conducted two surveys in order to evaluate the potential of the smartwatch platform for nutrition monitoring. The surveys were conducted online, with 221 respondents in the first and 55 in the second. In the first survey, we asked subjects various questions about their general habits with respect to watches. For example, subjects were asked which hand they prefer to wear a watch, and whether they were willing to wear a watch on the opposite hand on which they were accustomed. In the second survey, respondents provided information about their opinion of various wearable form factors. 55 subjects rated their receptiveness to smartwatches, necklace-based wearables, custom wrist-worn hardware, and smart glasses.

3.2 System Architecture

Our proposed system does not require any custom hardware: the Android application runs on Samsung Galaxy Gear smartwatch running Android 4.2.1. This device, shown in Figure 3.2, features an 800 MHz ARM-based processor, 512 MB of RAM, and a 320x320

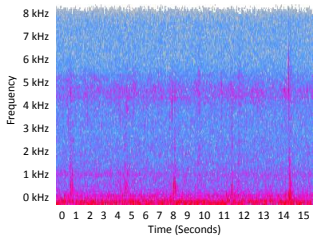


Figure 3.3: Spectrogram for five swallows using a Hamming window.

pixel 1.6 inch display. The device also supports transfer of data using the Bluetooth LE protocol, and can be configured to access the Internet using Bluetooth tethering with compatible smartphones. Once the on-board algorithm detects that a bite has been taken, a web-service call is made to store the data in a database for access by caregivers. In the case of algorithm inaccuracies and errors, subjects are permitted to manually make modifications and add annotations to the data.

Data was recorded using the Samsung Galaxy Gear microphone in MPEG-4 Part 14 (m4a) format at a rate of 96 kbps, as prior research has shown that the spectral energy for many common foods is between 0-10 kHz, with highest amplitude ranges between 1 and 2 kHz for water [BPB92][HDG88]. Of note is the availability of additional sensors on the Samsung Gear platform, including accelerometers and gyroscopes, which can be used for improved classification accuracy in future work, based on hand and wrist motion associated with eating behavior.

The Samsung Galaxy Gear has a 315 mAh capacity battery. This is significant because audio recording and transmission is a relatively energy-intensive task that may compromise battery life. This is partially mitigated by the decision to acquire data at a low sample rate. A more comprehensive evaluation is provided in Section 6.

3.3 Algorithm Design

3.3.1 Frequency-Domain Evaluation: Liquids

We begin our algorithm analysis with the objective of detecting liquid ingestion using a smartwatch. Because we have a-priori knowledge about the kind of data we would like to identify, we could pre-process the recorded data before classification, as we describe in

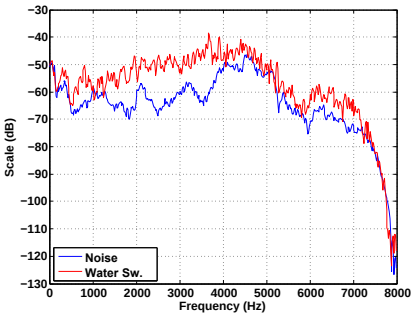


Figure 3.4: Frequency distribution of a water swallow vs. silence (noise).

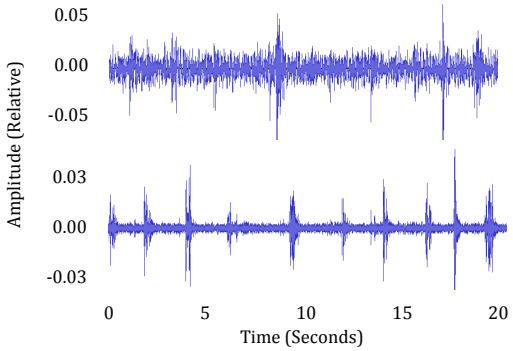


Figure 3.5: Post-processing of the audio signal corresponding with water can dramatically improve signal-to-noise ratio.

this section.

Figure 3.3 shows a spectrogram corresponding with an audio clip consisting of five water swallows acquired from the smartwatch. A spectrogram is a visual representation of the frequency spectrum over time, and is an ideal representation for extracting distinguishing features in many classification problems. A spectrogram is typically generated using a short-time Fourier transform (STFT) with a fixed window size, the squared magnitude of which yields the spectrogram. Fundamentally, a spectrogram allows easy identification of changes in the frequency spectrum of a signal, over time. Figure 3.4 shows a more detailed comparison between a brief interval of noise (1s) and a water swallow. Generally speaking, the data of interest is between 600 Hz and 1 kHz, as shown by the deviation between the signals at this time, and confirmed by the spectrogram shown in Figure 3.3. We conclude that analysis of this frequency range is critical for classification of liquid swallows. This observation is confirmed by Figure 3.5, which shows the transformation of an audio signal corresponding with ten swallows. The top waveform is

the original, while the bottom is the post-processed filter output in which noise is substantially reduced. This is achieved by band-pass filtering the audio data with cutoffs of 600 Hz and 1 kHz and a rolloff of 48 dB- meaning the amplitude decreases by 48 dB for each octave outside the filter threshold.

While the resulting signal clearly shows the swallows, marked by pronounced peaks, this technique is not very generalizable to other foods besides water, because the data is pre-processed. In the case of the frequency distribution of a one second window around the initial bite of a potato chip, compared to an equal period of chewing, the amplitude of the bite signal is greater from 600 Hz to 4 kHz. However, the pattern is not as distinctive as for liquids, and may certainly vary between individuals with different eating styles. Therefore, a simplistic filter-approach may not be sufficient for foods with less uniformity. More significantly, removing a frequency band to simplify the recognition of one food may also remove crucial information necessary for identifying another, in systems which attempt to classify between very different food types. Therefore, a more generalizable approach is described in the next subsection.

3.3.2 Generalizable Feature Extraction

Detection of eating habits differs significantly from that of liquid consumption, as the smartwatch will not necessarily be near the throat during a swallow. When an individual is drinking water, the swallows happen almost immediately after each sip. However, chewing food takes a significant amount of time. Typically, the smartwatch would be brought toward the mouth during the first bite, after which it would be lowered oncemore during the chewing process. Once the individual swallows the food, it is difficult to predict the location of the microphone. Therefore, in these cases we attempt to identify when an individual bites into a food item rather than chewing. The smartwatch platform is particularly well suited for this application because the microphone will be nearest to the sound source during the times at which the signal is of interest. The proposed model must be flexible to identify biting and swallowing for many different foods and drinks, between individuals with varying eating styles.

The Munich open Speech and Music Interpretation by Large Space Extraction toolkit, known as openSMILE [EWS10], is a feature extraction tool intended for producing large

audio feature sets. This tool is capable of various audio signal processing operations such as applying window functions, FFT, FIR filterbanks, autocorrelation, and cepstrum. In addition to these techniques, openSMILE is capable of extracting various speech related features and statistical features. A partial list of extracted features is shown in Table 3.1 and 3.2, respectively. More "low-level" audio-based features include frame energy, intensity, auditory spectra, zero crossing rate, and voice quality. Therefore, the capabilities of this tool are significantly more extensive than that of the spectrogram based approach described earlier, which relied only on statistical features from time-frequency decomposition. After data is collected from a variety of subjects eating several foods, feature selection tools can be used to identify strong features that are accurate predictors of swallows and bites for various foods, while reducing the dimensionality by eliminating redundant or weakly correlated features.

A microphone on a Smartwatch can either constantly record data, or be configured to record audio based on motion-based triggers indicative of eating-related gestures, in order to save battery life. The recorded audio is stored on a buffer in Smartwatch memory with storage for 4096 samples, corresponding with 0.25 seconds of data. Once the buffer is full, features are extracted using openSMILE (elaborated upon in subsequent sections), and the audio clip is classified divided into several distinct categories corresponding with the various foods the system has been trained to detect. A counter is incremented corresponding with the food type detected, which is necessary for long-term record keeping. In the event that eating behavior is detected, subsequent detection is disabled for a period of two seconds to prevent duplicate records caused by the same event. The algorithm is presented in Figure 1, with $\beta = 4096$ samples and $\tau = 2$ seconds.

To minimize the overlap between neighboring segments for performance reasons, the last 50ms of buffer data are cleared after each classification activity, and classification resumes when the buffer is full once again (not shown).

3.4 Experimental Procedure

3.4.1 Data Collection for Recognition

A total of ten subjects were used for data collection, with ages ranging from 22 to 35 in order to develop a model for identifying swallows. The subjects included 8 males and two females. Each subject was asked to eat the following foods, in order: three apple slices with at least two bites per slice, one 8 oz. glass of room-temperature water, and one bag of potato chips. The moments at which the food was bitten into (or swallowed as in the case of the water) were manually annotated by the subject, though these events were clearly audible on the resulting waveform. The hand on which the smartwatch was worn was used to pick up the food items and water, which happened to be the left hand for all subjects.

Data collection took place in a laboratory environment which had a minimal level of background noise including talking and doors opening, most of which is barely audible in the recording. However, pre-recorded background noise from a public shopping square was combined with the original data, to produce clips that more accurately reflect a real-world use case. It was assumed that the background noise should be quieter than the original waveforms because in our experiments, the watch was inches away from the mouth at the time of the extracted audio clips. Regardless of the food or activity type, each sample was exactly 0.25 seconds in length, and the peak of the wave amplitude was not necessary centered in the window. In some cases, such as during the biting of an apple, one quarter of a second was not sufficient to capture the entire bite. Therefore, the relevant information was partially truncated. Subjects were then asked to read a brief passage from a Wikipedia article, with no particular instruction about the rate at which they should read. The data was then automatically split into 0.25 second audio fragments using an audio processing program. Therefore, some samples were collected between phrases, and were relatively silent. Other fragments had periods of silence as well as vocalizations.

In order to evaluate if the classifier can distinguish between background noise and other classes of data, we added a separate "noise" class that we present in our classification results. However, the background noise used was relatively uniform: a 50 second clip of

cafeteria noise consisting of movement, background chatter, and silverware noise. The clip was divided into 50 quarter-second samples. The environment was busy, and the noises were quite pronounced in comparison to the relatively quiet sounds associated with the other classes.

3.4.2 Smartwatch Feedback: A Survey

Before the system development phase, we had several important questions about how individuals feel about smartwatches. As described previously, a wearable device must have both high accuracy, and high rates of user adherence for the subject to reach his or her intended goals. Furthermore, we proposed several questions about which hand a subject prefers to wear a watch. For example, our experimental evaluation requires that subjects wear a watch on the same hand with which they typically eat food such as chips or raise a glass of water. Though preliminary results suggest that data from individuals who pick up food with the hand on which they do not wear the watch can still be useful for classification, a thorough evaluation is left to a future work.

An online survey was conducted with a total of 221 responses in which various questions were posed with respect to how individuals feel about wearing a smartwatch. The participants in the study were anonymous, but represented a diverse set of ages, cultures, and genders. The study was originally conducted on January 28th for an internal data collection on smartwatch usage applied to the domain of medication adherence, but we found the majority of the questions were also applicable to food-intake monitoring as most questions pertained to smartwatch usage in general. This survey consisted of a total of 9 questions.

A separate online survey, with a total of 55 subjects, was later conducted to specifically investigate the attitude of individuals towards wearables in various form-factors. Specifically, subjects were asked to rate their receptiveness to smartwatch-based systems, custom wrist worn devices such as FitBit, necklace-based wearables, and smart glasses. Survey results and discussion can be found in Section 6.

3.5 Results and Discussion

3.5.1 Audio Classification

Results for classification between apples, chips, water, speaking, and ambient noise are shown in Table 3.3 based on 50 unprocessed samples collected from each of these foods, using the Random Forest classifier with 6555 extracted features from each sample. The Random Forest classifier consisted of 100 trees, each constructed using 13 random features, and was validated using leave-one-subject-out cross validation. Classifiers are generally evaluated on the basis of precision, recall, and F-measure. These terms are defined in Equation 3.1, where t_p is the number of true positives, and f_p is the number of false positives. The weighted average precision, recall, and F-Measure from our experimental results were 94.7%, 94.4%, and 94.4% respectively. In this case, the weighted average refers to the accuracy of the classifier across all different food groups, weighted according to the number of samples in each group. The majority of classification errors were between apples and potato chips. It should also be noted that while the ambient noise was disambiguated from the other classes in every sample, the ambient noise data was all recorded at the same location and therefore quite similar. Therefore, further work must be done to validate the ability of the proposed algorithm to recognize eating in real-world environments.

$$\begin{aligned} Precision &= \frac{t_p}{t_p + f_p} \\ Recall &= \frac{t_p}{t_p + f_n} \\ F - Measure &= 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \end{aligned} \tag{3.1}$$

Several other classifiers were also evaluated, which many of which provided strong results using leave-one-subject-out cross-validation. The full comparison of classifiers is presented in Figure 3.6. Though the RandomForest classifier produced the best results, the SimpleLogistic technique produced comparable results. The J48 decision tree classifier also performed well, with a precision, recall, and F-measure of 91.56%, 91.6%, and 91.5% respectively.

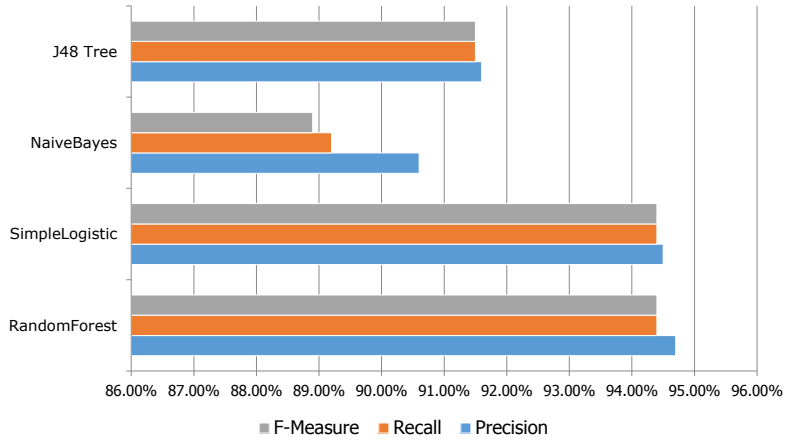


Figure 3.6: Precision, recall, and F-measure are common measures of classification accuracy.

3.5.2 Feature Extraction

From the 6555 extracted features, the Correlation Feature Selection (CFS) Subset Evaluator was used to evaluate 991,139 subsets of features. This is necessary to select the features best associated with the desired classifier outcomes. This subset evaluator considers both the individual predictive ability of features, as well as the redundancy between them, and found the merit of the best subset to be 0.948. The search was stale after 5 node expansions. In other words, the subset evaluator aggregates the best features linearly beginning with those that show the highest correlation, and terminates after five consecutive subsets show no improvement in classification accuracy.

The top ten features are listed in Table 6.1. The first feature is the skewness of the logarithmic signal energy, in which skewness is defined as the asymmetry of the variable in comparison with a normal probability distribution [PR11]. More formally, skewness is defined in below, where μ_i is the ith central moment about the mean.

$$\gamma_1 = \frac{\mu_3}{{\mu_2}^{3/2}} \tag{3.2}$$

For a probability density function f(x), the first moment about the mean is always zero (with s = 1), while the second moment is the variance. The third central moment is defined as skewness such that a distribution skewed to the right has a positive value, while one shifted towards the left has a negative skewness. The second most highly correlated

Table 3.1: Partial List of openSMILE Speech Features from [smi]

Speech-Related Features		
Signal Energy	Loudness	Mel/Bark/Octave Spectra
MFCC	PLP-CC	Pitch
Voice Quality	Formants	LPC
Line Spectral Pairs	Spectral Shape	CENS and CHROMA

Table 3.2: Partial List of openSMILE Statistical Features from [smi]

Speech-Related Features		
Means	Extremes	Moments
Segments	Samples	Peaks
Zero Crossings	Quadratic Regression	Percentiles
Duration	Onset	DCT Coefficient

feature is the mean peak distribution, which is defined as the mean distance between peaks for the logarithmic representation of the signal energy. The third feature is the number of non-zero values of the normalized log-energy signal.

Features 4-10, preceded by MFCC, are Mel-Frequency Cepstral Coefficients, which represent the spectral characteristics of the signal. A cepstrum is the result of the Inverse Fourier Transform of the logarithm of a signal spectrum. Mel-Frequency Cepstral Coefficients are based on the mel scale, which is a perpetual scale of pitches judged by listeners to be equidistant from one another [OS87]. The relationship between the frequency and mel scales is logarithmic, and can defined by the following formula (though other variations exist) [OS87]:

$$MEL(f) = 2595 \cdot log_{10}(1 + \frac{f}{700}) \tag{3.3}$$

However, the human ear can discern differences in frequency at low frequency ranges with a much higher resolution than at higher ranges, due to the physical properties of the cochlea. Therefore, a triangular Mel Filterbank is applied to the Discrete Fourier Transform of the original signal. Next, a dot product is computed between the filterbank and vector P(k), which yields N intermediary coefficients- one for each triangle window

function in the filterbank. Because humans do not perceive loudness on a linear scale, the logarithm is calculated for all N coefficients. Finally, the Discrete Cosine Transform (DCT) of the log powers is applied in order to decorrelate the energies of the overlapping filterbank energies. The resulting coefficients are used to extract statistical features as shown in Table 6.1.

3.5.3 Battery Life Implications

To realize the intended goal of minimizing burden, it is desirable for wearable devices to remain powered for weeks, or months, without interruption. Required nightly charging can be considered a burden to the user, which is undesirable because high user burden is typically associated with low compliance. Furthermore, energy-intensive applications can drain the battery completely, long before the user has an opportunity to recharge the device. Subsequently, the user will either uninstall the application or be unable to make proper use of it. Therefore, many activity monitoring devices have carefully factored power-efficiency into their design. Examples include the Misfit Shine activity monitor claims a battery life of four months [misa]. Other wearables devices such as the Jawbone UP24 claim their devices can sustain seven days of continuous use [Jaw].

Power-efficiency is a matter of particular concern in audio signal-processing applications such as the nutrition monitoring approach described in this chapter. Audio signal processing typically requires that the signal be sampled at the Nyquist frequency, which is rather high compared to approaches that rely on inertial sensors such as accelerometers and gyroscopes. The Samsung Galaxy Gear has a 315 mAh capacity battery, which is significantly smaller than that of most mobile phones. In this section, we briefly describe our evaluations of the battery life implications of recording audio using the Samsung Galaxy Gear.

We evaluated the battery life of the smartwatch in three different use cases. In the first case, the screen of the phone was off and the watch was idle and unused. In the second case, the watch was idle but the screen was on. In the third case, the screen was on and the watch was recording audio at the same rate (96kbps) as required for our audio analysis algorithms. Therefore, it can be inferred that the overhead of audio recording is the difference between the screen on, and the screen on while recording. Our results

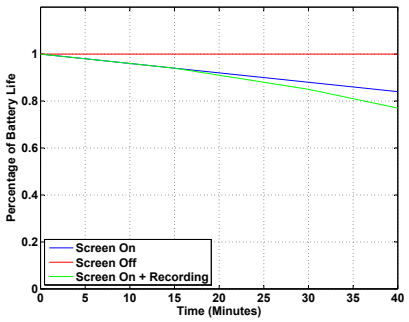


Figure 3.7: A graph of battery life for three different use cases is shown above.

are shown in Figure 3.7. From these results, we can draw several conclusions. First, it is evident that the Smartwatch in a static mode with no computation and the screen off, consumes very little energy. Secondly, the overhead of recording audio is significant, and has a substantial effect on battery life. As the graph shows, an hour of recording audio with the screen on will consume 38% of the battery life, which amounts to approximately 119.7 mAh. Therefore, it can be assumed that the audio recording functionality consumes 10% of the watch's battery life per hour, as a rough approximation. Figure 3.7 shows that the power dissipation of the screen is significantly larger than that of the audio recording and processing. Nevertheless, for long-term applications, the energy overhead of consistent audio recording may be prohibitive.

Algorithm 1: Simplified Classification Scheme

```
RecordAudio(Buffer);
if Buffer.Utilization = β then
    d = Buffer[1:β];
    f = ExtractFeatures(d);
    s = {Water, Talk, Apple, Chips, Other};
    c = Classify(f, s);
Counter_c++;
if c ≠ Other then
    PauseRecording(τ)
```

Table 3.3: Audio: Confusion Matrix (Random Forest)

	Predicted Class					
True Class	Apple	Chips	Noise	Water	Talk	**Recall**
Apple	40	9	0	1	0	80%
Chips	3	47	0	0	0	94%
Noise	0	0	50	0	0	100%
Water	0	1	0	49	0	98%
Talk	0	0	0	0	50	100%
Precision	93.0%	82.5%	100%	98%	100%	

Table 3.4: Partial List of Selected Features

Rank	Feature Name
1	Log Energy: Skewness
2	Log Energy: Mean Distance Between Peaks
3	Log Energy: Zero Crossings
4	Mel-Freq: Simple Moving Average[0] Quartile 3
5	Mel-Freq: Simple Moving Average[0] Mean Distance Between Peaks
6	Mel-Freq: Simple Moving Average[0] Zero Crossings
7	Mel-Freq: Simple Moving Average[1] Quartile 2
8	Mel-Freq: Simple Moving Average[1] Mean Distance Between Peaks
9	Mel-Freq: Simple Moving Average[1] Arithmetic Mean of Peaks
10	Mel-Freq: Simple Moving Average[1] Arithmetic Mean

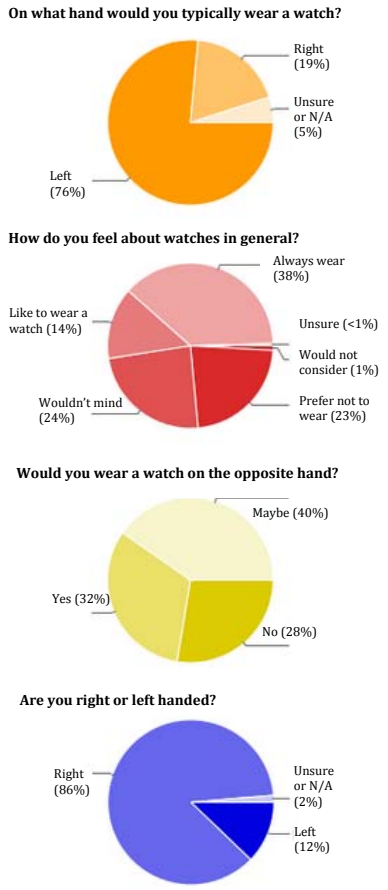


Figure 3.8: Partial survey results are shown above.

3.5.4 Smartwatch Feedback: A Survey

Figure 4.10 provides several of the most pertinent questions from the survey. From the total sample of 221 respondents, 86% claimed to be right handed, 12% left-handed, and the remaining responded that they were 'unsure' or the question was 'not applicable'.

In the following question, a total of 76% of respondents stated that they generally would wear a watch on their left hand, with an additional 19% who preferred to wear the watch on their right hand. The remaining 5% of those surveyed expressed no preference.

The next question asked respondents how they felt about wearing watches in general. Most individuals stated that they always wear a watch (38%). However, 23% claimed that they preferred not to wear a watch, 24% stated that they would not mind, and 14% stated that they like to wear a watch. Only 1% of individuals claimed that they would not consider wearing a watch. The next survey question revealed that those who drank water out of a glass would use their primary hand to lift the cup for their mouth (69%),

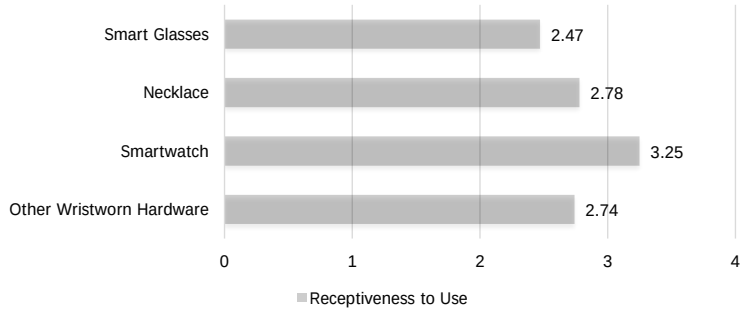


Figure 3.9: Survey results for the smartwatch-based scheme.

rather than the secondary hand on which the watch is worn (20%) with a remaining 10% claiming to be unsure. The final question asked respondents if they would be willing to wear a watch on the opposite hand to which they are accustomed. 40% of respondents answered 'maybe', 32% answering 'yes', and 28% answering 'no'.

These results are generally promising: almost no individuals expressed an adamant refusal to wear a watch. Furthermore, results suggest that most subjects show some flexibility about which hand they wish to wear a watch. Consider foods that require both hands to be raised towards the mouth, such as large sandwiches or hamburgers. In such cases, the eating can be detected regardless of which hand the subject prefers to wear the watch upon. This is the case because during the initial bite, the watch will be close to the mouth and the microphone can detect the pertinent signals. However, failing to use the hand on which the watch is worn to raise a glass of water or eat potato chips may pose a challenge to detection, as the source is not as close to the microphone, and it is possible that the signal-to-noise ratio may be lower. The feasibility of detecting the ingestion of foods consumed with the secondary hand should be explored in future work.

It appears that enough individuals are willing to change which hand they wear their watch, to make detection of most eating habits possible if the algorithm settings are customized to their personal habits. However, generally speaking, the results suggest the importance of an adaptive algorithm that can be used to detect eating habits regardless of the hand on which the device is worn. This can potentially be achieved by detecting the distance between the watch and the audio source, and performing amplification and filtering accordingly. This will be explored in future works.

To evaluate the receptiveness of the public to the smartwatch platform, we conducted

a separate survey to specifically investigate the attitude of individuals towards wearables in various form-factors. A total of 55 subjects participated in the online survey, of which 45.5% were male and 50.9% were female, and 3.6% who did not identify. 25.5% of subjects were 17 and younger, 27.3% ranged from 18 to 23, 27.3% were from 24 to 3, 12.7% were from 30 to 40, and 7.2% were over 40 years of age.

Subjects were asked to rate their willingness to wear health-monitoring wearable devices in four forms: glasses (such as Google Glass), smartwatches (such as the Galaxy Gear), custom wrist-worn hardware (such as FitBit), and necklaces (such as WearSens [KAL15a]). The scale ranged from 1, "not at all interested", to 5, "I would be completely comfortable wearing it". The results can be found below, in Figure 3.9. As the data suggests, "Smart Glasses" was the least favorable option, with an average score of 2.47. The "Necklace" and "Other Wristworn Hardware" options scored similarly, at 2.78 and 2.74, respectively. The highest score was associated with the smartwatch, with a rating of 3.25 out of 5.

With respect to the number of individuals who assigned a rating of 5, the highest possible score, the smartwatch was also the favorite. 25.5% of individuals assigning the smartwatch a rating of 5, compared to 9.1% with glasses, 10.9% for the necklace, and 10.9% for the "Other Wearables" option.

3.6 Conclusion

This chapter presents a novel approach to detecting ingestion of foods and liquids, using a smartwatch for identification of bites and swallows from acoustic signals. We also present a survey of users about smartwatch usage which confirms that a substantial portion of individuals would be willing to wear a watch on the hand with which they primarily eat. Future works will attempt to analyze eating behavior from the secondary hand, and explore the integration of audio-based detection of eating with inertial sensors for gesture recognition.

CHAPTER 4

Gesture Recognition

Poor adherence to prescription medication can compromise treatment effectiveness and cost the billions of dollars in unnecessary health care expenses. Though various interventions have been proposed for estimating adherence rates, few have been shown to be effective. Digital systems are capable of estimating adherence without extensive user involvement and can potentially provide higher accuracy with lower user burden than manual methods. In this chapter, we propose a smartwatch-based system for detecting several motions that may be predictors of medication adherence, using built-in triaxial accelerometers and gyroscopes. The efficacy of the proposed technique is confirmed through a survey of medication ingestion habits and experimental results on movement classification.

4.1 Introduction

It is well established that poor adherence to prescription medication can limit the benefits of medical care and compromise assessments of treatment effectiveness [MGH02]. Poor adherence is associated with increased hospital readmissions, medical complications, and even death [GB11]. It has been estimated that lack of adherence causes approximately 125,000 deaths in the United States, and costs the health care system been $100 and $289 billion pear year [VGJ12].

A significant body of research has been conducted to improve adherence to prescription medications through various interventions. These techniques vary tremendously from reminder-based systems, simplified pill packaging, positive reinforcement, financial incentives, and counseling. However, these systems typically suffer from high complexity, user burden, and inaccurate estimations of adherence [HHA]. One survey of major in-

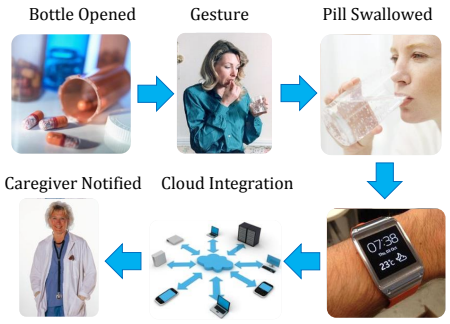


Figure 4.1: Illustration of methods in which a SmartWatch or similar wrist-worn device can be employed to detect medication intake.

terventions concluded that less than half of evaluated interventions were associated with statistically significant increases in adherence [BAK96].

In recent years, a greater emphasis has been placed on the role of technology in detecting non-adherence to medications. Because patient behavior can be monitored passively, user burden is potentially less than other methods that rely on patient record keeping, phone calls, and self-reporting. Furthermore, these digital systems have a potential to provide a better adherence assessment than self-reporting. However, these digital system suffer from several substantial limitations. Though they employ sensors to perform activity recognition, it is not always possible to accurately estimate adherence by recognizing a single action such as opening a pill bottle, or removing a capsule. For example, an individual may remove a pill from a medicine bottle, receive a phone call immediately thereafter, and neglect to return to swallow the pill. These factors suggest the need for systems capable of identifying multiple motions or activities associated with medication adherence, rather than relying on a single predictor.

Recently, smartwatches have become widely available on the commercial market. From a user-acceptance standpoint, these systems have a clear advantage over other proposed solutions based on custom hardware such as the wrist-worn accelerometry proposed by Chen et al. in [CKJ14] or audio-based ingestion monitoring systems proposed by Sazonov et al. and Amft et al. in [SMS10][AKT09]. Clearly, the multitude of sensors available on the smartwatch platform, wireless connectivity, as well as the comfort and social acceptance of the form-factor warrant further study into their potential applications in the medical domain. For example, several use cases of the smartwatch platform

are shown in Figure 4.1; the watch can be used as an end-to-end system for characterizing medication adherence by detecting gestures associated with ingestion and capsule removal, relaying this information to caregivers through web services.

In this chapter, we propose a system that can detect several motions associated with medication adherence using a custom Android application running on a Samsung smartwatch. The activities that are detected are shown in Figure 4.2. Using a tri-axial accelerometer and gyroscope, we can determine when a bottle is opened and a pill is retrieved. Furthermore, the proposed system can be used with any standard twist-cap prescription bottle, without requiring that each bottle to be equipped with sensors and wireless connectivity as in the case of the Vitality Glowcap [vit14].

Many other works describe various approaches to classifying motion using accelerometers and gyroscopes [ABM10][TCS08]. However, there are several novelties to our particular approach. First, we propose a method for tracking medication adherence using a commercial hardware device, rather than cumbersome custom hardware solutions that have limited applicability in real-world environments. Second, we are able to detect an extremely subtle wrist motion that is significantly more challenging to identify than the fitness-related activities that are emphasized in other works, such as walking, running, and climbing stairs. This is achieved by increasing the recall of the first-stage of the algorithm at the expense of precision, and filtering out the false-positives in the second stage.

Furthermore, we achieve high classification accuracy of the wrist motion associated with opening a pill bottle, using three simple features from each axis of the accelerometer. The majority of other works related to activity recognition extract hundreds of mathematical features from each axis, and perform computationally complex feature selection and classification. These techniques are more burdensome from the perspective of real-time implementation, require more processing power, and can significantly impact battery life as a result of their complexity. Lastly, our classification algorithm runs in real-time on a commercial smartwatch device, while many other works on activity recognition simply use the hardware for signal acquisition, and perform classification offline.

Figure 4.2: (A) The wrist motion necessary to twist the bottle cap open is detected using a tri-axial accelerometer. (B) The act of turning the palm upward to pour medicine from the bottle is detected using a gyroscope.

4.2 Related Work

4.2.1 Mobile-Phone Solutions

Several SmartPhone applications such as MyMedSchedule, MyMeds, and RxmindMe, provide advanced functionality for medication reminders. These applications issue reminders, allow users to manually enter their dosage information, and record when they have taken their medication [DHA13]. In [SM06], Sterns et al. mounted a pill bottle onto a personal digital assistant running the RxmindMe software, and successfully trained elderly subjects with an average age of 72 to operate the software used to monitor adherence. This work suggests that users from a variety of age groups and backgrounds have the ability and motivation to use electronic monitoring devices if given adequate training.

4.2.2 Hardware Approaches

This proposed work is an extension of our prior work described in [KS15b]. However, other hardware approaches have been proposed in recent literature. The work described in [HHA] describes a portable, wireless-enabled pillbox suitable for elderly and those suffering from dementia. Similar approaches for electronic detection and smart pill boxes have also been proposed [VRR13]. These devices generally suffer from the same shortcom-

ing: they cannot determine if the medication is ingested or simply removed and discarded [BFK05]. In another work, Valin et al. successfully identified medication adherence using a series of images and associated image processing algorithms [VMS06]. Very recent work by Chen et al. in [CKJ14] describes a system in which inertial sensors worn on the wrist are used for detection of gestures associated with medical intake, based on a Dynamic Time Warping (DTW) algorithm. In [KAL15b] and [KAL15a], the WearSens necklace is used to detect different kinds of swallows, including those that may be associated with medication capsules and tablets.

The Vitality Glowcap is a wireless-enabled pill bottle that can report when medication is removed [vit14] using a cellular network, while a recent product from Amiko [ami14] is one of the few systems that can monitor the ingestion of medication directly, based on a smart-inhaler technology. Other notable technologies include the Smart Blister from Information Mediary Corporation [bli15], which can detect when medication is removed from a blister-packet.

Several digital systems have been proposed for evaluation of swallow disorders and monitoring eating habits using audio processing techniques. These techniques give credence to future smartwatch-based systems which can combine audio-based ingestion monitoring with inertial detection of user activity. Analyzing wave shape in the time domain or feature extraction and machine learning by Okazaki et al. [TOY10] resulted in an 86% swallow detection accuracy in an in-lab controlled environment. Similarly, the work featured in [NS11] by Nagae et al. distinguishes between swallowing, coughing, and vocalization using wavelet-transform analysis of audio data.

4.3 System Architecture

4.3.1 Hardware Description

The SmartWatch application is capable of predicting if a pill has been swallowed using the on-board inertial sensors available on the Android SmartPhone. The application runs as a background service: data is collected and processed even while the user is interacting with other applications on the watch.

The hardware platform used is the Samsung Galaxy Gear SmartWatch running Android 4.2.1. This phone features an 800 MHz ARM-based processor, 512 MB of RAM, and a 320x320 pixel 1.6 inch display. The device also supports transfer of data using the Bluetooth LE protocol, and can be configured to access the Internet using Bluetooth tethering with compatible Smartphones. Once the on-board algorithm detects that the medicine has been ingested, a web-service call is made to store the data in a database for access by caregivers. Though the sample rate of the on-board sensors can be configured, a rate of 16.66 Hz was determined to be sufficient for activity recognition through experimentation. Higher sample rates increase computation power and decrease battery life with no significant effect on accuracy.

4.3.2 Android Application

The Android application works as follows. First, samples are acquired from the accelerometer and gyroscope in the X, Y, and Z axis. These values are buffered in a vector-based data structure in memory, which is of a fixed size and operates in a first-in-first-out format. That is, the oldest sample is removed from the structure to make room for each new sample. After every new data point is acquired from the inertial sensors, the Detect-Motion function is called. This function implements the feature extraction, processing, and thresholding techniques described in Section IV. This function sets a flag when the accelerometer data suggests that a bottle has been opened, and when the gyroscope data suggests that the hand on which the watch is worn has been turned such that the palm faces upward. A timestamp accompanies each of these flags, which can then be used to detect the time interval between these two motions. If the time interval is less than the set threshold (T=30 seconds), the application reports that the medicine has been taken. Subsequently, a notification email can be transmitted, or a web service call can be made, based on the users individual requirements.

4.4 Algorithm Design

In this section, we describe the algorithms running on the Android Service, which predict if medication has been ingested based on the recognition of two activities: (1) The bottle

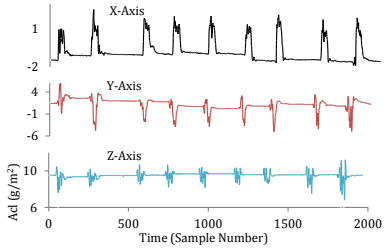


Figure 4.3: Accelerometer data from opening the pill bottle nine times.

being opened while the SmartWatch is worn on the wrist by detecting the twisting motion of the bottle cap, and (2) the wrist being rotated for the palm to face upwards, in order to pour medicine capsules into the secondary hand. All results are based on data acquired from tri-axial accelerometer and gyroscope samples acquired at 16 MHz. Figure 4.2 shows the actions the proposed system was designed to identify.

In the following formulas, we refer to the window size as β, and the set of original sensor data as D. $\bar{D}(j)$ refers to the transformed sensor data after being processed in Equation 4.1, $\acute{D}(k)$ after Equation 7.2, and $\tilde{D}(n)$ referring to the output after Equation 7.4. Symbols j, k, and n refer to individual data points in the first, second, and third phases of the algorithm respectively. The constant α refers to a predefined threshold for separating the different peaks.

4.4.1 Bottle Opening: Data Transformation

Figure 4.3 shows the waveforms acquired from the SmartWatch accelerometer for each axis which correspond with a bottle being opened nine times. Each bottle-opening event corresponds with a different peak. Successful identification of the event is dependent on analysis of the features of each peak in all three dimensions. Therefore, the data must be transformed to decouple the perturbations of the signal from the offset, and limit the effects of drift and noise. This new waveform, shown in Figure 4.5, provides a more objective representation of the features of a bottle opening event.

This signal transform is first achieved by generating a new waveform using a sliding-window average of the original data. The relevant equations for each axis are shown in Equation 4.1. It was determined that 70 is an appropriate value of β, as significantly smaller values are too sensitive to minor fluctuations.

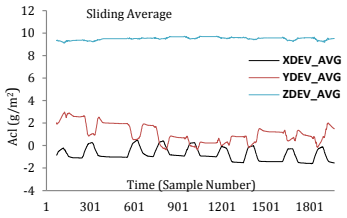


Figure 4.4: Data converted to a sliding window representation.

$$\begin{gathered} \forall D \in \{X, Y, Z\}, \\ \forall j \in D, \\ \bar{D}(j) = \frac{1}{\beta} \sum_{i=j-\beta}^{j} D(j) \end{gathered} \tag{4.1}$$

After the moving-average representation of the data is generated, each point is then assigned a numerical value with respect to the average value in the previous window. This essentially removes the offset from the data and combats the effect of drift, while preserving the critical features of the original waveform. This is shown in Equation 7.2.

$$\begin{gathered} \forall k \in D, \\ \acute{D}(k) = \left| D(k) - \bar{D}(k) \right| \end{gathered} \tag{4.2}$$

The next transformation simply separates the continuous data into different peaks separated by spans in which the data is zero, based on a simple thresholding technique. This allows different instances to be more easily identified. The relevant equation is shown in Equation 7.4, and the corresponding waveform (with additional smoothing) is shown in Figure 4.6. It was experimentally determined that an α value 0.5 g/m^2 of visually preserved the critical features of the waveform while removing noise during periods of inactivity.

$$\forall n \in \acute{D},$$

$$\tilde{D}(n) = \begin{cases} 0, & \acute{D}(n) < \alpha \\ \acute{D}(n), & \acute{D}(n) \geq \alpha \end{cases} \tag{4.3}$$

Subsequently, features from individual 'pulses' can be extracted, which each correspond with a different bottle opening episode. This is shown in Figure 4.7, which shows one individual pulse in the X axis. By performing a summation of each pulse, which is delimited by a value of zero as described in Equation 7.4 as a result of the thresholding technique, a distinguishing feature can be extracted from each axis. The width of the pulse, once again delimited by zero, is a secondary feature that can used to improve classification accuracy.

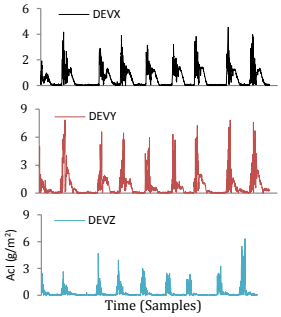


Figure 4.5: Phase (3) waveforms for processing smartwatch signals.

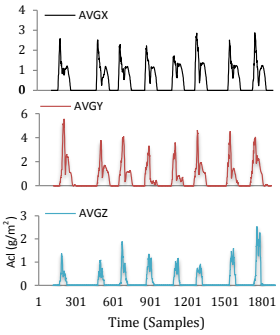


Figure 4.6: Phase (4) waveforms for processing smartwatch signals.

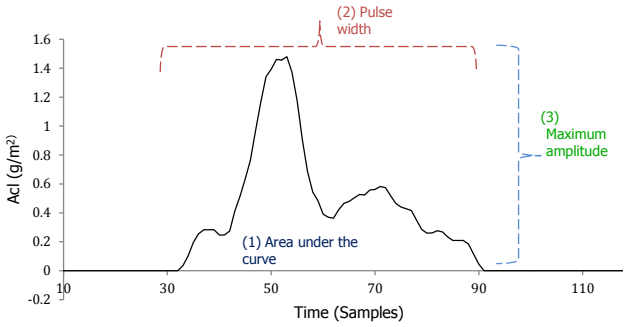


Figure 4.7: Output pulses from Phase (4).

4.4.2 Bottle Opening: Detection

Based on the previously collected features, we apply various constraints for the classification of each pulse, as shown in Equations 7.5. These constraints are formulated on the basis of their efficacy in detecting the twisting motion required to remove the bottle cap. Figure 4.8 shows the distribution of feature values such as pulse width for all three axes, and well as the area under the curve of each pulse, as users twisted the bottle cap during the initial phase of data collection. The observations that are made from the feature distribution associated with this activity are used to formulate the constraints for classifying an action as the opening of a bottle cap. Visually, it can be inferred that the data from the Y-axis of the accelerometer is very weakly coupled with the act of twisting the bottle. However, the standard deviation of the X and Z axis data appears to show significantly less variation.

As Equation 7.5 shows, the first requirement is that the standard deviation of indices of the first nonzero values of the accelerometer data in each axis to be less than three, to reduce the effects of noise and drift. The remaining constraints are the widths of the X, Y, and Z pulses, which correspond with the overall duration of the bottle cap opening event. The bounds on the integral of acceleration (velocity) constrain the intensity of the motion based on what is typical for the action. This is necessary to prevent motions of similar durations but varying intensity from being misclassified.

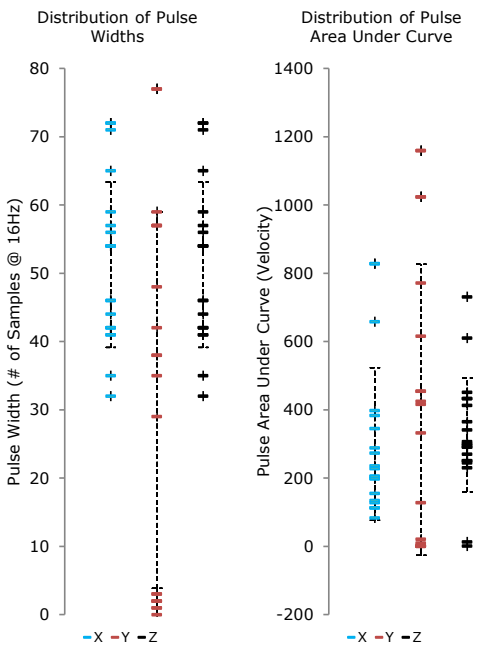


Figure 4.8: An analysis of the clustering patterns for different features.

$$\begin{aligned} 30 < WidthX < 75, 0 < WidthY < 80 \\ 10 < WidthZ < 80, 60 < SumX < 1800 \\ 0 < SumY < 1200, 20 < SumZ < 1600 \end{aligned} \tag{4.4}$$

Once it has been determined that the bottle has been opened with a high probability, the system makes a record of this event and begins detection of pill extraction. This is obtained using gyroscope MEMS sensors available on the SmartPhone, which are processed as described in the following section.

4.4.3 Medicine Removal: Data Transformation

In the case of most twist-cap medication bottles, it is not possible to reach inside to retrieve the medication. Typically, once the bottle is opened, it is turned upside down and a medication capsule is emptied on the secondary (non-dominant) hand. This requires that the individual turns their hand upside-down with their palm facing upwards for a brief period, as shown in Figure 4.2(B). If the SmartWatch is worn on the wrist of the secondary hand, this motion can be detected. Data is acquired from the SmartWatch's

built-in triaxial gyroscope at a rate of 16 Hz, which represent angular speed around the X, Y, and Z-axis in units of radians/second. The Android API provides output with built-in drift compensation algorithms, though raw data is also available.

The gyroscope data can be integrated along each axis to provide an estimation of rotation in a given unit of time. However, as in the case of the accelerometer processing used to estimate if the bottle cap is removed, the gyroscope data must be transformed for effective activity identification. However, the transformation is much simpler. Equation 7.6 shows the simple summation of the last β values acquired from the gyroscope. In this equation, x_n corresponds with the n_{th} sample of acquired data, and the same convention is used for the Y and Z axis. The chosen value of β is 12 samples, which corresponds with 750 ms of data at a 16 Hz sample rate. These values are selected based on the observation that most individuals will perform the hand motion in significantly under one second; longer sample rates would distort gyroscope data with extraneous movements and produce false positives.

$$\begin{aligned} &\forall Sample_i \in \{Buffer\}, \\ &x_{sum} = \sum_{k=i-\beta}^{i} x_k \end{aligned} \tag{4.5}$$

Because the required sample rate for inertial-based activity recognition schemes can be quite low (around 16 Hz in our case), a high computational complexity does not necessarily preclude an algorithm from practical real-time application when the collected data is in the range of several seconds. That being said, the complexity for this particular algorithm is linear. Each acquired data point is another value that must be summed to find the average value of a window. This point must then be subtracted from the window average to find the magnitude difference (Phase III). The number of these operations, along with those associated with the subsequent thresholding and peak detection, do not grow exponentially with the size of the dataset.

4.4.4 Removing the Medicine: Detection

Detecting that an individual has poured the medicine into his secondary hand is relatively simple, after the preprocessing shown in Equation 7.6. The detection of this movement does not imply that any medication was removed- simply that the palm was turned to face upward. Therefore, this is not a primary heuristic for medication adherence, and is used as a supplement to the bottle cap detection mechanism. The constraints on which this movement is detected are shown in Equation 7.7. First, some time interval ΔT must have elapsed since the last recorded event, to prevent duplicate records of the same event. The absolute value of the movement in the y and z directions must also be less than some arbitrary threshold, to ensure that random hand movements are not considered. Lastly, x_{sum}, the movement around the x axis over the last 12 samples (16 Hz) in radians/second, must be less than the threshold of -28, or greater than 28, depending on which arm the watch is worn. Experimentally, it was determined that lower threshold values could not differentiate relatively minor turns of the wrist to the full action of turning the palm upward that is required to pour medication from the bottle into the hand.

$$\begin{aligned} \Delta T &> 1s \\ |y_{sum}| < 5, |z_{sum}| &< 5 \\ x_{sum} : \begin{cases} < -28, & LeftHanded \\ > 28, & Right\ Handed \end{cases} \end{aligned} \tag{4.6}$$

4.5 Secondary Motions and Other Future Works

Figure 4.9 shows a illustration of different motions relating to medication ingestion. Starting from an initial condition, the watch being worn on either the primary or secondary hand, the bottle cap is twisted open and poured into either hand. The green markers denote actions that our algorithm is capable of detecting using the previously outlined techniques. The red markers are examples of motions that are not as pronounced. For example, if the watch is being worn on the primary hand, the secondary hand will move

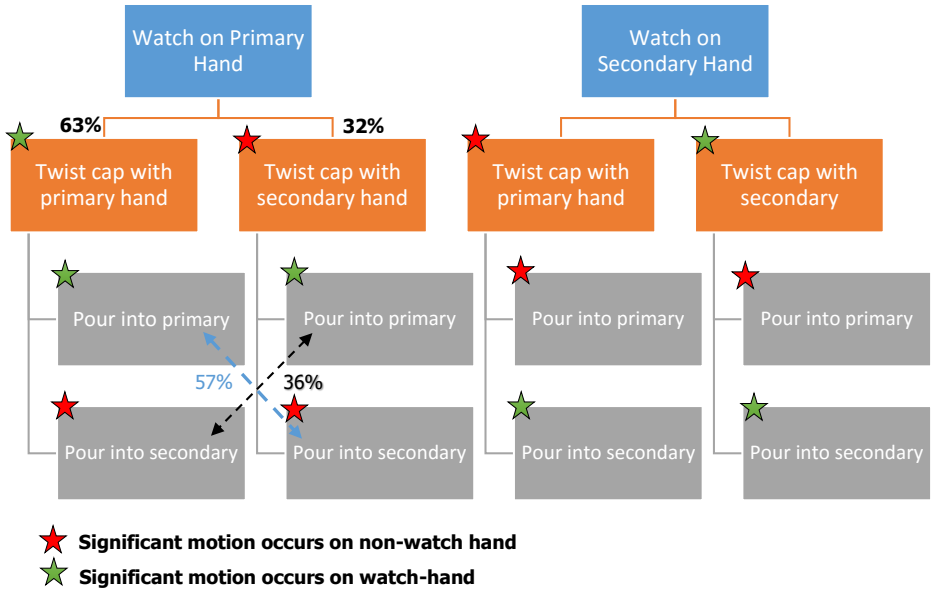


Figure 4.9: Different motions observed during data collection and reported by users in the survey.

very slightly if the primary hand is twisting the cap open. Furthermore, if the watch is worn on the primary hand and the medication is poured into the secondary, the watch-hand is tilted more subtly to allow the medication to slide out of the bottle. As the survey of medication ingestion habits has determined that either hand can be used, detection of these alternative motions warrants a closer look.

Another aspect of inertial sensing associated with medication intake is the act of raising the pill to the mouth, which can be done with either hand. However, the secondary hand will most likely not exhibit any identifying characteristics in this case. The detection of this action can be explored in future work to reduce the false positive rate, but will be less useful for reduction of false negatives as the absence of this motion does not necessarily suggest that the medication has not been taken.

4.6 Experimental Procedure

Training data was collected from five subjects between the ages of 21 and 25, all of which were right-handed. The subjects wore the watch on their left hand in their preferred configuration, and were asked to open the pill bottle using the hand on which the watch

is worn. The results were used to formulate the algorithm constraints, which were then tested on the remaining subjects.

4.6.1 Gesture Recognition

Twelve subjects were asked to perform several activities while wearing the SmartWatch including walking, opening a medicine bottle, and opening a bottle of water.

The data collection occurred in two separate sessions to increase the diversity in motion patterns. The medicine bottle used was a standard prescription variety containing empty gel capsules (Size 00). As in the case of most standard prescription bottles, opening the lid requires the application of downward pressure while twisting the cap in the counter-clockwise direction. However, the subjects used in the study were not given any instruction on how the bottle was to be opened, in order to avoid influencing activity patterns. After opening the pill cap, the subjects were asked to pause briefly for a period of three seconds, before pouring the medicine out of the bottle.

4.6.2 Online Survey of Habits

In order to design an appropriate activity recognition scheme, it is necessary to validate various assumptions about how people take their medication, as well as their opinion on smartwatch devices. An online survey was conducted with a total of 221 responses, in which various questions were posed with respect to how individuals feel about wearing a smartwatch, on what hand they would typically wear it, and how they retrieve and ingest a medication capsule. The participants in the study were anonymous, but represented a diverse set of ages, cultures, and genders. The survey results were used as a basis for algorithm design.

4.6.3 Observational Survey

To validate the results of the online survey, twenty subjects were asked to open a pill bottle and consume an empty gel capsule while being observed. No instruction was provided on how the medication should be taken, how the watch should be worn, or how the bottle should be opened. Thus, individuals were allowed to take the medication in

a relatively natural environment. This is necessary because anonymous online survey results can be error prone. Online surveys can be particularly challenging because it may be difficult for an individual to ascertain their medication habits without a pill bottle in front of them.

4.7 Results

4.7.1 Online Survey

From the survey based on responses from 221 individuals, 86% claimed to be right handed. A total of 76% of individuals claimed that they generally would wear a watch on their left hand, with an additional 19% who preferred to wear the watch on their right hand. The remaining 5% of those surveyed expressed no preference.

The next question in the survey asked subjects how they felt about watches in general. 72% of responses were positive, as 38% claimed they always wear a watch, 14% preferred wearing a watch, and 53% stated that they would not mind. Subjects were then asked to estimate what percentage of the time they would remove medicine from the bottle and not consume the pill within the next minute. 12% answered that this would occur occasionally, 6% often, and 1% always. 76% of individuals stated that this would happen very rarely.

Figure 4.10 shows other relevant survey questions. The first question reveals that though most individuals open a bottle by twisting the bottle with the primary hand, a significant percentage (32%) preferred to steady the bottle with their primary hand, and twist with the secondary hand. Therefore, the bottle cap would more frequently be twisted by the opposite hand on which the watch is worn. This is confirmed by another survey question, which established that only 11% of subjects opened the bottle by twisting the bottle base, rather than the cap.

The next question evaluated what happens after individuals open the pill bottle. As hypothesized, most individual's poured the medicine into the palm of their hand, as opposed to another surface such as a napkin or table. However, there was little homogeneity in responses, with 57% who stated that they would pour the medicine into

the hand that twisted the cap, and 36% that originally held the bottle.

Generally, the results suggest that some individuals will need to adapt their watch usage in order to recognize the motions suggested in this chapter. This can be partially mitigated by developing detection strategies for a broader range of motions and applying template matching, though this is left to a future work. The remaining survey results were promising, as only 28% of subjects claimed that they would not consider wearing a watch on the opposite hand of what they are generally accustomed, compared to 40% who claimed that they would consider it, and 32% who stated that they would be willing.

4.7.2 Observational Survey

Table 4.1: Observational Survey of medication ingestion habits

Hand used to twist bottle cap	
Dominant	15
Secondary	5
SmartWatch Placement	
Dominant	1
Secondary	19
Pill Extraction Method	
Pour into dominant hand	5
Pour into secondary hand	15
Other	0
ΔT Between Actions	
0-5 seconds	16
5-12 seconds	4
Which End is Twisted	
Cap	17
Bottle	3

Observational study results indicated that 75% of subjects used their dominant hand to twist the bottle cap in the observed study, compared to 63% in the online survey. Furthermore, 85% twisted the cap (rather than the bottle) in the observed study, com-

pared to 91% in the online poll. Lastly, 57% of individuals stated that they would pour the medicine into the hand that twisted the cap, compared to 75% who poured into the secondary hand (which is the dominant hand in 95% of cases). Though there are some discrepancies due to limited sample size, the online and observational study results are generally in accord with one another. Full survey results are shown in Table 4.1.

4.7.3 Motion Classification Results

The classification results are shown in Table 4.2 and 4.3. The results indicate that while accuracy of wrist rotation detection is good, the false-positive rate of pill cap opening detection is very high. This design tradeoff is necessary to ensure that nearly all real pill opening events are detected; false positives will be filtered out in the second stage of the algorithm. Table 4.2 shows that despite very low precision across categories, the recall for the action of 'medicine bottle opened' is very high. The remaining false positives are filtered out in the next stage of the algorithm shown in Table 4.3 in which the precision of the 'other' category, which comprises the other four listed actions, is 100%. Note that, because no traditional classifier was used for activity recognition, there is no cross-validation scheme to separate the test and training data. The algorithm was designed and tested on one subject, who later did not participate in the final data collection in order to avoid overfitting the data.

Table 4.2: Confusion Matrix using Accelerometer Data

	Predicted		
Actual	Med. Bottle Opened	Other	*Recall*
Med. bottle	21	3	87.5%
Raise Arm	14	6	30%
Walk	1	23	4.1%
Open door	14	10	41.6%
Water bottle	20	4	16.6%
Precision	30%	6.5%	

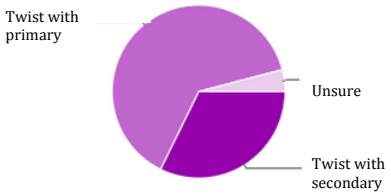
Table 4.3: Confusion Matrix using Gyroscope Data

	Predicted		
Actual	Palm Up	Other	*Recall*
Palm Up	24	0	100%
Raise Arm	2	22	91.6%
Walk	1	23	95.8%
Open door	2	22	91.6%
Water bottle	0	24	100%
Precision	82.7%	100%	

4.8 Conclusion

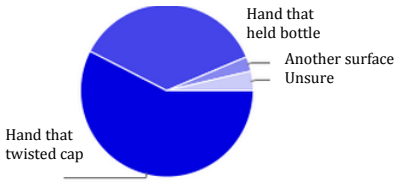
In this chapter, a survey was conducted to understand how individuals take their medications from standard-sized twist-cap pill bottles in a normal environment. The results suggest that it is possible to use the Smartwatch as a platform for detection of medication adherence for many individuals. Using the tri-axial accelerometer and gyroscope on the Samsung Smartwatch, we are able to detect (1) the act of twisting the cap of a medicine bottle open, and (2) the removal of a tablet or pill by pouring the pill into the palm of the hand. Though the proposed system imposes some restrictions on how subjects should remove the pill bottle for successful recognition, the system nevertheless has much less human involvement compared to manual record keeping or phone calls from nurses and other forms of adherence detection.

How would you open a typical twist-cap bottle?



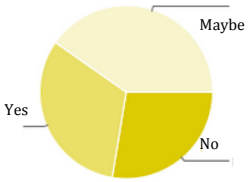
Steady bottle with primary hand, twisting with secondary.	32%
Steady bottle with secondary hand, twisting with primary.	63%
Unsure	4%

What would you typically do after removing the cap of the bottle?



Pour medication into same hand that twisted cap.	57%
Pour medication into hand that originally held the bottle.	36%
Pour medication onto another surface (napkin, table, etc).	3%
Unsure or N/A	4%

Would you be willing to wear a watch or similar wrist-worn device on the opposite hand to which you are accustomed?



Maybe	40%
Yes	32%
No	28%

In what way would you open a medicine bottle?

Hold the bottle steady while twisting the cap.	91%
Hold the cap steady while twisting the bottle.	11%
Unsure or N/A	4%

Figure 4.10: Partial online survey results are shown above.

CHAPTER 5

Classification and Segmentation for Wearable Activity-Monitoring Applications

Detecting short-duration events from continuous sensor signals is a significant challenge in the domain of wearable devices and health monitoring systems. Time-series segmentation refers to the challenge of subdividing a continuous stream of data into discrete windows, which can be individually processed using statistical classifiers or other algorithms. In this chapter, we propose an algorithm for segmenting time-series signals and detecting short-duration data in the domain of lightweight embedded systems with real-time constraints. First, we demonstrate an approach for signal segmentation using a simple binary classifier. Next, we show how a novel two-stage classification algorithm can reduce computational overhead compared to a single-stage approach. Our proposed scheme is benchmarked using an audio-based nutrition-monitoring case-study..

5.1 Introduction

This chapter addresses the issue of efficient *time-series segmentation and classification*: an important topic in real-time embedded systems such as those used in wearable health and fitness monitoring devices which collect and process continuous sensor data. The challenge of segmentation that we discuss in this chapter is particularly relevant in applications which detect short-duration events from relatively long data sets, such as brief moments of coughing in a day-long audio recording.

To understand why time-series segmentation is important requires a basic understanding of sensor systems and statistical classifiers. In this section, we begin with some of these preliminaries.

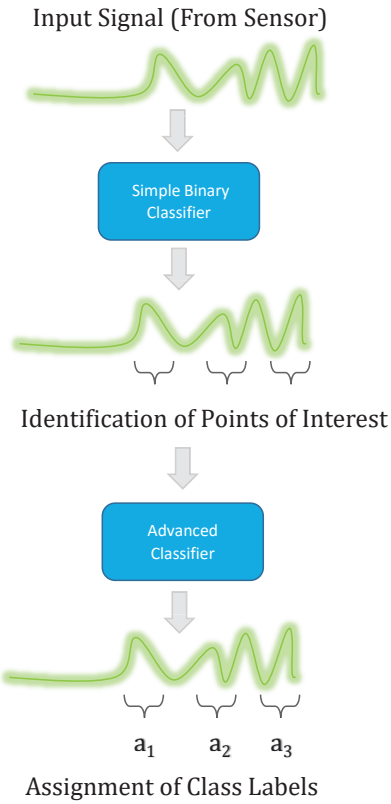


Figure 5.1: System architecture of the proposed two-stage segmentation and classification scheme.

5.1.1 Overview of Real-Time Wearable Sensor Systems

Real-time wearable sensor systems have become very popular in recent years, and address a variety of health needs ranging from fitness, health monitoring, and object tracking. Examples of such systems are everywhere, and can range from a simple FitBit that monitors physical activity, to more complex examples such as the wearable diet-monitoring devices proposed by Kalantarian et al. in [KS15b] and [KAS14a]. These devices acquire signals from sensors such as accelerometers, gyroscopes, and microphones, as individuals go about their normal daily activities. Subsequently, various algorithms are used to identify actions of interest from the continuous stream of real-time data.

Some of the often conflicting challenges associated with processing these signals is to improve the rate and accuracy of event detection, while optimizing computational and energy-related costs. This is a matter of particular concern in wearable devices such as the Misfit [misa], Jawbone [Jaw] and other similar devices which are often powered by small coincell batteries and must last hours or days on a single charge.

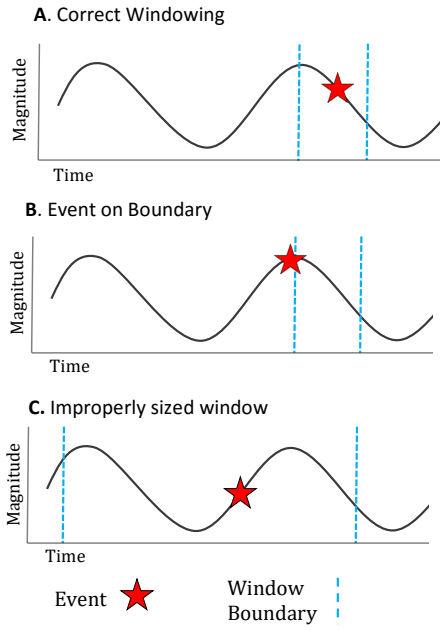


Figure 5.2: Potential problems when attempting to identify an action from a large discrete signal.

5.1.2 Classifiers

Various statistical classification techniques have been proposed to identify activities of interest from time-series sensor data. One example of such a system was proposed by Alshurafa et al. in [AXL14], in which the authors extract statistical features from sensor data, and determine the activity being performed using machine learning tools. However, in the case of a 24 hour stream of data it is not practical to assign a single class label to the whole dataset. Expectedly, the subject may be running for one hour, walking for one hour, and sitting for the rest of the day. Therefore, the data must be *segmented* into fixed or varying-length windows, each of which is assigned a separate class label.

A classifier may require dozens of features to accurately distinguish Activity A from Activity B. However, this approach is not always practical in embedded systems as there is a computational burden associated with deriving features from sensor data in real-time. Equation 6.5 shows the amount of time (Cost) to extract features from a continuous signal is a function of the number of features, N, as well as the cost of each feature, i, which is f_i, and the number of Windows that the signal has been divided into, W.

$$Cost = W \cdot \sum_{i=1}^{N} f_i \tag{5.1}$$

As this equation suggests, a large feature set may be prohibitive for embedded hardware applications with low power budgets. Furthermore, classifiers with large input-sets may be slower to produce an output than simpler approaches with smaller sets of input features. However, larger feature sets may also lead to higher classification accuracy in some cases. This is particularly true for classifiers with many class labels. These design tradeoffs are the basis for the development of our two-stage classification and segmentation algorithm.

5.1.3 Time-Series Segmentation

Time-series segmentation refers to the challenge of subdividing a continuous signal into separate windows. A very simple example of segmentation would be dividing handwritten text into separate letters for character recognition. In practice, the challenge of efficient segmentation is applicable to numerous other domains. In this subsection, we describe several ways in which an arbitrary time-series signal can be incorrectly segmented. Figure 7.4-A shows a correctly windowed signal- the bounds of the window are set such that the window holistically contains the activity that we aim to recognize. In Figure 7.4-C, this is not the case; the window bounds are much greater than the event to be recognized. An example of such a case would be attempting to detect running in a 5-hour window of data: most people would not run for this entire period of time, which suggests that assigning a single class-label to this window is not meaningful. Furthermore, some approaches average together statistical features of a window such as the works of Kalantarian et al. [KAP14] and Alshurafa et al. [AKP14]. Therefore, even small portions of the window that are not associated with the unique event could homogenize the distinguishing features of the activity, severely reducing classification accuracy.

Another example of incorrect windowing is shown in Figure 7.4-B. In this case, the window has been set such that it bisects the event of interest in half. In this case, there is no single window that holistically represents the activity, which may lead to a reduction in classifier performance. While overlapping windows is a potential workaround, increasing

the W value in Equation 6.5 will increase the total cost of the algorithm.

5.1.4 A Novel Segmentation and Classifical Scheme

As we have discussed, a classifier that relies on a large input feature set may have higher classification accuracy than a simple classifier that uses two or three features. However, a less sophisticated classifier may result in faster performance. More specifically, the classifier with the smaller feature set or simpler operating characteristics could produce an output class label in less time than a more complex classifier with a larger feature set. In this chapter, we propose an algorithm for segmenting a time-series signal using an adaptive classifier scheme to reduce the computational load on an embedded device. Our algorithm consists of two different phases:

In the first stage, we use a simple binary classifier C_{smp} to accurately segment the time-series signal into varying window sizes. That is, our algorithm can assign a class label to a given window $W \subset \{relevant,\ not\ relevant\}$. Through an interative process, we can select window boundaries such that the data within may be relevant. As the classifier does not require a large input feature set, computational overhead is reduced compared to a more traditional approach.

In the second stage, upon locating a window that may be relevant to the current analysis, we use a complex multiclass classifier C_{adv}, to assign a more specific label to the window. This computationally intensive process is minimized, as the analysis is conducted only for those windows that have not be filtered out by the first stage of the algorithm.

The primary contributions of our chapter are:

- A novel approach for detecting short-duration events from a continuous time-series signal.
- An efficient technique for time-series segmentation using an adaptive window-sizing technique.

This chapter is organized as follows. In Section 5.2, we present several related chapters that address the challenge of segmenting time-series data. In Section 5.3, we describe

our proposed algorithms in greater depth. In Section 7.5, we describe our experimental methods. Results and a discussion are provided in Section 7.6, followed by concluding remarks in Section 7.7.

5.2 Related Work

A comprehensive survey of time-series segmentation was provided by Keogh et al. in [KCH04]. The emphasis of existing work on segmentation is to develop a compact representation of a large signal, with applications in data mining and compression. The three primary methods in literature are sliding window approaches, similar to the baseline used in this chapter, and recursive top-down or bottom-up techniques that are either partition or merge signals until a stopping criteria is met. In [MOS12], Mithal et al. propose a technique for land-cover identification from EVI (Enhanced Vegetation Index) time-series data using a model difference segmentation score. However, their approach is applied specifically to satellite imagery and is not a power-aware technique. In [XZK12], Zu et al. propose an adaptive algorithm for online time-series segmentation. The authors approach the challenge from the perspective of representing complex data using a series of possible candidate functions as a form of data compression.

Lovri et al. provide a comprehensive overview of various time-series segmentation techniques in [LMS]. The primary difference between our techniques and those discussed in [LMS] is our application of a simple pre-trained binary classifier to identify relevant windows of data, rather than a comprehensive approach using a classifier with an expensive input feature set applied to all data windows. Adaptive window sizing has been explored in several other works. For example, in [OK92], Okutomi et al. present a signal matching algorithm that can adapt window size based on the uncertainty of the disparity between two signals. The issue of adaptive window sizing for image filtering using local polynomial approximation is presented by Katkovnik et al. in [KEA02]. In [KJ00], Klinkenberg et al. propose an approach for selection of window size to minimize generalization error based on leave-one-out estimation error.

Lastly, our experimental use-case of audio-based detection of eating habits has been explored in a variety of works [SSL08][RAZ14]. In [RAZ14], Rahman et al. present

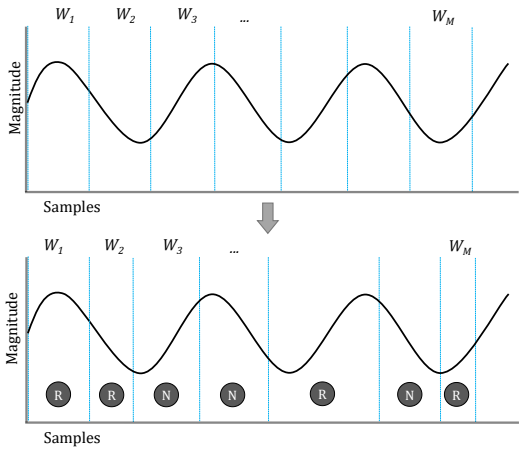


Figure 5.3: First stage of the algorithm, which adjusts the window boundaries and assigns labels of *Relevant* or *Not Relevant* to each window.

BodyBeat: a robust system for detecting human sounds. A similar work is presented by Yatani et al. in [YT12]. The BodyScope device can distinguish between twelve kinds of activities including eating, drinking, laughing, and coughing. Other approaches include that of Amft et al. in [AKT09], in which acoustic signals are used for bite-weight prediction, or [AT06] in which muscle activation and sound are used together to detect swallowing. Our objective is in this chapter is not to present a novel algorithm for detecting eating behavior, but rather to validate our proposed scheme with a realistic application.

5.3 Algorithms

5.3.1 Phase I - Window Selection

Our objective in the first stage of the algorithm is to subdivide a long-duration time-series signal in order to minimize the potential problems shown in Figure 7.4. We begin by subdividing the signal into fixed-length windows, as shown at the top of Figure 7.1. Subsequently, the windows are resized based on various criteria. The final result is shown at the bottom of Figure 7.1; some of the windows have been resized, and all of them have been assigned a class label of *Relevant* or *Not Relevant*. Those which are assigned a label of *Not Relevant* are discarded, while the remainder are processed in Phase II of the algorithm.

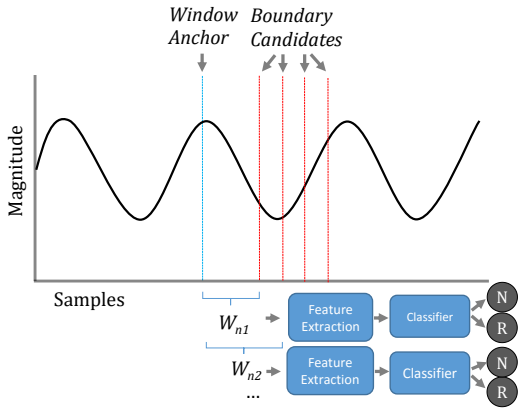


Figure 5.4: This figure shows how an individual window is resized several times. After each resize, it is input into a classifier to determine its relevance.

Though the objective of both Phase I and Phase II is to classify window of data that may be associated with events of interest, the emphasis in Phase I is to detect if a particular window is worthy of a closer look. This is achieved using a high performance classifier with a small input feature set, rather than an expensive and slow classifier. The motivation behind this optimization is the observation that detecting if *something* has taken place in a continuous signal is a much simpler problem than detecting exactly *what*.

Figure 6.3 shows a closeup of how one particular window would be processed. Each window is resized up to α times, with the left boundary fixed. After each resize, features are extracted from the signal and processed using a binary classifier. The boundary that is selected is that which minimizes the window length, while satisfying the constraint that the window is assigned a *Relevant* class label. If no boundary yields a window that receives this label, it is discarded. A more detailed reproduction of this algorithm can be found in Algorithm 6. From this algorithm, it is clear that a serious performance penalty will be incurred in applications with a high value of α; it is in these cases that the use of a simple, high-performance classifier is particularly important.

5.3.2 Phase II - Advanced Classification

In the second phase of the algorithm, we are given a series of windows derived from the original continuous signal that have been identified as potentially relevant. The primary challenge at this stage is to select an appropriate classifier to maximize the classification

accuracy. This procedure is shown in Figure 6.4. Note that the input to the algorithm is a series of windows that are of varying sizes; much of the signal has been discarded in Phase I.

The remaining signals are binned according to their window length. The logic behind this decision is simple: a large window is more likely to be associated with more complex or long-duration events, while a very small window is more likely to contain short-duration events. The process of binning is quite straightforward, as possible window sizes are multiples of a base length, l, and it is unlikely that this process is computationally demanding as the number of window candidates should be kept low as we discuss later.

The algorithm for classification is shown in Algorithm 7. Note that in this particular classifical scheme, the feature extraction algorithms and classifiers, **extractComplex** and **classifyComplex**, are different from those described in Algorithm 6. At this stage, the classifier must assign one of several class labels and therefore requires a more robust set of features than the simple implementation in Phase I.

Thus, in Phase II of the algorithm, most of the original windows of data have already been analyzed and discarded using a simple, lightweight classifier. For those data samples that remain, a higher-dimension classifier can be used to assign a specific class label to an event of interest. As we show in our case study on audio-based monitoring of eating behavior, the first step of the algorithm is to determine *if* the subject is eating, and the next (more challenging) step is to analyze *what*.

5.3.3 Cost Analysis

The last two subsections have discussed the general operation of the algorithm at a high level, but we have not properly motivated the design decisions of this methodology. The proposed algoriths, both Phase I and Phase II, make several hypotheses:

- Firstly, we assume that our first-stage classifier is able to properly discriminate between data that is *Relevant* and *Not Relevant* to the current classification problem. This is despite the limitations of a small feature set on the performance of the simple classifier used in Phase I.

- Secondly, we assume that we are able to better evaluate this decision by attempting

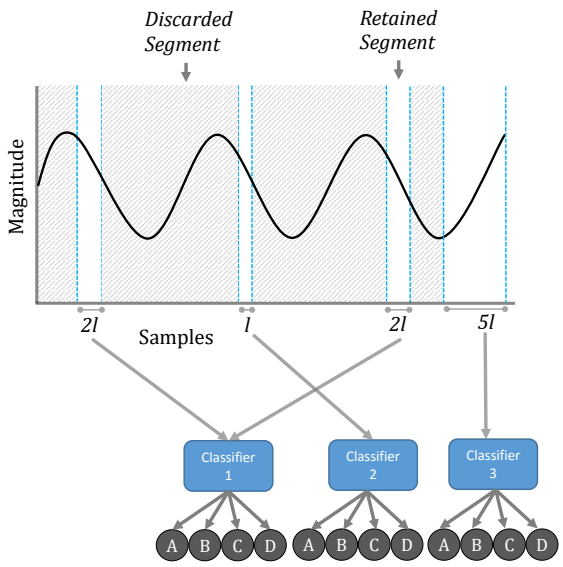


Figure 5.5: Phase II: discarding irrelevant segments of the window are discarded. The remaining windows are binned according to their length, and a more robust (and expensive) classifier produces the final class label.

a number of different window sizes.

- Thirdly, we assume that this approach is significantly less computationally intensive than an approach in which a single multi-class classifier is used to analyze every window of data.

The first two assumptions will be validated in the experimental methodology. The third assumption can be discussed more analytically. The general principle of the algorithm is to use the more expensive classifier and feature extraction algorithm on the most relevant data, and use the inexpensive Phase I algorithm to discard a larger amount of less relevant data. To preface this discussion, we use the term *cost* refer to the number of operations associated with an an operation; it is an estimate rather than an empirical measurement. The significant terms and symbols used in this discussion are presented in Table 5.1.

First, we define the cost of the simple and advanced classifiers used in Phase I and Phase II as C_{smp} and C_{adv}. Next, we define the number of features used by C_{smp} and C_{adv} as N and M. These values are significant because as shown in Equation 6.5, a higher number of features increases the cost of both the classifier and the feature extraction tool.

We can represent the total algorithm cost for processing a single window as the sum of the costs of Phase 1 (Segmentation) and Phase 2 (Class Label Assignment).

$$C_{tot} = C_{ph1} + C_{ph2} \tag{5.2}$$

We will now investigate C_{ph1} in more depth. For each original window, we define α possible boundary candidates as shown in Figure 6.3. For each boundary candidate, we must pay the penalty of extracting N features and classifying (C_{smp}). This formulation is shown below in Equation 5.3.

$$C_{ph1} = \alpha \cdot C_{smp} \cdot \sum_{i=1}^{N} f_i \tag{5.3}$$

$$\approx \alpha \cdot C_{smp} \cdot \frac{N \cdot f_{avg}}{2}$$

The per-window cost of the next stage of the algorithm, C_{ph2}, is similar to that of the first page except we use a different number of features, and the α parameter plays no role. This formulation can be found below, in Algorithm 5.4.

$$C_{ph2} = C_{adv} \cdot \sum_{i=1}^{M} f_i \tag{5.4}$$

$$\approx C_{adv} \cdot \frac{M \cdot f_{avg}}{2}$$

Thus, our final algorithm cost can be represented by substituting Equation 5.4 and Equation 5.3 into Equation 5.5. Simplifying, this yields the following per-window cost:

$$C_{tot} = \frac{f_{avg}}{2}(\alpha \cdot C_{smp} \cdot N + C_{adv} \cdot M) \tag{5.5}$$

Though the formula in Equation 5.5 represents the cost associated with one particular window, we extend our analysis for a signal comprised of W windows of some arbitrary size. To represent the total cost, we require a new parameter, β, which represents the percentage of windows in Phase I that are assigned the label of *not-relevant*, and are

therefore not processed by Phase II of the algorithm. Recall that all other windows are discarded in order to reduce overhead; the number of windows processed in Phase II should be significantly less than that of Phase I.

$$C_{tot} = \frac{W \cdot f_{avg}}{2}\big(\alpha \cdot C_{smp} \cdot N + C_{adv} \cdot M \cdot (1 - \beta)\big) \tag{5.6}$$

5.3.4 Comparison to Baseline

We now compare the cost associated with the proposed algorithm to a 1-phase system which essentially applies the Phase II algorithm to each window in the original signal. Therefore, the cost of the baseline can be approximated as:

$$C_{tot} = \frac{W \cdot f_{avg}}{2}(\alpha \cdot C_{adv} \cdot M) \tag{5.7}$$

Thus, with $\alpha = 1$, the following criteria must be met for the baseline to have a higher cost than the optimized algorithm:

$$\alpha \cdot C_{adv} \cdot M > \alpha \cdot C_{smp} \cdot N + C_{adv} \cdot M \cdot (1 - \beta) \tag{5.8}$$

In conclusion, the proposed scheme can outperform the baseline (on the basis of cost) when the constraints specified in Equation 5.8 are satisfied. The practicality of this is discussed further in subsequent sections as we solve for some of these coefficients through an experimental case-study.

5.3.5 Feature Selection

Feature selection allows us to reduce the values of M or N in Equation 5.8, as well as the classifier runtime C_{adv} and C_{smp} in some cases. Besides performance-minded applications, feature selection can eliminate redundancy and remove weakly correlated features, while reducing overfitting. The two general feature selection techniques are *filter* and *wrapper* methods. Filter methods use a specific metric to score an individual feature or subset of features, while wrapper methods use a classifier to evaluate features based

on their predictive power. Typically, wrapper methods may be too slow for systems with thousands of possible features; it is for this reason that we use the filter approach [LL06]. The specific algorithm we employed is the InformationGain feature extraction algorithm.

The attribute selection algorithm provides a sorted list of features, based on their predictive power with respect to the desired classifier outcomes. This approach allows us to train classifiers for Phase I and Phase II by modifying the thresholds for the attribute selection algorithm. Therefore, the Phase I classifier can be trained to rely on a smaller input set compared to the more robust, but computationally expensive classifier employed in Phase II.

5.4 Experimental Methodology

The case study we use to evaluate our proposed scheme is a nutrition monitoring application. In this study, we obtained audio recordings from 20 individuals who ate three different foods. From this data, we extract features and use various classifiers in an attempt to identify the correct food. The first challenge is to segment the original signal to distinguish eating *any* food from ambient noise and silence. The second challenge (Phase II) is to distinguish one food from another using a potentially more robust classifier. For further details about nutrition monitoring challenges and applications, we refer the reader to Alshurafa et al. [AKP15].

5.4.1 Data Collection

Data was collected from 20 individuals using a Hyperio Flexible Throat Microphone Headset placed in the lower part of the neck near the collarbone, connected directly to the mobile phones audio input port using a 3.5mm male audio cable. Among the subjects, 16 were male, and 4 were female. The ages ranged from 21 to 31 years old, with a median age of 22. Commercially available audio-recording technology was used to acquire the audio recordings from the microphone. Twenty subjects, who were given two miniature chocolate bars, followed by ten Pringles potato chips and 2 small celery sticks. The foods were consumed sequentially, in that order, and the subjects ate one potato chip at a time. These recordings formed the basis of the algorithm design and experimental evaluation.

The data collection took place in a lab environment; people can be faintly heard speaking in the background, and the microphone occasionally recorded doors closing and nearby footsteps. The collected data is a subset of the data collected in a previous trial on audio-based detection of eating behavior based on identification of chews and swallows [AKP14][KAL15a][AKP15].

5.4.2 Feature Extraction

The Munich open Speech and Music Interpretation by Large Space Extraction toolkit, known as openSMILE [EWS10], is a feature extraction tool intended for producing large audio feature sets. This tool is capable of various audio signal processing operations such as applying window functions, FFT, FIR filterbanks, autocorrelation, and cepstrum. In addition to these techniques, openSMILE is capable of extracting various speech related features and statistical features. Audio-based features include frame energy, intensity, auditory spectra, zero crossing rate, and voice quality. After data is collected from a variety of subjects eating several foods, feature selection tools can be used to identify strong features that are accurate predictors of swallows and bites for various foods, while reducing the dimensionality by eliminating redundant or weakly correlated features.

5.4.3 Selected Features

Table 5.2 shows the 20 features ranked according to their correlation with the desired classifier outcomes: their ability to distinguish between the two types of food. In this table, sma represents the simple moving average of a signal characteristic, while melspec refers to the mel-frequency cepstrum, which is a representation of the power spectrum of a sound on a non-linear scale.

5.5 Results and Discussion

In this section, we provide results as we apply our segmentation and classification algorithm to our audio-based nutrition monitoring dataset.

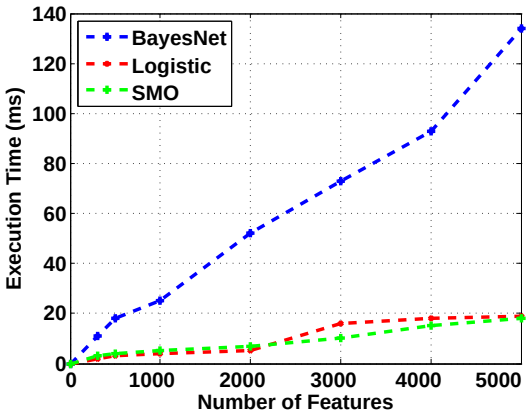


Figure 5.6: SMO [Pla99a], Bayesian Networks [FGG97], and LogicalRegression [HL04] classifiers are evaluated for speed as a function of feature set size.

5.5.1 Classifier performance vs. Input Size

Figure 5.6 shows the performance of three classifiers as input size is varied. The evaluated classifiers are: SMO (Sequential Minimal Optimization) [Pla99a], Bayesian Networks [FGG97], and LogicalRegression [HL04]. We do not evaluate the training time of each classifier as we assume that this is done offline for real-time systems. In our experiments, we used the WEKA implementation of these classifiers and used the WEKA Experimenter environment to generate our results [HFH09a]. Figure 5.6 reveals that, as expected, the amount of time to run a classification algorithm and produce a class label varies significantly with the number of features. It is also important to note that the variation in performance between classifiers is substantial. Therefore, it is imperative that a lightweight power-aware classifier select both an appropriate classification algorithm and feature size.

The results suggest that the Bayesian Network classifier has more overhead for a given number of features than the two alternatives; Logistic Regression and SMO. Though experimental constraints make it difficult to evaluate classifier performance with very low (single digit) feature sizes, there appears to be a roughly linear relationship between feature size and execution time in the case of the three evaluated classifiers. This, compounded with the overhead of *extracting* high-dimensional feature sets in real time, strongly biases performance-minded applications towards smaller feature sets.

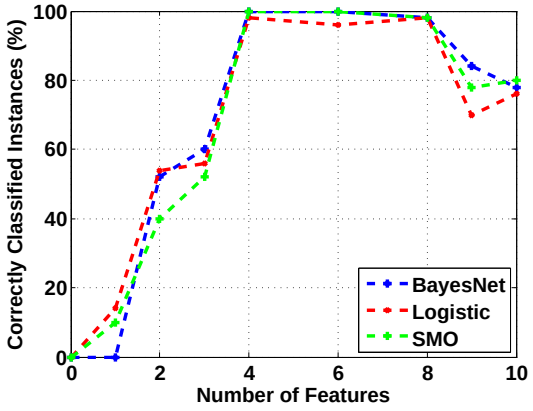


Figure 5.7: The ability of three classifiers to distinguish noise from relevant data (Phase I) as a function of number of features.

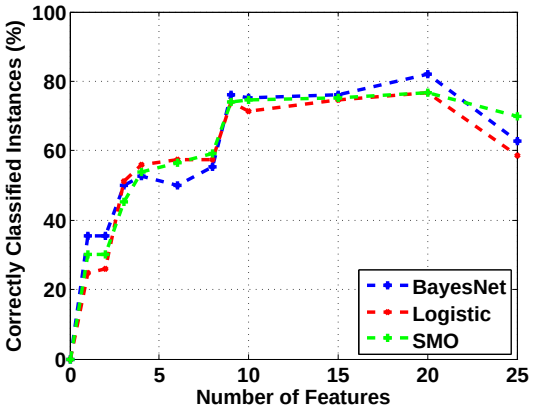


Figure 5.8: The ability of three classifiers to identify from three different classes (Phase II) as a function of number of features.

5.5.2 Classifier Accuracy

Figure 5.7 and Figure 5.9 show the classification accuracy of the Phase I and Phase II classifiers, respectively. As Figure 5.9 shows, a feature set of size nine is also a good tradeoff between accuracy and set size for distinguishing between the three foods. With nine features, the percentage of correctly classified instances between the three food types is just under 80%, and increases only slightly as the feature set size is increased to twenty before dropping off.

By comparison, Figure 5.7 shows that because of the simplicity of the classification problem, four features is sufficient to clearly identify eating-related sounds from ambient background noise and silence. Note that in the first case, in which noise is to be differen-

tiated from eating behavior, we are able to correctly classify virtually all instances with a feature set of size four. Through this observation, we can optimize computation time without diminishing classification accuracy by identifying relevant signals with a classifier with feature size four, and distinguish between various foods with a larger feature set.

5.5.3 Speed vs. Unoptimized Baseline

Let us assume that our baseline model classifies between all four classes of data (the three foods as well as noise) all at once. With no apriori knowledge about whether the data in the window was relevant, we would require nine or more features as shown in Figure 5.9. Referring back to Equation 5.8, we can calculate the ratio of the baseline execution time and the proposed scheme using Equation 5.9 below, along with the addition of the α parameter that reflects the number of boundary candidates.

$$T_{ratio} = \frac{\alpha \cdot C_{adv} \cdot M}{\alpha \cdot C_{smp} \cdot N + C_{adv} \cdot M \cdot (1 - \beta)} \tag{5.9}$$

As we have discussed previously, viable values for M and N (feature set sizes for both classifiers) are nine and four, respectively. Furthermore, we can assume that with assumptions of linearity (as shown in Figure 5.6), the ratio of $\frac{C_{smp}}{C_{adv}}$ also $\frac{4}{9}$. Therefore, the final ratio of performance can be approximated using Equation 5.10 below:

$$T_{ratio} = \frac{\alpha \cdot 81}{\alpha \cdot 16 + 81 \cdot (1 - \beta)} \tag{5.10}$$

A plot of Equation 5.10 under various values for β and α is shown in Figure 5.9 in which we compare the execution time of both algorithms in relative terms. That is, this equation does not consider the overall signal length or amount of time necessary to extract a feature as these values would be approximately equal for both the baseline and proposed scheme.

As Figure 5.9 shows, the proposed scheme vastly outperforms the baseline in cases which have a high number of boundary candidates (α) per window. However, in the case that boundary candidates are not evaluated and the size is fixed ($\alpha = 1$), the baseline outperforms the proposed scheme in some cases. Figure 5.9 also reveals the sensitivity of

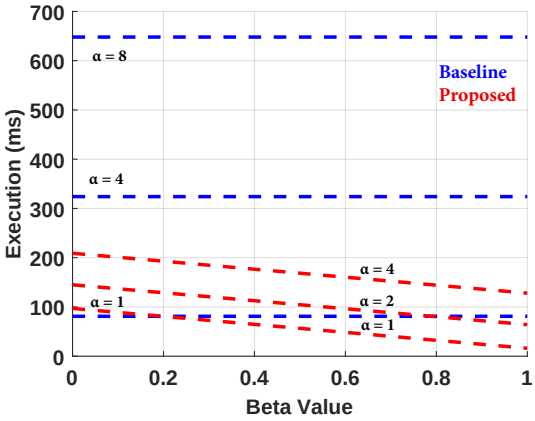


Figure 5.9: Comparison between the baseline and proposed scheme as we sweep across β values and vary the α parameter.

the algorithm to the β parameter. Recall that this parameter prefers to the percentage of original windows that are labelled as *not relevant*, and therefore are not processed in the second stage of the algorithm. As expected, the baseline is not sensitive to the β parameter as it is a one-stage algorithm. As β increases, the execution time of the proposed algorithm decreases as unnecessary computation is avoided. However, with a β value of close to zero, in which almost all windows of data must be processed using C_{adv}, the baseline outperforms the proposed algorithm. It is clear from these results that the proposed algorithm will outperform the baseline in cases in which a significant percentage of data windows contain no meaningful information (ie. noise), with only intermittent events which require class labels beyond *(relevant, not relevant)*.

5.5.4 Boundary Selection

We now direct our attention to the boundary selection problem. In our previous study on audio-based nutrition monitoring, we were able to classify between a variety of different foods based on the characteristics of the swallow sounds associated with each food type. However, the swallows were manually identified and segmented from a long audio-file consisting of chewing, swallowing, silence, and background noises. In this experiment, we apply our Phase I segmentation algorithm to the original audio datasets, to evaluate if the time-series segmentation algorithm is capable of dividing the signal such that the Phase II algorithm can accurately identify the food type. In order to do this, we evaluate

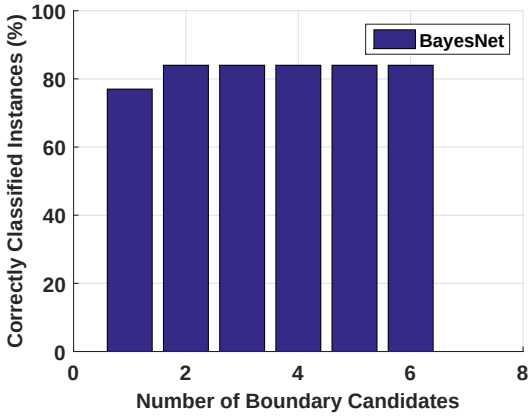


Figure 5.10: Classification accuracy vs. α parameter.

classification accuracy as a function of α: the number of window boundaries attempted for each signal.This experiment was conducted using the BayesNet classifier, with Phase I and Phase II feature counts of four and nine, respectively. Results are shown in Figure 5.10.

As these results indicate, the classification accuracy improves by approximately 6% as we consider two window candidates in comparison to one. However, there does not appear to be any improvement in the percentage of correctly classified instances as the boundary candidates (α) is increased beyond this number. This is most likely because the proposed algorithm breaks out of the iteration process of testing different window sizes as soon as the smallest available window size is given the *relevant* class label. As the focus of our work is on mobile low-power devices, this performance optimization is necessary for the practical realization of the proposed algorithm.

5.6 Conclusion

In conclusion, we demonstrate a new technique for segmenting and classifying time-series signals using a hierarchical approach that attempts multiple window-sizes using a small, efficient classifier, as well as a more robust classifier with a higher feature set for those segments that are deemed relevant for further analysis. We show, compared to a baseline approach, that our approach is capable of improving performance under various conditions with minimal impact to overall system classification accuracy. Future research will evaluate the efficacy of the proposed technique in more varied use-cases with different

properties.

Algorithm 2: Phase I - Segmentation Algorithm

```
/* We iterate through each window in the signal.
The identifier for each window is the left boundary. */
while LBoundary lb in Signal do
    /* Iterate through α possible window lengths for each window. */
    for i = 1:α do
        /* Extract features from a window of a particular length and assign a class
        label to it. */
        Window cur = getWindow (lb, lengthIndex(α))
        Features fts = extractSimple (cur);
        ClassLabel = classifySimple (fts);
        if ClassLabel == Relevant then
            /* The shortest length that is classified as relevant is saved. Otherwise,
            we discard. */
            Save (LBoundary, lengthIndex(α));
            /* Adjust starting index of next window. */
            adjustBoundaries ();
            break;
    /* Otherwise, discard non-relevant windows. */
```

Algorithm 3: Phase II - Final Classification

```
/* We iterate through each relevant window in the signal.
   The identifier for each window is the left boundary. */
foreach Window win in Signal do
    /* We retrieve a pre-trained classifier based on the length of the current
       window. */
    Classifier class = getClassifier (win.length());
    /* We extract features from the window, and input these features into an
       advanced (resource-heavy) classifier to receive a class label. */
    Features fts = extractComplex (win);
    ClassLabel cl = classifyComplex (class, fts);
    /* Finally, we associate the returned label with the current window and proceed
       to the next window. */
    AssignLabel (win, cl);
```

Table 5.1: Definition of Terms

Symbol	Definition
C	Cost: the amount of time necessary to execute an operation.
f_{avg}	The average cost required to extract a feature from a particular window.
M	The number of input features required by the Phase II Classifier.
N	The number of input features required by the Phase I Classifier.
C_{smp}	Average cost associated with executing the simple classifier for a particular feature set.
C_{adv}	Average cost associated with executing the advanced classifier for a particular feature set.
α	The number of boundary candidates evaluated in Phase I.
β	The average probability that the Phase I classifier will classify a particular window as *not relevant*.
W	The number of windows processed for a particular signal.

Table 5.2: Top 20 Features Selected by the Feature Reduction Algorithm

Rank	**Attribute (FFTMag)**	**Correlation**
1	melspec sma[1] quartile1	0.951
2	melspec sma[0] nonzero geom. mean	0.885
3	melspec sma[1] nonzero geom. mean	0.877
4	melspec sma[1] quartile2 numeric	0.858
5	melspec sma[1] nonzero arith. mean	0.833
6	melspec sma[1] arithmetic mean	0.833
7	melspec sma[1] abs. value mean	0.833
8	melspec sma[0] quartile1	0.815
9	melspec sma[1] nonzero quadr. Mean	0.807
10	sma[1] qmean	0.807
11	sma[1] linregc2	0.807
12	sma[1] quartile3	0.807
13	sma[0] linregc2	0.794
14	sma quartile1	0.77
15	sma[1] qregc3	0.767
16	sma quartile2	0.761
17	sma[1] peakMean	0.758
18	sma[0] quartile2	0.747
19	sma[1] percentile98.0	0.726
20	sma[0] quartile3	0.718

CHAPTER 6

Probabilistic Time-Series Segmentation for Real-Time Classification Applications

To further bolster the quantified-self movement and provide timely user interventions, sensor signals from body sensor networks need to be processed in real-time. Time subdivisions of the sensor signals are extracted and fed into a supervised learning algorithm, such as Support Vector Machines (SVM), to learn a model capable of distinguishing different class labels. However, selecting a short-duration window from the continuous data stream is a significant challenge, and the window may not be properly centered around the activity of interest. In this work, we address the issue of window selection from a continuous data stream, using an optimized SVM-based probability model. To measure the effectiveness of our approach, we test our algorithm against audio signals from a wearable nutrition-monitoring necklace. We compare our optimized time-series segmentation algorithm against a naive and exhaustive approach. Our optimized algorithm is capable of correctly classifying 92.6% of instances, compared to a baseline of 73% which segments the time-series data with fixed-size non-overlapping windows.

6.1 Introduction

Sensing and monitoring technologies have become increasingly popular in recent years, motivated by advances in wearable technology and a rapidly aging global population [LSS08]. The applications of wearable technologies range from monitoring cardiac activity, eating habits, blood pressure, physical activity, brain activity, speech patterns, and more[misa][LKC10][TIH07]. These devices use integrated micro-sized wearable sensors, which provide data that can be analyzed to track user activity. Recently, many works have employed machine learning and data mining techniques to identify health-related

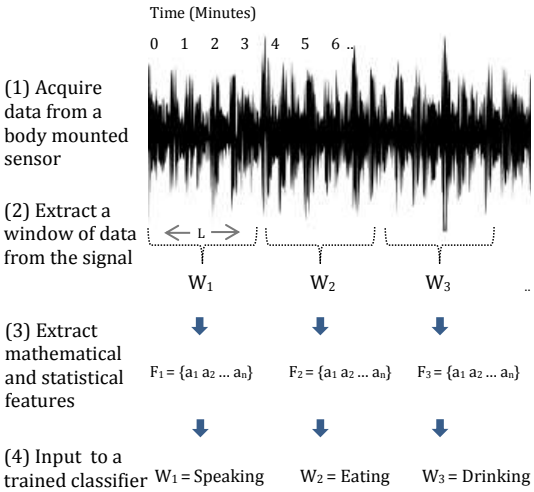


Figure 6.1: Typical system flow for classifying sensor data.

events from digital signals [PM07][SKC12][KLM13]. These devices and technologies have the potential for significant impact in real-word environments, by enabling *preventative healthcare*. However, the detection of finite events in a large stream of data is not without its challenges. In many cases, sensor data analysis can be quite simple. For example, determining heart rate from heart rate sensors could in some cases require little more than amplification and thresholding[SBT02]. In other cases, such as continuous audio-based monitoring of eating habits, sleep apnea, and coughing, much more advanced techniques are required to discriminate between activities.

Machine learning tools based on classifiers such as Support Vector Machines (SVMs) are often used to transform data to find an optimal boundary between class labels [AKP14]. Such tools enable the development of models to identify the subtle and interrelated conditions needed to distinguish between class labels. However, identifying the intended activity and extracting data points for classification from a continuous time-series signal is quite challenging. Some existing techniques identify peaks and troughs in the signal to identify each data sample, while others perform fixed time subdivisions of the data [PCC14]. The division of a large continuous signal into individually-processed fragments is known as *time-series segmentation*, which is the focus of our work.

This chapter is organized as follows. In Section II, we describe the motivation for our work. In Section III, we present a brief introduction to the challenges of time-series segmentation. In Section IV, we present related work in this field. In Section V, we

provide an overview of Support Vector Machines and the notion of classifier confidence. In Section VI, we describe our proposed window selection algorithm. In Section VII, we describe the experimental setup. In Section VIII, we present our results. We provide a discussion of future work in Section IX, and concluding remarks in Section X.

6.2 Motivation

Consider a 24-hour stream of audio data from which we intend to identify eating behavior that we must disambiguate from other sounds. A typical system flow is defined as follows. First, audio is recorded using a microphone that is placed somewhere on the body. Examples include a smartwatch [KS15b], smartphone, or a custom hardware device such as a neckband [KAS14a]. After audio is recorded, it is either buffered or transmitted to an aggregator for analysis. However, assigning a single class label to an entire days worth of data is impractical for many cases, which attempt to identify brief actions throughout the day. Therefore, the large audio stream is divided into shorter windows, in order to identify activities with a finer granularity and avoid averaging out unique statistical features associated with finite-length events such as coughing, chewing, and swallowing.

After the audio signals are divided into shorter windows, statistical features can be extracted from the window. Examples include mean, interquartile range, standard deviation, variance, kurtosis, and skewness. These features are processed using a pre-trained classifier, which can identify the action being performed based on the distinguishing characteristics of the signal represented by the features selected by the training process. This system flow is shown in Figure 7.1. At the top, we have an audio waveform that is several minutes in duration. In order to identify multiple short-duration events from this signal, we extract windows W_1, W_2, and W_3 from the data. We then extract statistical features from each window, and input these features to a trained classifier, which outputs a class label corresponding with that window. In this case, our classifier reports that the user is speaking at times $\tau = 0$ through $\tau = 4$, followed by four minutes of eating, and four minutes of drinking.

Figure 7.1 presents an example of a working system with ideal segmentation, but consider a case in which W_1 contains both data corresponding with speaking and eating,

with roughly the same proportion. In this case, no class label can holistically represent the contained data. Rather than introducing a new class label corresponding with this particular combination of events, the window size and boundaries could be adjusted to properly contain the events of interest. In the next section, we describe the shortcomings of arbitrary window selection in more detail.

6.3 System Challenges and Considerations

Most signal classification approaches in academic literature extract windows that are manually centered on the action of interest, with a window size large enough to fully contain the distinguishing features of the signal. Thus, the system has a-priori knowledge about the action being performed. Figure 7.1 shows an approach in which the system has no a-priori knowledge about the signal, which is a more realistic scenario for real-world environments. However, there are several significant challenges associated with this approach, which are shown in Figure 7.4. In this figure, the sinusoid represents the raw signal, while the red star denotes the short-duration event that we wish to detect. The dashed line represents the boundaries of the extracted window from the original signal. The major challenges are described below:

6.3.0.1 Window Size Selection

First, we may fail to select an appropriate window size, as shown in Figure 7.4-C. Consider an audio-based classification system designed to identify an activity of short duration such as a cough. If the system window size is ten minutes long, the audio features associated with the cough will be averaged together with the rest of the data in the window. Therefore, certain statistical features, such as those which evaluate power spectral density at different frequency ranges, may be rendered ineffective. Furthermore, a window of large length will be assigned a single class label, while multiple actions of interest could have taken place in this time period. Alternatively, a small window size may fail to holistically represent the event in the extracted snippet.

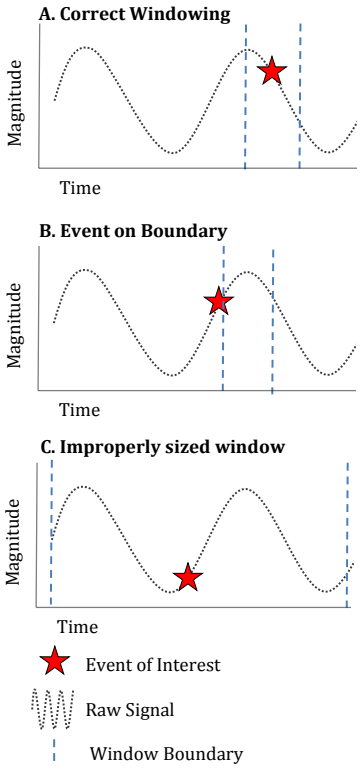


Figure 6.2: Incorrect segmentation scenarios.

6.3.0.2 Window Boundary Selection

We must appropriately place the boundaries of the window. Consider the case in which the first half of the cough is found in window W_1, while the second is in window W_2. An example can be seen in Figure 7.4-B, in which only a portion of the event of interest is represented in the current window. Therefore, no window may have enough information for a correct classification. However, the classifier must still output a class label for the data, and the accuracy of the system will decrease. A potential solution is to overlap the windows, as shown in Figure 6.3. However, this approach is associated with high computational overhead; if we define$L$ as the window size, this approach requires a factor of L more time to complete. An embedded application may not have the power budget or processing power to justify this approach, considering some signals such as audio have sample rates in the kHz range.

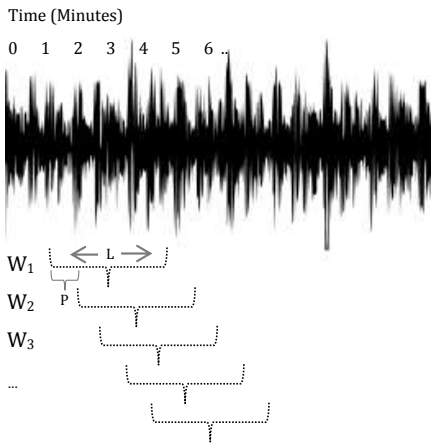


Figure 6.3: Window overlapping to avoid boundary issues.

6.3.0.3 Binary Output

Our classification system does not return a probabilistic output. A class label is assigned to each window, regardless of the classifier confidence that this label is correct. In some cases, it may be preferable to avoid assignment of a class label unless the classification confidence exceeds a predefined threshold. The end user of the classification model can subsequently choose to retain the class label or discard the sample, based on the specific properties of their application.

The key question we aim to address is: *How can we correctly window a continuous signal, in order to maximize classification accuracy of short-duration events?* Fortunately, various algorithms have been proposed for mapping a raw classifier output to a probability, or classification confidence [Pla99b][LLW07]. In this chapter, we propose a new architecture that can be used for real-time monitoring of sensor signals such as audio, by optimizing window selection based on the classification confidence for that particular instance.

6.4 Related Work

A comprehensive survey of time-series segmentation was provided by Keogh et al. in [KCH04]. The emphasis of existing work on segmentation is to develop a compact representation of a large signal, with applications in data mining and compression. The three primary methods in literature are sliding window approaches, similar to the baseline used

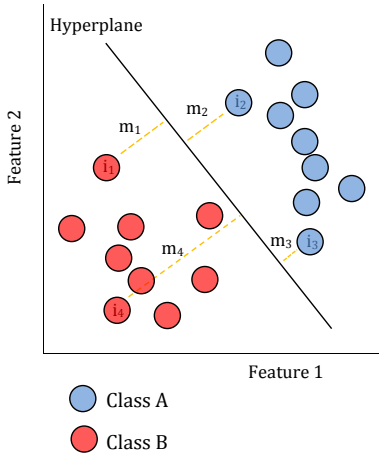


Figure 6.4: SVM hyperplane-based separation of instances.

in this chapter, and recursive top-down or bottom-up techniques that are either partition or merge signals until a stopping criteria is met.

Analysis of digital signals, particularly audio signals, has been a subject of many recent works [SSL08][RAZ14]. In [RAZ14], Rahman et al. present BodyBeat: a robust system for detecting human sounds. A similar work is presented by Yatani et al. in [YT12]. The BodyScope device can distinguish between twelve kinds of activities including eating, drinking, laughing, and coughing. This device is based on a microphone worn on the neck. By analyzing the magnitude of different frequency ranges over time, the authors are able to recognize various activities using classifiers such as Support Vector Machines [HDO98]. However, the window size from which they extract features is of fixed length, at five seconds. This window size is arbitrary, as some actions are of significantly shorter duration, while others are more protracted. Furthermore, the activities in question were manually annotated, and the audio fragments were extracted by hand.

Adaptive window sizing has been explored in several other works. For example, in [OK92], Okutomi et al. present a signal matching algorithm that can adapt window size based on the uncertainty of the disparity between two signals. The issue of adaptive window sizing for image filtering using local polynomial approximation is presented by Katkovnik et al. in [KEA02]. In [KJ00], Klinkenberg et al. propose an approach for selection of window size to minimize generalization error based on leave-one-out estimation error. However, to the best of our knowledge, applying posterior probability of a classifier to the issue of window selection has not been explored in academic literature.

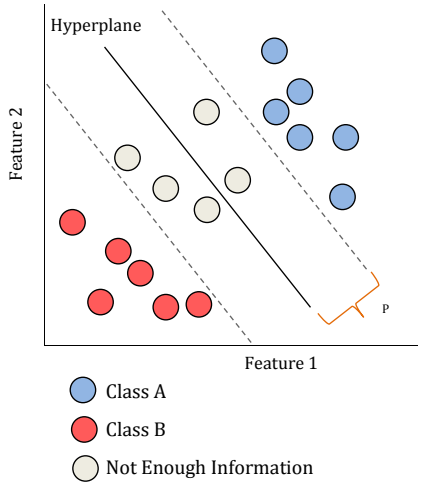


Figure 6.5: Proposed scheme in which instances close to the hyperplane are not assigned class labels.

6.5 Posterior Probabilities for SVM

6.5.1 An Introduction to Support Vector Machines

Support Vector Machines (SVMs) are supervised learning models commonly used for classification problems. Given a set of data, the SVM will attempt to separate the two classes using a hyperplane, which is a subspace one dimension less than the ambient space. Various mathematical techniques such as dual quadratic programming and non-linear kernel transformations can be used to maximize the margin between the classes of data. A *hard-margin* SVM will enforce the principle that the data must be separable. However, it may not be possible to achieve perfect separation of the data in some cases. Therefore, a *soft-margin* SVM classifier can be used to maximize the hyperplane margin while allowing some misclassifications.

6.5.2 Motivation

Consider a system designed for distinguishing between two activities: A and B. Figure 6.4 illustrates a simple case in which an SVM model is used to linearly separate the two classes of data using a one dimensional hyperplane in a two-dimensional space. Thus, all 18 instances are labeled as class A or class B, with no distinction made about their original location on the feature plane. An *SVM Score* for an instance i is defined as

the distance from i to the decision boundary. Clearly, instance i_4 has a very large SVM Score, m_4, as seen by its distance from the hyperplane separating the two classes. In comparison, note that the margin separating instance i_3 from the hyperplane is quite small. It has been shown that the probability that a classification is correct is a function of the SVM score for that particular instance, m [Pla99b]. Therefore, this observation suggests some ambiguity about the true class label associated with i_3. This can be a result of several factors:

- The window size may be too small to effectively capture the distinguishing feature of the activity.
- The window size may be too large, polluting the unique properties of the short-duration activity with neighboring noise, or encompassing multiple events within the boundaries of a single window.
- The window boundaries may be bisecting the activity of interest into two separate windows, neither of which can definitively identify the event.
- The sample may not be associated with activities represented by either class. However, a class label will be assigned regardless, because a real time activity monitoring device cannot practically have thousands of class templates for each possible action a user could take throughout the day.
- The window selection is optimal, but the activity has unique properties that are dissimilar to other instances in that class.

We can potentially improve classification accuracy by considering the distance between an instance and the hyperplane to adjust the window index and size. That is, we are no longer exclusively concerned with the specific class an instance is associated with, but with the probability that the instance belongs to that particular class. Our motivation is the observation that window boundaries can be adjusted until the instance's SVM score increases beyond our desired threshold. As shown in Figure 6.5, we assign class labels to those instances which are likely to be correct classifications. Those instances associated

with lower probabilities, based on their distance from the hyperplane, are labeled as not enough information. In these cases, we refrain from assigning a classification output, and perform minor modifications to the associated window in order to disambiguate the class label. In the event that we are unable to improve the classification confidence score, we tag the data as ambiguous, leaving to the discretion of the user the decision of whether to discard the sample or assign a label.

6.5.3 Probability Model

In the previous section, we referred to the probability that an instance is classified correctly as a function of the distance between the instance and the hyperplane. However, the relationship between distance and probability is not necessarily linear [NC05]. The work by Platt et al. in [Pla99b][LLW07] explored the probabilistic outputs of classifiers and concluded that the relationship between SVM score and posterior probability assumes a sigmoid distribution. Platt Calibration is a well-known method for transforming classification model outputs to probability distributions. This technique, though generalizable, is often used in the context of Support Vector Machines. Platt Calibration can be used to compute the posterior probability of a class label given an instance x, using the following formulation:

$$P(y = 1|x) = \frac{1}{1 + e^{\alpha \cdot f(x) + \beta}} \tag{6.1}$$

In this case, *f(x)* is the raw, uncalibrated classifier output, the sign of which is used for binary class assignment. Constants α and β are coefficients obtained using a maximum likelihood estimation. These values are obtained by fitting training data to a sigmoid function, as the sigmoid model has been shown to be generally superior to other models such as Gaussian distributions for probability estimation. Typically, the sigmoid is trained using a subset of the training data (roughly 30%) [Pla99b] to ensure an unbiased training set. Thus, the Platt Calibration algorithm returns a posterior probability of classification accuracy for each instance, based on the signed distance between the instance and the decision boundary. Therefore, we can use the Platt Calibration algorithm to produce a classification confidence score for a particular instance, and adjust the window parameters

to disambiguate the true label. This procedure is outlined in the following section. For further details about Platt Calibration, please refer to Platt et al. [Pla99b].

6.6 Window Size Selection

Each window is a function of two parameters: the starting index at which the window boundary is placed, as well as the length of the window. Thus, we define a window as a function of these parameters, $w = (I, L)$. Function f uses a trained SVM classifier as a well as a calibrated Platt model to return a class label c and a classification confidence p, based on an input window defined by the length and starting index, I and L, respectively.

$$[c, p] = f(I, L) \tag{6.2}$$

The classification confidence is a number between 0 and 1, which reflects the estimated probability that the instance is labeled correctly. Thus, in a binary classification problem, p ranges from .50 to 1. Based on this formulation, we compare three algorithms for window selection.

6.6.1 Naive Window Selection

Naive Window Selection is the simplest model that we use as a baseline. This is the scheme shown in Figure 7.1, which is computationally simple, and does not use probability in the assignment of class labels. In this model, we simply subdivide the signal into non-overlapping windows and assign class labels to each window as described in the introduction. We use a fixed window length, and the default starting index, both of which are a function of the particular classification problem and cannot be generalized. This very simple scheme is shown in Algorithm 4. Though classification accuracy is likely to be quite low using this approach, the simplicity of this model is likely to execute quickly because the algorithm is not iterative in the case of a particular window; the time complexity is $O(\frac{n}{L})$ where n is the number of samples and L is the window size.

Algorithm 4: Naive Window Selection

```
/* We use the default window length and index*/
(c,p) = f (I, L);
/* Regardless of the probability of correct classification, we return the class label.
*/
return (c, p)
/* The next index will be I_new = I + L */
```

6.6.2 Exhaustive Search

In the Exhaustive Search approach, we emphasize classification accuracy over speed, by attempting every combination of window index and size to maximize the Platt classification probability. First, we define vectors **W** and **S** as the possible window sizes and starting indices, respectively.

Parameters α and β represent the allowed variation of window length and starting index, while parameters n and m represents the number of possible starting indexes and window lengths to iterate over. These parameters are typically a function of the properties of the particular event that we attempt to identify. The complete algorithm is shown in Algorithm 7. Note that this algorithm performs an exhaustive search of all combinations of W and S, and selects the window length and starting index based on the combination with the maximum classification probability, using Platt's probability model. The number of iterations, T, is defined in Equation 6.3 as the product of the lengths of vectors **W** and **S**.

There is no particular threshold at which the search concludes: the algorithm will iterate through every combination of window length and index until arriving at the optimal solution. Therefore, even if the original classification decision is associated with a posterior probability of over 99%, all other options will be considered. This inefficiency is addressed in the next subsection. The number of iterations necessary for the Exhaustive Search algorithm to terminate is shown below, in Equation 6.3. Thus, the exhaustive search will take approximately n*m more time to complete, compared to the Naive Window Selection model.

$$T = n * m \tag{6.3}$$

Algorithm 5: Exhaustive Search Algorithm

```
/* Array W represents the possible window lengths. */
W = {α, 2·α, 3·α, 4·α ... n · α}
/* Array S represents starting indices from the window. i represents the final
window boundary index from the previous window. */
S = {i+β, i+2·β, i+3·β, ... i+m · β}
/* We iterate through all combinations. */
for i = 0; i < n; i++
    for j = 0; j < m); j++
        /* We receive a class label and class probability from function f, for this
        particular combination of window length and index.*/
        (c,p) = f (W[i], S[j]);
        if p > maxProbability
            /* This combination of window size and window length yields the best
            estimated probability of correct classification. */
            maxProbability = p
            BestLength = W[i];
            BestIndex = S[j];
        end
    end
end
/* The search is complete. We return the output class and the corresponding
probability. */
return (c, maxProbability)
```

6.6.3 Optimized Search

The Optimized Search algorithm selects window parameters based on classification probability, without performing an expensive search across the entire feature space <I, L>.

Algorithm 6: Optimized Search Algorithm

```
/* Array W represents the possible window lengths. */
W = {α, 2·α, 3·α, 4·α ... n · α}
/* Array S represents starting indices from the window. i represents the final
window boundary index from the previous window. */
S = {i+β, i+2·β, i+3·β, ... i+m · β}
/* We define the initial conditions */
wi = W[1];
si = S[1];
/* We receive a class label and class probability from function f, for this particular
combination of window length and index. */
(c,p) = f (W[wi], S[si]);
if p > γ then
    return (c, p);
maxProb = 0;
while maxProb < γ do
    (c2, p2) = f (W[wi+1], S[si]);
    (c3, p3) = f (W[wi], S[si+1]);
    (c4, p4) = f (W[wi+1], S[si+1]);
    (maxProb, bestI, bestL, bestClass) =
    max([c2, p2], [c3, p3], [c4, p4]);
    /* We evaluate the directional derivative of f using three vectors */
    if maxProb < p then
        /* If neither yielded an improvement, we return the original class and
        probability value. */
        return (c, p);
    else
        /* Otherwise, we continue iterating until we hit a local maximum or exceed
        γ */
        wi = bestL;
        si = bestI;
return (bestClass, maxProb);
```

This may be necessary due to system constraints, as an Exhaustive Search approach could be impractical for wearable systems with battery constraints and basic low-power processors. The Optimized Search algorithm pseudocode is presented in Algorithm 6.

We begin with line 2, in which we assume a large stream of audio data as a prerequisite for the search. First, we define array **W**, which represents the possible window lengths, where the value α represents the smallest possible window length. The value of α is a function of the classification problem. In our particular application, we believed that 500ms was the smallest possible reasonable time window. Therefore, the range of possible time windows varied between 500 ms and 2.5 seconds, as our n parameter was 5. Recall that n represents the number of different window lengths to attempt before selecting the length that yields the highest classification confidence.

In addition to length, the window is characterized by its start index: the time corresponding to the first sample of the window. Therefore, array **S** is required to present possible starting times. The offset of each element index is i, which is the endpoint of the previous window. The possible start indices range from $i + \beta$ to $i + m \cdot \beta$. Therefore, β represents the range of possible start indices. As in the case of α, the value of β is a function of the classification problem. The intention of this parameter is to shift the window slightly, to prevent cases in which the window boundary is dividing the action of interest into two separate windows. Given that the goal in our particular application is to detect a chew or swallow, or perhaps a combination of both, any more than a few seconds of offset is unnecessary. Therefore, the value of β in our experimentation was fixed to 200ms with an m value of 5.

Lines 6 and 7 show that the algorithm selects default values of WindowLength and StartingPoint as 2.5 seconds and $\tau = \mathrm{i} + 200$ ms, respectively. Subsequently, we extract a window with those parameters for L and I, extract statistical features, and classify using SVM. The classifier results, shown on line 9, are the predicted output class and classifier confidence derived from the Platt model. In line 14, the classification confidence is compared to a predefined threshold, γ. If the margin exceeds the threshold, set to 0.8 in our experimentation for its relative balance between accuracy and efficiency, the classification result is retained and the search terminates. However, if the classification confidence is low, we evaluate the sensitivity of the classifier probability to each dimension,

and proceed along the direction of the maximum gradient in the positive direction. The gradient of function **f** is shown below, in which the unit vectors in both dimensions, length and index, are represented by **x** = <1,0> and **y** = <1,0>.

$$\nabla f = \frac{\partial f(x,y)}{\partial x}\mathbf{x} + \frac{\partial f(x,y)}{\partial y}\mathbf{y}$$

We then calculate the directional derivative $\forall\ \mathbf{u} \in \mathbf{K}$, where **K** is defined as a set consisting of the following vectors:

$$\mathbf{K} = \{<1,0>, <0,1>, <1,1>\}$$

The motivation for the selection of these particular vectors is the initial condition of the algorithm, which begins at the smallest possible index and vector size. If this were not the case, directional derivatives would be calculated in the positive and negative directions. Also, note that these vector values are array indices, rather than absolute values. Thus, applications which require a finer granularity of search could modify the values of variables α, β, m and n, as defined in Algorithms 7 and 6.

After computing the directional derivatives, we select vector **u** with the highest value, and continue iterating while two constraints are met:

$$\alpha < x < n \cdot \alpha \tag{6.4}$$

$$\beta < y < m \cdot \beta \tag{6.5}$$

These conditions ensure that parameters α and β remain within the valid range. However, if we find that the gradient value ∇f is negative for $\forall \mathbf{u} \in \mathbf{K}$, the search terminates, as we have reached a local maximum. This case is shown in Equation 6.6.

$$\nexists \mathbf{u} \in \mathbf{K} \ \mid\ \nabla f \cdot \mathbf{u} > 0 \tag{6.6}$$

Lastly, if we find x and y such that f(x,y) $\geq \gamma$, the search will terminate even if Equation 6.6 is satisfied, as the value of p is thought to have reached a point of diminishing

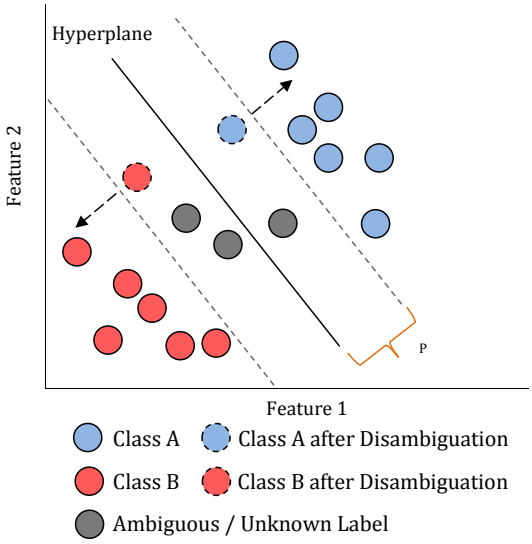


Figure 6.6: Relabeling instances original in the ambiguous region through parameter adjustment.

returns at this threshold, if selected appropriately. After the algorithm is completed, we have a final class label c and probability p. If p is low, it is likely that the instance is not a complete representation of an activity.

A possible outcome is shown in Figure 6.6. Note that several instances that were originally labeled as ambiguous are excluded from the list of false positives because adjustments to the window associated with these instances disambiguated their class label. However, in the case of several other instances, no modifications to the window were able to provide sufficient confidence of the node's true class label. Note that it is possible for a node to drift from one side of the hyperplane to another during the iterative process.

6.7 Experimental Setup

Data was collected from 20 individuals using a Hyperio Flexible Throat Microphone Headset placed in the lower part of the neck near the collarbone, connected directly to the mobile phones audio input port using a 3.5mm male audio cable. 16 of the subects were male, and 4 were female. The ages ranged from 21 to 31 years old, with a median age of 22. Commercially available audio-recording technology was used to acquire the audio recordings from the microphone. Twenty subjects, who were given two miniature chocolate bars, followed by ten Pringles potato chips. The foods were

Table 6.1: Partial List of Selected Features

Rank	Feature Name
1	Log Energy: Skewness
2	Log Energy: Mean Distance Between Peaks
3	Log Energy: Zero Crossings
4	Mel-Freq: Simple Moving Average[0] Quartile 3
5	Mel-Freq: Simple Moving Average[0] Mean Dist. Between Peaks
6	Mel-Freq: Simple Moving Average[1] Quartile 2
7	Mel-Freq: Simple Moving Average[1] Mean Dist. Between Peaks
8	Mel-Freq: Simple Moving Average[0] Zero Crossings

consumed sequentially, in that order, and the subjects ate one potato chip at a time. These recordings formed the basis of the algorithm design and experimental evaluation. The data collection took place in a lab environment; people can be faintly heard speaking in the background, and the microphone occasionally recorded doors closing and nearby footsteps.

6.7.1 Feature Extraction

The Munich open Speech and Music Interpretation by Large Space Extraction toolkit, known as openSMILE [EWS10], is a feature extraction tool intended for producing large audio feature sets. This tool is capable of various audio signal processing operations such as applying window functions, FFT, FIR filterbanks, autocorrelation, and cepstrum. In addition to these techniques, openSMILE is capable of extracting various speech related features and statistical features. Audio-based features include frame energy, intensity, auditory spectra, zero crossing rate, and voice quality. After data is collected from a variety of subjects eating several foods, feature selection tools can be used to identify strong features that are accurate predictors of swallows and bites for various foods, while reducing the dimensionality by eliminating redundant or weakly correlated features.

From the 6555 extracted features, the Correlation Feature Selection (CFS) Subset Evaluator was used to evaluate 991,139 subsets of features. This is necessary to select the features best associated with the desired classifier outcomes. This subset evaluator

considers both the individual predictive ability of features, as well as the redundancy between them. The top ten features are listed in Table 6.1.

6.7.2 Model Development

A total of 1000 data samples were generated from a smaller subset of 40 audio clips from the dataset containing information recorded from 20 subjects. The SVM kernel used in our evaluation was a radial basis function (RBF), which is used to map the original data to a higher dimension such that it is linearly separable by the hyperplane. The optimization technique used for the quadratic programming analysis necessary to generate the best-fit hyperplane was Sequential Minimal Optimization (SMO), which divides the original problem into smallest-subset problems that can be solved analytically. The implementation of these algorithms was provided by the WEKA Data Mining software, which was used for model development and evaluation[HFH09a].

In our experimentation, no labels were discarded; all instances were retained to allow an objective comparison between the Optimized Search, Exhaustive Search, and Baseline algorithms.

6.8 Results

6.8.1 Classification Results

Presented results were based on a total of 1000 data samples split evenly between each class, with 500 samples of chocolate and 500 samples of chips. 10-fold cross validation was used for algorithm evaluation. In other words, 90% samples were used for training and 10% for test for ten iterations, with a different test subset used for each iteration. The results were then averaged together for statistical convergence. The probabilistic model was derived from the same data as the training class, and was not developed using the full data set to prevent overfitting.

The classification results between the two food groups using the LibSVM classifier library are shown in Figure 6.7. Three results are shown in this figure based on the three evaluated algorithms: naive, exhaustive, and optimized. The naive approach was used

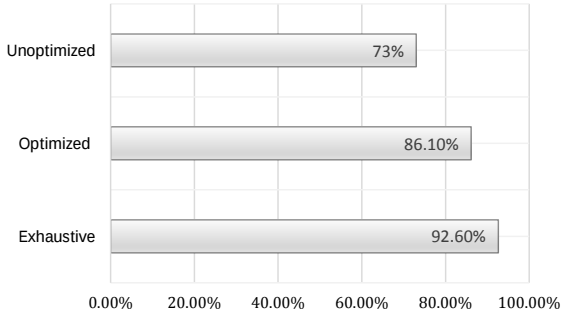


Figure 6.7: Percentage of correctly classified instances using a baseline with no optimizations, an exhaustive search, and optimized search.

as a baseline to evaluate classification accuracy without optimizations. Note that the relatively low classification accuracy is because the window selection was completely at random, and it is likely that there was no eating in some samples at all. The samples were generated at random from a larger data stream, in order to replicate a realistic use-case. With no optimizations, only 73% of instances were classified correctly. Using the optimized scheme, the percentage of correctly classified instances rose to 86.1%. However, as expected, the highest performance was achieved using the exhaustive search technique, in which 92.6% of instances were correctly classified.

In our experiments, the values of n and m were set to five. Recall that these values correspond with the number of possible audio clips generated from each original clip, with n different window lengths and m different starting indexes per length. The window sizes ranged from 500ms to 2.5 seconds, in increments of 500ms. The starting indexes varied by increments of 200ms, with a maximum of 1 second based on the m value of 5. The values for n and m chosen in our evaluation are a function of our particular classification problem; any more than a few seconds of window adjustment could re-center the window around a different activity. In other applications with longer duration events, or larger intervals of time between the activities of interest, these values may not apply.

6.8.2 Probability Model Validation

Figure 6.8 shows a comparison between predicted and measured classification accuracy. Recall that the training set was used to create a sigmoid-based model for mapping the SVM score to a probability. This model was then applied to the test set, to evaluate how

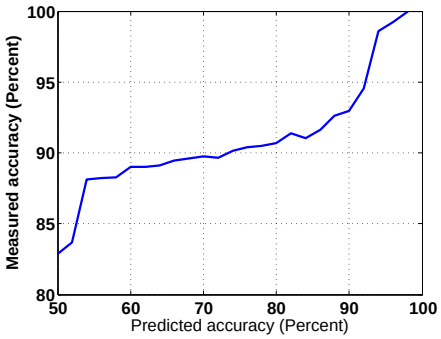


Figure 6.8: Validation of the trained probability model.

the posterior probability correlated with the reported probability. This figure was generated by listing all the test instances, along with their predicted classification accuracy using Platt Correlation. The predicted accuracy was then swept between 50%, the lowest possible accuracy for a binary dataset, and 100%. For each predicted accuracy value (x-axis), the y axis represents the percentage of instances with at least that high prediction confidence, which were correctly classified. For example, consider the point *(70,90)*. This can be interpreted such that, for all instances which had a predicted classification confidence of at least 70% based on the sigmoid model developed by the training set, 90% were classified correctly. Generally speaking, there were some discrepancies between the probability model and the observation, particularly at lower prediction accuracies. However, the correlation between predicted accuracy and measured accuracy is still sufficient for window size selection.

6.9 Future Works

We have validated that adjustments to the data window as a function of classification probability are capable of improving classification results. However, several relevant issues remain to be addressed in future work. First, the generalizability to non-acoustic signals should be explored. Second, the application of this technique to problems with multiple possible class labels should be investigated. Lastly, a more advanced heuristic for classification probability for other classifiers such as RandomForest and C4.5 decision trees should be developed to further generalize this approach.

6.10 Conclusion

In this work, we present a window-selection algorithm for identification of short-term events in continuous data streams, which we validate in an audio-classification testcase. By applying Platt Calibration to a Support Vector Machines (SVM) classifier, we are able to identify those instances most likely to be classified incorrectly. Through computationally efficient modifications of the data window, we are able to improve classification accuracy by identifying those instances whose class labels are most uncertain.

CHAPTER 7

Dynamic Computation Offloading for Low Power Wearable Health Monitoring Systems

Motivated by lowering user burden and increasing long-term adherence rates, optimization of battery life has emerged as one of the major challenges associated with the development of real-time wearable health monitoring systems. One of the major technique with which this objective can be achieved is *computation offloading*, in which portions of computation can be partitioned between the device and other resources such as a server or cloud. However, data offloading should only be used in specific situations as it is associated with higher RF power overhead than local processing. In this chapter, we describe a novel dynamic computation offloading scheme for real-time wearable health monitoring devices that adjusts the partitioning of computation between the wearable device and mobile application as a function of desired classification accuracy. By making the correct offloading decision based on current system parameters, we show that we are able to reduce system power by as much as 20%.

7.1 Introduction

7.1.1 Introduction to Wearables

As smartphones have entered ubiquity in recent years, various wearable wireless health monitoring gadgets have been proposed. These devices have broad applications ranging from physical activity monitoring [Misb, Jaw], to more experimental applications such as diet tracking [KAL15a, JCY13, SSL09], mental stress detection [SKC12], smoking [PCC14], and rehabilitation [PPB12]. Spurred by a rapidly aging global population and the emergency of lightweight, affordable microelectronics, wearable devices will have increasingly broad applications for consumers and clinicians [LSS08] in the years to come.

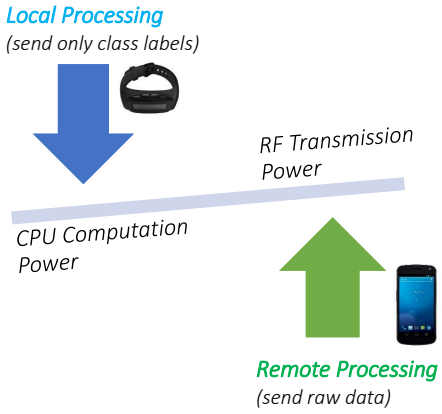


Figure 7.1: Wearable devices must evaluate the performance penalty of RF transmission when deciding to perform data computation locally vs. remotely.

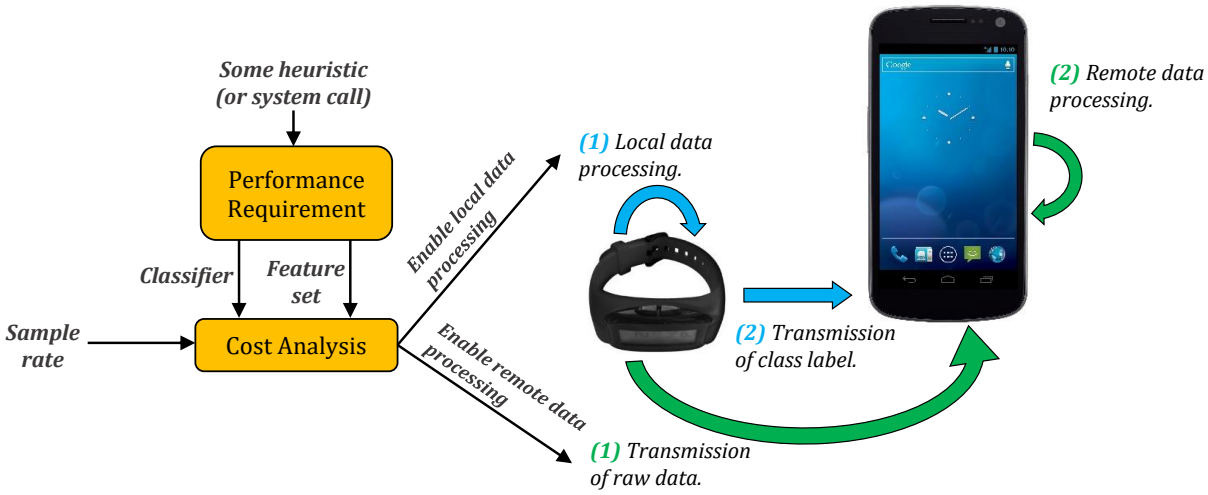


Figure 7.2: System flow of proposed computation offloading scheme.

One of the most significant challenges in medicine is *non-adherence*, as prior studies have shown that non-compliance to treatments is associated with a host of negative health outcomes [DM01, Bla76]. Wearables are not exempt from this challenge, as evidence overwhelmingly shows a lack of sustained use associated with mobile health devices and applications [MHK15]. Among other factors, battery life has been reported to be a significant contributor to non-adherence [AEN14, MHK15], as frequent battery recharging may present a repeated burden to the user. Moreover, the device cannot report user activity when the battery has been depleted, resulting in a loss of data. For these reasons, an increasing amount of attention has been directed towards improving the battery life of wearable health monitoring devices is recent years.

7.1.2 Energy Usage

Regardless of function or application, wearables have many fundamental architectural similarities. The primary components typically consist of:

7.1.2.1 Sensors

One or more sensors, sampled at a particular frequency depending on the application. Examples include accelerometers, gyroscopes, infrared sensors, and microphones. These sensors are the foundation through which the device can detect user activity.

7.1.2.2 Microcontroller

A microcontroller that performs the sensor sampling, processes the data, and interfaces with other peripherals. Generally, microcontrollers are designed for low power consumption some, such as the TI MSP430, can operate for weeks on a coincell battery at supply voltages such as 3.3v and 5.0v.

7.1.2.3 Transceiver

A wireless transceiver based on a technology such as Bluetooth, Zigbee, or Wifi, which transmits data remotely to a mobile phone or cloud services for analysis, visualization, and feedback. Bluetooth 4.0 LE appears to be one of the more predominant wireless technologies in modern wearable devices, and features several important power optimizations in comparison with the earlier Bluetooth standards.

There are two predominate forms of power usage in a wearable device: ***wireless transmission***, and ***local computation***. Wireless transmission overhead refers to the power necessary to transmit data from the microcontroller to a mobile application using a technology such as Bluetooth. Local computation overhead refers to the power requirements of data processing performed on the wearable device's microcontroller, before transmission. Unfortunately, as Figure 7.1 illustrates, it is often the case that optimizing the energy of the microcontroller and wireless transceiver are diametrically opposing goals. Assume there is some feature or property f, that we are interested in detecting

from a continuous signal. An example of f could be a step (for a pedometer), a bite (for a nutrition monitor), or a fall (for a gait monitor). There are two possible approaches to detect such an event. The first option is to process the data locally on the wearable device, and only transmit the minimal information: that we have detected f. This puts a significant burden on the computation resources of the microcontroller, as processing the sensor data can be expensive and resource intensive. However, in this approach it is only necessary to transmit the final *class label* to the mobile phone, rather than the raw data, which minimizes total Bluetooth overhead.

The second option is to perform no data processing on the local device, deferring the processing to the smartphone. This approach saves local computation energy on the microcontroller, as it is no longer required to do any expensive processing or classification; the microcontroller can transmit the sensor data as soon as it is acquired. However, this approach may result in higher wireless transmission overhead, which in many cases can be even more than that of local computation. The decision on whether to process data locally or remotely is a function of many parameters, such as Bluetooth connection interval, sample rate, classifier choice, and more.

7.1.3 Computation Offloading

Computation offloading is a broad paradigm in which data is outsourced from local computation to a server, cloud, or other form of aggregator. The primary objective of computation offloading is to reduce the energy demands of a small microcontroller with a low battery capacity, or to perform very resource-heavy operations on more powerful hardware for performance reasons. The motivation behind this approach is that mobile phones typically have much larger batteries than small wearable devices, just as servers have more hardware resources than personal computers.

Computation offloading techniques typically falls under the categories of *static* and *dynamic* schemes. In static schemes, the partitioning between the local and remote node is made in advance. By contrast, dynamic schemes adjust the offloading based on real-time conditions. In this chapter, we propose a novel algorithm for dynamic computation offloading, targeted towards real-time wearable health monitoring applications. In [KLL13], Kumar et al. provide a survey of computation offloading strate-

gies. Many other works have discussed different strategies for offloading computation [KHR03, KL10, GNM03, LKL10]. However, our work focuses specifically on modern cutting-edge wearable devices, emphasizing the tradeoffs between local computation and Bluetooth transmission overhead as a function of the required classification accuracy.

This chapter is organized as follows:

- In *Section II*, we describe the components of a typical classification system flow.
- In *Section III*, we present our energy models.
- In *Section IV*, we describe our offloading strategy.
- In *Section V*, we describe our experimental setup.
- In *Section VI*, we present our experimental results.
- In *Section VII*, we provide concluding remarks.

7.2 Classification Flow

In this section, we begin with the preliminaries of a modern real-time health monitoring system. An example architecture for such a system consists of the steps shown in Figure 7.3. The major components are: acquisition of data, segmentation, feature extraction, and classification.

7.2.1 Acquisition

In this stage, the microcontroller on the wearable device acquires data from a sensor and buffers it locally until the buffer is full. The choice of sample rate f_s is critical at this stage, and is dependent on the type of sensor used. For example, 100 Hz may be sufficient for activity monitors, while microphones may require sampling rates of up to 44 kHz. The choice of sample rate has substantial implications on total system battery life for a number of reasons. First, more data samples may require more Bluetooth transmissions in cases when computation is offloaded. In cases when computation is performed locally, there is still additional overhead associated with processing more data.

7.2.2 Segmentation

In the next stage, the signal is divided into shorter windows, each of which are typically processed independently of one another. This is necessary, because it is often impractical to assign a single class label to a very large dataset. In practice, the data must be divided into fragments that are each assigned a class label. Windows that are not associated with any particular activity are known as the *null* class, which can be discarded after processing. The three primary methods in literature are sliding window approaches, similar to the baseline used in this chapter, and recursive techniques that are either partition the entire signal, or merge small denominations of the signal, until a stopping criteria is met. Comprehensive surveys of time-series segmentation are provided by Keogh et al. in [KCH04], and Lovri et al. in [LMS]. We refer the reader to these works for a more detailed discussion of the topic [XZK12, OK92, KEA02, KJ00].

7.2.3 Extraction

From each window, a set of features are extracted. An example of a feature is to take the Fourier transform of the signal, and extract the magnitude of a particular frequency band. As expected, this extraction is associated with computational overhead. However, not all features have the same cost; many features are simple lookups, while others require complex transformations. The number of features that must be extracted at this phase can also have a significant impact on total system power; larger feature sets may provide higher classification accuracy in some cases. The features extracted during this step are pre-selected during the classifier training process using various feature selection and dimension reduction algorithms. However, the feature *extraction* must be performed in real-time.

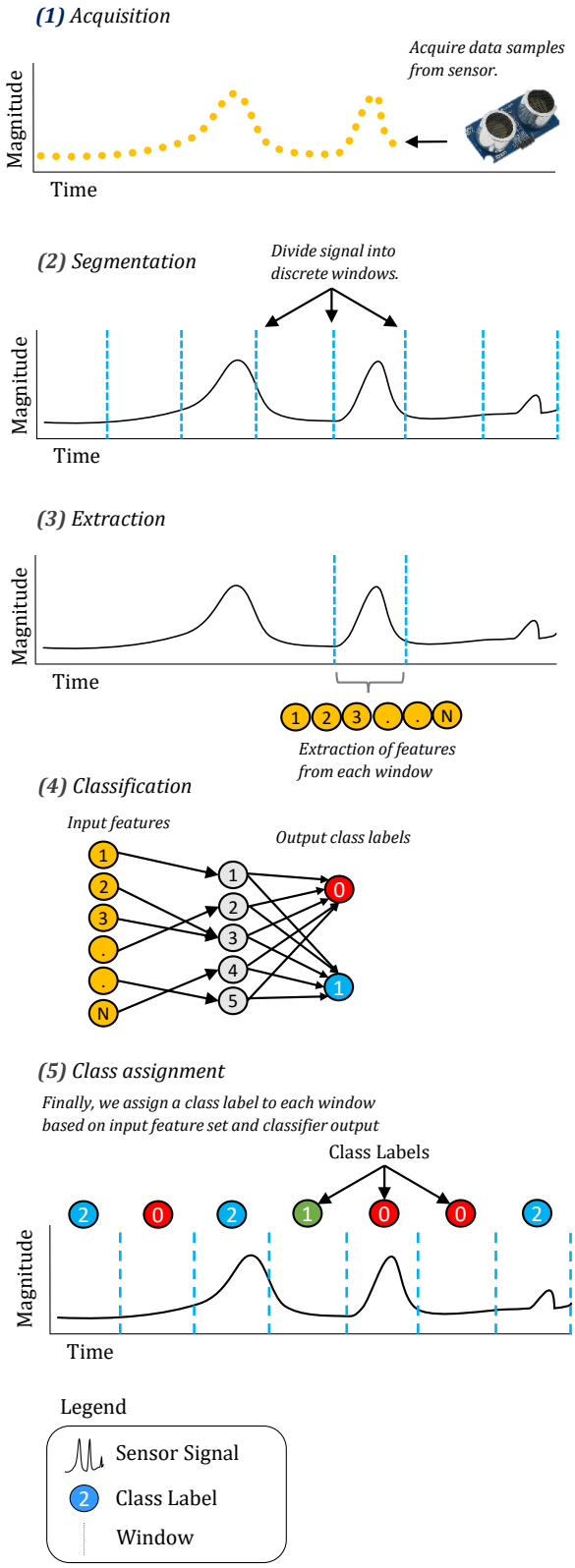


Figure 7.3: The overall classification model.

Table 7.1: Definition of Terms

Term	Description
C_x	Cost (in terms of power) to execute operation x
F	A set of features associated with a window.
f_s	Sample rate: the rate at which data is acquired from the sensor.
M	Function that maps feature set, classifier, and set size to a cost.
K	Variable that represents the classification algorithm used (eg. RandomForest), BayesNet.
n	Number of features extracted.
BW	Bandwidth: the rate at which data is transmitted wirelessly between device and phone.
λ	Connection interval: how often a connection is established between two Bluetooth devices.

7.2.4 Classification

The features are the inputs to a pre-trained classifier, which outputs a class label that describes the actions represented by the window. Once again, the classifier may be trained a priori. However, a real-time system would generally require that the classifier be run periodically for user feedback. However, not all of these tasks can be offloaded. In almost all cases, signal acquisition would take place on the wearable device. And while signal segmentation can be a complex challenge depending on the heuristic used, we assume a fixed-length segmentation for the purposes of this chapter. This leaves dynamic offloading optimizations to feature extraction, and classification, both of which can be performed locally or remotely.

7.3 Energy Modeling

In this section, we describe our model for power dissipation in a wearable health-monitoring device. An explanation of terms and symbols is provided in Table 7.1.

7.3.1 A General Model

A simplified model to represent power consumption in a system consisting of a wearable device and a paired smartphone can be represented as the sum of the cost of extracting features, running them through a classifier, and transmitting any necessary information between the phone and device. Though there are other sources of power dissipation such as sampling, segmentation, and leakage, we focus our analysis on the model shown in Equation 7.1 in this chapter.

$$C_{total} = C_{extraction} + C_{classification} + C_{transmission} \tag{7.1}$$

Assume we have a feature set F, consisting of a total of n features, f_1 through f_n. Similarly, assume each feature f_i has a cost, C_{f_i}. Thus, the cost of the feature extraction phase is:

$$C_{extraction} = \sum_{i=1}^{n} C_{f_i} \tag{7.2}$$

Modeling the cost of the classification is more challenging. Generally, however, it is a function of the input feature set F, feature size n, and the classifier algorithm used. We define function $\mathbf{M}$ as a function that maps these parameters to the total classifier cost.

$$C_{classification} = \mathbf{M}(F,\ K,\ n) \tag{7.3}$$

If we perform both the feature extraction and classification locally, the local energy can be modeled as the sum of the feature extraction and classification costs.

$$C_{local} = \mathbf{M}(F,\ K,\ n) + \sum_{i=1}^{n} C_{f_i} \tag{7.4}$$

We can also model the local cost of transmitting the raw data to the mobile device, and performing both the feature extraction and classification remotely. We assume these features are associated with a window of length L. Thus, the cost of transmitting a window of length L is: $\mathbf{C}_{tx}(L)$. Note that this relationship may not be linear; very high sampling

rates may change critical Bluetooth parameters such as the connection and advertisement intervals. Combining these equations gives us the equilibrium point, in which it is roughly equally costly to process the data locally and remotely:

$$\mathbf{M}(F,\ K,\ n) + \sum_{i=1}^{n} C_{f_i} = \mathbf{C_{tx}}(L) \tag{7.5}$$

An adaptive system could modify several parameters in real-time, to favor either local or remote processing:

7.3.1.1 Algorithm choice

Different classifiers, such as RandomForest, Support Vector Machines, and k-Nearest Neighbor, have different runtimes. Switching to a lighter classifier can swing the balance in favor of local processing at the cost of classification accuracy.

7.3.1.2 Feature count

Choosing a different subset of features may have dramatic performance implications. A higher number of input features may, in some cases, improve classification accuracy. However, the relationship is not linear; very large feature sets may overfit the training data and decrease total classification accuracy.

Each classifier **M** can be represented by its classifier choice and feature count, and is associated with a particular classification accuracy and power budget. The subsequent challenge is to identify the circumstances in which it is appropriate to vary the desired classification accuracy to save power. However, the specific scenarios are out of scope for this work; we refer the readers to [BP07, GJ13, SG08] for a discussion of these issues.

7.3.2 Connection Interval

The Bluetooth 4.0 LE standard, used in many wearable devices, allows peripherals to suggest a connection interval which specifies how often the device sends data. The Bluetooth standard supports connection intervals ranging from 7.5 ms to four seconds; higher connection intervals reduce system bandwidth, but significantly reduce power consump-

tion as well. Typically, during each connection interval, several Bluetooth packets can be transmitted. However, the Nordic nRF8001 used in our evaluation supports transmission of just one packet [Ole14] per connection event. The payload of each packet is 20 Bytes. According to [Ole14], the total Bluetooth connection bandwidth can be represented as a function of connection interval γ, as shown below in Equation 7.6:

$$BW_{BT} = \frac{20\ bytes}{packet} \cdot \frac{1\ packet}{conn.\ event} \cdot \frac{1\ conn.\ event}{\gamma\ seconds} \tag{7.6}$$

Generally, we would select a connection interval to satisfy a particular bandwidth requirement. Simplifying Equation 7.6 gives us:

$$\gamma = \frac{20\ bytes}{Bandwidth} \tag{7.7}$$

Expectedly, higher sample rates (f_s) are typically more expensive to process and transmit. In accelerometer-based activity monitoring applications, a sample rate of between 25 Hz and 100 Hz is typical, with some studies claiming that 45 Hz is optimal [YH10, KHM16]. Remote computation schemes will generally require that all sampled data is transmitted to the mobile phone for processing. If we assume each data sample is a 32-bit integer (four bytes), we can model the required connection interval as a function of sample rate below:

$$\gamma = \frac{20\ bytes}{4 \cdot f_s} \tag{7.8}$$

By comparison, local processing schemes can have almost negligible wireless transmission overhead, as it is only necessary to transmit a class label once an event is detected.

7.4 Algorithm

7.4.1 Classifier Accuracy Adjustment

Many prior works have scaled down accuracy to save power, for various applications including classification and health monitoring. For example, Benbasat et al. propose a

power efficient sensor system in [BP07], in which a wearable gate monitor is optimized by adjusting classification accuracy. In [GJ13], Ghasemzadeh et al. propose a two-tiered classification scheme in which preliminary classification is achieved using lightweight, low-power techniques. A similar two-stage scheme was proposed by Shih et al. in [SG08] in which a power-efficient screening stage precedes a more computationally expensive analysis stage. The key insight these works is that it is not always necessary to run the classifier at its highest accuracy setting; often, a low-power detection strategy can be used, which transitions into a more expensive recognition stage when various criteria are met [Hua12, JC04, MS03].

Dynamic optimization of classification accuracy for power reduction is particularly useful in scenarios with sparse, short duration events distributed across an entire day or week of data. For example, a heart-rate monitor could be optimized to enter low-power mode when a coupled accelerometer shows little physical activity. Moreover, adjusting the sample rate of an accelerometer when a subject begins to move has been shown to maintain high classification accuracy, while reducing power when more simple motions are performed [BlN15]. Though these prior works are able to successfully reduce power consumption, they generally do not evaluate the tradeoffs between wireless transmission and local computation upon changes in classification accuracy.

7.4.2 Dynamic Offloading

The proposed computation offloading scheme is as follows. First, the system pre-trains n classifiers, $\mathbf{M_1}...\mathbf{M_n}$ Each classifier has the objective of maximizing total classification accuracy for a given power budget, and may have a different number of input features. Based on the desired sample rate, we can predict the benefits of local and remote processing using Equation 7.5 and select one of the two schemes. When the user program specifies a need to improve the classification accuracy based on some external inputs, we iterate through the n possible classifiers and select classifier $\mathbf{M_i}$ that which minimizes power consumption based on the required accuracy threshold. Once this classifier is selected, its predicted cost is computed using Equation 7.5, and a decision is made with respect to local and remote processing overhead. This procedure is shown in Algorithm 7.

Algorithm 7: Computation Offloading

```
/* This function is called whenever the system specifies that the classification
accuracy can be adjusted. pct represents the new accuracy requirement. */
function OnChangeAccuracyRequirement(int pct)
Begin
    /* We select the optimal classifier which minimizes power but meets the
    performance requirement. */
    Classifier K = getClassifier (pct);
    /* We estimate local power based on the classifier and feature count. */
    Power LocalPower = EstimatePower(K.Classifier, K.FeatureCount);
    /* We estimate wireless power based on the Bluetooth connection interval and
    payload size. */
    Power RemotePower = EstimateRFPower(L, λ);
    if LocalPower > RemotePower then
        /* We determine that local processing is optimal. */
        RunLocally();
    else
        /* We determine that transmitting all the data is cheaper than processing it
        locally. */
        OffloadComputation();
End
```

Specifically, once we have selected a potential classifier and feature set, we can compare the power consumption of local data processing (in which the features are extracted locally, and only class labels are transmitted) to remote data processing (in which no features are extracted on the device, but all data is transmitted). This process continues until the next classification accuracy adjustment. Figure 7.4 shows the proposed scheme, which depicts both local processing regions, based on the desired classification accuracy. The point denoted by the star represents the condition shown in Equation 7.5; equal local and remote processing costs. The cost to process data locally is modeled as a function of two parameters: α and β. Parameter α represents the cost to extract a single feature, while β represents the cost to run a classifier on the local application. We assume the classification cost on the mobile device is negligible as the battery life is an order of magnitude larger, and the focus of our work is primarily in the optimization of the wearable device.

7.5 Experimental Methodology

7.5.1 Dataset

Our experimental methodology was derived from an audio-based nutrition monitoring dataset described in [KS15a]. Data was collected from 20 individuals using a Hyperio Flexible Throat Microphone Headset. The microphone was placed in the lower part of the neck near the collarbone, and connected directly to the mobile phones audio input port using a 3.5mm male audio cable. 16 of the subects were male, and 4 were female. The ages ranged from 21 to 31 years old, with a median age of 22. Commercially available audio-recording technology was used to acquire the audio recordings from the microphone.The primary challenges of nutrition monitoring is the identification of eating (such as chewing or swallowing) from other ambient noises, and identifying the specific food using various heuristics. The dataset used corresponded to eating of three foods: nuts, chocolate, and a vegetarian patty. Each food category consisted of sixty quarter-second samples, for a total of 180 samples. Evaluation was conducted using Leave-One-Subject-Out cross validation to avoid biasing the dataset.

Table 7.2: Top 10 Selected Features

Rank	Attribute (FFTMag)	Information Gain
1	mfcc sma[5] quartile2	1.292
2	fftMag melspec sma[5] quartile1	1.286
3	mfcc sma[5] amean numeric	1.239
4	fftMag melspec sma[8] quartile1	1.231
5	fftMag melspec sma[7] quartile1	1.21
6	fftMag fband250-650 sma quartile1	1.202
7	fftMag melspec sma[10] quartile1	1.194
8	fftMag fband0-650 sma quartile1	1.18
9	fftMag melspec sma[2] quartile1	1.17
10	fftMag melspec sma[6] quartile1	1.16

7.5.2 Feature Extraction

The feature extraction tool used on the audio dataset was provided by the OpenSMILE framework [EWS10]. This tool is capable of various audio signal processing operations such as applying window functions, FFT, FIR filterbanks, autocorrelation, and cepstrum. In addition to these techniques, openSMILE is capable of extracting various speech related features and statistical features. Audio-based features include frame energy, intensity, auditory spectra, zero crossing rate, and voice quality. After data is collected from a variety of subjects eating several foods, feature selection tools can be used to identify strong features that are accurate predictors of swallows and bites for various foods. The top ten selected features is shown in Table 7.2. These features are produced using an InformationGain attribute evaluation scheme provided by the WEKA data mining software [HFH09b]. For a detailed explanation of these features and a more qualitive explanation of what they represent, we refer the reader to the openSmile documentation [EWS10]; the specific setting used was the emo_large configuration

7.5.3 Bluetooth Modeling

Power simulations were conducted in the nRFgo Studio software, based on the nRF8002 integrated circuit by Nordic Semiconductor. The simulations assumed no advertisement period, and a default connection interval of 1000 ms. We initially simulated two use cases: one in which 20 bytes (payload) of data are transmitted at a rate of 100 Hz, and another in which data is transmitted at a rate of 1 Hz. These two conditions correspond with remote processing (sending all the data) and local processing (sending class labels) respectively. Subsequently, we experimented with various sensor sample rates and adjusted the connection interval accordingly based on the increase in bandwidth.

7.5.4 Computation Modeling

The WEKA data mining software is used to evaluate the performance of various classifiers based on the input features extracted from the openSMILE toolkit from each audio snippet from the nutrition monitoring dataset. The performance of all classifiers was normalized based on our prior results on the MSP430 platform in [KAP15], in which we measured the performance of various algorithms in real-time on a Texas Instruments development board. Similarly, the cost of extracting a particular feature is derived based on our prior work in [KAP15], with the simplifying assumption that each feature has the same cost.

7.6 Results and Discussion

7.6.1 Classifier Performance

Figure 7.5 shows the classification accuracy among four classifiers, as a function of feature set size. Note that the classification accuracy does not linearly increase with the number of features extracted; in some cases, more features are detrimental to performance due to overfitting. Moreover, some classifiers such the Naive Bayesian classifier, perform well with small feature sets but fail to improve significantly with greater numbers of attributes. Generally, this figure shows that approximately five features are sufficient for classification accuracy of over 75% for most classifiers, approaching 83% in the case of

Logistic Regression. The upper bound for feature count, beyond which there appears to be no significant improvement, appears to be approximately twenty features with 87-88% accuracy.

Figure 7.6 shows classifier runtime as a function of feature size and classifier choice. Classifier runtime on an embedded microcontroller can be an approximate heuristic for power consumption, as higher algorithm runtimes increase the percentage of time the device is in active mode, rather than one of the low-power modes in which output current is close to in the microamperes range [KAP15]. Simulations show that the Logistic Regression classifier was associated with the lowest runtime, across the entire range of features. Moreover, the runtime of the logistic regression classifier did not noticably increase as a function of number of input features. By contrast, the Naive Bayesian classifier was quick with a small number of features, but did not scale well with larger feature sets, but runtime increased linearly with the feature count. The RandomForest classifier had a somewhat high runtime, especially with small input feature sets. However, the runtime did not increase when sweeping the feature size between 2 and 20.

The implications of these results are shown in Table 7.3: for each desired accuracy level, we can select an optimal combination of classifier and classification feature size. In this case, Logistic Regression was the optimal choice for almost every scenario except accuracy of 85%, in which Sequential Minimal Optimization (SMO) was preferred by a small margin. Though Logistic Regression has a lower *runtime* cost, the *extraction* cost becomes much more significant for larger feature sets.

Table 7.3: Best Classifier and Feature Set Combinations For a Given Accuracy

Accuracy (%)	Lowest-Cost Classifier	Feature Set Size
70%	Logistic	2
75%	Logistic	2
80%	Logistic	6
85%	SMO	20
90%	Logistic	20

7.6.2 Comparison of Local and Remote Processing

Assuming a connection interval of 10 ms and a payload size of 20 bytes, simulator results show average power as 964 μw. Figures 7.7 and 7.8 show the power overhead of both techniques as a function of various values of α and β. More specifically, these graphs show how much power would be spent in a remote processing scheme compared to a local processing approach, at various desired accuracy levels. A power ratio of 1, shown in black, represents the equilibrium case in which local and remote processing powers are equal.

More specifically, we can observe that lower feature extraction costs, as well as lower desired classification accuracies, are dramatically cheaper to execute locally rather than remotely. However, as the required classification accuracy increases, it is often necessary to use more expensive classifiers and larger feature sets. Thus, these schemes favor remote processing, particularly in systems with more expensive features.

7.6.3 Variations in Sample Rate

Our previous experiment assumed a connection interval of 10 ms. This connection interval represents one Bluetooth LE connection per second, at which time only one 20-Byte payload can be transmitted based on the limitations of the nRF8001 integrated circuit. Using Equation 7.8, we know that this connection interval corresponds with a maximum bandwidth of 500 samples per second on our evaluation platform. Though there are applications for which a high sampling rate is sufficient, smaller sample rates can suffice for other uses cases such as environment monitors, dust particle detection, ambient light detection, or smart-home applications. In this section, we analyze the effects of sample rate on transmission power.

Figure 7.9 shows the Bluetooth power at a 3.3 V supply voltage as a function of sample rate. Interestly, this figure shows that as bandwidth needs increase, wireless power dissipation is less than linear. By comparison, Figure 7.10 shows the local power dissipation as a function of sample rate and accuracy thresholds. Our model shows a more linear relationship between power and sample rate, given parameters $\alpha = 0.1$ and $\beta = 0.5$. This may suggest high sample rates favor offloading, rather than local computation.

7.6.4 Variations in Sample Rate

For a baseline comparison, we consider a scenario in which the system attempts to detect a short-duration event from a large signal in a scenario with the baseline configuration. The system specifications are to operate at 90% accuracy for 10% of the time, and enter low-power mode (70% accuracy) for the remainder of the time. For this experiment, we assume parameters α of 0.1 and β of 0.5. This scenario is shown at the top of Figure 7.11. As this figure shows, the local processing scheme is optimal at lower accuracies, while remote offloading is favored at high accuracies. By adapting computation as a function of classification accuracy, we are able to reduce power from an average of 0.52 mW to 0.41 mW; a decrease of 21.1% compared to the local processing scheme.

7.7 Conclusion

In summary, we have demonstrated a novel scheme for selective computation offloading based on user-defined accuracy constraints. Our simulations show that with a baseline classifier execution cost of 0.2 mW and feature extraction cost of 0.1 mW, making a correct offloading decision can reduce power by over 20% in certain scenarios. Future work will evaluate these algorithms in a system feature an implementation of real-time classification accuracy adjustment to benchmark our proposed scheme.

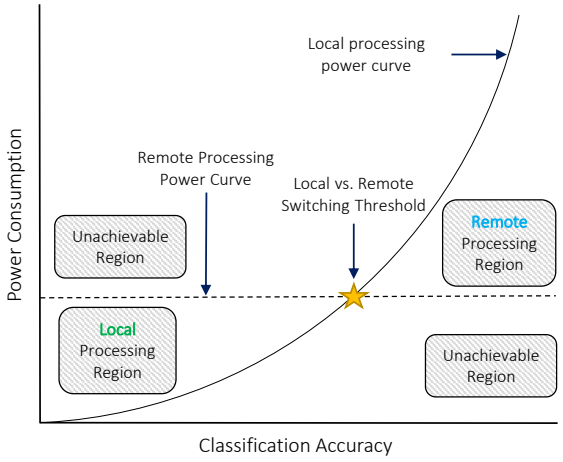


Figure 7.4: Tradeoff between local and remote processing.

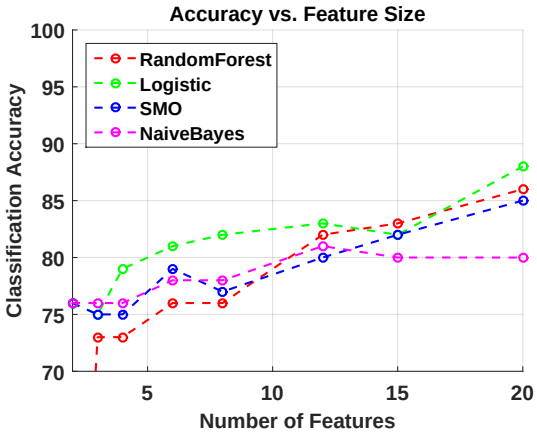


Figure 7.5: Variations in accuracy as a function of classifier, and feature set size.

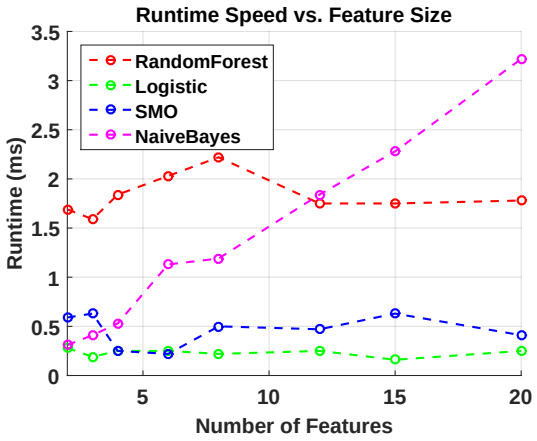


Figure 7.6: Variations in runtime speed as a function of classifier, and feature set size.

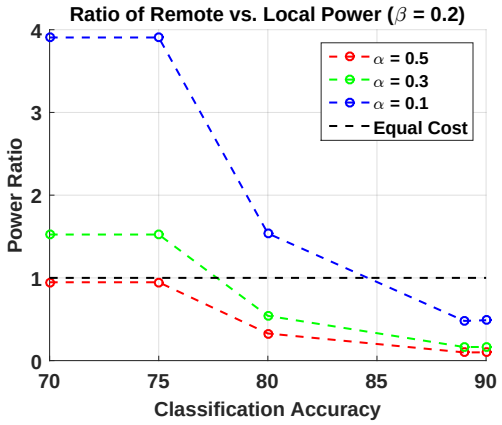


Figure 7.7: Ratio of power in the remote and local processing schemes at various values of α.

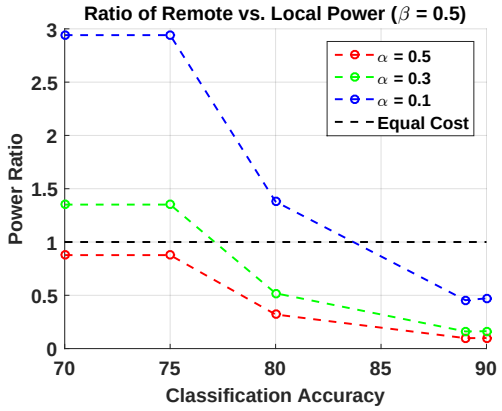


Figure 7.8: Ratio of power in the remote and local processing schemes at various values of α.

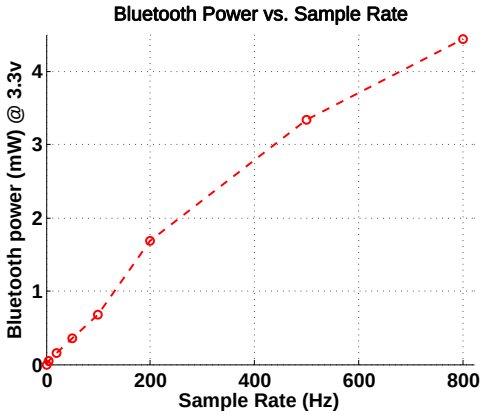


Figure 7.9: Nonlinearity between sample rate and Bluetooth transmission power.

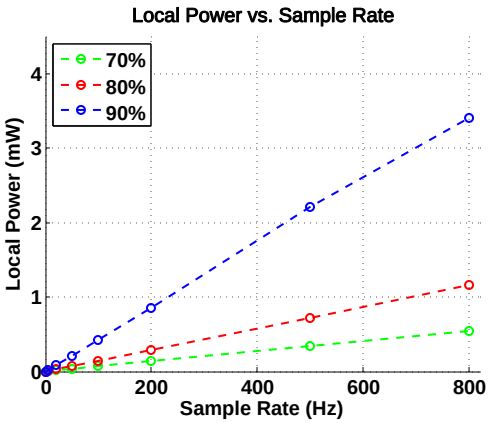


Figure 7.10: Relationship between local computation power and sample rate for three different accuracies.

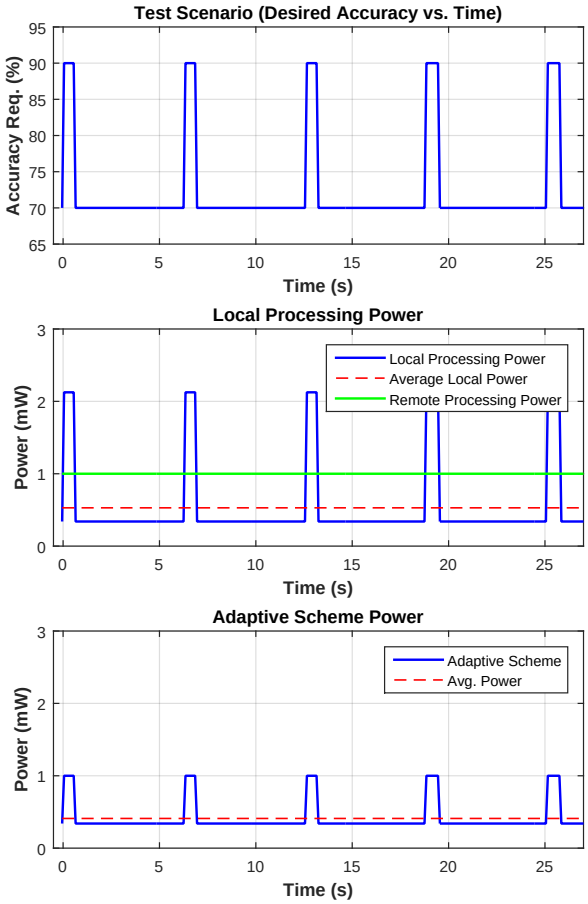


Figure 7.11: Comparison between local, remote, and adaptive schemes in a system which alternates between low and high accuracy requirements.

CHAPTER 8

Concluding Remarks

In this thesis, we have provided a survey of the major nutrition monitoring paradigms, as well as our own techniques for non-invasive assessment of dietary habits. We have introduced a necklace-based approach to detection of deglutition and classification between different foods, called Wearsens, in which a piezoelectric sensor detects vibrations of the neck that occur naturally during swallows. We have also described other techniques that can be applied to detecting eating behavior and medication adherence using audio signal processing and gesture recognition on the smartwatch platform. Lastly, we have introduced several techniques to efficiently segment and process sensor data in real-time using a stochastic segmentation approach as well as a computation offloading scheme to reduce power.

References

[ABM10] Akin Avci, Stephan Bosch, Mihai Marin-Perianu, Raluca Marin-Perianu, and Paul Havinga. "Activity recognition using inertial sensing for healthcare, well-being and sports applications: A survey." In *Architecture of computing systems (ARCS), 2010 23rd international conference on*, pp. 1–10. VDE, 2010.

[AEN14] Nabil Alshurafa, JoAnn Eastwood, Suneil Nyamathi, Wenyao Xu, Jason J Liu, and Majid Sarrafzadeh. "Battery optimization in smartphones for remote health monitoring systems to enhance user adherence." In *Proceedings of the 7th international conference on PErvasive Technologies Related to Assistive Environments*, p. 8. ACM, 2014.

[AKP14] Nabil Alshurafa, Haik Kalantarian, Mohammad Pourhomayoun, Shruti Sarin, Jason Liu, and Majid Sarrafzadeh. "Non-Invasive Monitoring of Eating Behavior Using Spectrogram Analysis in a Wearable Necklace." In *IEEE EMBS Healthcare Innovations & Point of Care Technologies (HIPT)*, 2014.

[AKP15] Nabil Alshurafa, Haik Kalantarian, Mohammad Pourhomayoun, Jason Liu, Shruti Sarin, and Majid Sarrafzadeh. "Recognition of Nutrition-Intake using Time-Frequency Decomposition in a Wearable Necklace using a Piezoelectric Sensor." *IEEE Sensors Journal*, 2015.

[AKT09] Oliver Amft, Martin Kusserow, and Gerhard Troster. "Bite Weight Prediction From Acoustic Recognition of Chewing." *IEEE Trans. Biomed. Engineering*, **56**(6):1663–1672, 2009.

[Amf10] O. Amft. "A wearable earpad sensor for chewing monitoring." In *Sensors, 2010 IEEE*, pp. 222–227, Nov 2010.

[ami14] "Amiko: Medication Management Made Easy." https://www.indiegogo.com/projects/amiko-medication-management-made-easy, 2014.

[ASL05] Oliver Amft, Mathias Stger, Paul Lukowicz, and Gerhard Trster. "Analysis of Chewing Sounds for Dietary Monitoring." In Michael Beigl, Stephen Intille, Jun Rekimoto, and Hideyuki Tokuda, editors, *UbiComp 2005: Ubiquitous Computing*, volume 3660 of *Lecture Notes in Computer Science*, pp. 56–72. Springer Berlin Heidelberg, 2005.

[AT06] O. Amft and G. Troster. "Methods for Detection and Classification of Normal Swallowing from Muscle Activation and Sound." In *Pervasive Health Conference and Workshops, 2006*, pp. 1–10, Nov 2006.

[AT08] Oliver Amft and Gerhard Tröster. "Recognition of dietary activity events using on-body sensors." *Artificial intelligence in medicine*, **42**(2):121–136, 2008.

[AXL14] Nabil Alshurafa, Wenyao Xu, Jason J Liu, Ming-Chun Huang, Bobak Mortazavi, Christian K Roberts, and Majid Sarrafzadeh. "Designing a robust

activity recognition framework for health and exergaming using wearable sensors." *Biomedical and Health Informatics, IEEE Journal of*, **18**(5):1636–1646, 2014.

[BAK96] R Brian Haynes, K Ann McKibbon, and Ronak Kanani. "Systematic review of randomised trials of interventions to assist patients to follow prescriptions for medications." *The Lancet*, **348**(9024):383–386, 1996.

[BFK05] Carol A Bova, Kristopher P Fennie, George J Knafl, Kevin D Dieckhaus, Edith Watrous, and Ann B Williams. "Use of electronic monitoring devices to measure antiretroviral adherence: practical considerations." *AIDS and Behavior*, **9**(1):103–110, 2005.

[Bla76] Barry Blackwell. "Treatment adherence." *The British Journal of Psychiatry*, 1976.

[bli15] "The Most Accurate Smart Blister in the World." http://www.informationmediary.com/med-ic, 2015.

[BlN15] Szymon Bobek, Mateusz layski, and GrzegorzJ. Nalepa. "Capturing Dynamics of Mobile Context-Aware Systems with Rules and Statistical Analysis of Historical Data." In Leszek Rutkowski, Marcin Korytkowski, Rafal Scherer, Ryszard Tadeusiewicz, Lotfi A. Zadeh, and Jacek M. Zurada, editors, *Artificial Intelligence and Soft Computing*, volume 9120 of *Lecture Notes in Computer Science*, pp. 578–590. Springer International Publishing, 2015.

[BP07] Ari Y. Benbasat and Joseph A. Paradiso. "A Framework for the Automated Generation of Power-efficient Classifiers for Embedded Sensor Nodes." In *Proceedings of the 5th International Conference on Embedded Networked Sensor Systems*, SenSys '07, pp. 219–232, New York, NY, USA, 2007. ACM.

[BPB92] DENISE BROCHETTI, MARJORIE P. PENFIELD, and SAMUEL B. BURCHFIELD. "SPEECH ANALYSIS TECHNIQUES: A POTENTIAL MODEL FOR THE STUDY OF MASTICATION SOUNDS." *Journal of Texture Studies*, **23**(2):111–138, 1992.

[Bre01] Leo Breiman. "Random forests." *Machine learning*, **45**(1):5–32, 2001.

[BWS05] L. E. Burke, M. Warziski, T. Starrett, J. Choo, E. Music, S. Sereika, S. Stark, and M. A. Sevick. "Self-monitoring dietary intake: current and future practices." *J Ren Nutr*, **15**(3):281–290, Jul 2005.

[CB08] A.M. Coulston and C. Boushey. *Nutrition in the Prevention and Treatment of Disease*. Academic Press/Elsevier, 2008.

[cdd14] "Centers for Disease Control and Prevention: Adult Obesity Facts.", 2014.

[CKJ14] Chen Chen, Nasser Kehtarnavaz, and Roozbeh Jafari. "A Medication Adherence Monitoring System for Pill Bottles Based on a Wearable Inertial Sensor." In *Proceedings of the* 36^{th} *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 4983–4986, Chicago, IL, August 2014.

[DHA13] Lindsey Dayer, Seth Heldenbrand, Paul Anderson, Paul O Gubbins, and Bradley C Martin. "Smartphone medication adherence apps: Potential benefits to patients and providers." *Journal of the American Pharmacists Association*, **53**(2), 2013.

[DHM09] Yujie Dong, A. Hoover, and E. Muth. "A Device for Detecting and Counting Bites of Food Taken by a Person during Eating." In *Bioinformatics and Biomedicine, 2009. BIBM '09. IEEE International Conference on*, pp. 265–268, Nov 2009.

[DM01] Jacqueline Dunbar-Jacob and MaryKay Mortimer-Stephens. "Treatment adherence in chronic disease." *Journal of clinical epidemiology*, **54**(12):S57–S60, 2001.

[DYN10] Kyle Dorman, Marjan Yahyanejad, Ani Nahapetian, Myung-kyung Suh, Majid Sarrafzadeh, William McCarthy, and William Kaiser. "Nutrition Monitor: A Food Purchase and Consumption Monitoring Mobile System." In Thomas Phan, Rebecca Montanari, and Petros Zerfos, editors, *Mobile Computing, Applications, and Services*, volume 35, pp. 1–11. Springer Berlin Heidelberg, 2010.

[EWS10] Florian Eyben, Martin Wöllmer, and Björn Schuller. "Opensmile: The Munich Versatile and Fast Open-source Audio Feature Extractor." In *Proceedings of the International Conference on Multimedia*, MM '10, pp. 1459–1462, New York, NY, USA, 2010. ACM.

[FGG97] Nir Friedman, Dan Geiger, and Moises Goldszmidt. "Bayesian network classifiers." *Machine learning*, **29**(2-3):131–163, 1997.

[FHS14] JM Fontana, JA Higgins, SC Schuckers, and E. Sazonov. "Energy intake estimation from counts of chews and swallows." *Appetite*, (85):14–21, 2014.

[FLK11] P. S. Freedson, K. Lyden, S. Kozey-Keadle, and J. Staudenmayer. "Evaluation of artificial neural network algorithms for predicting METs and activity type from accelerometer data: validation on an independent sample." *J. Appl. Physiol.*, **111**(6):1804–1812, Dec 2011.

[FMS98] P. S. Freedson, E. Melanson, and J. Sirard. "Calibration of the Computer Science and Applications, Inc. accelerometer." *Med Sci Sports Exerc*, **30**(5):777–781, May 1998.

[GB11] Bradi B Granger and Hayden Bosworth. "Medication adherence: emerging use of technology." *Current Opinion in Cardiology*, **26**(4):279, 2011.

[GJ13] Hassan Ghasemzadeh and Roozbeh Jafari. "Ultra Low-power Signal Processing in Wearable Monitoring Systems: A Tiered Screening Architecture with Optimal Bit Resolution." *ACM Trans. Embed. Comput. Syst.*, **13**(1):9:1–9:23, September 2013.

[GNM03] Xiaohui Gu, K. Nahrstedt, A. Messer, I. Greenberg, and D. Milojicic. "Adaptive offloading inference for delivering applications in pervasive computing environments." In *Pervasive Computing and Communications, 2003. (PerCom 2003). Proceedings of the First IEEE International Conference on*, pp. 107–114, March 2003.

[HDG88] W. E. LEE HI, A. E. DEIBEL, C. T. GLEMBIN, and E. G. MUNDAY. "ANALYSIS OF FOOD CRUSHING SOUNDS DURING MASTICATION: FREQUENCY-TIME STUDIES1." *Journal of Texture Studies*, **19**(1):27–38, 1988.

[HDO98] Marti A. Hearst, Susan T Dumais, Edgar Osman, John Platt, and Bernhard Scholkopf. "Support vector machines." *Intelligent Systems and their Applications, IEEE*, **13**(4):18–28, 1998.

[HFH09a] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. "The WEKA data mining software: an update." *ACM SIGKDD explorations newsletter*, **11**(1):10–18, 2009.

[HFH09b] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. "The WEKA Data Mining Software: An Update." *SIGKDD Explor. Newsl.*, **11**(1):10–18, November 2009.

[HHA] Tamara L Hayes, John M Hunt, Andre Adami, and Jeffrey A Kaye. "An electronic pillbox for continuous monitoring of medication adherence." In *Proceedings of the* 28^{th} *Annual International Conference of Engineering in Medicine and Biology Society, 2006.*, pp. 6400–6403.

[HL04] David W Hosmer Jr and Stanley Lemeshow. *Applied logistic regression*. John Wiley and Sons, 2004.

[HOK88] C. H. Horst, G. L. Obermann-De Boer, and D. Kromhout. "Validity of the 24-hour recall method in infancy: the Leiden Pre-School Children Study." *Int J Epidemiol*, **17**(1):217–221, Mar 1988.

[Hua12] Shuo Huang. *Adaptive Sampling with Mobile Sensor Networks*. PhD thesis, Michigan Technological University, 2012.

[Jaw] "Jawbone UP Technical Specifications." http://jawbone.com/store/buy/up24.

[JC04] Ankur Jain and Edward Y Chang. "Adaptive sampling for sensor networks." In *Proceeedings of the 1st international workshop on Data management for sensor networks: in conjunction with VLDB 2004*, pp. 10–16. ACM, 2004.

[JCY13] Wenyan Jia, Hsin-Chen Chen, Yaofeng Yue, Zhaoxin Li, John Fernstrom, Yicheng Bai, Chengliu Li, and Mingui Sun. "Accuracy of food portion size estimation from digital pictures acquired by a chest-worn camera." *Public Health Nutrition*, **FirstView**:1–11, 12 2013.

[KAL15a] Haik Kalantarian, Nabil Alshurafa, Tuan Le, and Majid Sarrafzadeh. "Monitoring Eating Habits using a Piezoelectric Sensor-Based Necklace." *Elsevier Computers in Biology and Medicine*, **58**(C):46–55, 2015.

[KAL15b] Haik Kalantarian, Nabil Alshurafa, Tuan Le, and Majid Sarrafzadeh. "Non-Invasive Detection of Medication Adherence using a Digital Smart Necklace." In *IEEE PerCom: Smart Environments Workshop*, 2015.

[KAP14] Haik Kalantarian, Nabil Alshurafa, Mohammad Pourhomayoun, Shruti Sarin, Tuan Le, and Majid Sarrafzadeh. "Spectrogram-Based Audio Classification of Nutrition Intake." In *IEEE EMBS Healthcare Innovations & Point of Care Technologies (HIPT)*, 2014.

[KAP15] Haik Kalantarian, Nabil Alshurafa, Mohammad Pourhomayoun, and Majid Sarrafzadeh. "Power Optimization for Wearable Devices." In *IEEE Percom: WristSense*, 2015.

[KAS14a] Haik Kalantarian, Nabil Alshurafa, and Majid Sarrafzadeh. "A Wearable Nutrition Monitoring System." In *IEEE Body Sensor Networks (2014)*, 2014.

[KAS14b] Haik Kalantarian, Nabil Alshurafa, and Majid Sarrafzadeh. "A Wearable Nutrition Monitoring System." In *IEEE Body Sensor Networks*, 2014.

[KCH04] Eamonn Keogh, Selina Chu, David Hart, and Michael Pazzani. "Segmenting time series: A survey and novel approach." *Data mining in time series databases*, **57**:1–22, 2004.

[KEA02] Vladimir Katkovnik, Karen Egiazarian, and Jaakko Astola. "Adaptive Window Size Image De-noising Based on Intersection of Confidence Intervals (ICI) Rule." *Journal of Mathematical Imaging and Vision*, **16**(3):223–235, 2002.

[KHM16] Aftab Khan, Nils Hammerla, Sebastian Mellor, and Thomas Pltz. "Optimising sampling rates for accelerometer-based human activity recognition." *Pattern Recognition Letters*, pp. –, 2016.

[KHR03] Ulrich Kremer, Jamey Hicks, and James Rehg. "A Compilation Framework for Power and Energy Management on Mobile Computers." In HenryG. Dietz, editor, *Languages and Compilers for Parallel Computing*, volume 2624 of *Lecture Notes in Computer Science*, pp. 115–131. Springer Berlin Heidelberg, 2003.

[KJ00] Ralf Klinkenberg and Thorsten Joachims. "Detecting Concept Drift with Support Vector Machines." In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, pp. 487–494, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.

[KL10] K. Kumar and Yung-Hsiang Lu. "Cloud Computing for Mobile Users: Can Offloading Computation Save Energy?" *Computer*, **43**(4):51–56, April 2010.

[KLL13] Karthik Kumar, Jibang Liu, Yung-Hsiang Lu, and Bharat Bhargava. "A survey of computation offloading for mobile systems." *Mobile Networks and Applications*, **18**(1):129–140, 2013.

[KLM13] H. Kalantarian, S.I. Lee, A. Mishra, H. Ghasemzadeh, J. Liu, and M. Sarrafzadeh. "Multimodal energy expenditure calculation for pervasive health: A data fusion model using wearable sensors." In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2013 IEEE International Conference on*, pp. 676–681, March 2013.

[KS15a] Haik Kalantarian and Majid Sarrafzadeh. "Audio-Based Detection and Evaluation of Eating Behavior using the Smartwatch Platform." *Elsevier Computers in Biology and Medicine*, 2015.

[KS15b] Haik Kalantarian and Majid Sarrafzadeh. "A Smartwatch-Based System well. for Audio-Based Monitoring of Dietary Habits." In *The Fifth International Conference on Ambient Computing, Applications, Services and Technologies*, 2015.

[Lep15] Jacques Lepine. "How ingestible sensors and smart pills will revolutionize healthcare." *HAPI.com*, sep 2015.

[LKC10] C-T Lin, L-W Ko, M-H Chang, J-R Duann, J-Y Chen, T-P Su, and T-P Jung. "Review of wireless and wearable electroencephalogram systems and brain-computer interfaces–a mini-review." *Gerontology*, **56**(1):112–119, 2010.

[LKL10] Jibang Liu, Karthik Kumar, and Yung-Hsiang Lu. "Tradeoff Between Energy Savings and Privacy Protection in Computation Offloading." In *Proceedings of the 16th ACM/IEEE International Symposium on Low Power Electronics and Design*, ISLPED '10, pp. 213–218, New York, NY, USA, 2010. ACM.

[LL06] Changki Lee and Gary Geunbae Lee. "Information gain and divergence-based feature selection for machine learning-based text categorization." *Information processing & management*, **42**(1):155–165, 2006.

[LLW07] Hsuan-Tien Lin, Chih-Jen Lin, and Ruby C Weng. "A note on Platts probabilistic outputs for support vector machines." *Machine learning*, **68**(3):267–276, 2007.

[LMS] Miodrag Lovrić, Marina Milanović, and Milan Stamenković. "ALGORITHMIC METHODS FOR SEGMENTATION OF TIME SERIES: AN OVERVIEW.".

[LSS08] Wolfgang Lutz, Warren Sanderson, and Sergei Scherbov. "The coming acceleration of global population ageing." *Nature*, **451**(7179):716–719, 2008.

[LW02] Andy Liaw and Matthew Wiener. "Classification and regression by randomForest." *R news*, **2**(3):18–22, 2002.

[MGH02] Heather P McDonald, Amit X Garg, and R Brian Haynes. "Interventions to enhance patient adherence to medication prescriptions: scientific review." *Journal of the American Medical Association*, **288**(22):2868–2879, 2002.

[MHK15] Elizabeth L Murnane, David Huffaker, and Gueorgi Kossinets. "Mobile health apps: adoption, adherence, and abandonment." In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM International Symposium on Wearable Computers*, pp. 261–264. ACM, 2015.

[misa] "Misfit Wearables FAQ." http://www.misfitwearables.com/supportl.

[Misb] "Misfit Wearables FAQ." http://www.misfitwearables.com/supportl.

[MOS12] Varun Mithal, Zachary O'Connor, Karsten Steinhaeuser, Shyam Boriah, Vipin Kumar, Christopher Potter, and Steven A. Klooster. "Time series change detection using segmentation: A case study for land cover monitoring." In *2012 Conference on Intelligent Data Understanding, CIDU 2012, Boulder, CO, USA, October 24-26, 2012*, pp. 63–70, 2012.

[MS03] A Djafari Marbini and Lionel E Sacks. "Adaptive sampling mechanisms in sensor networks." In *London Communications Symposium*, 2003.

[NC05] Alexandru Niculescu-Mizil and Rich Caruana. "Predicting Good Probabilities with Supervised Learning." In *Proceedings of the 22Nd International Conference on Machine Learning*, ICML 05, pp. 625–632, New York, NY, USA, 2005. ACM.

[NK08] J. Nishimura and T. Kuroda. "Eating habits monitoring using wireless wearable in-ear microphone." In *Wireless Pervasive Computing, 2008. ISWPC 2008. 3rd International Symposium on*, pp. 130–132, May 2008.

[NS11] M. Nagae and K. Suzuki. "A neck mounted interface for sensing the swallowing activity based on swallowing sound." In *Engineering in Medicine and Biology Society,EMBC, 2011 Annual International Conference of the IEEE*, pp. 5224–5227, Aug 2011.

[OK92] Masatoshi Okutomi and Takeo Kanade. "A locally adaptive window for signal matching." *International Journal of Computer Vision*, **7**(2):143–162, 1992.

[Ole14] Stefan Sverrisson Ole Morten. "How do I calculate throughput for a BLE link?", jul 2014.

[OS87] Douglas O'Shaughnessy. *Speech communication: human and machine*. Addison-Wesley, 1987.

[P12] Fischer WJ. Pler S, Wolff M. "Food intake monitoring: an acoustical approach to automated food intake activity detection and classification of consumed food." *Physiological Measurement*, pp. 1073–1093, jun 2012.

[PCC14] Abhinav Parate, Meng-Chieh Chiu, Chaniel Chadowitz, Deepak Ganesan, and Evangelos Kalogerakis. "Risq: Recognizing smoking gestures with inertial sensors on a wristband." In *Proceedings of the 12th annual international conference on Mobile systems, applications, and services*, pp. 149–161. ACM, 2014.

[Pla99a] John Platt et al. "Fast training of support vector machines using sequential minimal optimization." *Advances in kernel methodssupport vector learning*, **3**, 1999.

[Pla99b] John C. Platt. "Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods." In *ADVANCES IN LARGE MARGIN CLASSIFIERS*, pp. 61–74. MIT Press, 1999.

[PM07] Ramaswamy Palaniappan and Danilo P Mandic. "Biometrics from brain electrical activity: A machine learning approach." *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **29**(4):738–742, 2007.

[PMH11] Ross L. Prentice, Yasmin Mossavar-Rahmani, Ying Huang, Linda Van Horn, Shirley A. A. Beresford, Bette Caan, Lesley Tinker, Dale Schoeller, Sheila Bingham, Charles B. Eaton, Cynthia Thomson, Karen C. Johnson, Judy Ockene, Gloria Sarto, Gerardo Heiss, and Marian L. Neuhouser. "Evaluation and Comparison of Food Records, Recalls, and Frequencies for Energy and Protein Assessment by Using Recovery Biomarkers." *American Journal of Epidemiology*, **174**(5):591–603, 2011.

[PPB12] Shyamal Patel, Hyung Park, Paolo Bonato, Leighton Chan, and Mary Rodgers. "A review of wearable sensors and systems with application in rehabilitation." *Journal of NeuroEngineering and Rehabilitation*, **9**(1):21, 2012.

[PR11] T. Pfister and P. Robinson. "Real-Time Recognition of Affective States from Nonverbal Features of Speech and Its Application for Public Speaking Skill Analysis." *Affective Computing, IEEE Transactions on*, **2**(2):66–78, April 2011.

[RAZ14] Tauhidur Rahman, Alexander T. Adams, Mi Zhang, Erin Cherry, Bobby Zhou, Huaishu Peng, and Tanzeem Choudhury. "BodyBeat: A Mobile System for Sensing Non-speech Body Sounds." In *Proceedings of the 12th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys '14, pp. 2–13, New York, NY, USA, 2014. ACM.

[SBM14] Mingui Sun, Lora E Burke, Zhi-Hong Mao, Yiran Chen, Hsin-Chen Chen, Yicheng Bai, Yuecheng Li, Chengliu Li, and Wenyan Jia. "eButton: a wearable computer for health monitoring and personal assistance." In *Proceedings of the 51st Annual Design Automation Conference*, pp. 1–6. ACM, 2014.

[SBT02] SCOTT J Strath, DAVID R Bassett Jr, DIXIE L Thompson, and Ann M Swartz. "Validity of the simultaneous heart rate-motion sensor technique for measuring energy expenditure." *Medicine and science in sports and exercise*, **34**(5):888–894, 2002.

[SF12] Edward S Sazonov and Juan M Fontana. "A sensor system for automatic detection of food intake through non-invasive monitoring of chewing." *Sensors Journal, IEEE*, **12**(5):1340–1348, 2012.

[SG08] Eugene Shih and John Guttag. "Reducing Energy Consumption of Multichannel Mobile Medical Monitoring Algorithms." In *Proceedings of the 2Nd International Workshop on Systems and Networking Support for Health Care and Assisted Living Environments*, HealthNet '08, pp. 15:1–15:7, New York, NY, USA, 2008. ACM.

[SKC12] Feng-Tso Sun, Cynthia Kuo, Heng-Tze Cheng, Senaka Buthpitiya, Patricia Collins, and Martin Griss. "Activity-aware mental stress detection using physiological sensors." In *Mobile computing, applications, and services*, pp. 211–230. Springer, 2012.

[SM06] AnthonyA. Sterns and ChristopherB. Mayhorn. "Persuasive Pillboxes: Improving Medication Adherence with Personal Digital Assistants." In WijnandA. IJsselsteijn, YvonneA.W. de Kort, Cees Midden, Berry Eggen, and

Elise van den Hoven, editors, *Persuasive Technology*, volume 3962 of *Lecture Notes in Computer Science*, pp. 195–198. Springer Berlin Heidelberg, 2006.

[smi] "openSMILE FAQ." http://www.audeering.com/research/opensmile.

[SMS10] E. S. Sazonov, O. Makeyev, S. Schuckers, P. Lopez-Meyer, E. L. Melanson, and M. R. Neuman. "Automatic detection of swallowing events by acoustical means for applications of monitoring of ingestive behavior." *IEEE Trans Biomed Eng*, **57**(3):626–633, Mar 2010.

[SPS09] Andrea Santamato, Francesco Panza, Vincenzo Solfrizzi, Anna Russo, Vincenza Frisardi, Marisa Megna, Maurizio Ranieri, and Pietro Fiore. "Acoustic analysis of swallowing sounds: a new technique for assessing dysphagia." *Journal of rehabilitation medicine*, **41**(8):639–645, 2009.

[SS85] E. Stellar and E. E. Shrager. "Chews and swallows and the microstructure of eating." *Am. J. Clin. Nutr.*, **42**(5 Suppl):973–982, Nov 1985.

[SSL08] Edward Sazonov, Stephanie Schuckers, Paulo Lopez-Meyer, Oleksandr Makeyev, Nadezhda Sazonova, Edward L Melanson, and Michael Neuman. "Non-invasive monitoring of chewing and swallowing for objective quantification of ingestive behavior." *Physiological Measurement*, **29**(5):525, 2008.

[SSL09] Edward S. Sazonov, Stephanie A.C. Schuckers, Paulo Lopez-Meyer, Oleksandr Makeyev, Edward L. Melanson, Michael R. Neuman, and James O. Hill. "Toward Objective Monitoring of Ingestive Behavior in Free-living Population." *Obesity*, **17**(10):1971–1975, 2009.

[TCS08] Pavan Turaga, Rama Chellappa, Venkatramana S Subrahmanian, and Octavian Udrea. "Machine recognition of human activities: A survey." *Circuits and Systems for Video Technology, IEEE Transactions on*, **18**(11):1473–1488, 2008.

[TIH07] Emmanuel Munguia Tapia, Stephen S Intille, William Haskell, Kent Larson, Julie Wright, Abby King, and Robert Friedman. "Real-time recognition of physical activities and their intensities using wireless accelerometers and a heart rate monitor." In *Wearable Computers, 2007 11th IEEE International Symposium on*, pp. 37–40. IEEE, 2007.

[TK12] Mitsuru Taniwaki and Kaoru Kohyama. "Fast Fourier transform analysis of sounds made while swallowing various foods." *The Journal of the Acoustical Society of America*, **132**(4):2478–2482, 2012.

[TOY10] H. Tsujimura, H. Okazaki, M. Yamashita, H. Doi, and M. Matsumura. "Non-Restrictive Measurement of Swallowing Frequency Using a Throat Microphone." *IEEJ Transactions on Electronics, Information and Systems*, **130**:376–382, 2010.

[VGJ12] Meera Viswanathan, Carol E. Golin, Christine D. Jones, Mahima Ashok, Susan J. Blalock, Roberta C.M. Wines, Emmanuel J.L. Coker-Schwimmer, David L. Rosen, Priyanka Sista, and Kathleen N. Lohr. "Interventions to Improve Adherence to Self-administered Medications for Chronic Diseases

in the United StatesA Systematic Review." *Annals of Internal Medicine*, **157**(11):785–795, 2012.

[vit14] "Vitality Glowcap." http://www.vitality.net/, 2014.

[VMS06] M. Valin, J. Meunier, A. St-Arnaud, and J. Rousseau. "Video Surveillance of Medication Intake." In *Proceedings of the* 28^{th} *Annual International Conference of Engineering in Medicine and Biology Society, 2006.*, pp. 6396–6399, Aug 2006.

[VRR13] Glenn Vonk, Richard Rumbaugh, and Colleen Ryan. "Medication adherence system.", 2013. US Patent 8417381.

[XZK12] Zhenghua Xu, Rui Zhang, Ramamohanarao Kotagiri, and Udaya Parampalli. "An Adaptive Algorithm for Online Time Series Segmentation with Error Bound Guarantee." In *Proceedings of the 15th International Conference on Extending Database Technology*, EDBT '12, pp. 192–203, New York, NY, USA, 2012. ACM.

[YH10] Che-Chang Yang and Yeh-Liang Hsu. "A review of accelerometry-based wearable motion detectors for physical activity monitoring." *Sensors*, **10**(8):7772–7788, 2010.

[YT12] Koji Yatani and Khai N. Truong. "BodyScope: A Wearable Acoustic Sensor for Activity Recognition." In *ACM International Conference on Ubiquitous Computing*. ACM, September 2012.

[ZCS15] Bo Zhou, Jingyuan Cheng, Mathias Sundholm, Attila Reiss, Wuhuang Huang, Oliver Amft, and Paul Lukowicz. "Smart table surface: A novel approach to pervasive dining monitoring." In *Pervasive Computing and Communications (PerCom), 2015 IEEE International Conference on*, pp. 155–162, March 2015.