

Empath: A System for the Collection, Storage, Analysis, and Reporting of Objective Behavioral Measures

A Dissertation

Presented to

the Faculty of the School of Engineering and Applied Science

University of Virginia

In Partial Fulfillment

of the requirements for the Degree

Doctor of Philosophy (Computer Science)

by

Robert F. Dickerson

December 2013

Abstract

Remote health monitoring deployed in homes could help streamline the efficiency of the medical system by decreasing the costs of senior care and providing preventative care to keep people out of hospitals. Besides for caring for a particular person, these systems will generate tremendous amount of aggregate data that can be used to help diagnose diseases and run longitudinal studies on epidemiology. Most systems are not yet flexible enough to handle adding additional modalities into the system or to leverage the cloud's horizontal scalability for storage, analysis, and display of this data. This dissertation presents a novel framework that sets itself apart from existing remote health monitoring systems for its scalability and flexibility. In addition, this work advances two key technologies for home healthcare. The first is the creation and evaluation of cheap and non-invasive sleep monitoring systems and novel accelerometer-based systems from RFIDs and motes. Secondly, this thesis demonstrates that the general features from speech in the home can be a useful modality for measuring social interaction and mood and promotes a solution to existing technical problems. Ambient monitoring of speech in the home has not been successful primarily because distortion from a room's acoustics negatively impacts classification results. This work presents a novel matched-condition classifier using cuboid acoustic simulation to achieve accuracy comparable to ideal close-to-microphone conditions. Most health monitoring systems have only been tested in a lab and under very scripted scenarios. This system, however, has been used in three different applications: monitoring sleep behaviors and stress for those who suffer from severe epilepsy in a clinical study, another clinical application that studies the relationship of incontinence with sleep agitation for those suffering from Alzheimer's disease, and an in-home deployment monitoring

important factors of depression. We present the commonalities among these different applications and show how to adapt the system for these purposes.

Acknowledgements

To Valerie– for editing my incoherent writing, volunteering to be instrumented by all sorts of sensors, and keeping me sane these past few months. I love you!

To Mom and Dad– for teaching me stay “well balanced” and to always remain intellectually curious.

To Jack– who has supported me for many years and has taught me the importance of achieving excellence. Without your help, I would never have become a professor.

To Enamul– without your help on all of the projects- no data would have been collected.

To Ann, Jewel, Karen, and Windy– for finding patients and putting Empath into real studies.

To Ben– for taking a risk with taking in an undergraduate, teaching me to be a good mentor, and encouraging me to go to graduate school.

More thanks to John Lach, Steve Patek, Sang Son, Alf Weaver, Kamin Whitehouse, Gabe Robins, Timothy Hnat, Philip Asare, Shahriar Nirjon, Jackey Gong, Kyle Johnsen, Andrew Raij, Aaron Kotranza, Karen Schmidt, Greg and Tierney Fairchild, and David Stone.

Approval Sheet

This dissertation is submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy (Computer Science)

Robert F. Dickerson

Robert F. Dickerson

This dissertation has been read and approved by the Examining Committee:

John A. Stankovic

John A. Stankovic, Advisor

Gabriel Robins

Gabriel Robins, Committee Chair

Yanjun Qi

Yanjun Qi

John Lach

John Lach, Minor Representative

Stephen Patek

Stephen Patek

Accepted for the School of Engineering and Applied Science:

James H. Aylor

James H. Aylor, Dean, School of Engineering and Applied Science

December 2013

Contents

Contents	vi
List of Tables	viii
List of Figures	ix
1 Introduction	1
1.1 Application Examples	3
1.2 Technical Challenges	4
1.3 Contributions	6
1.4 Dissertation Organization	7
2 State of the Art	8
2.1 Home Medical Monitoring Systems	8
2.2 Activity Detection	12
2.3 Mood and Wellness Tracking	14
2.4 Sleep Monitoring Systems	14
2.5 Speech Monitoring	16
3 Empath System Design	19
3.1 Contributions	20
3.2 System Design	21
3.3 Home Area Network Design	21
3.4 Sensing Devices	24
3.5 Cloud Layer	32
3.6 Streams on the Cloud	33
3.7 Data Inference Design	38
3.8 Caregiver Displays	39
3.9 Conclusions	40
4 Passive Speech Monitoring	41
4.1 Introduction	41
4.2 Reverberant Environments	43
4.3 System Implementation	46
4.4 Evaluation	50
4.5 Conclusions	64

5	Improving Sleep Monitoring Options	65
5.1	WISP Sleep Sensor	66
5.2	Cheaper Mote and Accelerometer Solution	84
5.3	Conclusions	86
6	Case Studies	88
6.1	Epilepsy and Stress	88
6.2	Alzheimer’s Disease and Incontinence	93
6.3	Depression Monitoring	97
6.4	Conclusion	106
7	Conclusions	107
7.1	Key contributions towards home monitoring systems	107
7.2	Future Improvements	109
7.3	Other example applications	111
	Appendices	113
A	DSM-IV Depression Criteria	114
A.1	Major Depressive Episode and Major Depressive Disorder	114
A.2	Dysthymic Disorder	115
A.3	Bipolar Episode and Bipolar Disorder	115
B	PHQ-9 Questionnaire	117
C	Recommendation Items	118
	Bibliography	120

List of Tables

3.1	Stream Metadata Examples	25
3.2	PHQ-9 Scoring	29
3.3	Weight Factor Evaluation	30
3.4	RESTful HTTP Requests	35
3.5	Comparison of Database types	38
4.1	AIR database Room Characteristics	51
4.2	Classification Benchmarks	55
4.3	Experiment Room Characteristics	57
4.4	Training Time	60
4.5	Sounds Encountered	62
5.1	Summary of Results for 6 Datasets	83
5.2	Comparison of sleep monitoring systems	87
C.1	Recommendation Items	119

List of Figures

1.1	General Architecture	4
3.1	Home area network	22
3.2	X10 Devices	26
3.3	Touchscreen Device for Subjective Mood	29
3.4	Withings weight scale	30
3.5	XBee Radio	31
3.6	Zigbee communication from motes to basestation	31
3.7	Verizon Modem	32
3.8	Empath deployed on Google AppEngine’s Cloud infrastructure.	34
3.9	Clinician Display	40
4.1	Speech classification pipeline	44
4.2	Speech network architecture	47
4.3	Speech Monitoring Device	48
4.4	Speaker ID Accuracy on EmoDB	52
4.5	Mood Accuracy on EmoDB	52
4.6	Feature Selection	53
4.7	F-statistics for Feature Selection	54
4.8	RT estimation error	55
4.9	Relationship of the length and the speaker identification accuracy.	56
4.10	Room Floorplans	57
4.11	Conference Room	58
4.12	Living Room 1	58
4.13	Living Room 2	58
4.14	Kitchen 1	59
4.15	Longterm Study Rooms	61
4.16	Long term Study Results	63
5.1	WISP Tag	67
5.2	SpeedWay RFID reader	67
5.3	Accelerometer Reading for Empty, Lying, and Sitting	68
5.4	WISP Placement	68
5.5	Accelerometer Reading Variation for Different Lying Positions	69
5.6	WISP Study Participants	70

5.7	Accelerometer Reading Variation for Different Lying Positions	71
5.8	Average Classification Error for 5 Subjects for One of the Beds	72
5.9	Average Classification Error for All Mattresses	73
5.10	Accelerometer Reading along Y-axis During a Movement	75
5.11	WISP Movement in Controlled Experiment	76
5.12	Discrete Movement Events During 70 Minutes of Controlled Experiment	77
5.13	Pressure Pad Adapted from Dance Dance Revolution	77
5.14	DDR on Bed	78
5.15	Time Windowing of Sleep Epochs	78
5.16	Accelerometer Measurements for 1 Night of Sleep	79
5.17	DDR Measurements for 1 Night of Sleep	80
5.18	Comparison Between our System and DDR	80
5.19	SleepCycle iPhone App Results	81
5.20	Body Positions During 1 Night's sleep of Evaluation Set 1	82
5.21	Sleeping Monitoring Setup	85
5.22	Wixel Mote	86
6.1	Reflective Exercise	90
6.2	Epilepsy Architecture	91
6.3	Bedtime Density Plot	93
6.4	Wake Time Density Plot.	93
6.5	Incontinence System Design	95
6.6	DryBuddy Incontinence Sensor	96
6.7	Incontinence Study Setup	96
6.8	Incontinence Deployment 1	97
6.9	The inference system for creating a depression risk index.	99
6.10	Sleep Quality Factor	103
6.11	X10 Readings	104
6.12	House Occupancy	104
6.13	Activity Level	105

1 | Introduction

Health care in the United States is reaching a crisis where the increase of costs for healthcare will not keep up with the growth of the nation's wealth [1]. Health care spending is on the climb – in 2012 U.S. spending was \$8,936.80 and is projected to climb to \$13,708 per person in 2016. It is projected also that healthcare will consume 150% of the wealth Americans are expected to gain by 2050 [2]. Although the U.S. has one of the best healthcare systems in the world, which spends a total of \$2.5 trillion in healthcare each year, it only spends \$251 on public health measures that can prevent medical conditions before they occur. Adequate measures for routine monitoring and management of chronic conditions – before they proceed to advanced stages – will be one of the most vital factors in keeping these costs down.

For the past few years, health-minded people have increasingly used technology to proactively track their own health and well-being to prevent illness. The number of consumer health apps listed on the Apple Store has increased from 2,993 in 2010 to 13,619 in 2012. Along with the software, new sensors are available for purchase on the market. A recent report [3] predicts that 18.2 million health sensors will ship in 2017. Already 150 million mobile sensing health and fitness Apps have been downloaded, and by 2017 this number to climb to 1.4 billion. Cumulative revenues for these applications and subscriptions will reach \$975 million by 2017. However, currently the majority of these apps focus on tracking fitness or diet, and far fewer are geared toward tracking and managing health problems such as chronic disease.

Managing these chronic conditions, keeping people out of hospitals, and improving late-in-life care are important goals for the coming years. Technologies such as improved sensors, improved

data analysis algorithms, and more efficient delivery of medicine are anticipated to revolutionize the medical practice. Remote home monitoring will help streamline the efficiency of the medical system by decreasing the costs of senior care and providing preventative care to keep people out of hospitals. These systems will generate tremendous amounts of aggregate data that can be used to help understand the progression of diseases and evaluate the effectiveness of drugs and therapy programs.

This dissertation explores the appropriate design for deploying an integrated home healthcare system that can be useful for various health concerns. The system provides enough rich sensing to track multifactor syndromatic conditions such as depression and anxiety. Our system called *Empath* is a novel system that takes a different approach than the majority of current remote health monitoring systems. Primarily, *Empath* leverages quickly deployable nodes with Cloud-connectivity to allow data to be shared with many caregivers. Secondly, *Empath* focuses on enhancing the ability to collect comprehensive information regarding sleep factors and speech. Finally, and what makes this thesis unique, is that *Empath* has been adapted and employed for three different clinical applications: monitoring stress for those who suffer from severe epilepsy in Section 6.1, a study examining the relationship of incontinence with sleep agitation for those suffering from Alzheimer's disease in Section 6.2, and monitoring important factors of depression in Section 6.3. Whereas almost all current remote monitoring systems are solely proof-of-concept and have not left the labs, *Empath* is a valuable and proven tool currently used by clinical research teams.

Despite the differences in modalities used when monitoring epilepsy, Alzheimer's disease, and depression, we demonstrate that there are fundamental similarities among these different applications, and show that the system can be adapted for these purposes. It is important to note that this thesis and the content of the *Empath* system itself make no claim to be able to accurately diagnose the above conditions without assessment by clinicians, because that problem remains out of the technical scope of this work. However, we assume that the conditions above will typically manifest themselves in predictable symptoms (as enumerated by diagnostic books such as DSM-IV), so the system proves to be a valuable instrument for collecting and providing previously unavailable data

to clinicians. Providing the data that clinicians need for proper tracking and diagnosis requires tackling particular challenges in engineering that can only be ascertained and solved through real-life deployments. Along with the goals of collecting the most useful data for clinicians, we focus on also creating a system that is able to be quickly deployed at low cost, is unintrusive to the patient, and is flexible and extensible for future needs.

There are several reasons why current systems are ill-equipped for the applications listed above:

1. **Quickly deployable and modular:** Systems need to be easily and quickly deployed so that during installation and teardown it does not impact the lives of the residents.
2. **Multiple Disease Factors:** Applications such as depression monitoring require simultaneous attention to several disease factors (social interaction, sleep quality, body weight management, activities of daily living monitoring). Inferring these complex behavior changes requires aggregating multiple data streams.
3. **Scalable and Connected:** System must be designed to support hundreds of homes with a single web service, and must be fully connected so that caregivers and researchers can access the data.
4. **Speech-enabled:** For monitoring conditions that are related to social interaction and mood, information held in speech has been proven to be clinically useful. No systems we are aware of are able to use speech features that are captured under actual room acoustics, while monitoring both the amount of social interactions and the mood when speaking.

1.1 Application Examples

In Figure 1.1 we show a high-level diagram of the network architecture. Each house is instrumented by an appropriate combination of sensors according to the application domain the caregiver or researcher has decided. Each of the sensors uses a radio to broadcast the information to a base station. Each base station is then responsible for using an access point (such as broadband) to

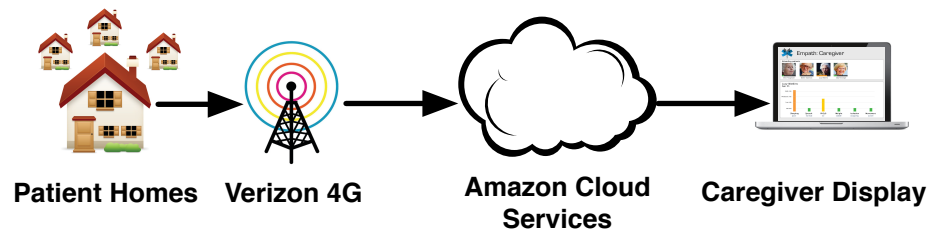


Figure 1.1: Data is collected from many homes and sent to the a webservice for later analysis by caregivers.

synchronize the information held locally to a request on the web service. The web service properly load-balances the request and stores the information to the backend. The system then performs inference routines on the streams, combining them if necessary to form higher level knowledge about disease risk factors in the home. Caregivers may then make queries to the web service either from their browsers or mobile devices.

1.2 Technical Challenges

This thesis addresses key technical issues that limit current home monitoring solutions today. Firstly is the ability to incorporate heterogeneous devices. To date, home monitoring systems use very few platforms for collecting data (some extant systems use the MicaZ and the Shimmer, for example). However, there are hundreds of companies producing new sensors each year. Future systems must be able to integrate these various sensors together in order to leverage more types of data to allow for more successful inference of disease patterns.

Secondly, future home monitoring systems need a flexible architecture in order to effectively evolve over time. New sensing modalities inevitably will be discovered to be important for the monitoring of chronic diseases, and in the future will need to be integrated into the system, ideally with limited impact to the system's current functionality.

The scale of existing systems is generally small, focusing on a single home and a single caregiver.

However, the reality is that hundreds, if not thousands, of homes will eventually be connected by a single service provider. The result will be an enormous volume of data collected over long timeframes which become a valuable resource for researchers investigating the symptomology or progression of disease. This proliferation of data is bound to cause challenges in storage, organization, and recall.

No home health systems currently use the sound in environments as a valuable resource. The speech in the home contains an enormous amount of important health information and ought to be harnessed and analyzed. Perhaps one of the reasons why speech has not been used is the great technical challenges it poses: when sound is captured in rooms there is considerable distortion from the acoustics such as reverberation and noise. This thesis presents a solution to the reverberation problem by developing matched-condition classifiers tailored to the reverberant qualities of each room, which perform speech identification and mood analysis in real time.

Next, many systems already collect data about sleep. However, all current systems require invasive and uncomfortable instrumentation, such as requiring the user to wear a device on their wrists or head, or to sleep on a large pressure pad placed on the mattress to track their movement. All of these existing options are expensive as well. To address these concerns, we developed a small and unobtrusive system using individual accelerometers that are placed on the bed. This system is not felt by the user during sleep, it can be deployed quickly, and it costs less than \$50. We also created new machine learning and signal processing algorithms for use by these accelerometer-based notes.

Finally, remote home health monitoring has not been investigated on larger scale. One area of concern is the accommodation of large number of homes. Our system is the first to use Cloud-based services for home monitoring, using all Web technologies such as HTTP and JSON through RESTful interfaces for delivery of data. The data are stored in distributed Cloud-based databases instead of a base station's SQL database, which allows for an integrated and universally accessible data system.

1.3 Contributions

This thesis provides a novel home monitoring system that provides new contributions to the area of home health monitoring. The technical contributions include capturing voice in the home and also cheap and accurate sleep monitoring, as well as easier deployment and management of data. We present the following contributions to this area in comparison to the state of the art:

1. An extensible, multimodal, largely passive behavioral monitoring system that is useful to caregivers in order to monitor their patient's behavior, and thereby track their well-being and their response to treatment and therapies.
2. A flexible system architecture that allows easy introduction of new custom components or commercial off-the-shelf products, and various web service abstractions and data abstractions for storing and sharing that data.
3. Advancements in addressing reverberation in ambient home speech monitoring, including the identification of important vocal features relating to mood, such as speaking durations and frequencies. Creation of a matched-condition classification strategy called RESONATE that uses room acoustic simulation to quickly generate models for a particular room. Additionally, a practical implementation that runs on embedded devices in a network for broader coverage in a home.
4. A passive and cheap sleep monitoring system that can collect useful features such as insomnia, restlessness, and sleeping patterns, which can be used to infer sleep quality. Relatedly, the evaluation of machine learning algorithms that can infer sleeping position. The sleep monitoring system is tested in each case for several weeks or months in three clinical studies and three non-clinical studies.
5. Results and observations from deploying the system in real clinical studies: examining the relationship of sleep and stress on the number of seizures that people with epilepsy experience, the relationships between nighttime agitation and incontinence events with those with

Alzheimer's disease, and finally case-studies with configuring the system for depression monitoring.

1.4 Dissertation Organization

The rest of the dissertation is organized as follows:

- Chapter 2 introduces the state of the art research related to home remote monitoring, speech monitoring, and sleep systems.
- Chapter 3 discusses the design for a heterogeneous and scalable system and its sensor components.
- Chapter 4 demonstrates how speech monitoring is an important element for home monitoring systems and how it can be achieved in reverberant acoustic environments.
- Chapter 5 presents highly deployable, cheap, and accurate sleep monitoring sensors and algorithms.
- Chapter 6 discusses adapting the system for different applications, and the experiences of deploying the system in real environments, primarily in clinical studies.
- Chapter 7 concludes this dissertation with the contributions and findings, discusses the limitations of this work, and provides a number of possible directions for future work.

2 | State of the Art

In this chapter, we describe the “state of the art” in remote home healthcare monitoring. First, we discuss several existing home health monitoring systems, such as those designed for assisted living care . Second, we discuss existing algorithms that are used in these home monitoring systems to track people and detect activities. Lastly, we introduce the latest in health trackers and sensors for use in the home and on smart phones. We conclude with a summary of sleep monitoring systems in Section 2.4 and speech monitoring in Section 2.5.

2.1 Home Medical Monitoring Systems

Various health-enabling technologies have emerged to lower the cost, increase quality and access to healthcare delivery [4]. One simple yet effective strategy is to connect patients and clinicians through real-time video streams [5]. It has been shown that telemedicine can reduce the cost of travel and increase efficiency [6].

As people age, they experience a variety of cognitive, physical, and social changes that can challenge their health. With the rising numbers of elderly, many companies and groups have pursued ways to keep people in their homes as long as possible, preventing the cost of assisted living or nursing care. Another reason is that many elderly do not live in places where family and loved ones are able to visit daily and check on their wellbeing. Since the early 1990s companies such as Phillips, GE, and IBM have presented telehealth systems– such as a telephone and a TV screen. With time, these systems were augmented with more advanced sensors and could track activities in the home.

Today, there are many various healthcare systems that are summarized well in a survey paper [7]. Most research in this area has occurred in university research testbeds. Georgia Tech's *Aware-Home* [8] combined context-aware and ubiquitous sensing, computer vision-based monitoring, and acoustic tracking of people. The University of Rochester built their *Smart Medical Home* which is a five-person house outfitted with infrared sensors, biosensors, and video cameras for use by research teams for work with research subjects. A computer-animated character called "Chester the Pill" was available to discuss their medical health, providing advice on which medicines to take and their symptoms. In addition, the system would also track the location of eyeglasses, coffee cups, and car keys which is especially useful for the cognitively declining. The system also had a gait monitor which could track the way the residents walk noting any precursors to Parkinson's disease. There were also worn devices that could take blood pressure readings automatically. Also environmental sensors detect levels of pollen or dust from air and provide steps to remedy the situation.

MIT had their own testbed called PlaceLab [9]. It consists of a one-bedroom condominium with hundreds of sensors on the walls, fixtures, and cabinetry. However, the goals of PlaceLab was much more broad than AwareHome, and focused on ways for people to control their environment, save resources, remain mentally and physically active, as well as stay healthy. The home was occupied by volunteer subjects for various lengths of time and introduced new tools for semi-automatically annotating the generated data. Harvard's CodeBlue[10] was designed to provide routing, naming, discovery, and security for various sensors including a portable 2-lead ECG, pulse oximeter, wearable *Pluto* mote with accelerometers, gyroscope, and electromyogram sensor for stroke monitoring. University of Washington's *Assisted Cognition* project focused on modelling the relationships among GPS readings, activities, and significant places.

The *Gator Tech* Smart House at the University of Florida was a laboratory-house created to assist older adults in maximizing their independence and maintaining a higher quality of life [11]. Ambient Assisted Living (AAL) focus on developing high-tech infrastructures that replace missing or fading human abilities with automated technical services (feeding robots). One system *SmartAssist* is an open platform for the creation of context-aware AAL services [12]. Users can subscribe to

and use AAL services, define peers in their social networks that will be informed about significant variations in their vital data. This system emphasized gamification methods.

To date, there have been a few companies that have begun to sell their systems directly to people who want to monitor either themselves or their loved ones. The cost includes a small installation fee, and then a recurring monthly subscription. There are many similarities in companies that provide home healthcare surveillance and for home security systems, and they often use similar sensors and network infrastructure. The WellAware [13] system provides commodity sensors to track sleep quality, activity levels, bathroom visits, and basic physiological information. Phillips's *Motiva* on the otherhand, takes a more active role in the patient's health. The system works by making personalized daily interactions with their residents by way of their home television. For instance, the system pushes educational material, actionable feedback, motivational messages, and health-related surveys to the patients. Intel-GE *Care Innovations* provides a similar service to *Motiva* and provides interactive patient health sessions, vital sign measurements, multimedia educational library, audio and visual notifications, and two-way video calls for patients and caregivers. *BeClose* [14] is another home monitoring system designed especially for the elderly. The system consists of a number of motion sensors as well as a bed pressure pad as well as a panic button that notifies authorities and kin if there is something wrong. The user interface is built on the web platform and it presents a dashboard showing caregivers their patient's sleep patterns, movement, and weight. If the patient's behavior is anomalous, such as if they are not getting out of bed after a certain time or whether they are leaving the house too little or too much, a concerned relative can check on them. *BeClose* is has a minimal installation fee of \$299 and includes a base station and three sensors, however more sensors can be added as needed. The service then costs \$49 a month for 24/7 monitoring.

HealthOS is a development and execution framework for pervasive health applications [15]. They address the sensor and system incompatibilities with *adapters*, and *pipelines* translates various protocols and to meet the application's specifications. The HealthOS *servers* and expose REST interfaces for data retrieval.

2.1.1 AlarmNet

The Empath project stemmed off of the existing testbed at the University of Virginia called *AlarmNet*. AlarmNet was an assisted living and residential monitoring network for pervasive and adaptive healthcare and based on an extensible, heterogeneous network architecture targeting ad-hoc, wide-scale deployments. It included custom and commodity sensors, an embedded gateway, and a back-end database for analysis. The system provided software that could learn the circadian rhythms of the occupants to better adjust the power scheduling of the motes for extending lifetimes. It supported realtime query-based online sensor data streaming and an inference system for recognizing anomalous behaviors. Perhaps one of the most novel aspects of AlarmNet was how context-aware protocols are able to dynamically configure the power-management and the alert-driven privacy controls based on the user's activity patterns. The focus of AlarmNet differs from CodeBlue and other existing systems in that it is designed for assisted living and residential environments where long-term (not just acute) behavior and symptomology is monitored.

For making queries into the system, a system called SenQ was created to address the fact that WSNs generate event-based data or continuous stream data. In-network filters could be created, and event based alarms could ensure that the proper caregiver would be notified when a patient's condition changes. SenQ was integrated into the context-aware privacy, power management, and activity analysis components. The data SenQ returns was made available to either PDA devices or special motes called the SeeMote with LCD displays [16]. One focus that AlarmNet had was a Circadian Activity Rhythm (CAR) program that could learn the patterns of the residents and can anticipate which sensors should be kept active and which could be temporarily disabled in order to conserve power. The architecture was able to accommodate heterogeneous sensors such as the MicaZ and Shimmer into a common architecture. A back-end computer was placed on the network that would receive all of the data streams, filter and process them, and produce new output streams in the database for other modules to later use.

A recent survey work describes some web-based sensor networks [17]. The SAPHE health

care system for the Sensor Web [18] is able to sense different physiological attributes such as blood pressure, temperature and send them to the web for a specialist or doctor. This system used the Open Geospatial Consortium (OGC) architecture. The system works by sending data to the basestation in the patient's home and the basestation then sends the information to the web. Another system was developed especially to monitor Parkinson's disease [19]. The system use sensors worn on the hand called Mercury, the data is then sent to a basestation to a web application called MercuryLive. Many Body Area Networks (BANs) use an architecture that the sensors aggregate the data to a mobile device, and the data are uploaded to the Cloud. BANs are out of scope for this thesis, but there are described in a recent survey [20].

2.2 Activity Detection

An important aspect of providing automatic detection regarding pathology is detecting if the activities in the home are becoming anomalous or following certain well-known trends of the progression of disease (such as cognitive decline). The accurate detection of daily activities in complex home settings using emplaced sensors is a highly researched problem. One approach is to use cameras and computer vision algorithms to help detect everyday actions such as stirring in a bowl or cooking [21]. Despite the high accuracy of this technique, studies have have shown that the introduction of microphones and cameras into the home is objectionable. However, people gradually came to be accustomed to surveillance even if they initially opposed it [22]. The Helsinki privacy experiment studies the effects of long-term ubiquitous surveillance in homes [22]. The homes were instrumented with video cameras, microphones, computer, wireless network, smart phone, TV, DVD, and customer card was logged. Anxiety, stress, concerns, were monitored in six months, and they show how people will gradually become accustomed to surveillance even if they oppose it. Another work [23] shows similar results.

Many existing solutions use simple sensors to detect movements as a resident moves from one room to another (e.g., motion sensors in the doorway) or changes in state of objects and devices (e.g., contact sensors). The analogy is often cited as the "invisible man", where a invisible person

will interact with objects, and the system must infer who has touched the object. Kasteren [24] used temporal probabilistic models to recognize activities from these binary sensor readings. Approaches from the machine learning discipline such as N ave Bayesian [25], Bayesian networks [26], Hidden Markov Models, particle filters, and conditional random fields are typical algorithms used. Other models are also being explored such as a Hidden Semi-Markov model that relaxes the Markovian assumption to consider an activity’s duration to improve the accuracy of activity recognition [27].

Intel Research Seattle and the University of Washington have built a prototype that can infer a person’s activities of daily living (ADLs) by using RFID sensor tags (both active and passive) on everyday objects such as a toothbrush or a coffee cup [28]. Users are expected to wear an iBracelet or iGlove, and when they touch an object that interaction is recorded. We later in Chapter 5 show that these sensors can be used on the bed to monitor sleep quality.

One of the most challenging problems associated with all the works is that they require accurate labeling of activities for training. This can be done either by the resident or by manual annotation after viewing the data – of course this is difficult to obtain for a long period. It is not convenient for users to record each activity while performing it and not practical to annotate each activity after viewing several hours of recordings. As an alternative, Kasteren presented a technique in [24] to use the ground truth collected in one house to train activity recognition systems in other houses. This technique works however, the details of activities may vary significantly from person to person and from home to home, in which case this technique may not perform well.

One way to infer who is interacting with which object is by narrowing down the possibilities of people in the room where the object rests. DoorJamb [29] was developed to track movement without requiring people to wear any sensor. The system works by using ultrasonic range finders mounted above each doorway pointing downwards. By measuring their heights, the system can infer the identity and direction of the walker.

2.3 Mood and Wellness Tracking

Almost all the above systems provide general surveillance in the home, but do not focus on more difficult to track diseases such as depression, stress, or other unhealthy mood or mental conditions. Very little work has focused on tracking or detecting mood and psychological health and wellbeing.

Because retroactive reports on mood taken in the therapists office several weeks after events occur are inaccurate, new tools need to be developed to more accurately assess this information. Mood charts have been recommended by mental health practitioners as tools for their patients to monitoring their own mental health, however more sophisticated technology could make this more accurate. The *Optimism App* can run on both desktop and mobile platforms and can help acquire instantaneous mood, medication use, exercise, and sleep quality. Researchers at the Rhode Island Hospital have developed a telemedicine-based depression protocol using simple telemonitor for in-home healthcare, with pilot studies showing that it could improve geriatric depression [30]. The subjects in the study were favorable to the technology, reporting that the frequent checks from the monitor were reassuring and helped them to better understand their condition. Autosense [31] provides objective measurements of psychosocial exposure to stress and alcohol in natural environments. They have created a body area sensor network to measure heart rate and variability, respiration rate, skin conductance, skin temperature, arterial blood pressure, and blood alcohol concentration. The readings are sent to a smart phone and features relating the stress and alcoholism are computed at real-time.

2.4 Sleep Monitoring Systems

To date, while there are many sleep monitoring systems there are very few low-cost, unobtrusive (comfortable) solutions. In this section, we review the major solutions and describe their characteristics and limitations.

Physiological signals are regarded as the most accurate means to differentiate between awake and sleep phases such as light, REM, and deep sleep. The electroencephalogram (EEG) can mea-

sure brain waves and is one of the best methods for determining sleep stages. Electrooculogram (EOG) and electromyogram (EMG) are also standard technologies for sleep monitoring. The electrocardiogram (ECG) is used to derive heart rate, which is well known to decrease at sleep onset. Some studies show that heart rate varies over different sleep stages [32, 33]. By using respiratory-derived features together with ECG-derived features, different sleep stages can be labelled automatically. All of these techniques have major limitations such as they must be performed by trained professionals in expensive clinical environments. In addition, these techniques require that the patients wear dozens of wires on their bodies, which can cause discomfort. Finally, these physiological signals do not support monitoring body positions during sleep which video capture is the best technique.

To overcome the limitations of the above techniques, there are many systems for sleep monitoring that are less accurate, but well designed for homes and daily use. Actigraphy [34] is a common technique for sleep monitoring where the user wears a watch-like accelerometer on the wrist (or ankle). The device monitors acceleration during different intervals and will label periods of low activity as sleep. Philip's Actiwatch is one of the most popular products for actigraphy. The Zeo is a headband that users wear each night that can detect electrical signals similar to a full-EEG.

Another more direct method for sleep monitoring is to instrument a mattress pad with sensors and passively infer body movements and sleep quality. The *Bed Alarm* sensor pad is a commercial bed pressure sensor that monitors the change in body pressure on the pad to detect movements. In [35] the authors use pressure and temperature sensors laid out in a grid pattern to determine quality of sleep. NAPS [36, 37] is a low-cost physiological sensor-suite that can passively acquire important physiological and environmental characteristics. Subjects lie on a mattress pad embedded with vibration sensors to obtain multidimensional data (e.g., body temperature, heart rate, respiration rate, positional mapping and movement). The main advantage of all these solutions is that users do not need to wear any device. But, in some cases batteries are needed and it may also be uncomfortable to sleep on a pad and thus, they can affect sleep quality. For patients with incontinence there is also

a problem with keeping the pads clean.

Audio and video signals can also be used to determine sleep quality. In [38] a combination of heart rate, audio and video sensors are used to infer sleep or awake periods. A system called Lullaby combines temperature, light, motion sensors, audio and phones, and an off-the-shelf sleep sensor to provide a comprehensive recording of a person's sleep [39]. Such systems raise privacy concerns among the users. *SleepCycle* is a popular iPhone based application that uses the accelerometer in the iPhone to monitor body movements and determine which sleep phase the user is in. The user just needs to put the iPhone in a suitable place on the bed. A more accurate system is presented [40]. Also, the iPhone can accidentally fall off the bed and it needs to be connected to the charger for the whole night.

In summary, the advantages of our WISP-based sleep monitoring system are that users do not need to wear any device, they do not need to sleep on any mattress pads instrumented with sensors, no batteries are needed, the system is wireless, and it avoids privacy violations of video solutions.

2.5 Speech Monitoring

Many groups are working in the area of developing automatic emotion classification systems. There are a number of data sets that have been collected with emotional speech, and different types of classifiers have been compared [41, 42, 43, 44, 45]. There has been a comparison of two learning methods, GMMs and SVMs [46, 47] in regard to emotion detection, which produced similar results to ours: 75% for GMMs and 76% for SVMs. Various other classifiers including neural networks and decision trees have been evaluated [48, 49], and the performance on real data shows that the choice of classifier does not make a large difference in performance [50, 51]. We chose to use a SVM classifier in our approach, however other classifiers could be adapted as well.

Emotion detection can be done by targeting a particular speaker (dependent), or by applying to any speaker (independent) - even if we do not have training data for him or her. When emotion detection is done speaker dependently, it is more accurate than when done speaker independently: accuracy levels are 89-93% versus 75-76% [46]. Some evaluation has been made comparing using

overall statistics for the speech contour or by using temporal information by applying continuous Markov models [52] with better accuracy using the temporal information. However, we decided not to use this approach, since reverberation affects the temporal characteristics of the signal considerably.

In addition to emotion detection, a large number of groups are developing speaker identification (ID) systems that can identify a specific person by characteristics of their voice. Like emotion detection, HMMs, GMMs, neural networks have been compared [53]. Speaker recognition has also been applied to work on mobile phones such as in SpeakerSense [54]. StressSense use microphones embedded in mobile phones, they track markers of stress from human voice [55].

Much of the work in emotion detection and speaker ID has been tested in controlled environments, but not tested in noisy and reverberant environments. Most of the work done in noisy and reverberant environments has focused on improving automatic speech recognition [56, 57]. The more successful strategies involve the use of multiple microphones (called beam-forming) to minimize the effects of noise and attenuation. Some groups have investigated using different features than MFCC, for instance, and instead use different features that are less environmentally sensitive such as Non-negative Matrix Factorization (NMF) [58]. One group makes a HMM classifier more robust by introducing varying levels of noise into the training set [59]. This strategy is similar to the solution provided in this chapter, however while it investigates automatic speech recognition using HMM classifiers in noisy environments, we look to target speaker recognition and mood detection using SVMs in the presence of a variety of acoustic distortions.

One problem with speech monitoring is capturing verbal interaction in large physical spaces. One group investigates collecting similar data that we are interested in such as total speaking time, total conversation time, number of turns, and mean energy in decibels and achieving it using microphone array [60]. They focus on achieving high source localization, however do not address doing classification. The other method of achieving high quality of speech processing is by worn microphones, such as mobile phones. One application called StressSense uses microphones on cellphones to detect stress in a person's voice [55].

Another problem with ambient sounds is that when sound sources are produced at the same time, it is difficult to isolate each of those sources to process their features independently. The classic issue of multiple people speaking at the same time is called the “cocktail party problem”. Blind Source Separation [61] attempts to separate independent signals in a mixture by using multiple microphones and assuming that the signals are statistically independent from one another and a weighted mixture of two sources. However, at this time, convolutive blind source separation does not perform well enough to be used in practice in non-controlled environments.

Crowd++ can accurately estimate the number of people talking in one place using an unsupervised machine learning analysis [62].

3 | Empath System Design

We have discussed the history and the state of the art of home monitoring systems in the previous chapter. Two notable trends have emerged in recent years: firstly, computer applications are leveraging service-based web architectures for providing services to end-users. Secondly, new sensors are saturating the market for health-minded consumers to purchase so that they can track their own health and wellness.

There have been many remote health monitoring systems available as discussed in 2.1, but they lack several important features necessary for long term inference based monitoring. First, they lack a consistent method for representing data. In most systems [63], the data are typically stored in statically defined tables defined by statically defined SQL schemas. The problem with this method is that as new device types or different stream types are developed, the system's database and logic needs to be overhauled. Second of all, most systems have one particular design intended for general monitoring, and a specific assortment of sensors that the company feels necessary must be installed or the system does not work as intended. We believe this assumption is limiting since many applications— particularly clinical research applications— require a varied assortment of sensors. Third, each system generally implements its own closed and proprietary network protocol stack, usually by pushing mote data to a base station, then having mobile devices access the base station for the data. This makes sharing the information with many caregivers or leveraging external data sources difficult. No other system to date uses the web for storing and sharing home collected data in the same way as our Empath system. Fourth, it is important to have a system that is quickly and easily deployable in homes without an existing internet connection. Many of

our study participants live in rural areas, and do not necessarily have their own Internet service provider, therefore no assumptions can be made about an existing internet connection. Also, when the system is deployed, it needs to be able to be set up quickly with very little configuration time to minimize intrusion into the participants' lives.

3.1 Contributions

In this chapter we present the following contributions to the state-of-the art when building remote monitoring systems versus existing systems:

1. **Sensor system tailored for depression:** Monitoring many syndromes such as depression, stress, and mood require many sensor types to infer the severity of the symptoms. We present a large combination of sensing components and required analysis modules that are based on the DSM-IV criteria for depression.
2. **Generalized database:** We present a stream abstraction designed for heterogeneous and multidimensional stream data that is targeted for storing polymorphic and multi-dimensional data in a unified way in document based, SQL-based, and web service backends.
3. **Hierarchical inference chain:** We show how inference modules can be wired together so that application developers can easily adapt the system to their needs. We present a novel strategy for inference evaluation by using lazy evaluation to minimize evaluation time.
4. **Fast deployment:** We demonstrate that the system can be deployed typically in less than 10 minutes for basic sleep monitoring and upwards to 30 minutes for a full depression monitoring setup. We later show in Chapter 4 that using our training and simulation algorithms, that a speech monitoring system can be added in less than 5 minutes per person for training.
5. **Case studies deploying on the Cloud:** We demonstrate how the system can be deployed on two popular cloud-based services: the Amazon Web Services platform and the Google App Engine, with minimal change to the basic system design.

3.2 System Design

We begin to describe Empath's design by describing the system at a high level. In the Empath framework, many homes are instrumented with various wired and wireless sensors attached to motes. These motes generate either continuous or event-based data from weight scales, contact reed, PIR motion detectors, sound, and user surveys. Often, relays are needed to convert the data from one protocol or medium to another. This is often required since each device has its own particular protocol designed by their manufacturer. Also, because some sensors generate high data-rate streams, we typically employ reducers which convert the high data rate or highly multidimensional data to lower data rate and fewer dimensions. For instance, this can be achieved by buffering continuous data over an epoch window (for instance 1 minute), and then the statistics for the epoch window— such as the mean, variance, min, max, count) are reported as output. When an epoch is instantiated, it is forwarded to a basestation over the network. Once the basestation receives the epoch, it is stored temporarily and staged for syncing to a web service. Once the data is stored on the webservice, the data are evaluated by a tree of data processors that are wired to one another to generate high-level reports of various disease risk factors. For instance, a `SleepQuality` processor would be responsible for transforming acceleration data on the bed to reports on sleeping trends such as bed time and wake time. Some processors do not transform the data, called aggregators, since they aggregate many streams together into multidimensional streams which are used to render composite reports. Finally, there are a series of possible actuators, which are mechanisms and devices that influence the environment. Some examples of these are recommendation systems and speech synthesizers. Finally, there are several varieties of displays that are responsible for rendering the data for display for patients, caregivers, researchers, and technicians.

3.3 Home Area Network Design

The sensor network developed in the home is ad hoc, and may contain many different components that need to be integrated and therefore the architecture must be flexible. We used a middleware

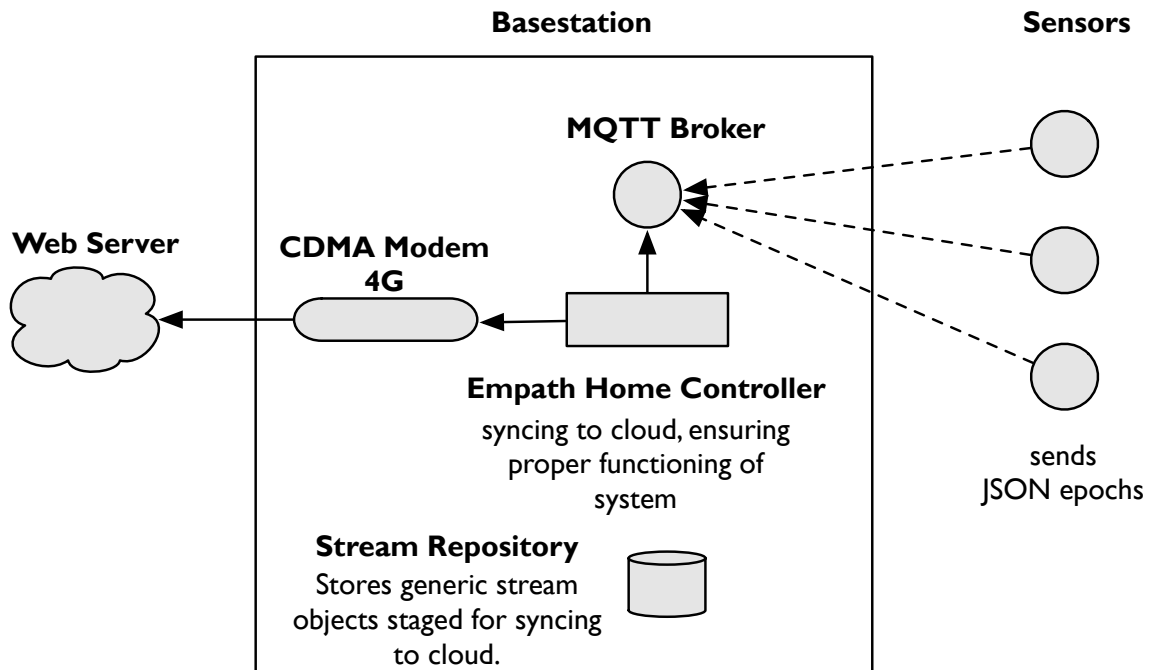


Figure 3.1: Home area network

communication layer to simplify the networking among the components. For communication, we defined a common protocol for sending data to other devices. In the home, one device is designated as a basestation, and the message broker runs on this device and listens for incoming connections. We use the MQ Telemetry Transport (MQTT) protocol [64] as implemented by Mosquitto for our message broker. MQTT implements a publish/subscribe message pattern to provide a one-to-many message distribution so that we can adequately decouple the modules from one another. In each of the packets, the payload is a series of Epochs serialized as a JSON string. MQTT uses TCP/IP to provide basic network connectivity with small transport overhead (the fixed-length header is just 2 bytes), and protocol exchanges minimized to reduce network traffic. There are three quality of service (QoS) levels that can be set for message delivery:

1. **“At most once”**, where messages are delivered according to the best efforts of the underlying TCP/IP network. Message loss or duplication can occur. This level could be used, for example, with ambient sensor data where it does not matter if an individual reading is lost as the next one will be published soon after.

2. “At least once”, where messages are assured to arrive but duplicates may occur.
3. “Exactly once”, where message are assured to arrive exactly once. This level could be used, for example, where duplicate or lost messages could be misleading.

3.3.1 Basic Operation

When a sensor is installed, a URL for the message broker must be specified as well as a device name. For example, a broker can be located on a machine at `tcp://10.0.0.1:1883` and the bed sensor is on `tcp:10.0.0.8` and the name is `bed1`. When the bed sensor collects data for one second, the mean and standard deviation are computed for each of axis of each accelerometer, resulting in a 12 dimensional feature vector. A new Epoch is created as shown in Listing 1 using the current timestamp, duration, and feature vector. This epoch is serialized to a JSON string and published to the topic: `sensors/bed1`. The Empath Controller subscribes to sensor data by subscribing to the topic group `sensors/+`. When a message arrives, the message is stored in the Controller’s local database for later syncing with the Cloud. This local database was implemented using SQLite 3¹ which is a self-contained, serverless, transactional SQL database engine. SQLite is the most widely deployed SQL database engine in the world, and is employed in the Android Operating system as the default persistence module and was chosen since it can run on both laptops and embedded devices such as the ARM (Android or Linux).

```
public final class Epoch implements Serializable {  
    private String ID;  
    private Date timestamp;  
    private long duration;  
    private List<Object> values;  
}
```

Listing 1: Basic storage unit for streams

¹<http://www.sqlite.org>

```
public class Stream implements Serializable {  
    private String ID;  
    private Map<String, Object> metadata;  
}
```

Listing 2: Data abstraction for a stream

3.4 Sensing Devices

Our application types such as depression, Alzheimer's monitoring, and epilepsy require many different sensor types and therefore the system must accommodate each easily without requiring considerable changes to the configuration. Each sensor program must be built as an independent module implemented in the programmer's choice of language (i.e. JAVA or Python) and be launched from the command line. The modules have been built to be cross-platform and have been tested on both embedded platforms such as the Raspberry pi and more powerful computers such as laptops. Besides modules that we create, there are others created by third parties such as the Fitbit and Withings scale that have their own mechanism to send data to their own server in the cloud. Currently, we have incorporated just two of these types of sensors, a Withings weight scale and the Fitbit, which publish the data to the Withings and the Fitbit web servers respectively. Because many commodity components release an API for their data, it can be easily incorporated into Empath's system along with the sensor data collected internally.

```
public final class BedSensor extends AbstractDataCollector {  
    public BedSensor(String brokerURL, String clientName, String serialPortName);  
    public void init() throws DeviceInitException { ... }  
    public void start() throws DeviceInitException { ... }  
    public void stop() { ... }  
    public Boolean isCollectingData () { ... }  
    public void shutdown() { ... }  
}
```

Listing 3: Implementation of a BedSensor module from the AbstractDataCollector

Table 3.1: Stream Metadata Examples

Key	Description
Creator	UUID of the user who created the stream
DeploymentID	UUID of the deployment the data came from
TargetID	UUID of a user the stream might relate to
PreferredRenderer	Bar plot, time series, table, etc
Device	Specification of the device make and model

When a developer creates a new sensor device for the Empath system, he must extend the abstract class `AbstractDataCollector` which implements basic functionality such as maintaining a connection to the message broker and handling the serialization and publishing messages to the broker. In addition, one must implement virtual functions such as initialization, starting, stopping, and shutting down the device. An example of the implementation of the module is shown in Listing 3 which is shortened for brevity, although the original only contains 300 lines of code. The Controller module implements a “heartbeat” ping to ensure proper data collection, so `isCollectingData` is useful to see if a device is collecting data or not. Each sensor module therefore subscribes to its own control topic: `sensors/<deviceID/control` so that other sensors or controllers can send messages to it to stop data collection, change sampling rates, or check basic operating statistics.

3.4.1 Sleep Monitoring Module

At least two accelerometers are attached to a Wixel mote. A program to sample the ADC pins was written in C and compiled for the Atmel microcontroller and loaded onto the firmware. When deployed, the accelerometers are taped on top of the mattress but under the bed sheet or mattress pad. One accelerometer is placed on each of the left and right sides of the bed. At the basic level, the accelerometer will capture events such as when a person gets in and out of the bed, as well as any movements due to restless sleep. The data can be sent wirelessly using a 2.4Ghz radio to another Wixel configured to be a radio receiver for the basestation. A detailed explanation of sleep monitoring hardware and analysis is presented in the Chapter 5.



Figure : (a) X10 motion sensor (b) X10 contact reed (c) X10 receiver

Figure 3.2: The X10 Home Security devices are used for collecting motion or activity events.

3.4.2 Activity Monitoring

Our activity monitoring subsystem uses wireless home-security motes placed on objects in the home and on the walls to track the movement through the home. Various options are available for wireless mote switches such as ZWave, MicaZ, and TelosB, but we chose the X10 line of devices because of their long lifetime. The motion detectors and contact reed switches use the X10 protocol such that when the switch is triggered, a pulse train of 5 duplicate packets are sent in a 1 second interval. No MAC-layer is implemented and there are no retries or guaranteed delivery. The delivered message is then captured by an X10 radio receiver (W800RF32A), and sent through the USB port to the base station. We reverse engineered the serial protocol from the signal received by the W800RF32A device to parse the packet for the **HouseID** (A-P) and the **UnitID** (1-9), which are the mote's ID. Unfortunately, the contact switch has a factory-assigned ID that cannot be reassigned. Ensuring that the devices deployed are all unique is very important. However, the motion switch does allow the user to change the **HouseID** and **UnitID** by using buttons on the device.

We placed motion detectors on doors, small rooms, and different places along the wall. The primary purpose of placing motion detectors is to track movement through the home. The motion detectors will fire whenever a person is close enough to the field of view of the sensor. It was necessary in many situations to tape off the periphery of the sensor to achieve a narrower sensing range. Contact reeds switches are attached to objects that move or have parts that separate such as a refrigerator door, cabinet door, and a stove switch. They fire only when two contact points of the reed are separated, which in turn shorts the current. Using the raw data from X10 devices,

information can be inferred about user's activities of daily living such as the time a user interacted with an object. The X10 relay produces a new **Epoch** consisting of the timestamp in milliseconds, the duration of the event, the signal strength of the message to a stream for the device and adds this epoch to a stream dedicated for that object. This signal strength is important for maintaining system health, and to make recommendations on better placement of the sensors.

3.4.3 Speech Module

Monitoring a person's speech in the home can be very useful for extracting components that correlated to different types of mood and for monitoring the level of social interaction the person engages in. All of this information relates to determining whether the person may be depressed or not. We created a novel speech monitoring system that uses USB microphones attached to embedded platforms such as the Beaglebone, Beagleboard, or RaspberryPi to collect speech samples and process it for the identity of the speaker. It also compares the signal to a corpus of emotional speech from various third-party speakers and selects the emotion most related to the component features. The microphones can be placed in unobtrusive areas of the room such as on top of the mantel or the corner of the room. Other options include as a centerpiece on a dinner table where conversations occur in the home. It is important to note that no actual conversation content needs to be transmitted or stored— the user's privacy can be respected while at the same time extracting useful but generalized features from it. This component is explained in great detail in Chapter 4.

3.4.4 Subjective Mood Scoring

Subjective evaluation through questionnaires are vital for monitoring psychological disorders. Sensors cannot fully replace this important element in psychological assessment. However, it has been shown that questionnaires inaccurately report data when responses are gathered retroactively. Due to a hindsight bias, people typically underestimate their mood. *Experiential sampling* is a technique where questions can be gathered immediately when the event occurs. In line with this, we created a mobile app that runs on a touchscreen or an Android tablet. We show the screen in Figure 3.3.

This app serves as a “sensor” for gathering subjective assessments from the patient at any time. The first questionnaire we implemented for the device is the PHQ-9 (Patient Health Questionnaire) which asks the patient some specific questions related to occurrence and severity of depression. The user can complete this evaluation at any time, however it is frequently set by the caregiver or study coordinator to administered on a two week schedule. The device displays the prompt, “Over the past 2 weeks, how often have you been bothered by any of the following problems?” to which the user responds to 9 questions related to their mood and depression. The complete questionnaire is shown in Appendix B. Many different basic questionnaires can be added to the application, and these are not limited only to depression, and could include questions dealing with diet. In fact, when a new survey needs to run on the app, it receives as a JSON message from the webserver composed of a vector of strings for each of the responses. The types of responses are in themselves a list of strings. When a patient completes a questionnaire, a new Epoch is instantiated, the timestamp set to the current date, the duration being set to the time taken to complete the questionnaire, and the responses being a vector of polymorphic responses. In the case of the PHQ-9, the responses would be values between 0-4 producing a 9-dimensional vector, but other questionnaires would accept Strings for responses.

The stream used for questionnaire responses are typically wired to a PHQEvaluator module that processes and scores these 9-dimensional Epochs into a stream containing just a single value. The PHQ-9 and many other questionnaires will typically take the sum of each of the items to form a composite score. This score as shown in Table 3.2 is useful for recommending treatment options and assessing the possible level of severity.

3.4.5 Weight monitoring

We use the \$100 WiFi weight scale shown in Figure 3.4 to track the user’s weight. A WiFi connection must be established on the scale. When a user stands on the scale, and his weight measured, the value sent wirelessly to the home’s router and to the Withing’s web server ². Later, when care-

²<http://www.withings.com>

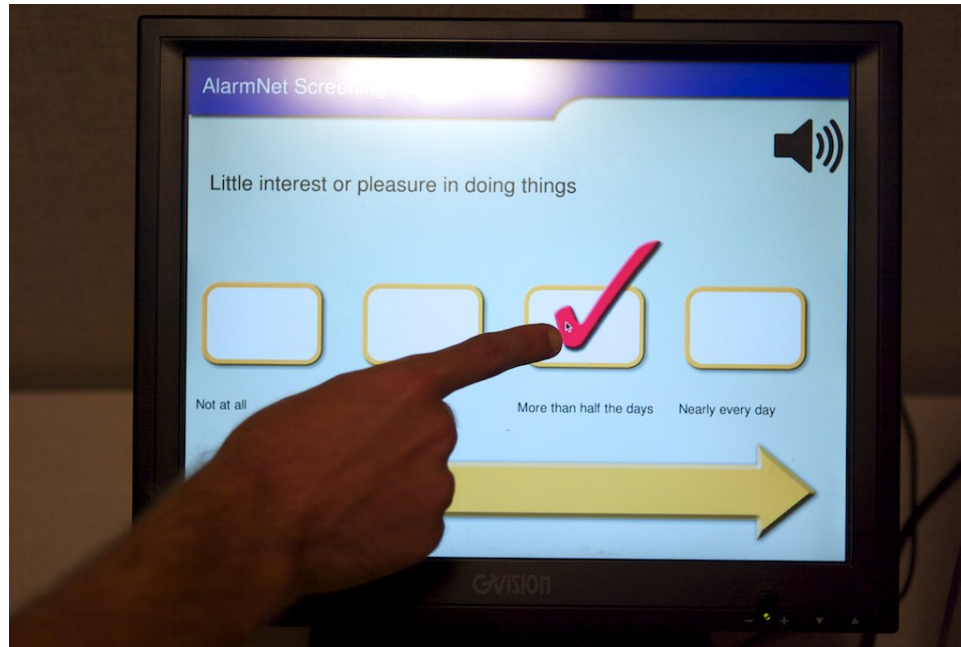


Figure 3.3: The subjective mood can be input into the system using a touchscreen device shown, the system has been ported to an Android App for various tablets or phones.

Table 3.2: PHQ-9 Scoring

PHQ-9 Score	Minimal Symptoms	Treatment Recommendation
5-9	Minimal Symptoms	Support, educate to call if worse, return in one month
10-14	Minor Depression, Dysthymia, Major Depression (mild)	Support, watchful waiting, antidepressant or psychotherapy
15-19	Major Depression, moderately severe	Antidepressant or psychotherapy
>20	Major Depression (severe)	Antidepressant and psychotherapy

givers need access to the information, the Empath data layer pulls the data from their web service exposed as a SOAP webservice. The bodyweight stream is typically wired to a `BodyWeightEvaluator` which gathers the past two weeks of historical weight data and detects any significant weight gain or loss. As per the DSM-IV criterion[65], if the patient's body weight changed at least 5% in the past two weeks, it could be a sign of depression. If no new measurements have been taken for a week, an alert appears on the patient's touchscreen device. The following scoring system implemented by `WeightEvaluator` is used for the weight factor:



Figure 3.4: The Withings weight scale takes body weight and transmits the information to their web server for keeping track of weight over time.

Table 3.3: Weight Factor Evaluation

0 pts	within 5% gained or lost
1 pt	over 5% gained or lost
2 pts	over 10% gained or lost
3 pts	over 15% gained or lost

3.4.6 Integrating other Mesh Wireless Networks

Many other sensors in the home require being wireless and battery powered such as chair sensors or tablemat sensors. For this purpose, we implemented a Zigbee-based mesh network³ using the Xbee motes from Digi International. Each mote in the Zigbee network needs a common Personal Area Network (PAN) number set during configuration. Each mote has a preconfigured and unchangeable 64-bit value as its ID. One device must be configured as a Coordinator in order to establish a routing tree for the Zigbee network, and the other devices are set as Routers and End-devices depending upon whether duty cycling is required. A relay was created for the Coordinator mote and the basestation so that when a packet is received it converts the source ID into a stream ID for the particular device that was used.

³<http://www.zigbee.org>



Figure 3.5: XBee Pro Series 2 Radio and mote capable of 1 mile transmission range 215 mA transmit current only 2.5uA when in sleep mode.

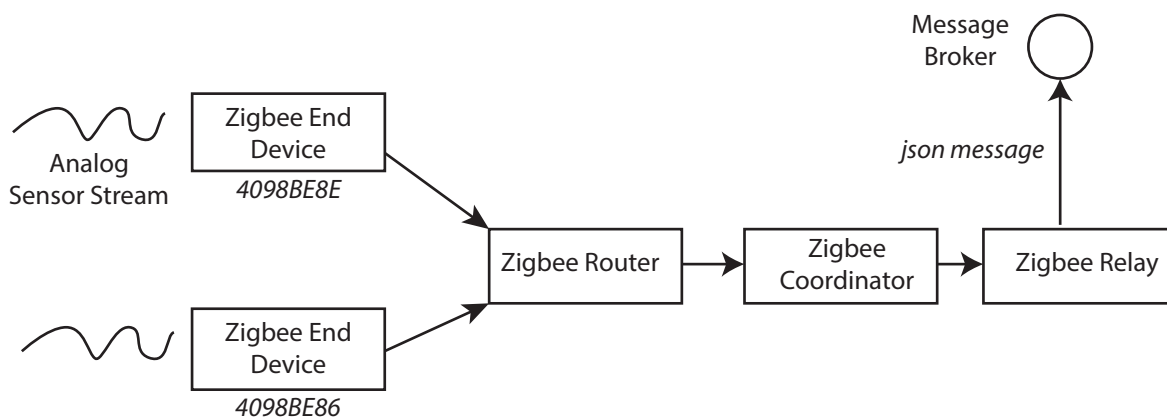


Figure 3.6: Zigbee communication from motes to basestation

3.4.7 Syncing to the Cloud

To make the information available to caregivers and technicians, internet connectivity is necessary. One important aspect to connectivity is that it improves our ability to ensure that the data collection is running correctly, and enables us to intervene using remote desktop or remote shell if there is a problem. In preparation for our clinical studies, we spent several months testing the reliability of the connection by monitoring the number of lost packets at different time periods. We created a script that pings our web server to detect the number of lost packets over a time period. Because of the intermittent disconnections, we decided to employ a synchronization based communication



Figure 3.7: UML290VW a 4G CDMA modem used for Verizon internet connection.

protocol. Instead of directly sending the Epoch directly to the server when they are generated, the Epochs reside locally on the basestation's SQLite database until a connection can be established for synchronization. Many mobile and web apps employ a similar strategy for their communications with the cloud. Staged Epochs are sent to the web service using a HTTP POST request, and the HTTP response must be a success (200) for the basestation to tag those epochs as synced so that they are not sent again.

3.5 Cloud Layer

Empath's server is implemented by a JAVA web application, thus when the code is compiled it is a deployable WAR file that can be installed into any JAVA web server container (Tomcat, Jetty, JBoss, etc). We used the Spring3 ⁴ framework for handling the Model View Controller (MVC) pattern for handling requests from the clients and handling serialization of JSON messages. The Spring Security extension was used to implement Empath's authentication and access control using the bcrypt cipher for hashing the passwords on the database. Each user is given a set of roles such as **Patient, Clinician, Technician, Administrator, Coordinator, Researcher** which is enough to

⁴<http://www.springframework.org>

handle course-grain requests to resources. For a more robust access-control mechanism, these roles can be predicated, so that a user can be a Clinician for X , and X is a Patient of study Y and Researcher is a member of Study Y . Before any resources are served, a user must sign in, and a session ID is created and stored as a cookie in the HTTP client, and the communication provided through an HTTPS tunnel.

The primary interface to the web is the Request Dispatcher. The role of this request dispatcher is to translate the request pointing to a URL to the appropriate Controller to handle processing the task. The current implemented Controllers are `StreamController`, handling all access to data in a stream, the `UserController` for handling any information about a `User`, and `ProcessorController` for `Processor` objects.

We implemented the system on two Cloud platforms: the **Amazon Web Services** and the **Google App Engine**. For the Amazon Web Service, we launched a two “small” virtualized EC2 instances with 2 GB of memory. One EC2 instance ran a Jetty9 instance for our application, and another ran only the MongoDB database. The instance with the database mounted a RAID10 array with 8 GB of Elastic Block Storage formatted with the XFS filesystem. For the Google App Engine, we used their native components, such as the Jetty web server and the Google Data Store for the database. Changing the deployments only required changing the Spring’s XML configuration file describing what DAO implementation was to be used for the local stream data. More about the databases are described in Section 3.6.1.

3.6 Streams on the Cloud

There are few abstractions for stream-like objects on the Web. We presented an abstraction that meets the special needs of sensor streams called a StreamFeed [66]. This new abstraction is similar to a regular web resource such as HTML or an image that it can be referenced by an URI. As such, they can be both the source and the target of hyperlinks. All data in the *Empath* system are exposed simply by a URL, where the **UUID** is a globally unique 128-bit string:

<http://www.empathproject.com/stream/<UUID>>

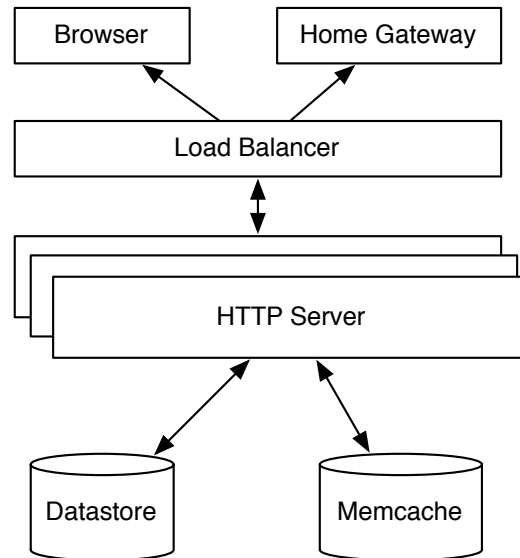


Figure 3.8: Empath deployed on Google AppEngine’s Cloud infrastructure.

Likewise, all Epochs inside of a stream also can be retrieved using a URL:

<http://www.empathproject.com/stream/<UUID>/epoch/<ID>>

Each of these URLs point to a *StreamFeed* that can be fused, processed and filtered to create new *StreamFeed*. The result can be repeated to eventually produce a *inference tree* as described in Section 3.7. For example, data from multiple sensors in the home can be combined to form better estimates of the occupant’s behaviors. Because the returned stream is often quite large, a series of time range parameters are encouraged, also a particular attribute can be filtered. Consider the following HTTP GET request to:

<sensor/<UUID>/epochs/filter/?min=100&day=Tuesday>

This query restricts the query response to only contain the values in the data stream that have a **value** attribute greater than 100 and a **day** attribute equal to “Tuesday.”

This URL-based resource interface conforms to RESTful (Representational State Transfer) principles listed in Fielding’s thesis [67]. Unlike RPC-based web services with XML-RPC and SOAP, where custom methods are defined in some WSDL document, a RESTful application promotes a standard interface of using a set of four commands: POST, GET, PUT, and DELETE to handle each of the needs of a resource such as CREATE, READ, UPDATE, and DELETE. The

advantage of using a RESTful interface is that there is an inherent standardization placed on the operations that can be applied to the resources, without needing to explicitly define descriptions of the methods. The URL contains all the information that is needed to return to a particular state of a web service.

The following table lists the types of HTTP requests that the resources support:

Table 3.4: RESTful HTTP Requests

HTTP Command	Description
POST	to create a resource on the server
GET	To retrieve a resource
PUT	To change the state of a resource or to update it
DELETE	To remove or delete a resource

When a user is logged in and wishes to view the listing of all the streams he has access to, an HTTP GET request is directed to `streams/` and a JSON message of a list of `Stream` object is returned. When a new stream needs to be created, a POST is directed to `streams/` with a JSON message describing the name and metadata for the stream. When a new Epoch needs to be added to an existing stream, a POST is placed on the `streams/<UUID>/epochs` where the UUID is the stream ID. Other web servers can subscribe to the streams by polling for new data using a GET command to `streams/<UUID>/epochs`. Lastly, when a stream needs to be deleted, a DELETE command is sent to `streams/<UUID>`.

3.6.1 Stream Storage

There are three basic types of streams in Empath. First, there are persistent streams that are stored in a database of some type. Second, and very common, are memory streams that do not have persistence and are populated upon request. This is useful for streams are only needed to produce some report to a caregiver. This allows the Evaluator objects to store the results temporarily like a scratchpad, so that clients can quickly query for the information without requiring the entire inference chain to be recomputed. Third, there are web streams which are data sources that are not stored locally in the Empath system, but rather through another webserver on the Internet.

The default persistent data dao is targetted for MongoDB ⁵ which is one of the most popular document-based databases. In addition, we implemented the Google Datastore which is another document-based database for storing non-relational data using the powerful and clustered Google servers. For web services, we implemented a basic Fitbit ⁶: an API for gathering the activity level and step counts from a Fitbit tracker. For temporary or cached data streams, we implemented our own hashtable-based storage, or for the Google App Engine, we used the *Google Memcache* service which is a high performance scalable distributed in-memory data cache. When a developer creates a new source for streams, they must implement `StreamListingDaoInterface` that stores the stream descriptions and metadata for the streams and `StreamDataDaoInterface` which holds the actual Epochs. The following are the interface definitions:

```
public interface StreamDataDaoInterface {  
    public void addEpoch(String streamName, GeneralEpoch e)  
        throws StreamNotFoundException, StreamTypeNotSupportedException;  
    public List<GeneralEpoch> getEpochs(String streamName, Date beginDate, Date endDate)  
        throws StreamNotFoundException, StreamTypeNotSupportedException;  
    public List<GeneralEpoch> getAllEpochs(String streamName)  
        throws StreamNotFoundException, StreamTypeNotSupportedException;  
    public void clearStreamData(String streamName)  
        throws StreamNotFoundException, StreamTypeNotSupportedException;  
    public void createStreamData(String streamName)  
        throws StreamNotFoundException, StreamTypeNotSupportedException;  
}
```

Listing 4: Stream Data Access Interface

⁵<http://www.mongodb.org>

⁶<http://dev.fitbit.com/>


```
public interface StreamListingDaoInterface {  
    public String createStreamByType(String type)  
        throws StreamTypeNotSupportedException;  
    public String createStreamByID(String id, String type)  
        throws StreamTypeNotSupportedException;  
    public void deleteStream(String streamID) throws StreamNotFoundException;  
    public Stream findStreamByID(String id) throws StreamNotFoundException;  
    public List<Stream> findAllStreams();  
    public Object getMetadata(String streamName, String key)  
        throws StreamNotFoundException;  
    public void setMetadata(String streamName, String key, Object value)  
        throws StreamNotFoundException;  
}
```

Listing 5: Stream Data Access Interface

Empath grew from the ALARMNET system, and as such we used the MySQL⁷ database backend. Although this did not cause many problems initially due to the small number of streams we collected then, development was stifled by working with unique table schemas. In order to minimize the amount of refactoring when new stream types need to be added to the system, we designed a `StreamService` abstraction that allows the users to store the information in a unified way, preventing them from adding new schemas or table types to the database and allowing streams to be handled in a unified way.

We moved away from using relational databases because our application domain did not need their consistency requirements that they provide, and we could relax those restrictions to get higher performance and scalability. Most relational databases (such as MySQL) enforce ACID guarantees for atomicity, consistency, isolation, and durability. However, our data being generated rarely changes once being committed, also even if it is changed, the data does not need to be consistent across the database replicas, just eventually consistent. Document-based databases achieve much higher performance in the write once, read many times use case. The following table from [68] compares relational databases with document-based databases:

⁷<http://www.mysql.org>

Table 3.5: Comparison of Database types

	Relational Database	Document-based
Data size	Gigabytes	Petabytes
Access	Interactive and batch	Batch
Updates	Read and write many times	Write once, read many times
Structure	Static schema	Dynamic schema
Integrity	High	Low
Scaling	Nonlinear	Linear

3.7 Data Inference Design

When a deployment is setup, many stream `Processors` are created and their operating parameters are set. Afterward, streams are “wired” to the input and output ports for these `Processors`. Consider a simple example with scoring questionnaires. A `PHQ9Evaluator` `Processor` is created and the input port ‘PHQ-9 Responses’ is wired to a persistent stream `A` holding the item responses, and to the output port ‘PHQ-9 Score’, a memory stream `B` is wired to the port. When the `PHQ-9` stream is queried for the first time, the stream holds no `Epochs` and is marked ‘dirty’. Because of this, the `StreamService` invokes the `PHQ-9 Evaluator`’s `evaluate` function which will query `A` for the all epochs in that time range. Because `A` is persistent and not marked dirty, all the values are available and evaluation does not need to be taken to another level. Next, for each of the epochs, a score is produced and the result added to stream `B`. This evaluation method uses a *lazy evaluation* strategy for fetching stream information because the rate of querying for higher level data is much less frequent than the production of lower level data. When a lower level stream gets appended to, the streams above it are marked ‘dirty’ for reevaluation. There are some requirements to this structure, most importantly, there cannot be any cycles in the inference chain or evaluation will never halt. More examples of a complex tree is described in the case study for depression in Section 6.3.

3.8 Caregiver Displays

We developed a user interface especially for caregivers and clinicians such as therapists, nurses or doctors. The caregiver's screen is shown in Figure 3.9 and is implemented as a dynamic webpage using AJAX (asynchronous JavaScript) for communicating with the webserver through the RESTful interfaces described in 3.6. We chose to implement most of the rendering on the client side versus the server side because it is often advantageous to download the data once, but be able to display it in different manner through the client's UI. The JavaScript client enables the data to be cached, and the user can adjust their graph types or time region as appropriate.

When the system is configured for depression monitoring, after the caregiver signs in, the dashboard displays a list of attending patients each with a risk factor of 1 – 5 rating for depression. When a patient is selected, a summary of the current behavioral factors: sleeping quality, social isolation, PHQ-9 score, weight, movement levels, and speech factors are presented as a column plot. Each factor is represented on a scale (colored from green to red) representing the risk for a particular factor. When the caregiver selects a particular factor, another plot appears displaying the detailed information in an appropriate time-series plot, bar chart, or table. Each stream has an optional metadata tag called `preferredRenderer` in which the data can be at first displayed as time series, bar plot, table, or more specific types can be added as plugins such as sleep quality and questionnaire results.

To give an example of the operation, when the PHQ-9 is selected, the patient's historical tests and the responses for each of the items can be individually viewed and evaluated. For sleep, detailed statistics can be shown, such as bed time, number of interruptions, and sleep duration. To put each patient's history in context and to see if a patient is improving, annotations can be added to the display indicating when a patient started new therapy or medication. It is important to understand that this system does not perform diagnosis, but rather it exposes all the factors and data in a cohesive way in order to improve diagnosis.

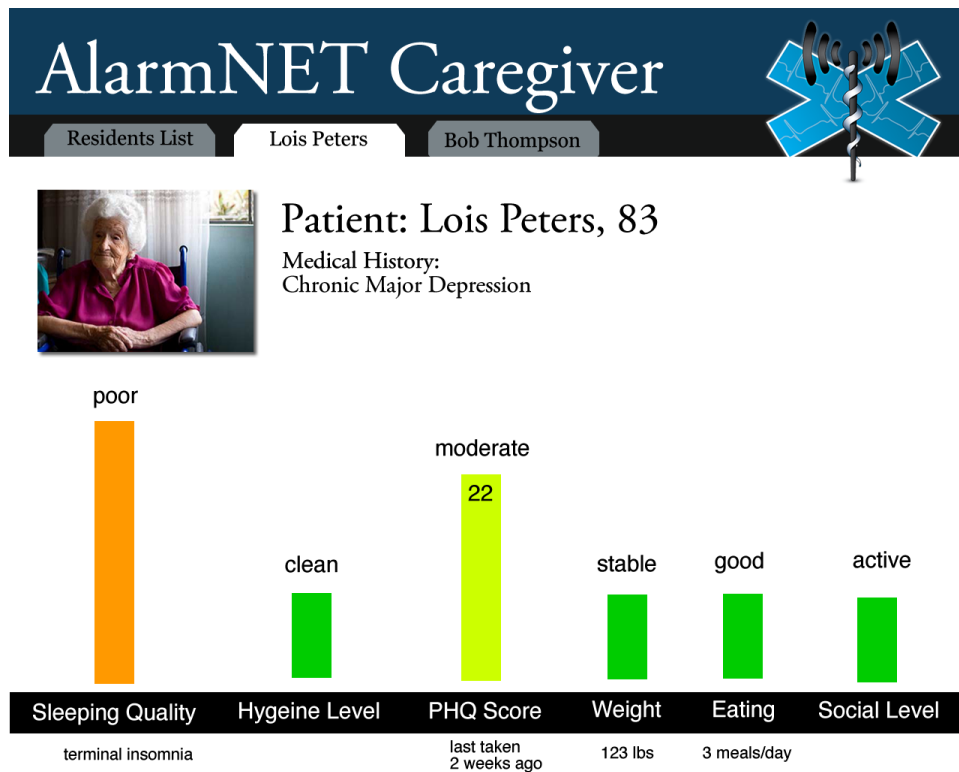


Figure 3.9: The clinician display shows an overview of the various factors related to the monitoring, and allows for drilling down to lower-level information

3.9 Conclusions

In this chapter, we presented the system design of Empath. Each of the core sensing modules were shown with examples of typical operations for processing the data. We presented unique way of representing that data on the web, stored in distributed document-based databases. Finally, we have shown how it is deployed on the Amazon Web Services and Google App Engine Cloud platforms.

4 | **Passive Speech Monitoring**

We have shown in the previous chapter how various modules can be created and linked together to combine various modalities to report more information to the caregiver. In this chapter, we present a novel system for the collection of speech features taken ambiently from rooms to offer valuable information to complement the other “traditional” home sensing modalities one sees in smart homes.

4.1 Introduction

Numerous studies show that mental illness or disturbances manifest themselves in speech [69, 70, 71, 72]. In fact, during a mental status examination, a clinician makes a psychological assessment by observing and describing his patient’s speech. The report usually includes some comments on its features such as loudness, rhythm, prosody, intonation, pitch, phonation, articulation, quantity, rate, spontaneity, and latency. Some features may indicate a neurological problem: for example, stroke or dementia can slow speech or produce aphonia or dysarthria. People with autism spectrum disorders or Asperger’s syndrome show abnormalities in their speech. People with mania or anxiety may have rapid, loud, and pressured speech, while people with depression show prolonged speech latency and speak in a slow, quiet, and hesitant manner and also use only small changes in intonation.

In addition to the speech’s features, other information such as how often the patient has conversations with others, and how often the patient actually speaks during these interactions provides a picture of the level of engagement for the speaker. Because conversations involve multiple people, a system must be able to identify who is speaking at any given time in order for a personalized report

of speech features to be generated for each user. Although speech information is generally seen as sensitive private information, we employ a strategy in this work where the content of the speech is not needed, only the high-level features.

There are many technical challenges in designing a system that capture these features accurately and without distortion in real environments. First, there may be significant ambient noise in the home, including that from music, television, appliances and air systems. Second, as with any propagating signal, increasing the distance between the emission source and the microphone attenuates the signal, resulting in a low signal to noise ratio. Third, and the focus of this chapter, is that when sound travels through rooms, it becomes distorted by an effect called reverberation. The amount of reverberation is related to the amount of time the original sound spends bouncing off of surfaces before being captured by the microphone. The amount of distortion depends largely on the acoustic characteristics of the room, which are related to the presence of acoustically insulating or reflective materials such as hardwood, carpet, furniture, and drapes. The final challenge is that the system is dynamic: users will change their position as they move about the house.

There is a large existing body of work for creating classifiers and completing necessary feature extraction for obtaining the identity of the speaker [54] and the speaker's mood [73], however they all make very limiting assumptions such as that the microphone and speaker are in fairly anechoic (non-reverberant) and non-noisy conditions. Previous studies show how mood detection is very challenging when audio is captured in realistic environments and standard classifiers (SVM and GMMs) are employed [73, 74]. We also show later in our evaluation, Section 4.4, many examples of how reverberation degrades the performance of SVM classifiers for both speaker identification and mood classification from 80-90% accuracy in non-reverberant conditions to only 20-50% with reverberant conditions.

Speech processing in open, realistic environments is an active and open research problem, but the majority of work to date has concentrated on automatic speech recognition – the task of producing text from speech content. One notable example is how to achieve accurate automatic speech recognition to allow hands-free mobile device interaction while driving a car. A recent survey paper

describes the state of the art for controlling reverberation for automatic speech recognition (ASR) [74]. Whereas ASR uses only MFCC features and HMMs for classifiers, mood and speaker recognition approaches use hundreds of features, taken over several frames of audio, with different types of classifiers such as the SVM and GMM.

The main contributions presented in this chapter are the following:

- We present a design for a practical platform for monitoring speech: such as speaker identification and mood, for use in home and office environments that can be deployed, trained, and configured quickly.
- We present and thoroughly evaluate a novel system called RESONATE, which combines a matched condition training approach with a unique reverberation impulse response simulator. This system allows a single training corpus to be adapted for various environments, minimizing necessary training and configuration time. We demonstrate that RESONATE performs close to the ideal baseline for accuracy, both in controlled experiments (six different rooms in houses and offices) and in uncontrolled long term deployments in both a home and an office.
- We demonstrate and evaluate how additional knowledge about the environment further improves accuracy, including data about room dimensions and position of the speaker in the room.
- We benchmark various stages of the classification task on different platforms, and offer an analysis of its performance. We show best performance when capture, feature extraction, and classification occurs on-node, while training and simulation is done off-node on a base station or cloud service.

4.2 Reverberant Environments

Addressing environmental reverberation can be tackled in two main ways, [74]: The first strategy is to modify the front-end which tries to reverse or mitigate the effects of reverberation in the the

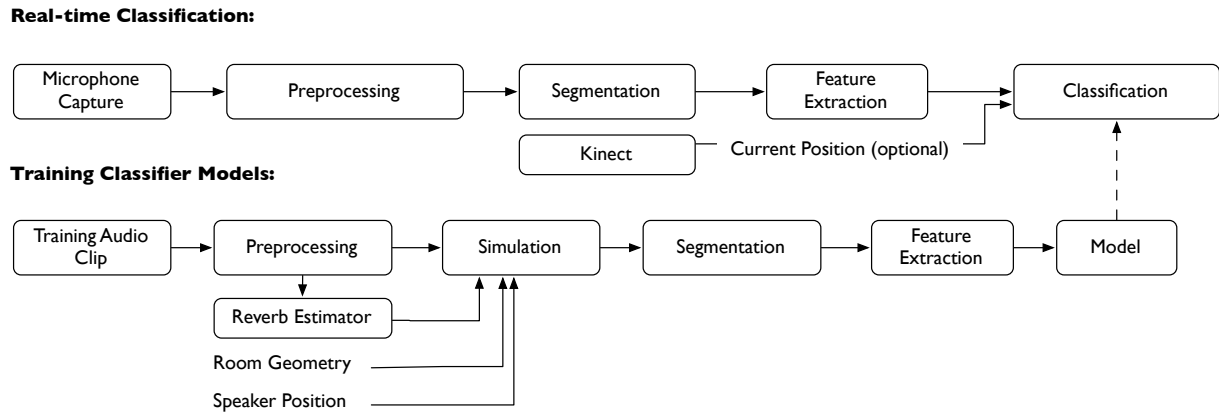


Figure 4.1: The classification processing pipeline extends the traditional approach by adding a simulation modeling step that introduces additional knowledge such as the room geometry.

preprocessing and feature extraction stages. Either the audio is preprocessed to explicitly reverse the reverberation, or only features that are robust to reverberation are selected. In this case, the classification model is left untouched. The second strategy takes the opposite approach by changing the classification model in some way to adapt it to handle reverberation.

4.2.1 Our Approach: RIR Simulator

Our approach called RESONATE, for Reverberant Environment Simulation, does not change the frontend nor the classifier, instead it works by transforming the training set to match the testing conditions. The advantage of this approach is that it can work alongside existing approaches for improving the frontend or backend of a classification system, but augmenting the pipeline to first match the testing condition correctly. Our classification system consists of a pipeline of components shown in Figure 4.1. One basic way to obtain matched training samples would be to record the subjects speaking in a number of different environments and locations. However, in this chapter we show that recording each speaker in the requisite number of locations and orientations can be a tedious process which involves over 30-60 minutes for each room in the house, thus makes this approach quite impractical. Of course, this time scales linearly with the number of possible speakers. Ideally, one small set of well conditioned recordings should be captured for a person and it can work in all environments.

With RESONATE, we use acoustic physical models to characterize reverberation for a particular room if the dimensions and some basic parameters are known. The result of this physical model is a room impulse response (RIR) which is essentially an FIR filter. Once this RIR is obtained, each *clean* recording can then be convolved with this filter to obtain a simulated reverberated sample for training. The difficulty now becomes obtaining these impulse responses. In [74], this process can be done by empirically by emitting a very short duration signal into the environment and capturing the signal after it has propagated through the environment for several milliseconds to infer the impulse response. This is a complex and long process that requires expensive audio equipment to do properly because synchronization is important.

Our solution is to use acoustic physical models to synthetically generate RIRs using a unique impulse response simulator that produces acceptable accuracy despite using rooms that are not quite perfectly cuboid or homogeneous in their wall reflectivity. We generate the RIRs by extending Habet’s implementation [75] of the Image Method [76]. The necessary parameters for this model are the sound velocity (usually 340m/s, but varies by temperature and humidity), the position of the microphone and the speaker, the room dimensions, and an estimate of the reverberation time \overline{RT}_{60} . The technique models the wave function as shown in Equation 4.1, where X and X' is the position of the source and receiver respectively and R represents the 6 wall geometry.

$$p(t, X, X') = \sum_{p=i}^8 \sum_{P=-\infty}^{\infty} \frac{\delta t - (|R_p + R_\tau|/c)}{4\pi|R_p + R_\tau|} \quad (4.1)$$

Obtaining \overline{RT}_{60} “blindly” by analyzing only the received signal is an active research problem. Reverberation is characterized by two components: the early reflections, which depend on the relative positions of the speaker and the microphone and which can be handled by the model, and \overline{RT}_{60} which is independent of these parameters, but depends on the nature of the materials of the room surfaces, which are not specified to the model. Since \overline{RT}_{60} only depends on the room’s properties, it can be inferred from a sample that has been reverberated by a real room with material properties similar to the room being simulated. We used Löllmann’s [77] algorithm of blind reverberation time estimation. The approach uses a simple statistical model for the sound decay and

\overline{RT}_{60} is estimated by a maximum-likelihood (ML) estimator. We tested this algorithm by creating reverberated samples by convolving the RIRs from the Aachen Impulse Response (AIR) database [78] with 60 speech recordings. We then compare our blindly determined \overline{RT}_{60} value with the true \overline{RT}_{60} value (from the annotation in the AIR database), and found it to be within 60 ms of the true RIR in most cases.

For handling the case where the speaker is positioned in various parts of the room, we used our model to synthesize several RIRs one for each of the various locations the speaker can occupy. This was practically achieved by subdividing the room into a grid pattern on the X-Y plane with a 1 meter offset (the height was set to the average height of a person). In certain setups, sensors, such as trackers, chair sensors, Kinect, may be available to estimate the location of the speaker in the room. In such a situation, only one of the RIRs above are selected based on position. Several training models are stored in a classifier bank, and during runtime the system adaptively selects the best classifier to use based on the current position.

4.3 System Implementation

We show a working example of the system operation in Figure 4.2. First, a node (with a microphone, beaglebone, and WiFi) is placed in each room where conversation typically occurs. Every person who lives in the home will train on one of the microphone devices close to the microphone to minimize reverberation and distortion. The recordings are sent to the basestation. Frequent visitors to the home can also do training, perhaps on their personal computer or phones, and have their training samples uploaded to the model generator which can either be on a webserver or home basestation. Each of the training samples are transformed to sound as if they came from a particular kind of room, and a tailored classification model is generated and that model is sent to the corresponding node in the system so that classification can be done in realtime without transmitting raw signals from the node. Finally, the classification result is sent back to the basestation.

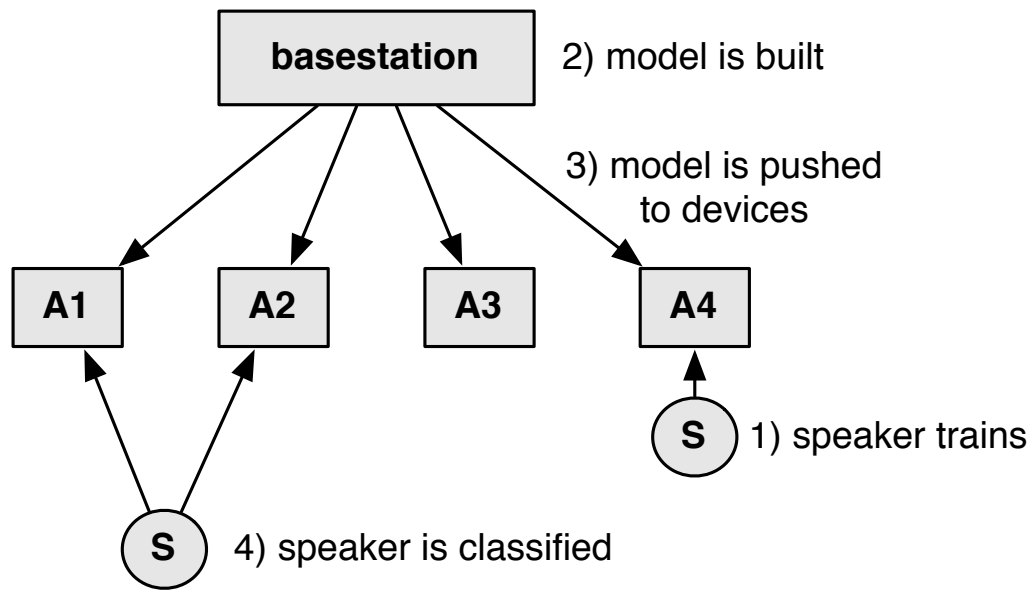


Figure 4.2: Four microphones and Beaglebones are installed in different rooms. The speaker trains in front of the microphone in room A4 and the base station builds a model that is pushed down to the other devices for classification later.

4.3.1 Audio Capture

There are many varieties of microphones and they are often referred to by their transducer type such as a condenser, being dynamic, or using MEMS. Most microphones also have a specific directionality (polar pattern) it was designed for (i.e. omnidirectional, unidirectional, cardioid, and shotgun), indicates how sensitive it responds to sounds arriving at various angles about its central axis. Microphones also have a unique dynamic range. Although microphone intrinsic are important to consider, evaluation of microphone selection is out of the scope of this chapter. For our testing, we used two types of microphones: high-end dynamic microphones and a desktop USB dynamic microphone, and both have cardioid response patterns.

We built a distributed system based on the device shown in Figure 4.3. For each node, we used a Beaglebone Black platform equipped with an ARM7 Cortex-8 microcontroller and 512 MB of memory. We loaded a Linux kernel (v.3.81) compiled for the ARM. We also created a similar device with the RaspberryPi.

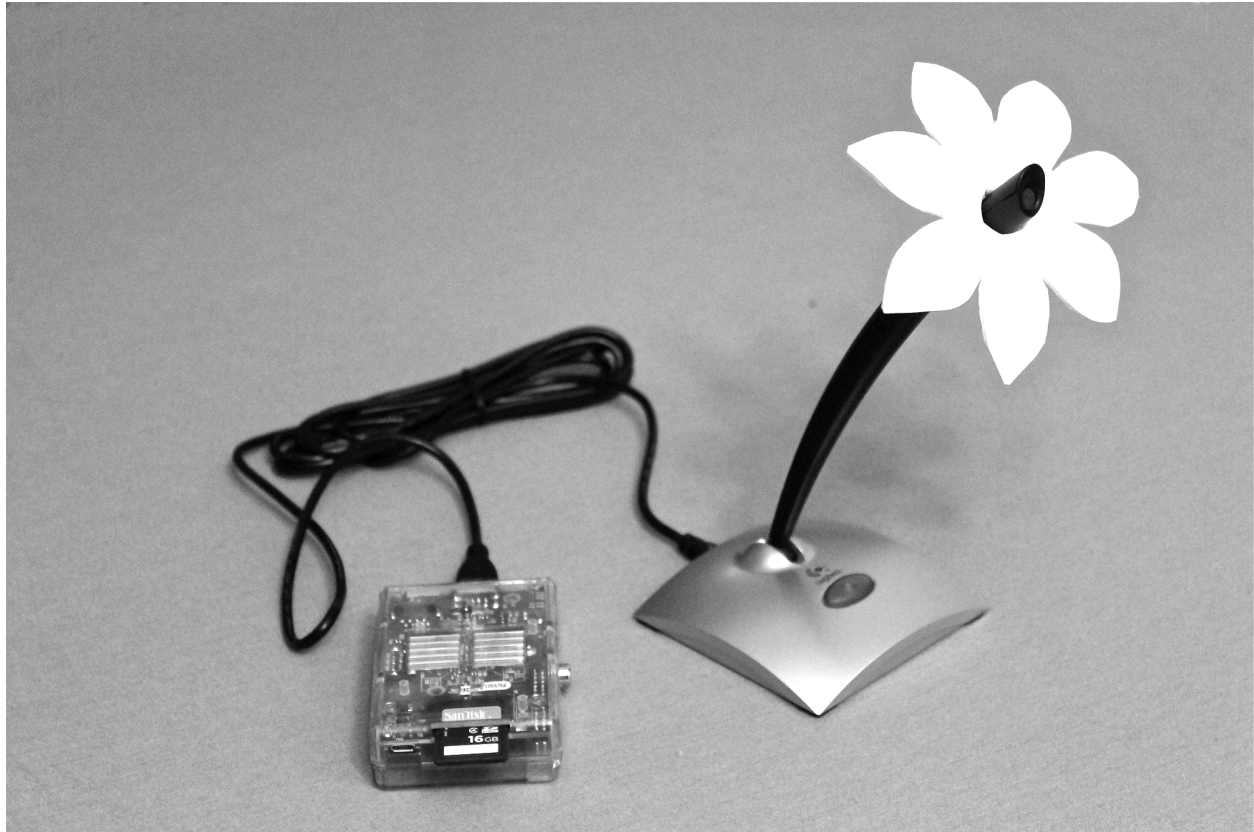


Figure 4.3: Our unobtrusive audio capture and classification device uses a USB desktop microphone with a Beaglebone.

4.3.2 Preprocessing

Noise plays a large part in the success of the classification and also for reverberation estimation. After capturing the audio, we perform normalization to remove the DC offset and to keep the maximum amplitude capped at -1.0dB . Preemphasis is applied to reduce the adverse effects of noise and attenuation. All environments have some level of baseline noise typically from the HVAC. Since most signals will capture non-speech (over 95% in our experience), we can build a noise model of the uniform noise in the room and use spectral subtraction. Our system used a Wiener noise suppressor with two-step noise reduction (TSNR) technique [79]. Their approach uses harmonic regeneration noise reduction (HRNR) to refine the SNR a priori to compute a spectral gain to preserve speech harmonics. More sophisticated machine learning-based noise subtractors can extend this approach.

4.3.3 Segmentation

The next step, segmentation, obtains discrete chunks of speech for processing. There are many voice activity detectors (VADs), silence detectors, and turn-taking options in the literature [80, 81, 82, 83]. We used a combination of volume, spectral energy, fundamental frequency (F_0), and spectral flatness for creating a predictor for speech segments. The spectral flatness can be used for characterizing an audio spectrum for how tone-like a sound is, and hence can eliminate signals with a large mixture of sources (such as multiple people talking at once, or music or TV in background).

4.3.4 Feature Extraction

We ported the OpenSMILE library (already written in C++) to be compiled on the ARM7 platform (with NEON optimizations). Doing classification on-node decreases network traffic, but also improves privacy concerns about transferring conversation data through the air in which eavesdroppers can intercept, or to the Cloud where other unwanted parties could access the information. We configured the feature extractor to extract a total of 384 functional features, the min, max, mean, stdev of each of the 16 low-level features. The device sends the classification result encrypted over WiFi to the base station. Because frame level features are not sent, reconstructing the speech content using automatic speech recognition would be very hard if not impossible to achieve.

For each segment, we extract the acoustic features described in the Interspeech 2009 Emotion Challenge [84]. By aggregating a series of low level descriptors (such as pitch) recorded at each instance, we compute general statistics over the duration of the utterance, resulting in a total set of 384 features. The OpenSMILE audio feature extractor [85] was used for extracting the features. First, the signal is framed into 20ms chunks using a sliding window of 10ms. A Hamming window is applied to each frame before the fast fourier transform (FFT) is taken. The mel-frequency cepstral coefficients (MFCC) are derived by mapping the powers of the spectrum to the mel scale, taking the logs of the powers at each mel frequency, then finally taking the discrete cosine transform of those log powers. The result of the FFT is also passed to a autocorrelation processor in order to

estimate the fundamental frequency (F_0) from the relationship of its harmonic frequencies. The root-mean-square (RMS) energy and the zero crossing rate (ZCR) are also extracted.

We smooth the values of the features into speech contours by using a moving average filter of three window lengths, which minimizes any pops or any abrupt fluctuations in the signal. For each of these contours, various statistical functionals are computed including the maximum, minimum, range, arithmetic mean, standard deviation, skewness, and kurtosis. Additionally, the contour is approximated by a linear approximation, resulting in a value for the slope and offset as well as the quadratic error this approximation with the actual contour.

4.3.5 Classification

We use a support vector machine (SVM) for classification because of the large feature size. The LIBSVM library is used for both the training and testing. Before fitting the model, all features are scaled to the range $[-1, +1]$ so that attributes in greater numeric ranges do not dominate those in smaller numeric ranges. The radial basis function (RBF) kernel then maps the samples onto a higher dimensional space. We configure the parameters of the RBF kernel, C and γ , using the grid-search method using cross-validation to find the best combination with the highest accuracy.

4.4 Evaluation

We evaluated RESONATE with three separate sets of experiments: First, we used an impulse response database on a dataset of emotional speech to investigate the effect of reverberation on both speaker identification and mood detection classification, and demonstrated how our system improves accuracy in the presence of reverberation. In the second category of experiments, we collected speech samples from four volunteers in homes and offices in a controlled manner, with a script and a predefined configuration of speaker positions. Finally, we conducted two case studies with the system running continuously for multiple weeks in real environments (one in a home, another in an office). For the controlled study and case study, we only evaluated the speaker identification because of the difficulty of assessing the mood of our speakers empirically.

Table 4.1: AIR database Room Characteristics

Room	Dimensions (m)	\overline{RT}_{60} (s)
Office room	5 x 6.4 x 2.9	0.43
Meeting room	8 x 5 x 3.1	0.23
Lecture room	10.8m x 10.9m x 3.15	0.78

4.4.1 Public Data Set Evaluation

We investigated the effects of reverberation on both speaker identification and mood detection by artificially introducing reverberation by convolving empirically collected impulse responses (AIR dataset) with recorded speech segments from a popular emotional speech data set (EmoDb). There are limited emotion datasets that are freely available [86]. We selected EmoDb [87] because it contains large number of speakers, is freely available, and is widely accepted in the affective computing community. It contains a collection of utterances spoken by 10 different actors (5 male, 5 female) using a variety of emotions. The recordings include various short phrases taken in a non-reverberant (anechoic) chamber.

The empirical set of RIRs came from the Aachen Impulse Response (AIR) database [78]. A summary room types of the collected RIRs are shown in Table 4.1. For each room type, there were 5 different RIRs corresponding to 5 different speaker positions.

We considered four different scenarios for this evaluation: for the baseline, we assumed that in the real scenario we would have access to clean recordings of the person’s speech to properly train our classifiers on, and that we would have a clean sample to test against such as gathered from a worn microphone. This is a “best case scenario” for the ability to properly classify the user and her mood. The second case introduced reverberation, but no correction was applied i.e., we trained with the clean samples, and tested with the reverberated samples created by simulation. In the third case, RESONATE₁, we used simulation to form a better training set and assumed that we know the dimensions of the room. The final case, RESONATE₂, assumes that along with the dimensions we could determine also the position of the speaker in the real room when speaking.

We evaluated the accuracy of the speaker identification and mood detection classifier under

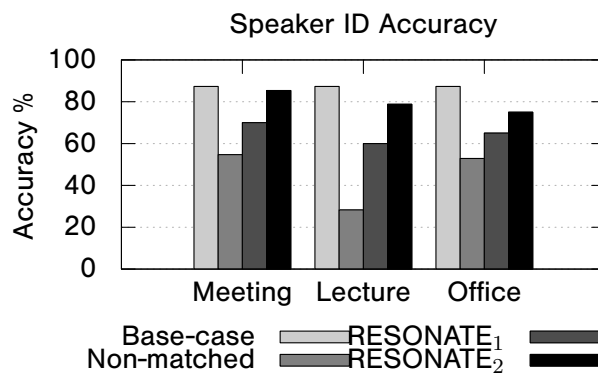


Figure 4.4: Speaker identification classifier performance using the EmoDB corpus. The effects of reverberation are most notable in larger rooms.

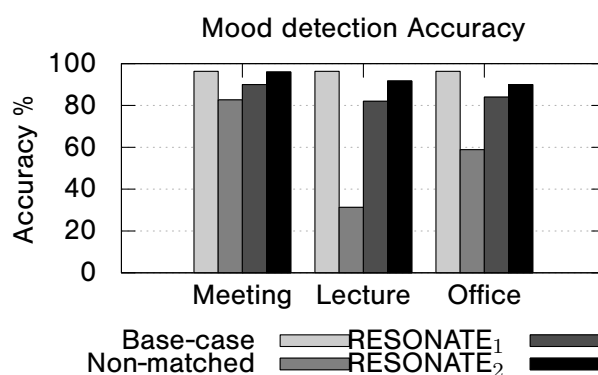


Figure 4.5: The results of the mood detection classification using the EmoDB corpus. Larger rooms are most impacted from reverberation, but RESONATE can improve the accuracy close to baseline.

these four different scenarios. We used 10-fold cross-validation on the training and testing set for each scenario. The results are shown in Figures 4.4 and 4.5. What we found was that the speaker identification accuracy varied considerably depending upon the room, however the RESONATE approach consistently gave better results, often near the baseline.

In addition to speaker identification and mood detection accuracy, we also evaluated the effects of two important parameters in the system. The first is the estimation error of \overline{RT}_{60} , since our training system must estimate this parameter from the training samples. The second is the effect of the utterance length on classification accuracy since this can vary. We present these evaluations next.

Feature Selection

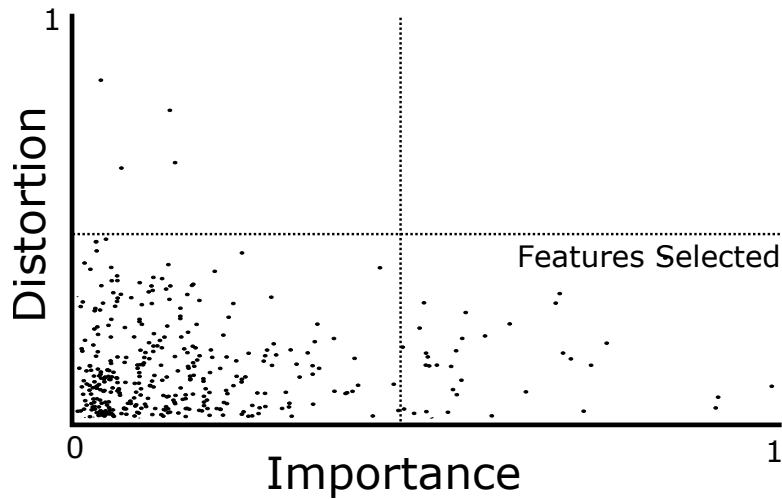


Figure 4.6: The 384 features extracted are sorted by their importance for identifying the speaker and the amount of distortion occurring from the effects of reverberation.

Not all of the 384 features are distorted to the same degree by the effects of reverberation and noise. Also, not all of them are useful for the classification task. To evaluate which features are most important for classification, we used an algorithm by Chen et al [88] that uses F scores to compute the importance of the feature for correct classification. Our goal is to choose features that maximize the number of important features and minimize the number of features prone to distortion.

We plot the importance of feature (from its f-statistic) and the normalized level of distortion in Figure 4.6. The features in the bottom-right portion of the graph should be selected because they exhibit low distortion, but high importance. In general, among the highest importance features are those related to the MFCC, particularly the higher band frequency (in the 11th band). However when reverberation is introduced, those MFCC suffer from the largest distortions. We discovered that the set of features related to PCM and F_0 to offer a balance between high importance and low distortion.

Next, we evaluated whether choosing a smaller subset of the original 384 features could offer better performance than the list as a whole. In Figure 4.7 we show how selecting a smaller set of features that have low distortion, but high classification importance improves accuracy. For the

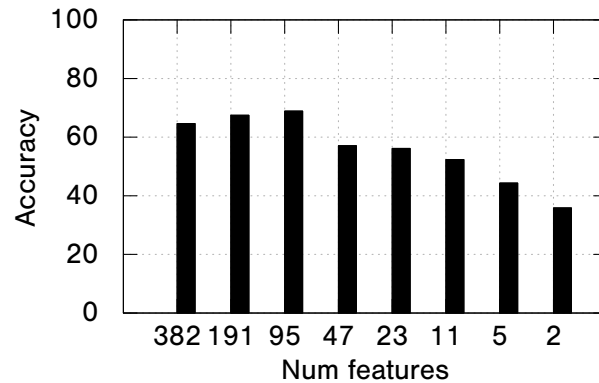


Figure 4.7: Using the f-statistic criteria we decreased the number of features used for classification. The peak accuracy was 68.86% with 95 features.

speaker identification task, we saw a maximum accuracy of 68.86% when 95 features were chosen versus 63% accuracy when all 384 were used. It is important to note that these accuracy numbers are from uncontrolled long-term collection in real environments, which would explain the low accuracy numbers.

4.4.2 Benchmarking

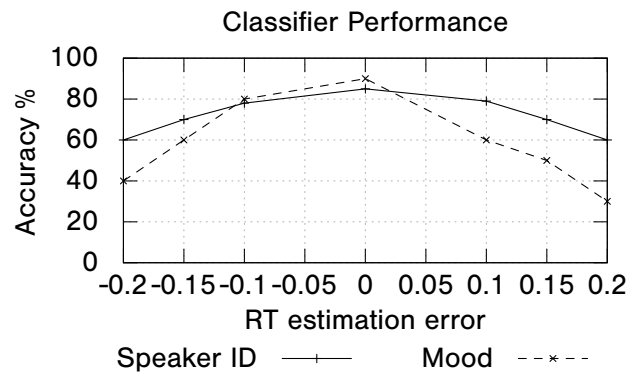
Model building is a computationally intensive task and the Beaglebones take several minutes to complete feature extraction, reverberation simulation, and SVM model fitting on the node, however the base station (multi-core machine with several gigabytes of memory) completes this task in a few seconds as shown in Table 4.2. For these benchmarks, we recorded the time the processing thread spends inside of the user-level of the OS. The Beaglebone has a Cortex A8 ARM processor, and all of our C++ code was compiled for the architecture using aggressive optimizations and the NEON extensions. The results show that realtime classification and feature extraction can be done on-node, however the corpus training and reverberation simulation should be done on a more powerful platform such as a base station (or web service).

Reverb Estimation and Accuracy

In practice, the \overline{RT}_{60} parameter must be estimated by having no additional information other than the signal itself. The algorithm for blind estimation maximum likelihood estimates the \overline{RT}_{60} , but

Table 4.2: Classification Benchmarks

Task	Beaglebone	Base station
Feature extract (clip)	2.51 sec	0.10 sec
Feature extract (corpus)	18.25 min	20.84 sec
SVM Training	4.88 sec	0.25 sec
Classification	5 ms	0.3 ms
Fast Conv. (corpus)	17 min	5 sec
Sim. building (room)	4.92 s	0.40s

Figure 4.8: \overline{RT}_{60} estimation error effect on speaker ID performance.

will result in some errors – especially when there is noise in the signal. Therefore, in this section, we evaluate how a poor estimation of \overline{RT}_{60} effects overall accuracy. Because the Aachen Impulse Response database reports the ground truth \overline{RT}_{60} value, we evaluate the accuracy of the speaker identification task as a function of error. We vary the error in milliseconds in steps of 0.05ms. The evaluation was done using the EmoDB speech samples and the results are shown in Figure 4.8. The results show that if there is zero or very small error in \overline{RT}_{60} , then accuracy is above the 80% level. If the \overline{RT}_{60} error is large, e.g. 0.2, then speaker ID accuracy drops to about 60% and mood accuracy drops to about 30%. In the EmoDb data set evaluation, because of the quality of the original recording, rarely did the error exceed 0.05ms. The estimation error our system noticed were within ± 0.1 s in the living rooms, but in the lecture hall the error was over 0.2s.

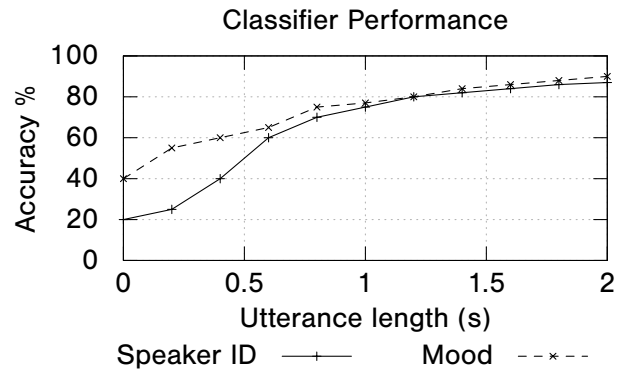


Figure 4.9: Relationship of the length and the speaker identification accuracy.

Length of segment and Accuracy

While collecting in-home and in-office audio data for many weeks, we observed that the speaking segments vary greatly in duration. Because the classification works by extracting the statistics of the features across frames for the entire utterance length, a large utterance size will increase the accuracy of the classification. We again used the EmoDb corpus, but varied the segment size and observed the classifier accuracy. In Figure 4.9 we see that if the utterance length is above 2 secs we obtain well over 80% accuracy for the classifier, while utterances under 0.5 secs are in an unacceptable 20-40% range.

4.4.3 Controlled Testing in Real Environments

The controlled experiments above show the potential for the RESONATE method for producing favorable accuracy, and in this section we demonstrate how well it performs when collecting audio from our system's microphones in actual environments. We selected a variety of rooms based on where conversation typically occurs, 2 living rooms, 1 office, 1 conference room, and 2 kitchens. The basic geometry and features of the rooms used in the experiments are shown in Table 4.3. It is important to note that these rooms were furnished with sofas, desks, curtains, and other objects that affect the acoustics of the space. Our test rooms also had typical noise sources such as the hum of the refrigerator and air conditioning system, which could be consistent or intermittent throughout the recordings. Additionally, none of these rooms were precisely cuboid in geometry, and often had

Table 4.3: Experiment Room Characteristics

Type	Dimensions (m)	Floor
Living room	3.9x4.3x3.05	Wood & Rug
Kitchen	2.3x4.3x3.05	Linoleum
Living room	3.7x4.7x2.43	Wood
Kitchen	3.7x4.4x2.43	Linoleum
Meeting	10.0x6.9x2.74	Carpet
Office	15.0x10x2.94	Linoleum

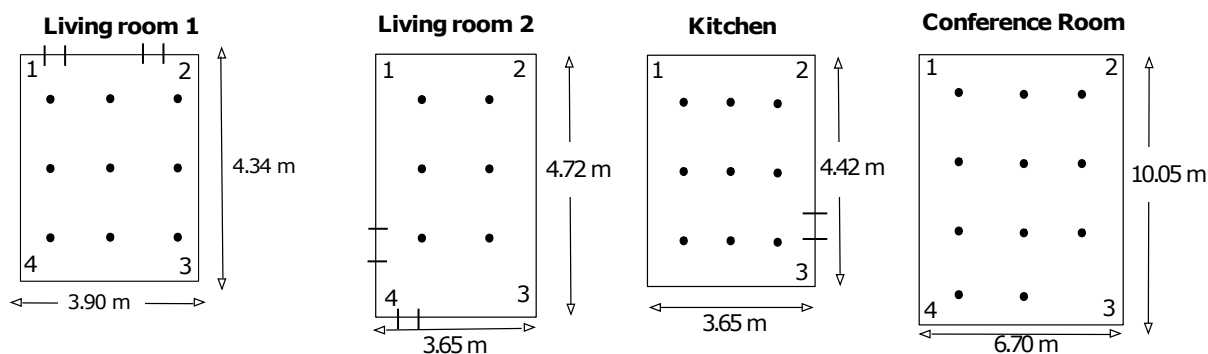


Figure 4.10: Our approach was evaluated in 2 living rooms, a kitchen, and an office for the controlled experiments.

open doorways and openings to other rooms, as most real environments do. One of the significant results of this section is that modeling these rooms as simple cuboids in our simulation, despite their small geometric aberrations, was quite successful for accurate speech analysis.

We recruited four volunteers (2 male, 2 female) and recorded them in each of the rooms. We placed four microphones in each of the corners of the room. Additionally, the speaker carried a hand-held microphone in order to simultaneously capture the signal with minimal reverberant effects. This signal was used as the ‘clean’ sample for base case training and testing, as well as later for the signal on which simulated RIRs would be applied. We divided the room into a grid (similar to the method described in producing simulated sampling), and at each point, the speaker spoke facing the four ordinal orientations (approximately north, south, east, and west). Speakers read the same three-sentence passage from a book to ensure consistency in our experiment. The speakers were instructed to remain in a neutral speaking tone. We did not evaluate the case of a moving speaker in this experiment.

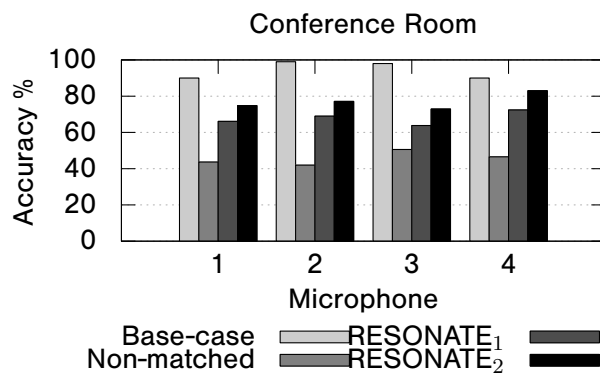


Figure 4.11: Conference Room

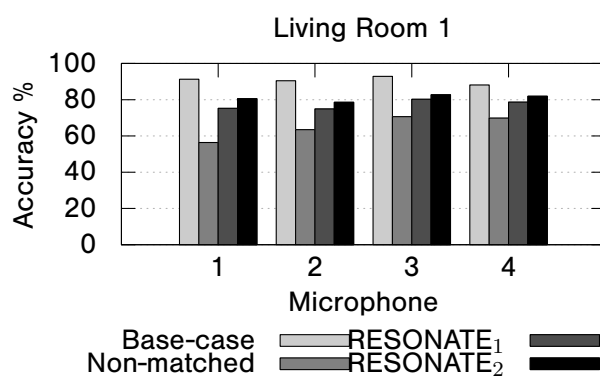


Figure 4.12: Living Room 1

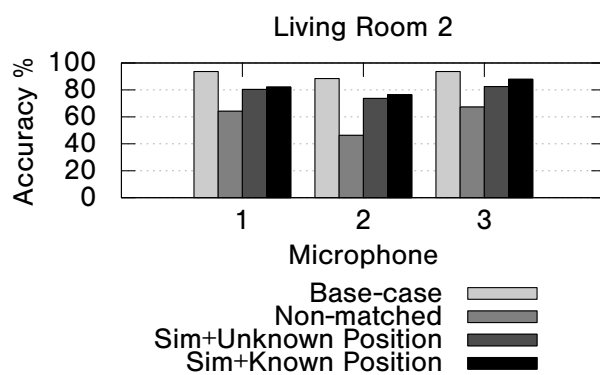


Figure 4.13: Living Room 2

Here, in the ‘Baseline’ case, we trained and tested with the samples recorded by a particular microphone at any corner. In the ‘Non-matched’ case we trained with the ‘clean’ samples recorded by the handheld microphone, and tested with the samples recorded by a particular microphone at any corner. In the RESONATE₁ case (simulation without known location), we converted the

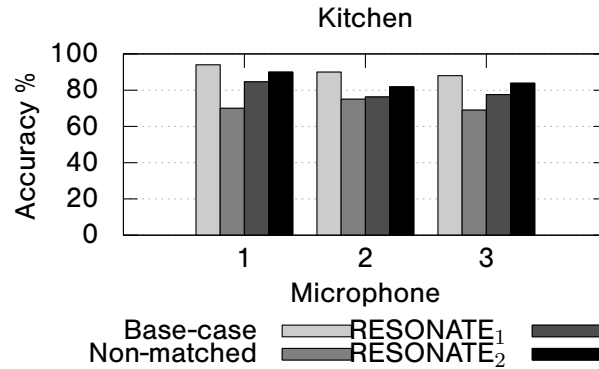


Figure 4.14: Kitchen 1

‘clean’ recordings of the speaker’s microphone to a reverberated version, assuming the receiver is a corner microphone and the speaker can be in any of grid positions of the room; then we tested on the recordings of the corner microphones. Finally, in the RESONATE₂ case (simulation with known location), we used the location of the speaker in the room and generated a reverberated signal accordingly from the ‘clean’ signal from the speaker’s microphone. In all four cases, we first removed the ambient noise from the recordings by all microphones using a noise removal algorithm [79] before processing. We applied 10-fold cross-validation as before, and again only evaluated speaker identification accuracy. The results are presented in Figures 4.11, 4.12, 4.13 and 4.14.

The accuracy for the ‘Baseline’ scenario was typically around 90% in each room. However, this scenario requires training for each user in each room where the system will be deployed. The ‘Non-matched’ scenario (where nothing was done to compensate for reverberation) shows results below 70% and as low as 40%. This shows the dramatic impact of reverberation on speaker identification accuracy. In the RESONATE₂ scenario (simulation with known location), the classification accuracy rises to within 5% of the base case for most of the tests, except for the conference room. In addition to having an unusual variety of sound-reflective surfaces, the conference room had considerable HVAC noise in the background. This noise was difficult to subtract using our Wiener filter and when reverberation was applied, the noise was amplified. This is another example why noise subtraction is a fundamental step to this strategy. As we do not know the location of the speaker in the RESONATE₁ scenario, the speaker identification accuracy drops. However, it is still signifi-

Table 4.4: Training Time

Case	Training Time	Accuracy
Base Case	45 min	90%
Do Nothing	1.5 min	60%
RESONATE ₁	1.5 min	80%
RESONATE ₂	1.5 min	85%

cantly better than the non-matched scenario. Our results here show that our techniques to address reverberation significantly improve speaker identification accuracy compared to the non-matched scenario by roughly 30%, and in most cases closely approach the baseline accuracy of training all of the speech in the environment that it will be tested in.

Training Time

Our results show that the baseline case still provided the greatest speaker identification accuracy. However, this came at the cost of a lengthy training period. For example, in one room, each speaker must occupy 34 total positions at 4 orientations each, totalling 136 recordings. At 20 seconds per recording, the minimum amount of time it would take to complete the training for 4 rooms would be approximately 45 minutes per person. Although this method provides good accuracy, this time investment is not always convenient especially considering multiple rooms and many speakers. In addition, there are also some situations where training in the real environment is not even possible. One such example is where the classifier has been trained from a preexisting corpus (such as EmoDB) that cannot be trained in an environment. RESONATE is able to solve this problem by giving close to baseline accuracy with minimal training time (1.5 minutes), without requiring access to the real environment for training.

4.4.4 Long-term Real Deployment Evaluation

We now test the system in a completely uncontrolled manner in the long term, by testing it in two real deployments: one in a living room in a home, and the other in our office space. The

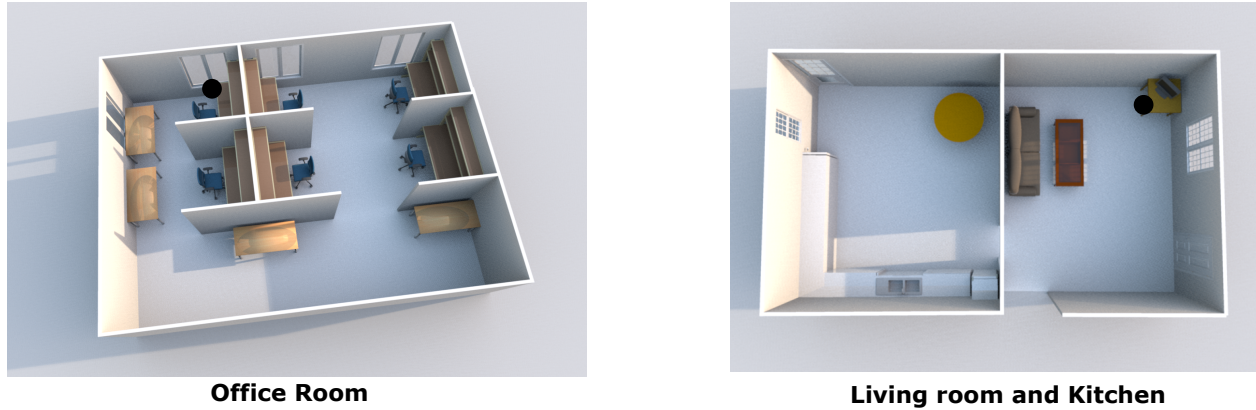


Figure 4.15: The two environments used for the uncontrolled long-term studies

Beaglebone system shown in Figure 4.3 was used, and was ideal for this purpose since it is compact, unobtrusive, and powered by a wall outlet. Although the system can do classification on-node, for post-experiment analysis of the data, we captured the signal and compressed it using the libVorbis codec at 44.1 kHz sampling rate and stored onto the 16 GB microSD card.

For the home, data was collected for 4 weeks and for the office for 6 weeks. The floor plan of the living room and office room is shown in Figures 4.15. In the living room was a large sofa, a TV, and an electric keyboard. The microphone was placed on the table next to the TV. Adjacent to the living room was a long hallway and an entrance to the kitchen. The office space was a large room (almost 10m x 10m) with cubicles down its center line, and the microphone was placed in the last cubicle next to the far wall.

Since our system does not have a robust signal selector, we selected only speech segments that were over 2 s long and did not process laughter or TV noises in the background. The signals often had pops, knocks, and clicks in the audio, there were also examples of typing and some appliances that were filtered out as well. We show a table of the types of sounds we came across other than voice in our listening stage in Table C.1. Voice vs. noise detection is outside of the scope of this chapter, however the literature uses many machine learning approaches similar to speaker identification and mood detection, and with similar feature sets and classifier types. The RESONATE approach might help augment those machine learning techniques as well.

Table 4.5: Sounds Encountered

Type	Examples
Physiological	Sneezing, nose blowing, sniffing, clearing throat, hiccup, eating, burp, humming, laughter, drinking
Objects	phone vibrating or ringing, typing, mouse wheel, unwrapping food, papers rustling, clothes rustling, television, piano, moving furniture, doors opening and closing, objects dropping or moving, footsteps, pouring liquid, coffee percolation, dishwasher, cleaning sounds
Ambient	truck backing up, siren, birds chirping, passing airplane, traffic, motorized tools (lawnmower, etc)

Challenges and Solutions

Real deployments offered a number of challenges that did not occur in the controlled recordings. For speech itself, many observations came to light: First, that real utterances are most often brief statements averaging 1 second long. This duration is insufficient for reasonable accuracy from our classifier. However, since mood detection only requires a small number of longer speech samples over the entire day, eliminating short samples might not affect overall mood detection. Second, the speech of multiple people is often mixed and overlapping. If these instances of speech are not separated by silence, the system cannot detect that they are separate utterances by different people. Third, in a real environment, people do not speak in the same consistent manner as they do when creating their training set. The occupants of the home in particular often took on different affects, and raised their voice into a higher register when talking to their cat. People are also prone to making many vocal noises that could be confused for speech, such as laughter and coughing. These issues have a negative effect on the system's overall accuracy. Another problem was that the microphone in the living room was able to pick up sounds and speech signals from adjacent rooms like the kitchen and the hallway. However, when using the networked configuration of Beaglebone devices, one in each room, only the cleanest signal is used for analysis. Advancements made in the area of blind source separation (BSS) can be used to separate mixed sources in a signal because the received signal is a linear mixture of statistically independent sources. However, to date, BSS tends

to not produce good results in reverberant environments.

Speaker ID Performance

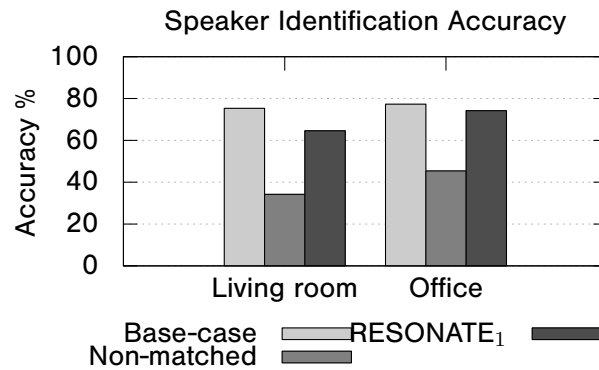


Figure 4.16: Speaker ID classification for the long-term 2-month studies in the home and the office.

For establishing ground truth for the speaker identification, an occupant living in the home and also working in the office listened to each audio clip and assigned a speaker label. The classifier was trained with three people who work in the office, and the two people who live in the home. In Figure 4.16, we show the performance of the classifier in the home and the office. Since the speakers' location was not tracked in this experiment, we did not use distance information as in our previous evaluations for RESONATE₂. The accuracy was roughly 75% for classifying among four speakers in the best case. It is important to consider that the accuracy even for state-of-the-art approaches for speaker identification is poor in the case for an unconstrained freeform speech. However, we demonstrate that RESONATE is able to resolve almost all the problems resulting from reverberation to match within 5-10% of the baseline. The accuracy of speaker ID in the controlled setting was significantly higher than in the long-term deployment because there was more consistency in the input to the classifier. As discussed, there are two main sources of variation: the first, which our system helps to overcome, is the reverberation and sounds from the environment. The second, however, is the variety of different ways that speakers talk in a real environment, in comparison to the consistent tone and content used during a training session. In our controlled experiments, the scripted content and tone during the training and testing cases were identical;

however, if testing had been performed instead on spontaneous speech, it can be projected that the accuracy would have been much lower.

4.5 Conclusions

We present and thoroughly evaluate a novel system called RESONATE, which combines a matched condition training approach with a unique reverberation impulse response simulator. This system allows a single training corpus to be adapted for various environments, minimizing necessary training and configuration time. We have practically demonstrated how it mitigates the negative effects of reverberation in real home or office environments for speech classification applications such as speaker identification and mood detection. Our results show that reverberation has a significantly negative effect on the performance of these applications in real environments, and we also show how our approach improves performance considerably in the presence of reverberation using only very basic room information. RESONATE minimizes training efforts for users using a shared large corpus of voices and then creates a tailored training set by generating reverberated samples of their voice considering different room acoustics, based solely on simple room acoustic models. We believe that this solution is extensible and in the future can be used in conjunction with other machine learning strategies such as multiple classifier models, improved feature sets, improved noise elimination, and blind source separation (BSS). We have evaluated RESONATE using public data sets, collecting voice samples from volunteers in different rooms in homes and offices in controlled settings, and finally by deploying our system for two long-term studies.

5 | Improving Sleep Monitoring Options

Among the most important elements of tracking a person's wellbeing is their sleep patterns. In numerous clinical studies, depression has been shown to frequently disrupt sleep patterns. For instance, depressed individuals show difficulty falling asleep or staying asleep and will also wake up early. These patterns are present among 80% of people with major depression [89]. The prevalence of sleeping disorders among the depressed shows the importance of monitoring sleep. Problems with sleep could also be related to stress, disease, or sleep apnea. In this chapter, we present solutions for sleep monitoring that are cheaper and more easily deployable than current systems.

The contributions provided in this chapter are:

1. We present a novel system using an RFID-based accelerometer called the *WISP* which allows battery-less operation and be placed directly on the mattress without wires.
2. In Section 5.2, we present another accelerometer-based solution with motes to be used in most of our studies for its price and versatility.
3. We compare various types of sleep systems with our systems showing the tradeoffs between privacy, accuracy, cost, and comfort.

5.1 WISP Sleep Sensor

RFIDs are an important technology for many application areas in particular for product inventory tracking. A new technology introducing sensors to RFIDs, called WISP tags, makes many new applications possible. One important area for WISP's use is for smart homes. There are potential applications for WISP tags in application areas such as saving energy or continuous health monitoring.

Wireless and batteryless sensors are now possible with this technology making the sensing component in the homes more transparent and can be run for longer periods of time before maintenance. This chapter explores the use of the WISP sensor on bed mattresses in order to determine the quantity and quality of sleep.

Many sleep monitoring systems have been developed and they attempt to recognize sleeping disorders by providing healthcare providers with quantitative data about irregularity in sleeping periods and durations or the amount of agitation and restlessness experienced during the night. These solutions vary in cost, comfort and accuracy. Our WISP system does not require any specific action from patients. In this system, we attach several WISP tags to the bed mattress and collect accelerometer data. Using the data we can infer body positions, movements, and entries and exits from the bed. We compare the performance of our system with several baseline systems including using pressure pads, video recording, a popular iPhone based sleep monitoring application, and the Zeo™ headband.

5.1.1 Design

An picture of the WISP tag is shown in Figure 5.1. The device's antenna and power harvesting circuitry enable off the shelf EPC Gen 2 RFID readers, shown in Figure 5.2 to power the nodes enough for a sample to be taken and returned by backscattering to the transceiver. To an RFID reader, a WISP appears as a normal RFID tag, however the tag is equipped with a 16-bit fully programmable ultra low-power microcontroller. The microcontroller can sample a variety of sensing

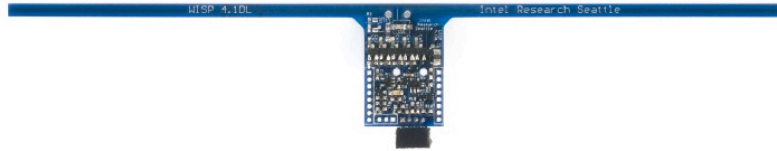


Figure 5.1: The WISP is an RFID-based mote that can report acceleration values



Figure 5.2: SpeedWay RFID reader

devices including triaxis accelerometers, lights, and temperature sensors. In our system, we only use the accelerometer capability. The WISP tags send these readings by encoding the values in the bitstring for the RFID's identifier.

For each bed we monitor, we attach three WISP tags along the edge of the mattress as shown in Figure 5.4. An example of the accelerometer readings, in shown in Figure 5.3: here the y-axis reading is shown as someone enters the bed. This raw data are used to differentiate between when the bed is empty, someone is lying on it, or someone is just sitting on the bed watching television or reading. When the bed is empty, the y-axis accelerometer of the tag is aligned perpendicular with respect to gravity, but when someone lays on the bed, because of the impact of the body on the mattress, the orientation changes. These orientations are different from the one when someone is just sitting on the bed.

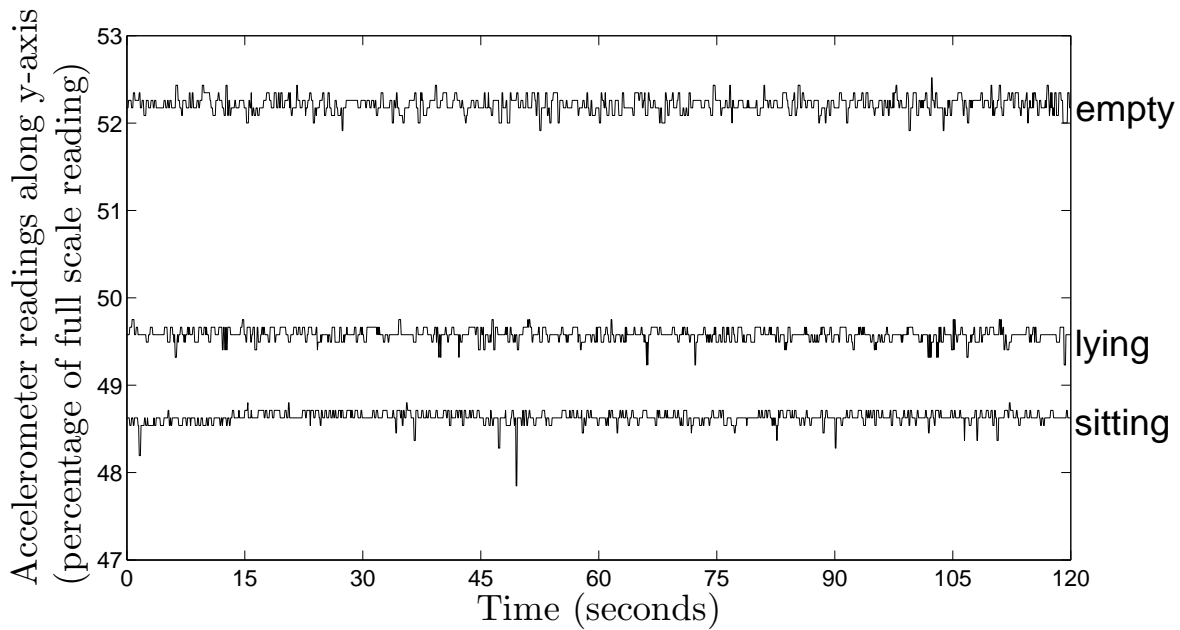


Figure 5.3: Accelerometer Reading for Empty, Lying, and Sitting

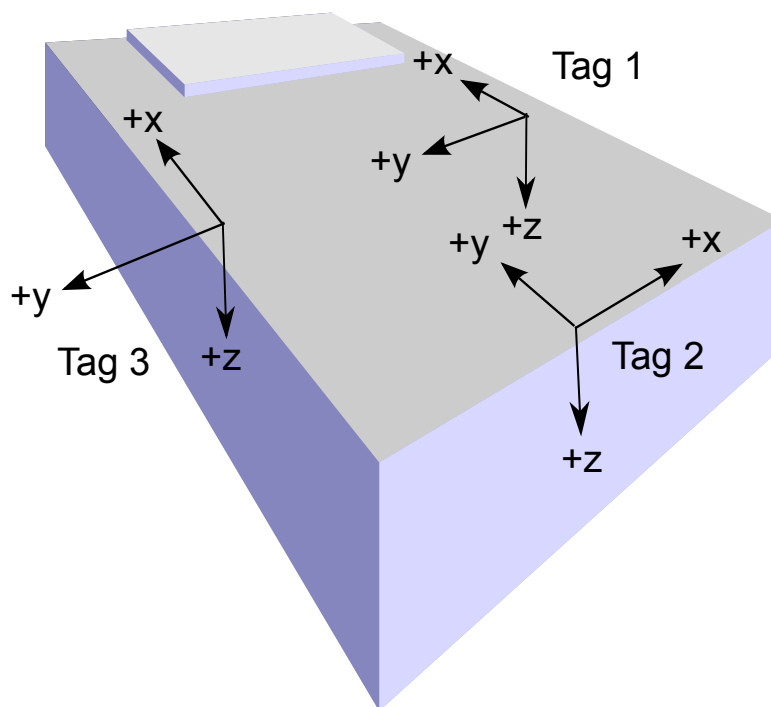


Figure 5.4: WISPs are attached to the corners of the bed.

Using the accelerometer readings, we distinguish among four positions: lying on the back, stomach, left, or right sides (shown in Figure 5.5) In the Figure 5.10 as an example, we show the accelerometer readings along the y-axis. Note that the readings along the z-axis (which is parallel to

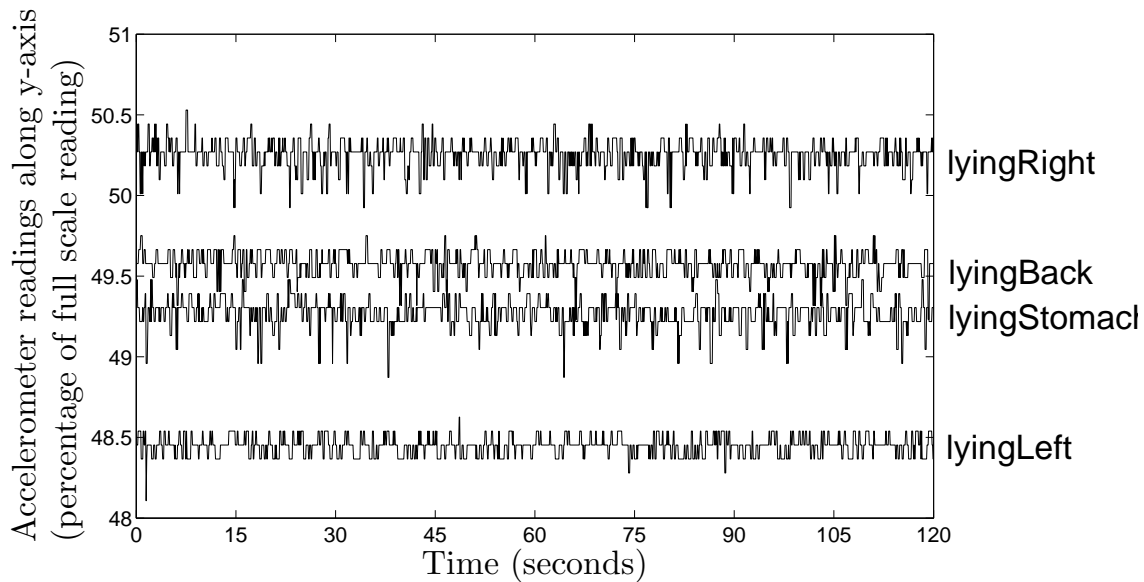


Figure 5.5: Accelerometer Reading Variation for Different Lying Positions

gravity) show similar variation. The readings along the x-axis do not show too much variation, but if we combine them with the readings along the y and z-axes, together they accurately differentiate among the four positions. The evaluation sections demonstrate the accuracy of this technique.

Our system can also detect the amount of movement to discover periods of restlessness and agitation. Each time someone moves on the bed, the accelerometer readings change rapidly. Using the standard deviation in these readings we can determine the amount of movement in time windows. The system monitors how many times a person tosses and turns during the course of the night and how many times the person leaves the bed. If someone is lying on the bed and does not move for a significant amount of time, then we assume the person is asleep. Frequency of movements is also different for different sleep stages and thus can be related to which sleep stage a person is currently in [90]. Transitions between different sleep stages also correspond to change in frequency of body movements. Thus, based on a summary of movements made during each night, doctors can infer duration, quality of sleep and irregular sleeping patterns.

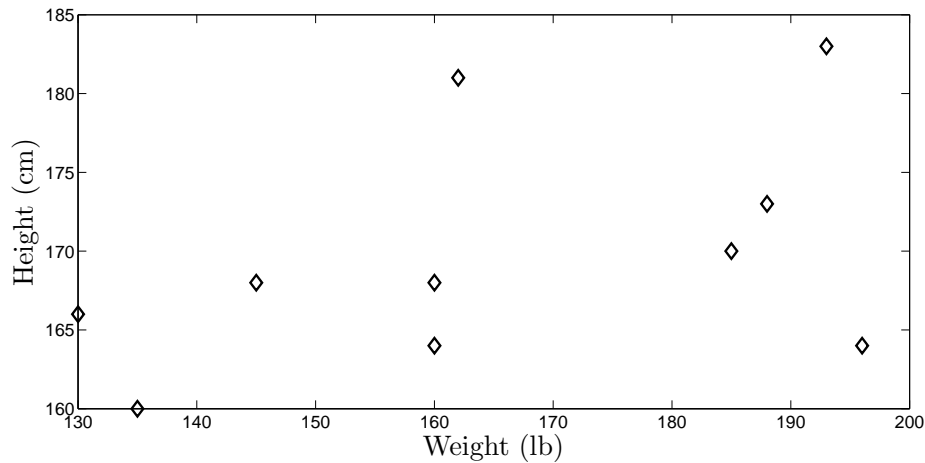


Figure 5.6: The volunteers for our controlled experiments had diverse heights and weights.

5.1.2 Controlled Experiments

Our evaluation of using WISP for sleep monitoring has three parts: controlled experiments for body position in Section 5.1.2 and for movement (Section 5.1.3) as well as real overnight experiments (the following section).

For our controlled experiments, 10 graduate students volunteered as subjects. The subject population was diverse in height and weight (shown in Figure 5.6) All participants were volunteers and were informed of the experimental procedures and the study's goals prior to participation. We conducted our experiments in three different beds to consider how different mattresses affect the measurements. Five subjects were evaluated on a twin-size bed in AlarmNET's medical testbed [63]. Five other subjects participated in each of the other two beds that were in a graduate student's apartment. All three of them were twin mattresses. So, for each bed, five subjects participated in the experiments. For two of the three beds the participating subjects were the same.

For each experiment, we attached 3 WISP tags to the mattress of a bed. Again, in Figure 5.4 we show the bed along with the positions of the tags. Note that we also investigated the use of a fourth tag near the head. However, this tag did not improve the accuracy and so we eliminated it. We placed the tags in such a way that when someone lies on the bed, there is one tag on each side of his body and one tag near the legs. We used two antenna for reading from the tags. The

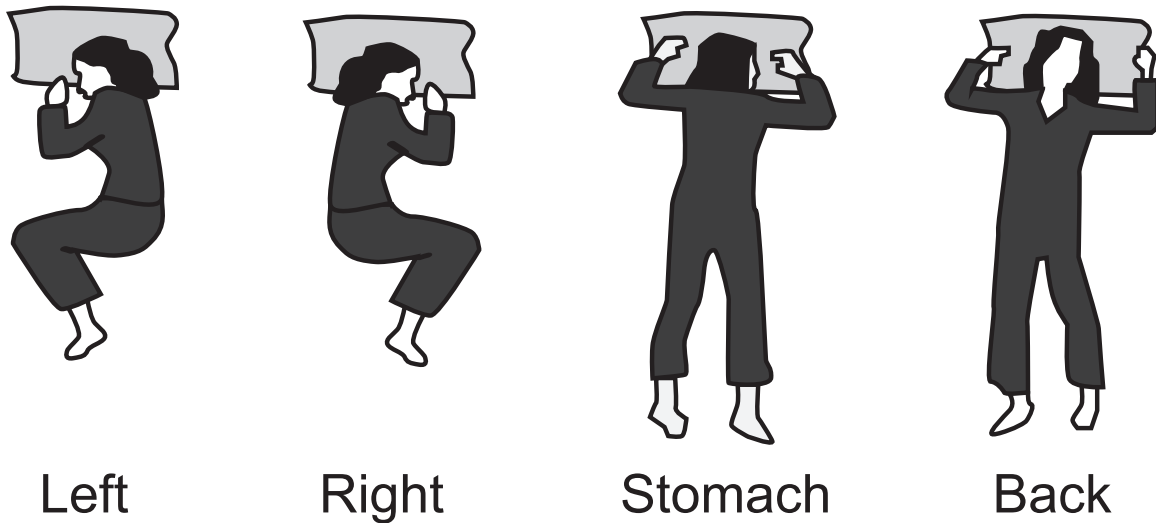


Figure 5.7: Accelerometer Reading Variation for Different Lying Positions

reader sends 10 read requests per second. The read rate from each of the tags were 4-7 reads per second during all our experiments. When we use more than one antenna, the total transmission rate drops. Another disadvantage of using the WISP tags is that they need to be placed within 1-2 meters of the antenna of the reader. To meet our range requirements and not to obstruct the residents' movement, we placed the antenna below the bed. The antennas were wired to the reader and the the device was connected to a laptop through a USB cable.

In the experiment, each subject lay on the bed in the following four positions: on the back, on the stomach, on the left side and on the right side. These four positions are shown in Figure 5.7. Each subject also sat on the bed with his or her back on the wall and faced towards the camera shown in Figure 5.4. This position modelled the position when someone lies in bed while watching television or reading a book while sitting in bed. For each position, we recorded data for two minutes. For each WISP tag, we obtained the acceleration along all 3 axis. From the readings of all three tags, we get a 9-tuple feature vector. Note that, the three tags do not report their values synchronously. We combine the readings from the three tags within each second and construct each possible 9-tuple. We associate all the 9-tuples collected during these two minutes to that particular body position. We also recorded the readings from the tags when the bed was empty. We use the collected data to train our system.

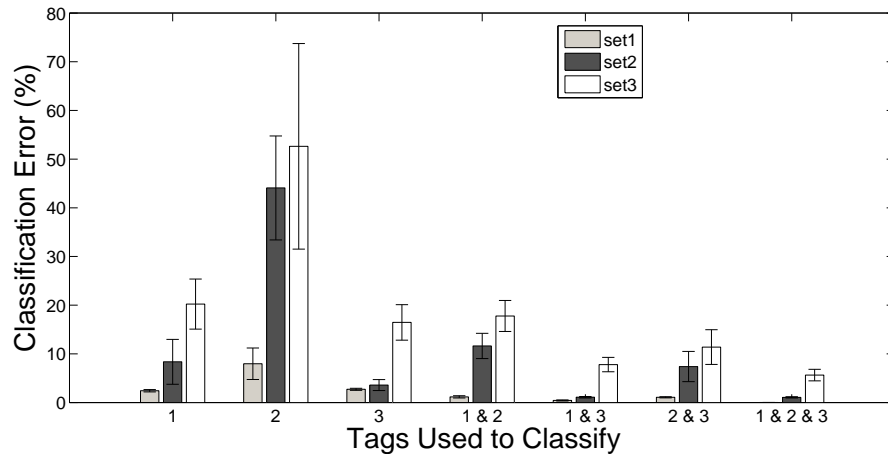


Figure 5.8: Average Classification Error for 5 Subjects for One of the Beds

After the training phase, the subject repeats the tasks again and we record data for 30 seconds for each position. Our system then classifies the new data based on previous training. For training and classification, we use the open source software, Orange Canvas, [91] which supports a number of classifiers. We decided to use the Naive Bayesian Classifier. Note that for each subject, first we train our system based on the subject's training data and then classify his or her remaining data.

5.1.3 Controlled Experiments - Body position

For each subject, we classify the collected data under three different settings. In the first setting, we test whether it is possible to differentiate between the bed being empty and someone lying on it (in any position). So we label all data collected during a subject lying on the bed in four different positions as lying. We do not include the data when the subject was sitting on the bed. In the second setting, we include the data for sitting and test whether it is possible to differentiate among the bed being empty, or someone lying on it (in any position), or someone sitting on it. In the last setting, we test whether it is possible to differentiate among all six cases: empty, lying on back, lying on stomach, lying on back, lying on left side, lying on right side and sitting. We name the above three cases as set1, set2 and set3.

For each setting, first we train and classify based on the data collected from one tag only (tag no. 1 or 2 or 3 of Figure 5.4). Then we use data from a combination of two of the three tags. Finally, we

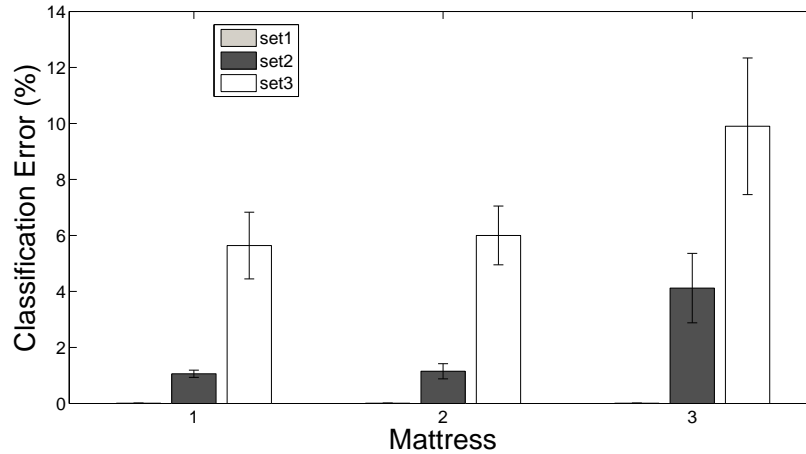


Figure 5.9: Average Classification Error for All Mattresses

use data from all three tags. Our goal is to test how increasing the number of tags helps in reducing classification error. The results of our experiments on one of the three beds are summarized in Figure 5.8. Five of the 10 subjects participated in the experiments on this bed. For each case, the y-axis shows the average of percentage classification errors for all five subjects. The error bars represent the standard deviations of the errors for each experiment. As we see from Figure 5.8, if we increase the number of tags, the classification error decreases. When we use data from only one tag, the performance of tag 2 is the worst. This is expected, because it is placed near the leg, and so it fails to capture enough of the variation of body impact on the middle portion of the mattress for different positions. When we use data from any two of the three tags, we see that the combination of tags 1 and 3 performs best. This is because both of them are placed in the middle parts of the two opposite edges of the mattress. When we use data from all three tags, the error for set1 becomes almost zero. For set2 and set3, average percentage errors are 1.06% and 5.64%, respectively. For the other two mattresses, we also observe similar trends, i.e., increasing the number of tags increases classification accuracy.

We also check how classification error varies over different mattresses. Figure 5.9 shows average classification error for all mattresses. Here we calculate the average over the classification errors for all subjects that participated in the experiments on a particular bed. As we see from the figure, classification error for set1 is almost zero for all mattresses. But for the other two sets, classification

error is greater for mattress 3 than the other two mattresses. This mattress is the one that is in our university testbed. The testbed quality is different than the other two. It is hard and inflexible. So, the impact of the body weight does not change the orientation of the WISP tags immediately. As mentioned earlier, we classify the body positions for 30 seconds of data for each subject. Later we used the data from the last 20 seconds and the classification error went down significantly and was approximately same as the other two mattresses. So for such mattresses, we need to classify the body position after the body settles in to a new position. One implication of these results is that we could build a new mattress with the correct flexibility and embedded WISP tags that is especially targeted for those wanting or needing sleep monitoring.

Now, we analyze what body positions are misclassified the most. Here, we consider misclassifications for set3 only. For every mattress, the case when the bed is empty is classified correctly. For the other positions, the average of misclassifications for each mattress is shown in Figure 5.9. Here, we see that classification error is most prominent for the two body positions where a subject lies on back and on stomach. The reason is that sometimes one of these is classified as the other. For both these positions, the impact of body weight on the mattress remains almost same. For the other three positions, the classification error remains less than 10% for each mattress.

Note that, for these controlled experiments, the training period is only two minutes for each body position. For practical use, we need to train the system for longer periods. During our realistic overnight experiments, we train our system for several nights (about seven hours per night) and then run the system. The results are much better and shown in the realistic overnight experiments.

5.1.4 Controlled Experiments - Movement Detection

As we see from Figures 5.10 and 5.11 when a subject lies on the bed in a particular position or when the bed remains empty, the accelerometer values returned by the WISP tags remain within a noise level of a particular value. This is true for acceleration values along each of the three axes. To find the maximum deviation in the readings, we calculate the derivative of all the readings when a subject remained in a particular position. The derivatives show that if the subjects remain in a

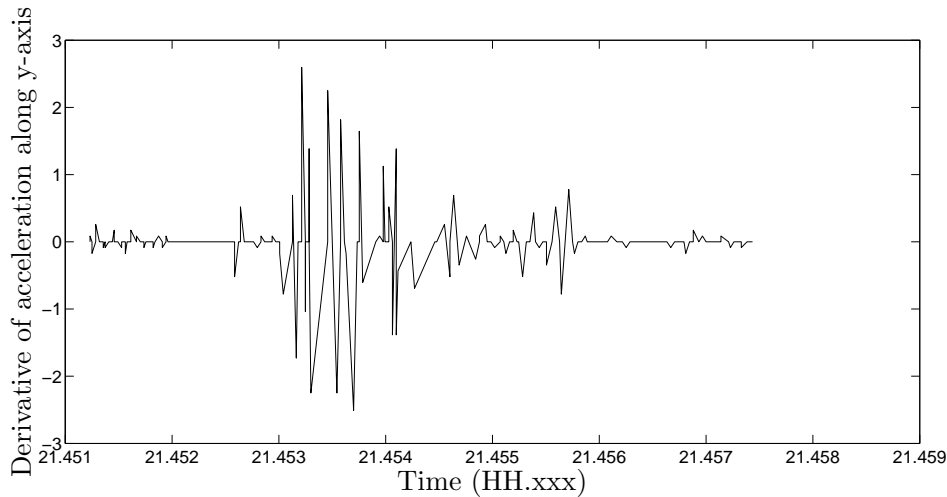


Figure 5.10: Accelerometer Reading along Y-axis During a Movement

particular position or if the bed is empty, the deviation remains in the interval $[+a, -b]$. The values of a and b vary for different tags, axes and mattresses, but remain same for different subjects. We calculate these values from the data collected during the controlled experiments of the previous section.

If the subject moves to a new position or makes significant movements while remaining in the same body position, the derivative of the accelerations of all three tags along both y -axis and z -axis become higher than the corresponding $+a$ or lower than the corresponding $-b$. So during the movements, the derivatives of y and z acceleration values cross the threshold values ($+a$ and $-b$) several times. Figure 5.10 shows y -axis accelerometer readings during such a move. Here the values of both a and b are 1.

Our algorithm to extract movement events from derivatives of y and z -axes accelerations of the three tags is as follows: For each axis of each tag, we record timestamps when the reported reading is outside the interval $[+a, -b]$. We consider each of these moments a possible movement. Note that the three tags do not report values synchronously. We calculate the total number of movements reported by the three tags within each two second time window. If the total number of movements within a time window is less than a predefined threshold, we consider those as discrete movements that do not affect sleep quality. We then cluster the other time windows, when a significant number of movements take place, using the *DB-SCAN clustering algorithm* [92] to compute discrete move-

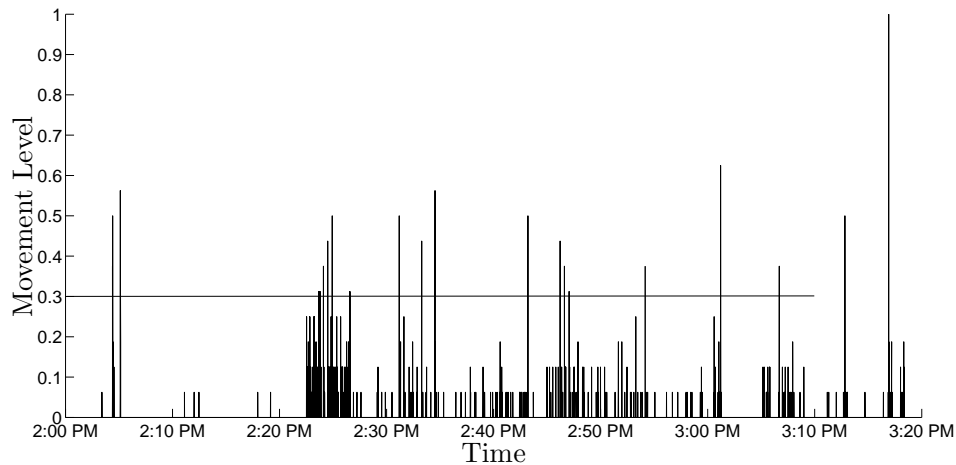


Figure 5.11: The number of movements in a 2s time window during a 70 Minute controlled experiment.

ment events. The clustering also ensures that discrete movement events that happen within a short amount of time are combined as a period of restlessness. For each cluster, we set the movement level as the maximum of movement levels of all the time-windows belonging to that cluster.

Figure 5.11 shows the number of movements for each 2 second time window during 70 minutes of a controlled experiment. During the experiment, the subject got on the bed, laid there for 70 minutes during which he made several movements and finally got off the bed. Some movements were from one body position to another and in some cases, the subject made significant movements while remaining in the same body position. We normalize the y-axis by dividing the number of movements for each time window by the maximum number of movements in any time window to get the movement level. We use 0.3 as the threshold to filter out the time windows where movement level is insignificant.

Figure 5.12 shows the discrete movement events as clustered by DB-SCAN. All the discrete movement events during the controlled experiment were successfully detected by our system. As we can see from Figure 5.12 some movement events span several minutes. During these movement events, the subject made a number of movements in quick succession. We comprehensively validate the performance of our movement detection algorithm by realistic overnight experiments that we present in the next section.

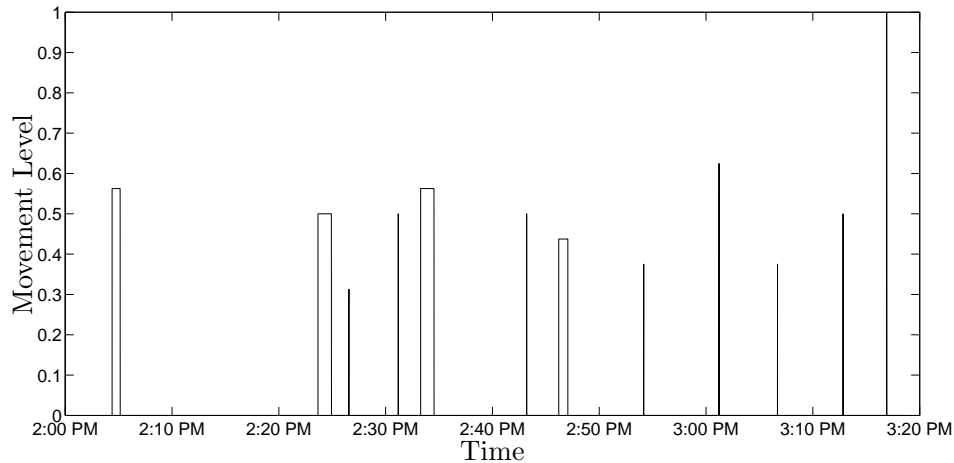


Figure 5.12: Discrete Movement Events During 70 Minutes of Controlled Experiment

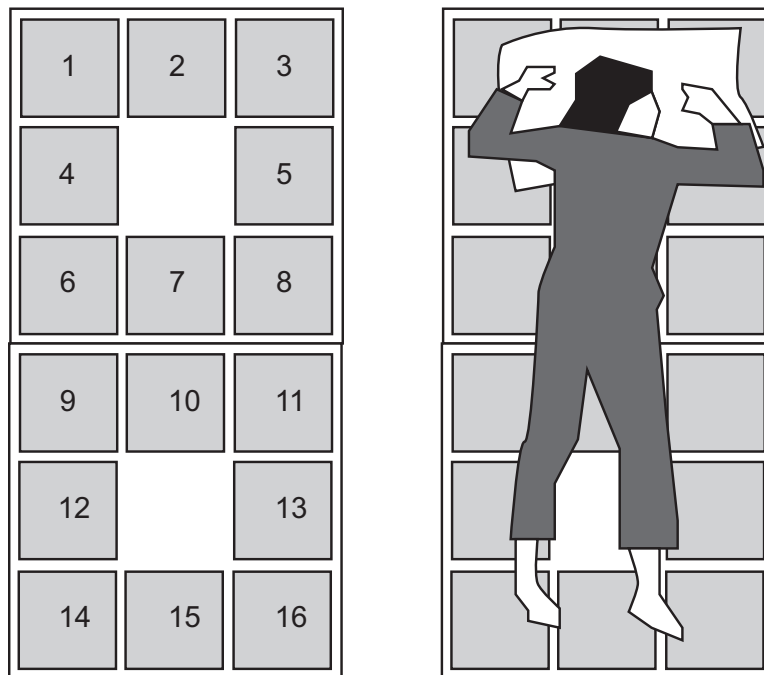


Figure 5.13: A DDR Pad has 8 binary pressure sensitive buttons on the side of a square. When two pads are placed next to one another they can cover a twin size bed.

To evaluate the performance of our movement detection algorithm, we compare it with a baseline system that uses pressure pads to measure the movement levels. The pressure sensor we used was a USB interface Multi-Platform Dance Dance Revolution (DDR) pad typically used in the popular DDR video game series. The configuration of the pad is shown in Figure 5.13. Two pads were tiled to cover the area of a twin size bed. Data collected from the DDR pad is a bit-vector of



Figure 5.14: A DDR is placed under the bottom sheet on the bed.

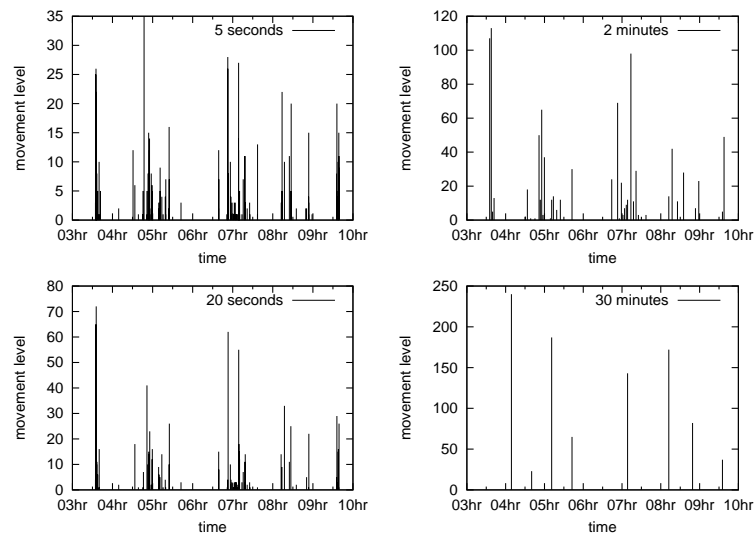


Figure 5.15: We windowed the high sampling rate accelerometer readings to form epochs of various lengths.

size 16 representing which of the 16 buttons are activated. Our algorithm examines a time window, and takes the sum of the number of changes occurring in this bit-vector in that window. We chose a window size of two seconds, same as we did for WISP tags. After calculating the number of movements during each two-second time window during the night, we clustered them in the same way as discussed in the previous section.

We also compare the performance of our system with an iPhone-based sleep monitoring application SleepCycle that uses accelerometer data to infer sleep quality. The application requires the

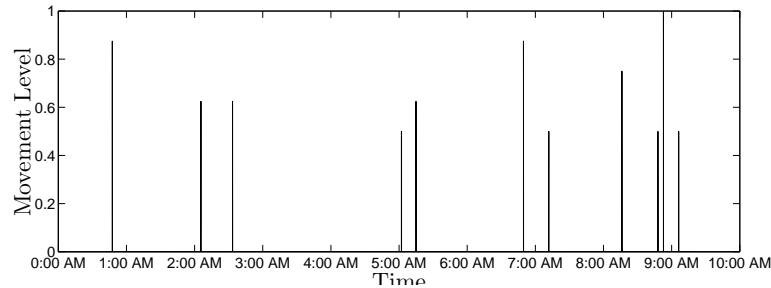


Figure 5.16: Movement determined by our system during One Night’s Sleep of Evaluation Set 1.

iPhone to be placed on a suitable position of the bed (e.g., beside the pillow) all night and it collects data from the accelerometer of the iPhone for the whole night. Based on the data, it produces sleep quality related data that includes transitions between different sleep cycles. Durations of different sleep cycles over the course of the night are part of a person’s sleeping pattern. So monitoring the transitions between sleep cycles helps in identifying irregular sleeping patterns. Our hypothesis is that transitions between the sleep cycles will correspond to higher number of movements per time window. So from our overnight report of number of movements during each time window, we can infer the transitions between sleep cycles and the duration of each of them. We test our hypothesis in this section.

The study participant slept on the same bed for six nights. We collected and logged data from the DDR pads and the WISP devices simultaneously, and also placed an iPhone on the bed (beside the pillow) during each of these six nights. The SleepCycle application recorded sleep quality data and produced a report for each night. We also videotaped the sleeping period of the subject for each night after being given the subject’s consent. We first validated the performance of the DDR pads in detecting movements during sleeping by comparing it with the video data for the first three hours of the recorded data for the first night. The validation result confirmed that the DDR pads can be used as ground truth to detect movements during sleeping. For evaluation, we use a cross validation approach. For each evaluation set, we choose five nights’ data to train our system and evaluate the performance for the remaining night’s data. So, there are six possible sets of training data. Thus, we have six sets of evaluation.

For each evaluation set, training of the movement detection algorithm includes calculating the

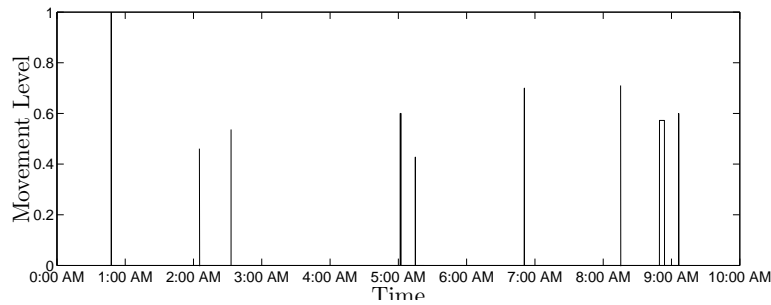


Figure 5.17: Movement Determined from the DDR Pad during one night’s sleep of Evaluation Set 1.

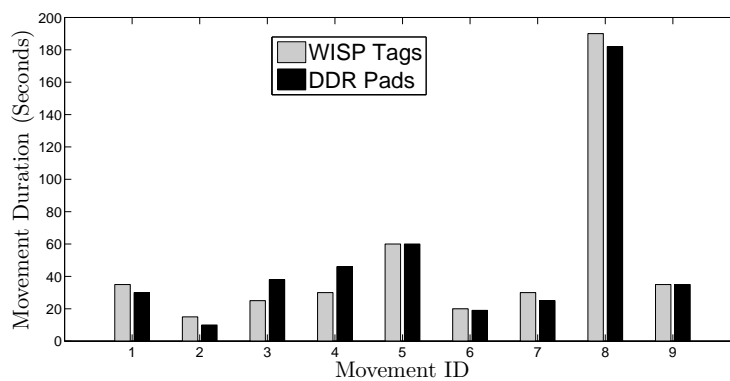


Figure 5.18: Comparison of sleeping duration between our system and the DDR pads.

thresholds of rate of change of acceleration values (i.e., values of a and b) along each axis for each tag and also the threshold to filter out the time windows where movement level is insignificant. During training, we consider movements detected by the DDR pads as ground truth. Training of the body inference algorithm includes training the Bayesian classifier with the accelerometer readings collected during the five nights with the corresponding body position. Collecting the actual body position for each time instant of each of these five nights is challenging. One option was to monitor the recorded video for each night and assign body positions accordingly. But this requires significant effort. To reduce effort, for each night, we watch the initial body position from the video and from then on we assume that unless there is a movement detected by the DDR pads, the position remains unchanged. When the DDR pads detect a movement, we fast forward to that time instant and see the new body position from the video and we continue in this way. Thus, we collect the ground truth for body position. Figures 5.16 and 5.17 show the movement events during

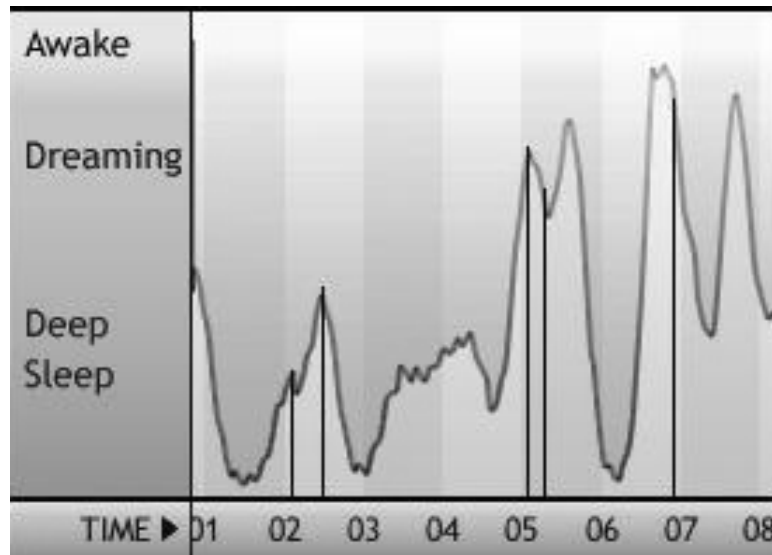


Figure 5.19: The SleepCycle App is placed on the mattress and can produce reports of awake, dreaming, and deep sleep stages.

one night's evaluation (from the first evaluation set) of the subject as detected by the WISP tags and DDR pads, respectively. If we compare these two figures, the first and last movements on both the figures represent the events when the subject got on and off the bed, respectively. Our system reported all movement events detected by the DDR pads. The timings of the movements are same in both figures. There was one movement that our system reported, but the DDR pads did not. It happened just after 7:00 AM in the morning. To investigate this incident, we fast forwarded to that specific time of the recorded video and observed that there was no significant movement during that time. So it was indeed a false positive.

Another notable difference occurred just before 9:00 AM, when our system reported two movement events and the DDR pads reported one movement event. However, the two events reported by our system are very close to each other and can be considered a part of the same movement. The duration of the nine movements during this night that both systems detected are shown in Figure 5.18 From this figure, we see that there are no notable differences between the duration of movements calculated by both the systems. We present a summary of results and their implications for all six evaluation sets at the end of this section.

Figure 5.19 shows the report produced by the iPhone application SleepCycle to show the sleep

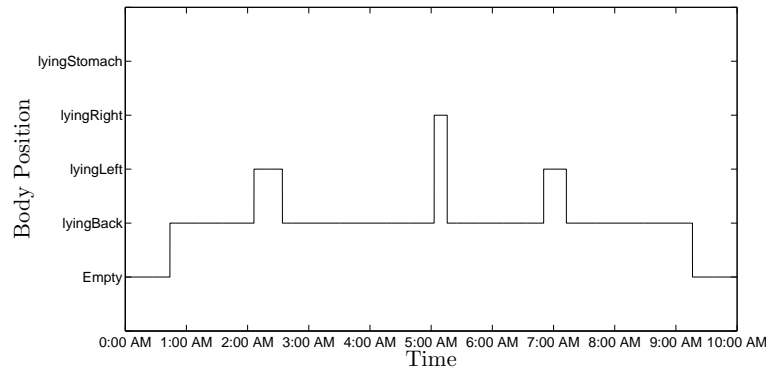


Figure 5.20: Body Positions During 1 Night’s sleep of Evaluation Set 1

quality for the same night as shown in Figures 5.16 . The application shows various sleep stages like *awake*, *deep sleep* and *dreaming*. These sleep stages are irrelevant for our comparison. The application recorded data up to 8:00 AM in the morning. The vertical bars show when movement events are reported by our system. As we know, transitions between different sleep cycles correspond to movements made by a person. From the figure we can see that the timings of the movement events match to those of transitions between sleep cycles. There are no vertical bars for two transitions: one that happened between 7:00 and 8:00 AM and the other in between 4:00 AM and 5:00 AM. During the latter one, the subject was in deep sleep stage before and after the transition. So this is why there were no major movements. We explain the reason of lack of movements during this transition at the end of this section. But this result proves our hypothesis that from the frequency of movements reported by our system, it is possible to infer transitions between sleep cycles. In addition, our system provides fine-grained body position monitoring which the Sleep Cycle application does not.

Figure 5.20 shows the body positions as inferred by our system for the same night that was considered in Figures 5.16 and 5.17. If we compare these four figures, we see that during each transition from one body position to another, there was a discrete movement event detected by our movement detection algorithm. Also, for the last three movements, the body position did not change. To ensure robustness against discrete erroneous classifications, we consider that the subject changed his body position if 20 successive instances are classified as the new body position. Also,

if the movement detection algorithm detects that a movement is taking place, the body position is considered to be the same as it was before the movement until the movement is complete.

To evaluate the performance of the body position inference algorithm of our system, we generate 10 random instances of time for each night and check the subject’s body position during each of those instances. For each night, the time instances are uniformly distributed over the course of the night. We define the accuracy of our inference algorithm to be the percentage of time instances when the body position inferred by our system match to the actual body position as seen from the recorded video data. We present the accuracy for each night as part of the summary of all results next.

Table 5.1: Summary of Results for 6 Datasets

Evaluation Set	1	2	3	4	5	6
False Negatives	0	0	0	0	0	0
False Positives	1	0	0	1	0	0
Avg. Error in Movement Detection	6.9s	6.2s	2.2s	5.2s	4.1s	5.2s
Sleep Cycle Detection Accuracy	71.4%	75%	80%	75%	90%	80%
Body Position Inference Accuracy	100%	100%	100%	90%	100%	90%

Table 5.1 presents a summary of results for our six sets of evaluation. False negatives refer to the number of movement events that are detected by the DDR pads, but not by our system. Similarly, false positives refer to the number of movement events that are detected by our system, but not by the DDR pads. For each night, we define ‘average error in movement duration’ as the average of absolute differences between the movement durations calculated by our system and the DDR pads. Sleep cycle detection accuracy refers to the percentage of sleep cycle transitions (as shown by the iPhone application) that correspond to increased number of movements detected by our system.

From Table 5.1 we see that for each set, our system detected all the movement events detected by the DDR pads. Average error in calculating movement duration is less than six seconds for each night. But, for two nights, we observe one false positive each in our system. This may be due to the threshold in change of acceleration that we selected to filter insignificant movements. We believe by training the system for more nights, we can get rid of such false positives.

Overall, our system shows 100% accuracy in detecting discrete movement events and calculates the durations of each movement with reasonable accuracy. The accuracy of the body position inference algorithm is at least 90% for all sets. Our evaluation was based on 10 randomly selected time instances that are uniformly spread over one night. More detailed evaluation is necessary to guarantee its performance. Therefore, we can say that, with proper training, our system performs as well as a system that uses pressure sensors and also is more comfortable for the users and completely unobtrusive. Moreover, our system provides fine grained body position monitoring which no existing pressure sensor based sleep monitoring system provides.

We also compare our system with the popular iPhone based application “Sleep Cycle”. Comparison results show that by only looking at the movement reports of our system, it is possible to identify most of the transitions between sleep cycles. Among the transitions that were not possible to identify, most of them were during deep sleep stages. The pressure sensors also did not identify them. So, these types of transitions do not correspond to significant body movements. We need to lower the value of the threshold for filtering out insignificant body movements which was set assuming the DDR pads’ detected movements as ground truth. Therefore, we can say that, by training our system with the transitions detected by the iPhone application, it is possible to detect all the transitions between sleep cycles by our system.

5.2 Cheaper Mote and Accelerometer Solution

The WISP solution, along with the algorithms for detecting movement on the mattress was sufficient for many sleep applications. However, there were two notable problems with using this system in practice. Although the RFIDs themselves did not require power, the reader device required a large amount of energy in order to power and use backscattering to receive the signal. There were notable warnings to exposing people to this amount of radiation over long periods of time, especially when the device is placed directly under the bed. In addition to this problem, the reader was prohibitively expensive, \$1200, for use for just a few tags on the bed.



Figure 5.21: The sleeping monitoring setup. Two tri-axis accelerometers are taped onto the bed under the mattress pad and connected to a wireless mote.

To monitor sleep cheaply and non-invasively, we built our own devices using embedded programmable microcontrollers. Our first is shown in Figure 5.21 and consists of a Wixel wireless module with a TI CC2511F32 microcontroller and a 2.4Ghz radio shown in Figure 5.22. We connected the two tri-axis accelerometers to the mote's 6 GP-IO ADC pins with a 6ft CAT-5 cable. A program written in C was compiled and loaded onto the flash to sample the ADC pins at 50 Hz and send the data to the serial adapter. When installing the sensors, the accelerometers are secured by duct tape to the mattress on the left and right side of the area where the user sleeps. Although

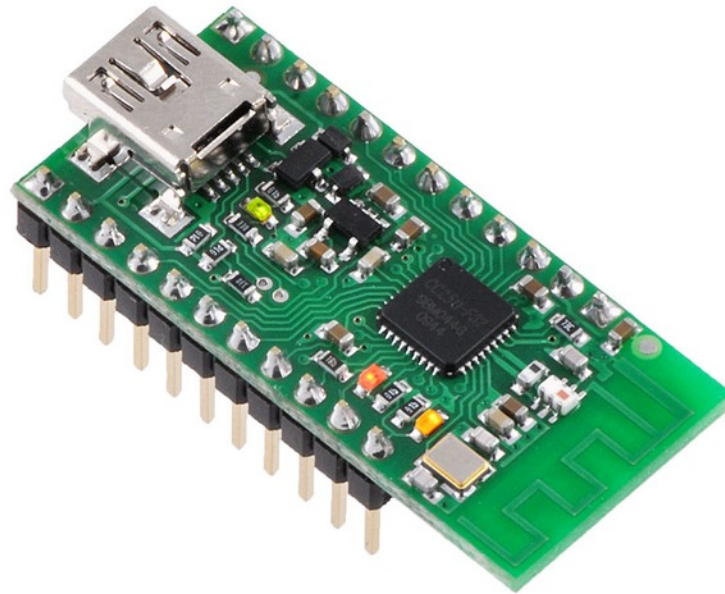


Figure 5.22: The Wixel mote has 4 ADC pins and can run on battery or wall outlet power.

the Wixel was equipped with a wireless radio, because the Wixel communication protocol did not have a MAC network layer, we used a direct USB serial connection to the mote to gather the data.

One of the most important higher-level data gathered using these devices were to record the time in and out of bed and periods of restlessness during sleep. We created an algorithm based on existing and clinically validated actigraphy algorithms [93, 94]. Time is divided into one minute epochs in which the σ and μ is recorded on each of the axis (x, y, z). The σ values are a good estimation of the level of movement during this epoch, and is robust to the sensor gradually changing orientation through the night. The μ values are valuable for recognizing the position of the person in the bed, since when a person's weight is applied to the mattress, the mattress deforms and tilts the sensor's normal vector inwards towards the center.

5.3 Conclusions

We have presented two sleep monitoring systems that advance the state-of-the-art in sleep monitoring. Additionally, we compared our systems with the most used solutions based on the level of comfort, perceived degree of privacy, the accuracy, and price. We show that the our solutions

Table 5.2: Out comparison of sleep monitoring systems show that our system can effectively balance comfort levels, privacy, and cost with high degree of accuracy.

Device	Comfort	Privacy	Accuracy	Price
Empath	High	High	High	Low
WISP Tag	High	Good	High	Moderate
Pressure Sensor	Medium	Good	High	Moderate
Camera	High	Very Low	Excellent	Medium
Zeo	Low	Good	Excellent	Medium
Actiwatch	Low	Good	Excellent	High

provide high accuracy, low cost, and high degree of privacy, and high degree of comfort than the Zeo™, the Camera, and the Actiwatch. are shown in Table 5.2.

6 | Case Studies

In this chapter we describe three adaptations of the Empath system for monitoring three different clinical problems, some of which are difficult to monitor with just one modality. Empath has been used on two long-term deployments of people with Epilepsy, three week-long deployments on studying Alzheimer’s disease and urinary incontinence, and one deployment configured for monitoring depression. For each study, we describe the motivation and medical hypotheses formulated, the design of the configuration of the Empath system components, and present the data gathered from each deployment and conclusions.

6.1 Epilepsy and Stress

The most commonly identified precipitant of seizures in people with epilepsy is stress, and the most common comorbidities associated with epilepsy are mood disorders. Elucidation of the psychological and biological mechanisms that link stress and anxiety to seizures may provide a foundation for preemptive treatment. The working hypothesis for this study was that a reduction in the amount of stress experienced by persons with epilepsy will lead to a reduction in the number of seizures reported and potentially improve overall health-related quality of life (HQoL) for these individuals. The overall goal of this mixed-methods study is to reduce stress, which can potentially reduce seizure frequency and enhance HQoL in patients with epilepsy. We investigated using an ancient Chinese healing art based in Eastern philosophy, internal qigong, referred to as Reflective Exercise in this study, as an alternative therapy intended to reduce the number of seizures and improve the HQoL of the participants.

Epilepsy and seizure disorders affect nearly 3 million Americans and more than 45 million people worldwide, making epilepsy a common chronic disease affecting about 1% of the population, with the lifetime incidence being about 3%. Approximately one-third of these individuals (nearly one million Americans) do not respond to medical therapy. Thus, any intervention to reduce seizure frequency in these patients can potentially have a huge societal impact. The treatment of epilepsy extends far beyond controlling the number of seizures a person has, given that a number of factors, including other existing health conditions, can affect seizure activity. In addition, Boro and Haut [95] report that mood disorders, including major depression, anxiety disorders, and psychosis, have a high prevalence in epilepsy.

Research shows that important and complex linkages exist between stress, sleep, and epilepsy. Synchronized non-rapid eye movement (NREM) sleep facilitates seizure occurrence, while desynchronized rapid eye movement (REM) sleep discourages it. Whereas REM sleep seems to suppress seizures, sleep disorders, particularly sleep apnea, exacerbate seizures. Seizures themselves can also disrupt sleep structure, particularly REM sleep. Insufficient sleep may contribute to decreased daytime functioning and increased seizure activity. Unfortunately, sleep disturbance in patients with epilepsy is frequently overlooked, and an understanding of these relationships is important in seizure control and in improving sleep and maximizing HQoL for patients with epilepsy. In a systematic survey, the most commonly identified precipitant of seizures in people with epilepsy is stress. Biological indicators of stress and anxiety include changes in heart rate, systolic blood pressure, and diastolic blood pressure. Individual baselines of these measures could serve to demonstrate hyperarousal states. Thus, the psychological and biological mechanisms that link stress and anxiety to seizures require further exploration and may provide a foundation for preemptive treatment. Rajesh and colleagues, in a study of a yoga meditation protocol, concluded that yoga (and perhaps other mind-body modalities that reduce stress) shows promise as a cost-effective and risk-free adjunctive treatment in patients with drug-resistant epilepsy.

Although a variety of anti epileptic drugs (AEDs) are available, 38% of new cases of epilepsy are resistant to AEDs. Even when effective, AEDs can produce toxic reactions, and use of these

medications is associated with mood and cognitive dysfunction. Thus, the development of complementary therapy approaches that could enhance the efficacy of current AEDs or perhaps reduce the dosage needed by reducing seizure frequency, with few or no side effects, and improving HQoL is of considerable importance. Given the limited effectiveness of conventional treatment for epilepsy, some patients have used complementary and alternative therapies, suggesting its acceptability to this population. Data from the 2007 National Health Interview Survey indicate that around 35% of individuals who self-report a physician diagnosis of seizures have used mind-body therapies, as compared to about 20% of the general US population. Of the modalities that patients reportedly used, Wahbeh and colleagues argue that meditative relaxation techniques are among the most promising mind-body interventions for control of epileptic seizures, although these researchers acknowledge that more research is needed. Sahaja yoga, which includes components of meditation, has been shown to clinically benefit patients with epilepsy by reducing seizure frequency and reducing the effects of stress on patients.



Figure 6.1: Reflective exercises combines elements of Qigong and meditation to try to improve symptoms of epilepsy.

6.1.1 System Design

The epilepsy study was designed to monitor the sleeping habits of the participants over a long period of time; sleep was the only modality used, since the study organizers did not want to burden the

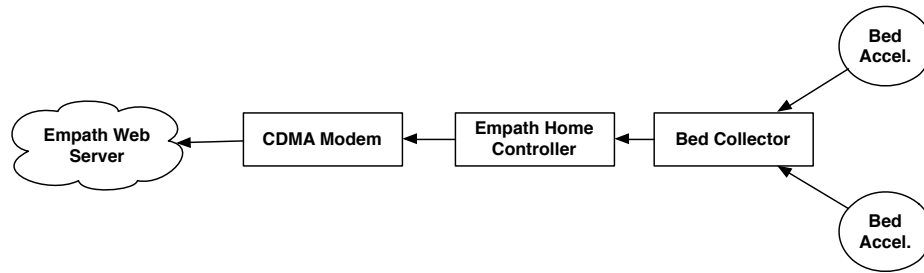


Figure 6.2: The Epilepsy study required only the sleep monitoring components of the Empath system.

subjects with overwhelming instrumentation. This deployment was a simple instantiation of the Empath flexible architecture where only a single modality is required.

To measure sleep quality and disturbances, we instrumented the bed with accelerometers. In our design, two tri-axis accelerometers, sampling at 50 Hz, are placed on the right and left sides of the bed. Because the accelerometers are placed just beneath the mattress pad, we are able to detect fine movements while the patient is unaware of their presence. The movement levels for one minute epochs are calculated by computing the mean and standard deviation of the samples. The weighted sum of activity values across a time window of seven minutes is used to estimate wake and sleep periods using Cole’s Actigraphy algorithm [93]. The following objective measures for sleep are determined: time in bed, sleep onset latency, wake and the number, degree, and duration of restlessness. The details of our accelerometer-based sleep monitoring systems are covered in Chapter 5.

We configured the Empath platform for this data collection application. The system is shown in Figure 6.2. When patients were recruited for the study, it came to light that many lived in rural areas and did not have internet service providers. In order to enable these subjects to participate, we adapted the Empath system architecture to use wireless access points. Even so, the wireless connectivity was intermittent and unreliable; therefore, we uploaded data in an opportunistic manner by synchronizing the local data with the server’s database. We chose the Amazon Cloud Services hosting because of our need for reliability, load balancing, cheap storage, and accessibility. An Elastic Compute (EC2) instance ran the Java application in a Jetty container. MongoDB was installed

on another dedicated instance. During the deployment, each one-minute epoch of bed movement data was stored as a record in the database. The data then can be accessed through a web interface for the engineers to ensure the deployment is running correctly, and for the clinical researchers to investigate the patients' data.

6.1.2 Results and Lessons Learned

We deployed the system in two homes for 6 weeks each. The first participant was a 19-year-old female who had experienced seizures since adolescence. She lived in a rural mobile home one hour from the University. The second participant was a female 28-year-old living in Charlottesville, who also started her seizures as a teenager. We tracked the participant's bedtime, wake time, and duration to discover any anomalies in sleeping patterns. The normal sleeping patterns could be represented as a density distribution as shown in Figures 6.3 and 6.4.

Our practical experience with these two deployments led to the following discoveries, which informed future designs and deployments:

- People do not necessarily sleep on their bed when they sleep, which affects the accuracy of the data. For example, one participant often fell asleep in front of the television on the sofa many nights, and also took frequent trips on weekends and did not sleep on the mattress.
- Accelerometers can only detect whether an individual is on the bed, not whether they are actually asleep. One participant, for instance, reported multiple cases of sleep walking during our data collection.
- Devices fail and the software will crash, therefore a system that allows remote administration is vital, so that problems can be resolved without traveling or entering the patient's home.

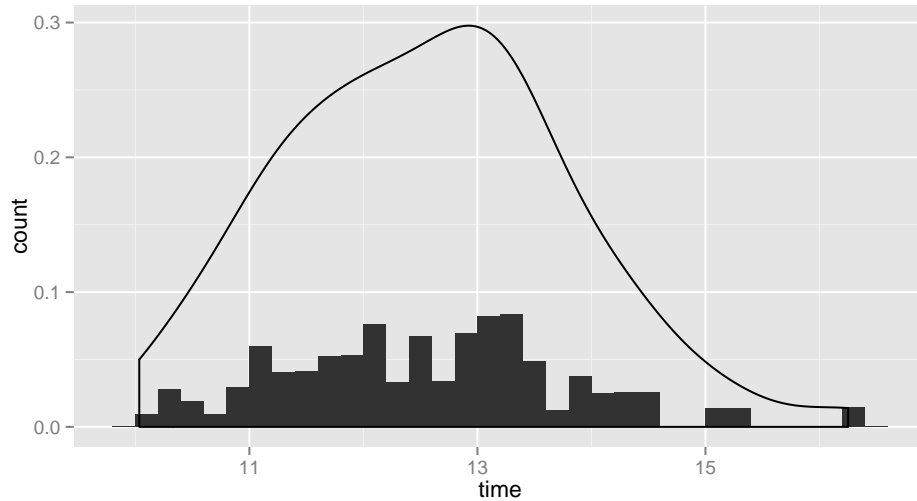


Figure 6.3: Bedtime Density Plot

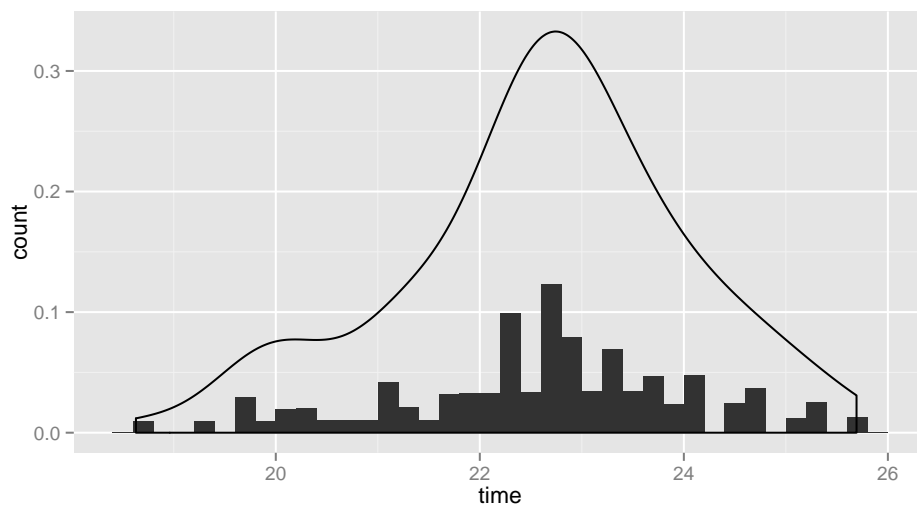


Figure 6.4: Wake Time Density Plot.

6.2 Alzheimer's Disease and Incontinence

Alzheimer's disease is the most prevalent type of dementia in the US, affecting 4-6 million people, and is estimated to expand to 17 million by 2050. Alzheimer's frequently presents with episodes of nighttime agitation, which are highly burdensome and costly for caregivers and for the health care system. Additionally, people with dementia such as Alzheimer's are also much more prone to urinary incontinence than others. Family caregivers report that coping with nighttime agita-

tion, associated sleep disturbances, and urinary incontinence can be overwhelming and that these disturbances are often a “tipping point” for seeking institutional care for their loved ones.

There is anecdotal evidence that there is a relationship among incontinence, sleep disturbance, and agitation in these patients. Urinary incontinence is thought to trigger awakening from sleep, with subsequent agitation, although there is a lack of systematic evidence that these phenomena are related. Based on previous work with urinary incontinence and sleep in persons with Alzheimer’s disease, this study aims to describe the relationships among the times of occurrence of nighttime agitation, sleep continuity and duration, and urinary incontinence in persons with Alzheimer’s disease by using innovative, non-invasive sensing technology.

Currently, nocturnal agitation in older adults with Alzheimer’s disease is most frequently treated with anti-psychotic and hypnotic medications, although these pharmacological therapies result in little success and have adverse effects on persons’ outcomes and quality of life. This study hopes to prove that simple, non-pharmacological nursing interventions such as regularly waking patients to use the toilet may reduce nighttime agitation, sleep disturbances, and urinary incontinence in persons with Alzheimer’s disease. The study will lead to interventions that can improve the quality of life for patients and their caregivers: delaying institutionalization, reducing caregiver burden, and resulting in substantial savings in the cost of caring for many persons with Alzheimer’s disease.

6.2.1 System Design

This case study shows that the Empath architecture is easily extended for multiple modalities. The system design for the incontinence study begins with the accelerometer-based sleep monitoring system that the epilepsy project required, plus three additional modalities shown in Figure 6.5. Perhaps the most important is the ability to detect when a wetness episode occurs. The *Dry buddy* is a small, lightweight, and wireless sensor that uses a magnetic locking system to keep the sensor in place on the outside of the undergarment or pajama. It uses electrical conductivity within an incontinence pad to determine if a wetness event has occurred. In Figure 6.7 we show the sensor and where it attaches. We monitor audible speech outbursts by way of a microphone. We additionally

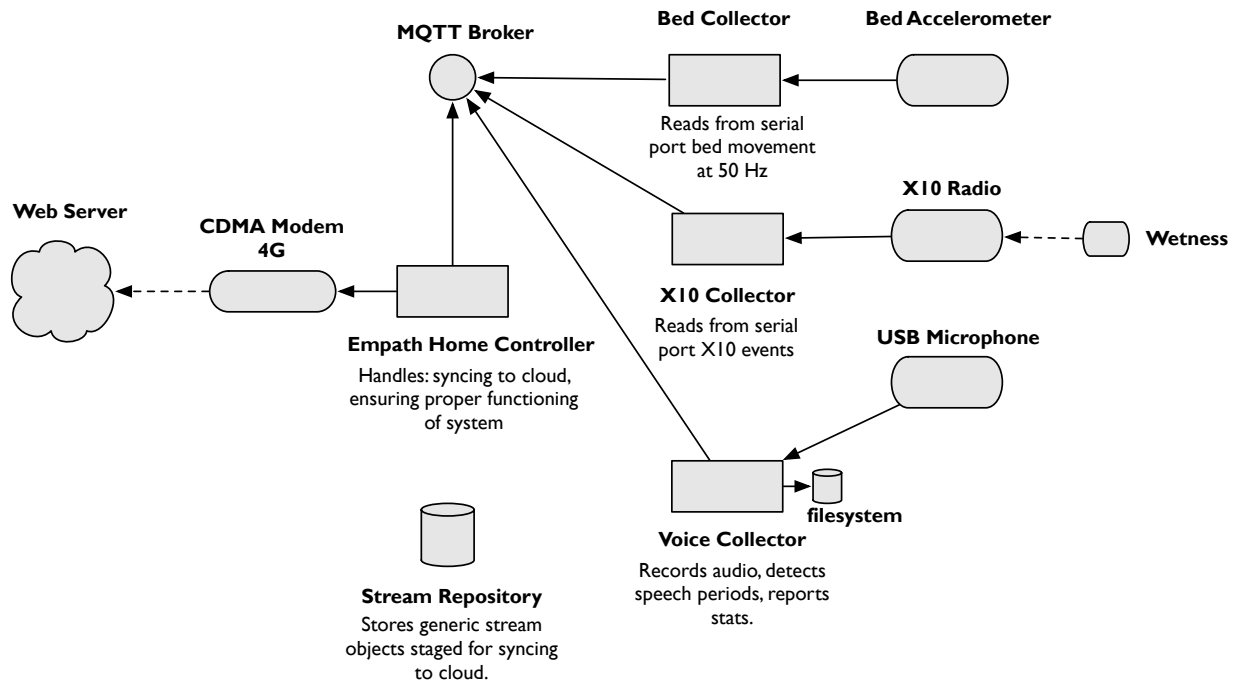


Figure 6.5: The incontinence study uses wetness sensor, bed accelerometer, and bedside microphone.

collect more detailed data about body movement by using TEMPO motes strapped to the left and right wrists of the patient.

6.2.2 Results and Lessons Learned

1. The voice agitation sensor picked up other sounds besides vocal outbursts, including the snoring of the participant. It was also not possible to automatically and accurately discern the character of all vocalizations, especially whether it was an intelligible utterance or a mumble.
2. The DryBuddy device had connectivity problems, in one instance producing a false negative despite an incontinence episode occurring. This was confirmed by both the husband attending



Figure 6.6: DryBuddy System detects incontinence episodes and sends event packets by radio by X10

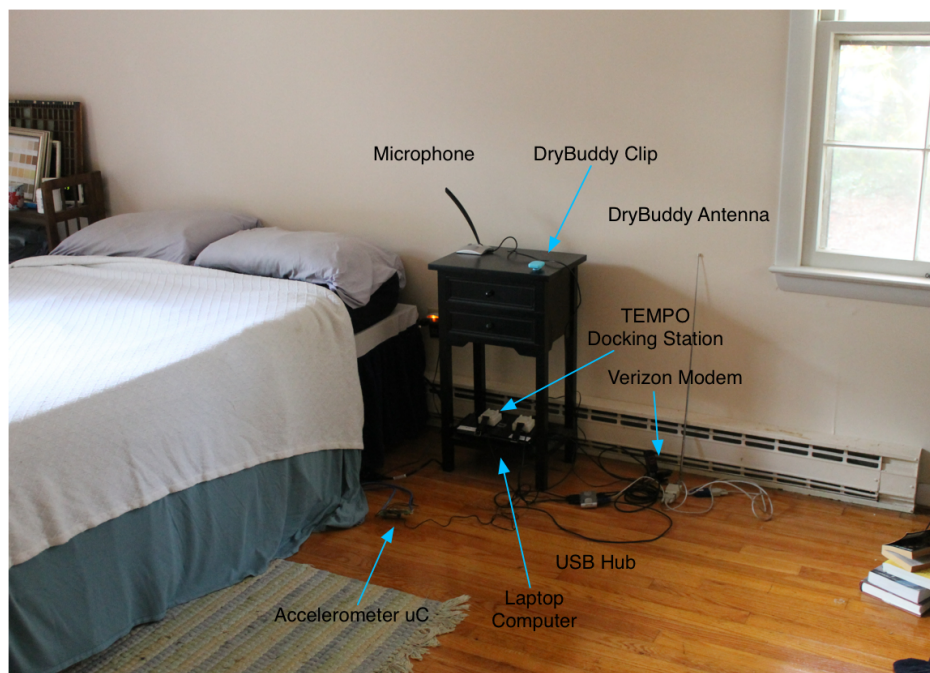


Figure 6.7: The system bed-side monitor combines mattress accelerometers, wrist accelerometers, a microphone, and a wetness sensor to investigate agitation before wetness episodes.

to the episode, and the weight of the pad afterwards. Although the DryBuddy was triggered, the signal was attenuated and lost in transmission. We fixed this problem by introducing

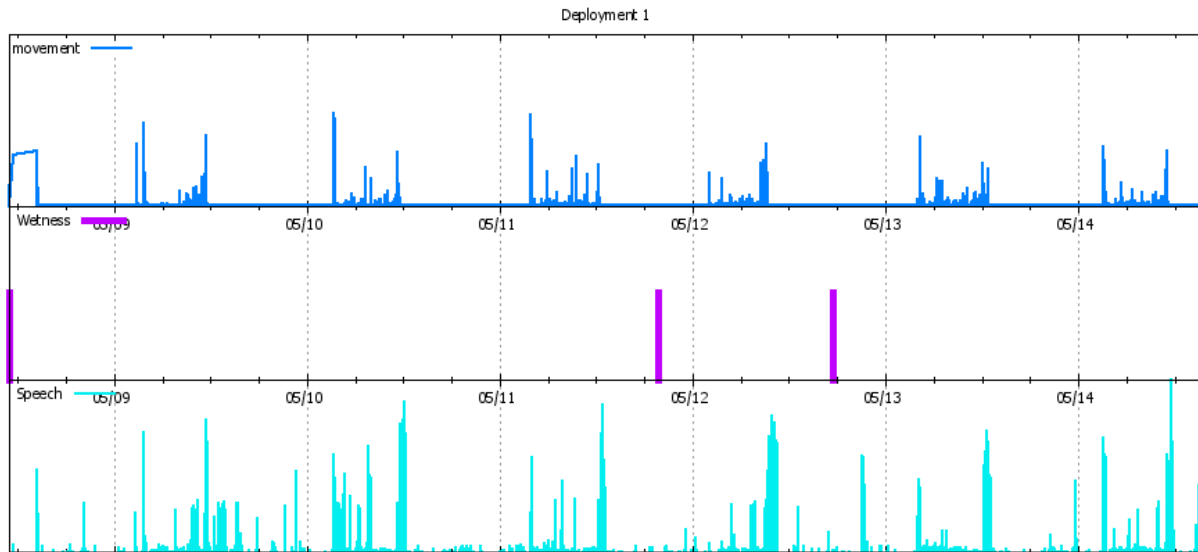


Figure 6.8: Incontinence study deployment 1: shows the relationship of the vocal outbursts, movement, and wetness episodes.

multiple X10 receivers on either side of the bed to more closely detect when the device is triggered, so the signal could not be blocked the patient's body. Adding more DryBuddy sensors on the pad was another potential option, however it was too uncomfortable for the user. This is an excellent example of the versatility and freedom of the Empath architecture.

6.3 Depression Monitoring

Depression is a major health issue that affects over 21 million American men and women each year. Depression often goes unrecognized and untreated, and even once treatment begins it is often difficult to monitor its effectiveness. This poses particular challenges for the diagnosis and treatment of depression, particularly for those who avoid visiting a doctor or therapist due to social stigmas or a lack of energy. Currently, depression diagnosis is based on subjective screening questionnaires or structured clinical interviews that rely on timely in-person visits as well as accurate recollections by the patient, these have been shown to be inaccurate since symptom reports are often exaggerated or left incomplete. This makes monitoring depression symptoms exceedingly difficult. Yet early detection and treatment of this debilitating disorder has been shown to improve patient outcomes

considerably [96, 97]. Also, the period of a few months after recovery is particularly prone to recurrence of depression, and that over time the longer one goes without relapse the less likely one is to develop a new episode. Therefore monitoring of symptoms even after they are ameliorated, recognizing early signs of an emerging episode is important to long term care of depression.

Along with depression's detrimental affect on mood, it can lead to other associated problems because of reduced social interactions, decrease in personal hygiene, increased alcohol use, and neglect of medications for current medical conditions. Depression is listed under the Americans with Disabilities Act (ADA), and equates to large amount of money lost due to missing work, disability, and lost productivity. According to a recent study, the cost to the workplace in terms of short-term disability alone, is greater than that of lower back pain, heart disease, high blood pressure, and diabetes together. Assessment and treatment are often hampered by a lack objective data to corroborate patients' retroactive self-reports about their current functioning; hence an objective symptom-monitoring tool such as Empath could complement subject self-report measurement and enhance diagnostic accuracy.

Depression has several behavioral and psychosomatic manifestations [98, 99]. Independently, each has been studied and is well documented in clinical research as well as in the widely used Diagnostic and Statistical Manual of Mental Disorders (DSM-IV) [65]. For example, depressed speech is marked by pauses, fewer words, monotone voice, and less eye contact. Depressive episodes frequently affect sleeping patterns, leading to increased or decreased sleep duration as well as diminished sleep quality (with frequent bouts of waking in the night, and more restlessness during sleep). Depressive episodes are also commonly characterized by lack of social interaction and signs of anhedonia, the lack of pleasure in doing things one previously enjoyed, and hence withdrawing from social, exercise, or recreational activities or deviating from one's usual patterns. Appetite changes and resulting weight gain and loss are other commonly observed symptoms of depression. Behavioral changes associated with depression onset also include listlessness, lethargy, and reduction in gross motor activity and slowing of gait. Each of these components on its own will not give caretakers a complete picture of an individual's condition, since depression is marked by a combination

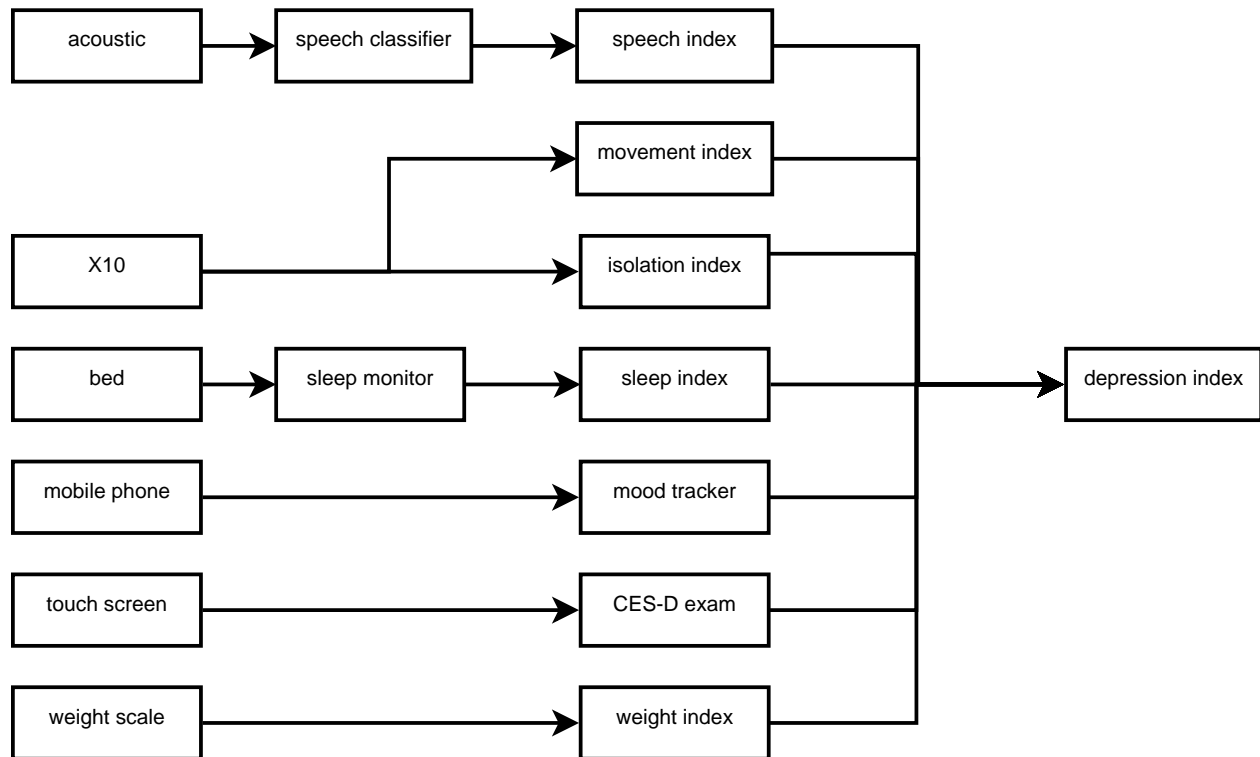


Figure 6.9: The inference system for creating a depression risk index.

of symptoms and each individual exhibits symptoms in different ways, and moreover, the way they are expressed is dependent on the severity of the case.

6.3.1 System Design

To address this problem, we adapt Empath with the necessary components for multi-factor depression monitoring. When properly configured, the system can assist with collecting and compiling necessary data for caregivers, predicting signs of a possible depressive episode, and can track symptom severity in order to evaluate a treatments effectiveness. The benefits over current diagnostic methods could result in lowered recurrence of episodes, shorter episode length, and ultimately improved quality of life for many sufferers of this disease.

The system was deployed in a real apartment for over 30 days. It took less than one hour to install Empath in the participant's home. 17 X10 devices were attached to the stove, freezer, refrigerator, kitchen sink, microwave, cabinets, bathroom sink, trash can, wardrobe closet, and

shower. A wireless weight scale was placed on the floor of the bathroom. A PC running the basestation software was in the living room. The total cost of the system excluding the PC is less than \$500. Twice each day, the participant recorded his subjective mood and once a day provided voice data through a touchscreen kiosk.

6.3.2 Activity Levels

We use passive activity detection sensors to detect symptoms that are related to loss of energy, anhedonia, and social isolation. In particular, Empath's activity detection module examines two key factors linked to depression: 1) home occupancy and 2) movement levels.

Depression can express itself as anhedonia or by social isolation. If this is the case, patients will leave their homes less than normal. We measure the percentage of time spent in the home versus away, and monitor for anomalies in this pattern. Many factors can contribute to a change in these activities, and the results are sensitive to false positives that do not have a mood component, such as going on a vacation, injuries or illnesses, and varying work schedules.

We created a simple algorithm to predict the time the patient would be in their home. The basic principle is that we can segment the periods where people are inside or outside of their homes by noting when a front door sensor fires. In our deployment, the front door was the only door through which someone could enter or leave the apartment. (However, we discovered that each time the door opened, the patient did not necessarily enter or leave their home, since the resident often kept the door to let a breeze or more light in.) Next, each segment between door opening events is labelled as 'occupied' or 'not occupied' by incorporating data from the other X10 devices in the home (motion detectors, kitchen cabinet sensors, etc.). If the sum of events within a segment exceeds a threshold, the segment is labeled as 'occupied.' This simple approach works sufficiently well, however it has a single point of failure. If the front door sensor malfunctions, the system cannot define crisp boundaries for home occupancy, and instead would have to rely on clustering of other activities to estimate occupancy. We updated the system architecture by using double redundancy on front door sensors to improve reliability. A score from 0 to 3 is generated for the

social isolation score, and one point is given for each increase in time spent at home by one standard deviation. Next, the movement factor is computed from the number of sensor firings that go off in a day. We scale the activity level to the amount of time spent in the home.

6.3.3 Activities of Daily Living

The activities of daily living are logged for viewing by a caregiver, but are not automatically used for various reasons. Firstly, the calculations of factors from these data streams that have a high mis-classification rate can make determining anomalies challenging. Secondly, it is difficult to determine what is an anomaly. This problem is addressed in the *Holmes* framework that we present in the paper [100, 101] and in Section 7.2.3. The activities that can be monitored include cooking, hygiene, and cleaning. We differentiate two types of cooking: preparing light meals and snacks, or preparing more complex meals. To do this, we placed contact switches on the microwave, the oven, and the cabinets (spice and sauces), the refrigerator, and the freezer. If the activity recognition module detects the opening of the spice cabinet or the use of the oven and stove during a cooking activity cluster, the whole cluster is labelled as a meal. However, if just the microwave or the freezer is opened but without the previous mentioned sensors, it is considered a light meal. Detecting whether someone is eating out, or not eating at all is challenging, and perhaps cannot be easily determined using only our in-home activity recognition system. For tracking regular cleaning and hygiene, we detect showering, using the bathroom sink, opening cleaning closets, and opening the trash lid. A motion sensor was placed in the shower unit and over the sink, and contact sensors are placed between the trash can and the lid and storage cabinets.

6.3.4 Patient Questionnaires

The patient interface shown in Figure 3.3 runs on a touchscreen inside the patient's apartment. Its primary purpose is to receive continuous subjective scores from the items in the CES-D exam. The test is available at all times, but encouraged to be taken once a week. The exam consists of 20-items, where each item is scored on a scale from 0-3 points. The sum of these items are used to predict

the severity of the episode. A score of 15–21 might suggest mild to moderate depression and over 21 a possibility of major depression. This application was implemented with Adobe AIR 2, which connects to the server and transmits the scores through JSON. Notifications and alerts can be sent from the server to the patient that appear on the screen. Some examples of alerts include: reminders to check body weight on the scale and to complete a late CES-D exam. The touchscreen's role could be expanded to serve as a mood coach, social planner, and mood journal. Personal behavioral factors similar to the caregiver is presented to patient for positive feedback. We created an iPhone application that serves as an sensor for instantaneous mood measurements. The patient to input instantaneous mood on a 1–10 ladder on the continuum of extreme depressed to extreme elevated mood. We created this mobile interface so that the patient does not necessarily need to be in his or her home to input data into the system. This experiential sampling approach [102] is useful for collecting instantaneous measurements. The iPhone's local notifications can be enabled to alert the patient when a new measurement is recommended. The application's capabilities can be expanded in the future to record other types of emotions, such as levels of anxiety or irritability which are also typically experienced during depression.

6.3.5 Results

In this section, we present a case study of the first version of the Empath system configured for tracking depression and deployed in a real apartment over a period of 4 weeks. Although these results are not meant to investigate any medical hypotheses, it however shows an example of the system in operation and how it is able to collect useful data about a depressive episode continuously in the home. Adding additional modalities involved registering an X10 receiver to the broker to add activity recognition and the a touchscreen client that will report the questionnaire. In addition, on the server, the weightscale API was configured to pull information from the webservice.

It took less than one hour to install Empath in the subject's home. X10 devices were attached to the stove, freezer, refrigerator, kitchen sink, microwave, spice cabinet, plate cabinet, glasses and cups cabinet, front door, cleaning closet, medicinal closet, bathroom sink, trash can, wardrobe

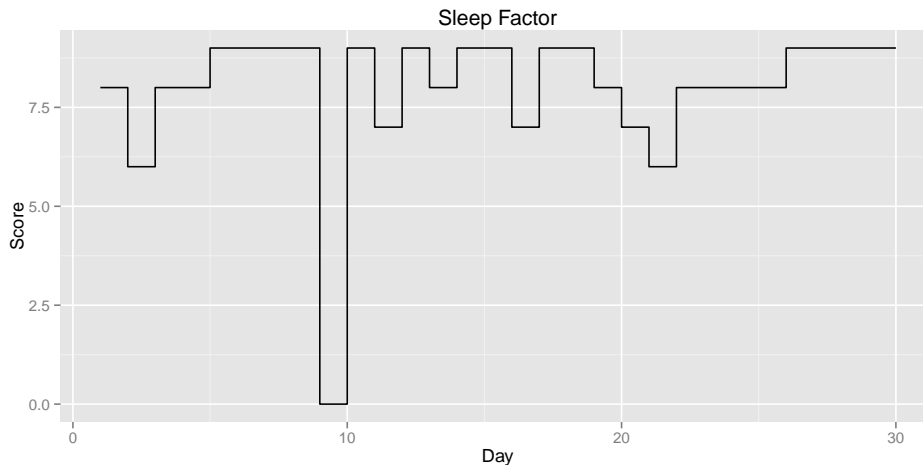


Figure 6.10: The sleep quality factor measure for each day for a month.

closet , and shower. The weight scale was placed on the floor of the bathroom. A computer with the client software was placed in the living room. The total cost of the system excluding the laptop and phone is less than \$500. The subject used the iPhone App to record his mood twice daily. Due to the diurnal variation of symptoms during depression, these measurements can vary greatly. We took measurements more frequently than would be typically needed by a patient using Empath. But the high-granularity of data is useful for comparing against the objective factors.

Each morning, the subject reported his subjective rating of the previous night's rest as being good or poor. Figure 6.10 shows the sleep quality rating for each night. We inverted the sleep score by taking the difference from 9, since we wanted to present in the graph poor sleep quality with a lower number. The nights where the subject responded that his sleep was poor were on days 2, 3, and 10, which appears to correlate with our sleep quality index. The graph suggests that for this subject, the previous day's mood highly affects the sleeping quality that night. These results show how Empath's sleep monitoring solution can approximate sleep quality with some degree of accuracy. However, one challenging problem we aim to solve, is determining the sleep efficiency , the amount of time spent in bed attempting to sleep rather than actually sleeping. We plan to run studies showing the relationship between actual sleep times and bed motion

The occupancy detection algorithm was used on 11 days worth of data. Figure 6.12 shows for

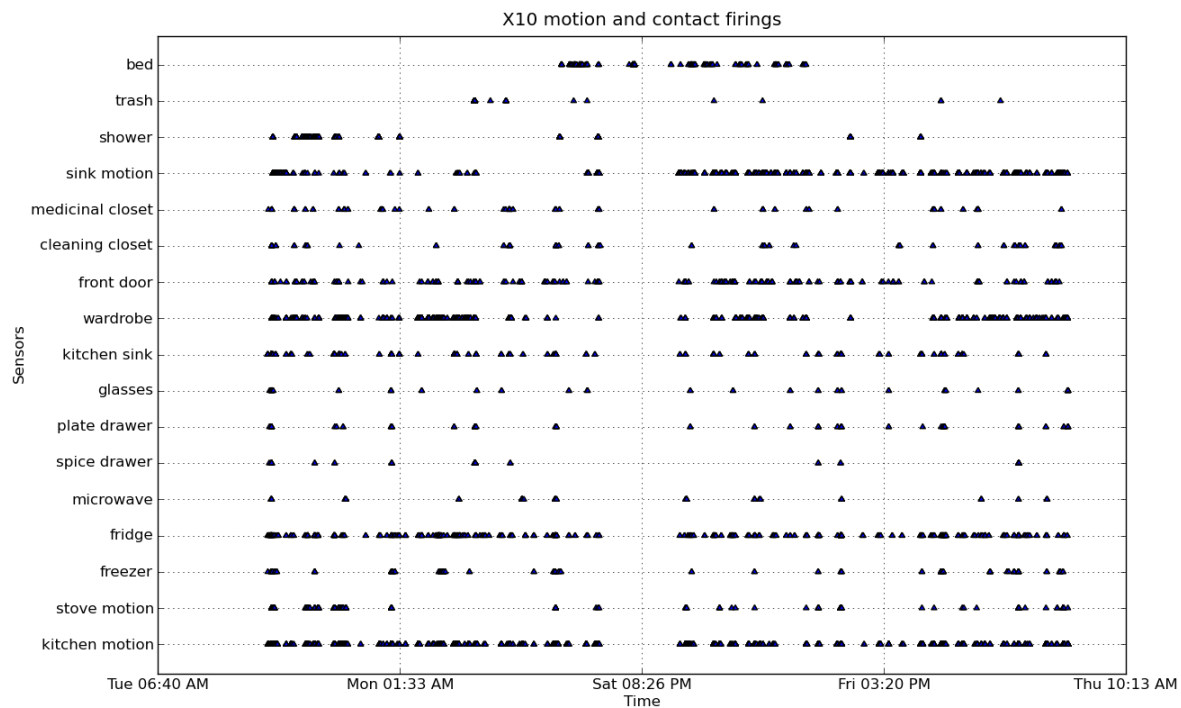


Figure 6.11: The X10 Readings from the deployment.

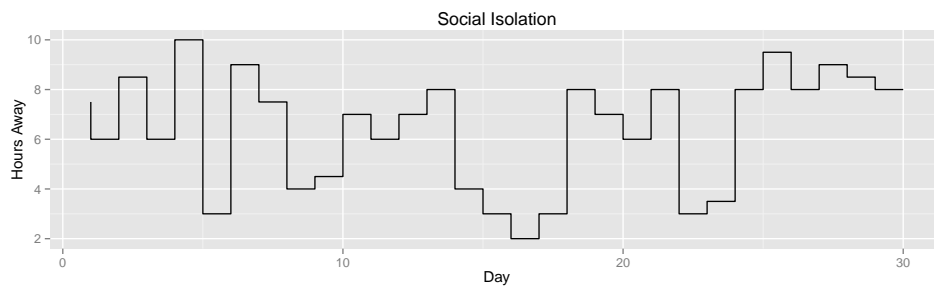


Figure 6.12: The house occupancy can be used to estimate social isolation.

each of the 11 days the amount of time spent away (vacant) each day. For this particular dataset, we found no relationship between mood and time spent away from the home, by running an ANOVA on the linear relationship between mood and duration. The assumption here is that higher levels of vacancy correlate to less social isolation. There are complications to this measure as if the subject stays at home, but receives visitors, the factor will be lower than it should be. In addition, times spent on vacation can produce errors in this estimation. We see that this is where other factors are

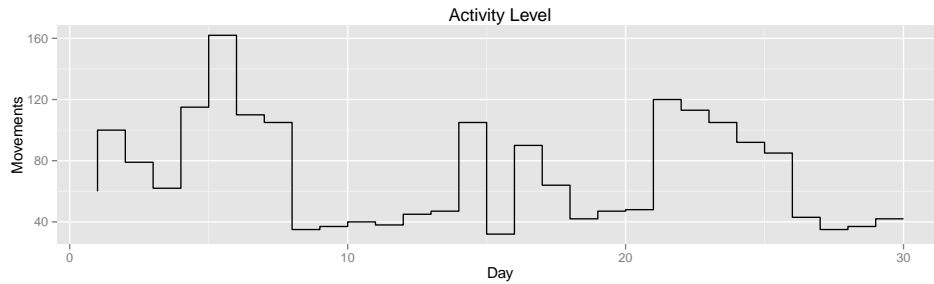


Figure 6.13: The movements in the home are related to depression.

important in this measure.

For each day, we recorded the number of sensor firings in the home to give us a gross estimate of the amount of motion and activities occurring in the home. Those who stay still, and therefore do not interact with many devices, and will receive a lower movement factor. We realized that the number of firings do not give us a fair measure of the activity level, since a person who scurries about their apartment for a few hours would receive a lower score than someone who spent the entire day in the apartment but spending most of the time on the couch. So we normalized the score based on the apartment occupancy times. Figure 6.13 shows the results of producing this factor against the reported mood. The sixth day was the most active for the participant, since day was spent cleaning the apartment. We ran an ANOVA on the linear model again to find a relationship between the movement factor and mood levels, but no significance were found. This method gives us an approximation of energy levels, that may correlated heavily to psycho-motor retardation that depressed individuals experience.

We use various factors together for arriving at a depression risk index. People exhibit depression in different ways, so relying on single measures is not accurate. In this case study, for instance, the speech factor and sleeping factors were most highly indicative of depression, while the weight, movement, and isolation were within healthy limits. The depression index is a weighted sum of various subcomponents, however should any of the sub-factors be in extremely high risk, the depression index should be elevated to a high level. Studies on the population need to be run first to make decisions on the appropriate weighting factors.

6.3.6 Lessons Learned

- Although the touchscreen device was able to capture voice data, the mood was sampled only once a day in artificial situations and not through real conversations happening in the home. This motivated our research direction of applying the same voice classification strategy but captured ambiently as discussed in Chapter 4.

6.4 Conclusion

In this chapter, we presented 3 different configurations of the Empath system for handling collecting useful information to clinicians. This adaptability was useful since all of the applications required ease of deployment, minimal invasiveness, and reporting of data to the cloud. New modalities were able to be introduced such as the case when migrating the existing platform for studying epilepsy and augmenting it with acoustics, wetness episode, and TEMPO. For the incontinence project, a new X10 receiver had to be added with minimal reconfiguration to the existing system.

All of the architectures required reporting time series information collected over various different time frames from one minute epochs to infrequent event-based data. Finally, for many applications transforming the data from low level epochs to higher-level epochs represented by different streams was trivial.

7 | **Conclusions**

This thesis presented many advancements to home monitoring systems. For each, we place them in context of existing approaches.

7.1 Key contributions towards home monitoring systems

7.1.1 Extensible, multimodal, easily deployable

Most home monitoring systems are expensive and hard to deploy. They are aimed for general monitoring, but more applications are needing a system to be adaptable to targeted monitoring scenarios. For instance, many applications do not need all of the sensors useful for a depression monitoring setup, some systems require specialized devices to be integrated into an existing setup. We have presented an extensible, multimodal, largely passive behavioral monitoring system called Empath in that is useful to caregivers in order to monitor their patient's behavior, and thereby track their well-being and their response to treatment and therapies. It is unique because it provides a flexible system architecture that allows easy introduction of new custom components or commercial off-the-shelf products to be integrated using popular web service abstractions such as RESTful interfaces and JSON. This enables 3rd party APIs to be integrated into Empath, or Empath can integrate into their services.

7.1.2 Passive acoustic monitoring

Acoustics are a valuable information source for clinicians attempting to track complicated diseases. A major hurdle to the integration of speech monitoring into these systems has been because of reverberation. We were able to effectively target the unwanted effects of reverberation captured by deployed room microphones by creating a matched-condition classification strategy called RESONATE that uses room acoustic simulation to quickly generate models for a particular room. We show that this kind of classification can be done in realtime, on-node in a distributed architecture so that this information can be generated without raw data being sent to a base station or cloud service.

7.1.3 Sleep monitoring

We showed that existing sleep monitoring systems are either too burdensome, expensive, or inaccurate. We have presented a passive and cheap sleep monitoring system that can collect useful features such as insomnia, restlessness, and sleeping patterns that are necessary for many applications. Besides engineering new hardware, we have evaluated the use of machine learning to infer sleeping position. The sleep monitoring system is tested in each case for several weeks or months in three clinical studies and three non-clinical studies.

7.1.4 Tested in the field

By deploying various iterations of the system in different settings, the Empath system has results and observations from deploying the system in real clinical studies: examining the relationship of sleep and stress on the number of seizures that people with epilepsy experience, the relationships between nighttime agitation and incontinence events with those with Alzheimer's disease, and finally case-studies with configuring the system for depression monitoring. Despite the differences in the design, we demonstrate that each data source can generate discrete epochs of data that enter

into the system at various points including a publish-subscribe channel at a message broker or a URL at a RESTful interface.

7.2 Future Improvements

There are many notable extensions and improvements to the research we have presented in this research.

7.2.1 Privacy

Due to the dynamic nature of sensor data and techniques for inference in an attacker's arsenal, there is an overwhelming threat to an individual's confidentiality of private data. Emerging personal health record applications such as Microsoft's Health Vault and Google Health have created controversy over the security of medical information. But as more devices such as mobile devices, body sensor networks, implantable medical devices and smart environments are integrated, and as the number of 3rd party applications using subsets of that data, the challenge of keeping information secure will be worse. Although advancing research is addressing maintaining access control, confidentiality, and anonymity in respect to generic medical databases, there has been limited work in regard to stream-centric data collected from these home environments. There is a considerable challenge since these system generate vast amounts of highly correlated information, and medical applications would not be able to run effectively if stringent policies are held regarding the release of data.

7.2.2 Recommendation Systems

The system could be enhanced to provide encouragement or behavior modification. Recommendations could be sent with or without the assistance of a caregiver. We formed a list of possible activities shown in Appendix C taken from a local Senior Activity center and labeled each with the category that the corresponding activity targets, such as *Social*, *Mental*, and *Exercise*. Behavior

modification is a burgeoning research area that poses considerable challenges in developing proper techniques for influencing human behaviors both effectively and ethically. Many businesses have successfully used *Gamification* to leverage people's natural desires for competition, achievement, status, self-expression, altruism, and closure by influencing behavior in a game-like framework. Typically, gamification will reward badges or unlock achievements for doing certain activities. Fitocracy is an example of a health and wellness app that has effectively used Gamification. The app allows people to record the type and duration of the exercise they do, and the system rewards experience points commensurate with the difficulty of the exercise; after enough experience points have been earned, the user will gain a level and be able to share their achievements with their social network. Similar techniques could be employed in home monitoring systems, where points could be awarded for making social contact with someone, attending events, doing regular exercises, preparing home-cooked meals, or keeping a regular schedule.

7.2.3 Semantic Anomaly Detection

With *Semantic Anomaly Detection*, activities of daily living can be monitored by detecting anomalies in their behavior. But unlike traditional anomaly detection systems, the aim is to reduce false positives in anomaly detection with the help of semantic rules. Some of these rules are predefined based on expert knowledge and the rest are learned by the system with the help of resident/expert feedback. In our co-author's work [100], trend of change in different activities can be used to improve anomaly detection. In addition to monitor statistical deviation from regular behavior, they detected deviation from healthy and social norms (defined by experts) as anomalies. Future work is investigating whether the concept of *entropy* can be used to describe the amount of anomalous behavior in the home. Integration of this work with Empath could help finding difficult to find anomalies and minimize the number of false positives.

7.2.4 Acoustic Monitoring

This thesis presented a solution to the reverberation problem, however, there are many open challenges for ambient and passive sound monitoring in home environments. There are many sounds in the home that must be filtered out in order for the system to effectively only monitor the speech. In addition, when multiple people are talking at once, or if television or music is played in the background, the system will fail. Advancements in Blind Source Separation (BSS) could eventually disentangle different sound streams in a clean environment. Unfortunately, it is hard to get BSS to produce good results in reverberant environments. We have shown that other physiological sounds such as coughing, laughter, and snoring are picked up by the microphone as unclassifiable speech and therefore new classifier chains must be developed.

7.3 Other example applications

We have demonstrated in this thesis that Empath is an extensible, multimodal and easily deployable home health system that is able to provide useful data for tracking a variety of health conditions. There are numerous other potential avenues where this system could be useful. Besides the three application examples we mentioned above, there are many additional biomarkers that could be tracked with similar components to the existing ones in Empath:

Any mood disorder such as PTSD, anxiety, bipolar, or schizophrenia could be tracked with very similar components to the ones used in the depression configuration because these disorders produce similar erratic behaviors in activities of daily living and sleep. General wellness monitoring is another example, such as keeping a healthy lifestyle, which could be measured through exercise, recreation, and healthy meals. The system could also be adapted to provide targeted tracking of drinking, obesity, or eating disorders. In addition, ambulatory tracking could be achieved with integration of body sensor networks into the current system, to assess the progression of ALS, arthritis, or Parkinson's disease. This is just a cursory list of other immediate applications for Em-

path; we believe that in the future will show many opportunities for this system to make a positive impact on modern healthcare.

Appendices

A | **DSM-IV Depression Criteria**

A.1 Major Depressive Episode and Major Depressive Disorder

Major Depressive Disorder requires two or more major depressive episodes. Depressed mood and/or loss of interest or pleasure in life activities for at least 2 weeks and at least five of the following symptoms that cause clinically significant impairment in social, work, or other important areas of functioning almost every day:

1. Depressed mood most of the day.
2. Diminished interest or pleasure in all or most activities.
3. Significant unintentional weight loss or gain.
4. Insomnia or sleeping too much.
5. Agitation or psychomotor retardation noticed by others.
6. Fatigue or loss of energy.
7. Feelings of worthlessness or excessive guilt.
8. Diminished ability to think or concentrate, or indecisiveness.
9. Recurrent thoughts of death.

A.2 Dysthymic Disorder

Depressed mood most of the day for more days than not, for at least 2 years, and the presence of two or more of the following symptoms that cause clinically significant impairment in social, work, or other important areas of functioning:

1. Poor appetite or overeating.
2. Insomnia or sleeping too much.
3. Low energy or fatigue.
4. Low self-esteem.
5. Poor concentration or difficulty making decisions.
6. Feelings of hopelessness

A.3 Bipolar Episode and Bipolar Disorder

Bipolar disorder is characterized by more than one bipolar episode. There are three types of bipolar disorder:

1. Bipolar 1 Disorder, in which the primary symptom presentation is manic, or rapid (daily) cycling episodes of mania and depression.
2. Bipolar 2 Disorder, in which the primary symptom presentation is recurrent depression accompanied by hypomanic episodes (a milder state of mania in which the symptoms are not severe enough to cause marked impairment in social or occupational functioning or need for hospitalization, but are sufficient to be observable by others).
3. Cyclothymic Disorder, a chronic state of cycling between hypomanic and depressive episodes that do not reach the diagnostic standard for bipolar disorder.

Manic episodes are characterized by:

1. A distinct period of abnormally and persistently elevated, expansive, or irritable mood, lasting at least 1 week (or any duration if hospitalization is necessary)
2. During the period of mood disturbance, three (or more) of the following symptoms have persisted (4 if the mood is only irritable) and have been present to a significant degree:
 - (a) Increased self-esteem or grandiosity
 - (b) Decreased need for sleep (e.g., feels rested after only 3 hours of sleep)
 - (c) More talkative than usual or pressure to keep talking
 - (d) Flight of ideas or subjective experience that thoughts are racing
 - (e) Distractibility (i.e., attention too easily drawn to unimportant or irrelevant external stimuli)
 - (f) Increase in goal-directed activity (either socially, at work or school, or sexually) or psychomotor agitation
 - (g) Excessive involvement in pleasurable activities that have a high potential for painful consequences (e.g., engaging in unrestrained buying sprees, sexual indiscretions, or foolish business investments).

B | PHQ-9 Questionnaire

Over the past 2 weeks, how often have you been bothered by any of the following problems?.

The response given is one of the following: *Not at all, Several Days, More Than Half the Days, Nearly Every Day.*

1. Little interest or pleasure in doing things
2. Feeling down, depressed or hopeless
3. Trouble falling asleep, staying asleep, or sleeping too much
4. Feeling tired or having little energy
5. Poor appetite or overeating
6. Feeling bad about yourself - or that you're a failure or have let yourself or your family down
7. Trouble concentrating on things, such as reading the newspaper or watching television
8. Moving or speaking so slowly that other people could have noticed. Or, the opposite - being so fidgety or restless that you have been moving around a lot more than usual
9. Thoughts that you would be better off dead or of hurting yourself in some way

C | Recommendation Items

Table C.1: Recommendation Items

Name	Type
Aerobic Dance	S, E
Alzheimer's Caregiver's Support Group	H
Art Group	S, M
Ballroom Dance	S, E
BBQ	S
Bingo	S
Blood pressure screenings	H
Book discussion	S, M
Bowling	S, E
Bridge	S
Canasta	S
Ceramics	S, M
Chess Club	S, M
Crafty Ladies	S
Current Affairs Discussion	S, M
DanceFit	S, E
Essential Tremmor Support	H
Falun Gong	E, M
Fitness Exercise	E
Golf	S, E
Happy hour	S
Hiking	S, E
Jazzercise	E
Laughter Gathering	S
Line Dancing	S, E
Massage	H
Maj-jongg	S, M
Men's Night Out	S
Movie Night	S
Parkinson's Yoga	E, H
Poker	S, M
Qigong	E
Racquetball	S, E
Scrabble	S, M
Sing Along	S, M
Socrates Cafe	S, M
Tennis	S, E
Walking groups	S, E
Water Workout	E
Wine Club	S
YogaLight	E

Bibliography

- [1] New Scientist. Healthcare costs. *New Sci.*, 203(2726):6, September 2009.
- [2] Michael E Chernew, Richard a Hirth, and David M Cutler. Increased spending on health care: long-term implications for the nation. *Health Aff. (Millwood)*., 28(5):1253–5, 2009.
- [3] Mareca Hatler, D. Gurganious, and Charli Chi. Health & wellness wireless sensor networks. Technical report, ON World, San Diego, CA, 2013.
- [4] Rashid Bashshur, Gary Shannon, Elizabeth Krupinski, and Jim Grigsby. The taxonomy of telemedicine. *Telemed. J. e-health*, 17(6):484–94, 2011.
- [5] Michael Fuchs. Provider attitudes toward STARPAHC: a telemedicine project on the Pagogo reservation. *Med. Care*, 17(1):59–68, 1979.
- [6] Abdel Raof Sharfi, Seif Nour, Hanif Motiwala, and Omer Karim. Low-cost telemedicine. *BJU Int.*, 107(11):1701–1702, 2011.
- [7] J.G. Ko, Chenyang Lu, M.B. Srivastava, J.A. Stankovic, Andreas Terzis, and Matt Welsh. Wireless sensor networks for healthcare. *Proc. IEEE*, 98(11):1947–1960, 2010.
- [8] Julie a. Kientz, Shwetak N. Patel, Brian Jones, Ed Price, Elizabeth D. Mynatt, and Gregory D. Abowd. The Georgia Tech aware home. *Proceeding twenty-sixth Annu. CHI Conf. Ext. Abstr. Hum. factors Comput. Syst. - CHI '08*, page 3675, 2008.
- [9] Stephen S. Intille, Kent Larson, J. S. Beaudin, J. Nawyn, E. Munguia Tapia, and P. Kaushik. A living laboratory for the design and evaluation of ubiquitous computing technologies. *CHI '05 Ext. Abstr. Hum. factors Comput. Syst. - CHI '05*, page 1941, 2005.
- [10] Marco Messina, Yen Yang Lim, Elaine Lawrence, Don Martin, and Frank Kargl. Implementing and Validating an Environmental and Health Monitoring System. In *Fifth Int. Conf. Inf. Technol. New Gener. (ITNG 2008)*, pages 994–999. Ieee, April 2008.
- [11] Sumi Helal, William Mann, Hicham El-Zabadani, Jeffrey King, Youssef Kaddoura, and Erwin Jansen. The Gator Tech smart house: a programmable pervasive space. *IEEE Comput.*, 2005.
- [12] Daniel Burmeister, Andreas Schrader, and Darren Carlson. A Modular Framework for Ambient Health Monitoring. *Proc. ICTs Improv. Patients Rehabil. Res. Tech.*, pages 3–6, 2013.

- [13] Diane Dumas. Advanced sensor technology enhances operational effectiveness in senior living. Technical report, WellAware, Glen Allen, VA, 2011.
- [14] BeClose. Return on Investment of In-Home Assistive Technology. Technical Report 866, BeClose, Vienna, VA, 2012.
- [15] Jong Hyun Lim, Andong Zhan, Evan Goldschmidt, Jeonggil Ko, Marcus Chang, and Andreas Terzis. HealthOS : A Platform for Pervasive Health Applications. In *ACM mHealthSys*, Toronto, ON, Canada, 2012.
- [16] Leo Selavo, Gang Zhou, and John a. Stankovic. SeeMote: In-Situ Visualization and Logging Device for Wireless Sensor Networks. *2006 3rd Int. Conf. Broadband Commun. Networks Syst.*, pages 1–9, October 2006.
- [17] Klaihem Al Nuaimi, Mariam Al Nuaimi, Nader Mohamed, Imad Jawhar, and Khaled Shuaib. Web-based wireless sensor networks : a survey of architectures and applications. In *Ubiquitous Inf. Manag. Commun.*, Kuala Lumpur, Malaysia, 2012.
- [18] Gavin E Churcher, Adastral Park, Martlesham Heath, Ipswich Ip, and Jeff Foley. Applying and Extending Sensor Web Enablement to a Telecare Sensor Network Architecture. In *COMSWARE*, pages 6–11, 2009.
- [19] Shyamal Patel, Bor-rong Chen, Thomas Buckley, Ramona Rednic, Doug McClure, Daniel Tarsy, Ludy Shih, Jennifer Dy, Matt Welsh, and Paolo Bonato. Home monitoring of patients with Parkinson’s disease via wearable technology and a web-based application. In *IEEE EMBS*, volume 02139, pages 4411–4414, 2010.
- [20] Min Chen, Sergio Gonzalez, Athanasios Vasilakos, Huasong Cao, and Victor C. M. Leung. Body Area Networks: A Survey. *Mob. Networks Appl.*, 16(2):171–193, August 2010.
- [21] J.K. Aggarwal and M.S. Ryoo. Human activity analysis. *ACM Comput. Surv.*, 43(3):1–43, April 2011.
- [22] Antti Oulasvirta, Aurora Pihlajamaa, Jukka Perkiö, Debarshi Ray, Taneli Vähäkangas, Tero Hasu, Niklas Vainio, and Petri Myllymäki. Long-term effects of ubiquitous surveillance in the home. *Proc. 2012 ACM Conf. Ubiquitous Comput. - UbiComp '12*, page 41, 2012.
- [23] Eun Kyoung Choe, Sunny Consolvo, Jaeyeon Jung, Beverly Harrison, Shwetak N Patel, Julie A Kientz, Vwdnhkroghuv Durxqg, W K H Dgrswlrq, D Q G Xvh, R I Vxvk, L Q Ode, Dfwlylwlhv Dqg, L Q Krph, and Fxowxudo Surehv. Investigating Receptiveness to Sensing and Inference in the Home Using Sensor Proxies. In *UbiComp*, Pittsburgh, PA, 2012.
- [24] Tim Van Kasteren, Athanasios Noulas, Gwenn Englebienne, and Ben Kröse. Accurate activity recognition in a home setting. In *Proc. 10th Int. Conf. Ubiquitous Comput.*, pages 1–9, 2008.

- [25] Beth Logan, Jennifer Healey, Matthai Philipose, Emmanuel Munguia Tapia, and Stephen S. Intille. A long-term evaluation of sensing modalities for activity recognition. In *Ubicomp*, pages 483–500, 2007.
- [26] Fahd Albinali, Nigel Davies, and Adrian Friday. Structural Learning of Activities from Sparse Datasets. *Fifth Annu. IEEE Int. Conf. Pervasive Comput. Commun.*, (c):221–228, 2007.
- [27] T. L. M. Kasteren, G. Englebienne, and B. J. a. Kröse. An activity monitoring system for elderly care using generative and discriminative models. *Pers. Ubiquitous Comput.*, 14(6):489–498, February 2010.
- [28] Michael Buettner, Richa Prasad, Matthai Philipose, and David Wetherall. Recognizing daily activities with RFID-based sensors. In *Proc. 11th Int. Conf. Ubiquitous Comput. - Ubicomp '09*, page 51, New York, New York, USA, 2009. ACM Press.
- [29] Timothy W Hnat, Erin Griffiths, and Ray Dawson. Doorjamb : Unobtrusive Room-level Tracking of People in Homes using Doorway Sensors. In *SenSys*, pages 309–322, New York, NY, 2012.
- [30] Thomas Sheeran. Viability and impact of telehealth-based depression care in home care. Technical report, Rhode Island Hospital, Providence, RI, 2010.
- [31] Andrew Raij, Patrick Blitz, A.A. Ali, Scott Fisk, Brian French, Somnath Mitra, Motohiro Nakajima, M. Nuyen, Kurt Plarre, Mahbubur Rahman, and Others. mStress: Supporting Continuous Collection of Objective and Subjective Measures of Psychosocial Stress on Mobile Devices. *ACM Wirel. Heal. 2010 San Diego, Calif. USA*, 2010.
- [32] S Redmond and C Heneghan. Cardiorespiratory-Based Sleep Staging in Subjects with Obstructive Sleep Apnea. *IEEE Trans. Biomed. Eng.*, 2006.
- [33] Z Shinar, S Akselrod, Y Dagan, and A Baharav. Autonomic Changes During Wake-Sleep Transition: A Heart Rate Variability Based Approach. *Auton. Neurosci.*, 2006.
- [34] A Sadeh and C Acebo. The Role of Actigraphy in Sleep Medicine. *Sleep Med. Rev.*, 2002.
- [35] H F M Van der Loos, H Kobayashi, G Liu, Y Y Tai, J Ford, J Norman, T Tabata, and T Osada. Unobtrusive Vital Signs Monitoring from a Multisensor Bed Sheet. In *RESNA*, 2001.
- [36] D C Mack, S W Kell, M Alwan, B Turner, and R A Felder. Non-invasive Analysis of Physiological Signals (NAPS): A vibration Sensor that Passively Detects Heart and Respiration Rates as part of a Sensor Suite for Medical Monitoring. In *Summer Bioeng. Conf.*, 2003.
- [37] D C Mack, M Alwan, B Turner, P Suratt, and R A Felder. A Passive and Portable System for Monitoring Heart Rate and Detecting Sleep Apnea and Arousals: Preliminary Validation. *Distrib. Diagnosis Home Healthc.*, 2006.

- [38] Ya-Ti Peng, Ching-Yung Lin, and Ming-Ting Sun. Multimodality Sensors for Sleep Quality Monitoring and Logging. In *ICDEW*, 2006.
- [39] Matthew Kay, Eun Kyoung Choe, Jesse Shepherd, Benjamin Greenstein, Nathaniel Watson, Sunny Consolvo, and Julie A Kientz. Lullaby : A Capture & Access System for Understanding the Sleep Environment. In *UbiComp*, Pittsburgh, PA, 2012.
- [40] Tian Hao, Guoliang Xing, and Gang Zhao. iSleep: unobtrusive sleep monitoring using smartphones. In *SenSys*, Rome, Italy, 2013.
- [41] Moataz El Ayadi, Mohamed S. Kamel, and Fakhri Karray. Survey on speech emotion recognition: Features, classification schemes, and databases. *Pattern Recognit.*, 44(3):572–587, March 2011.
- [42] Lijiang Chen, Xia Mao, Yuli Xue, and Lee Lung Cheng. Speech emotion recognition: Features and classification models. *Digit. Signal Process.*, 22(6):1154–1160, December 2012.
- [43] Layale Constantine and Hazem Hajj. A Survey of ground-truth in emotion data annotation. In *IEEE Int. Work. Pervasive Comput. Commun.*, number March, pages 697–702, Lugano, 2012.
- [44] Theodoros Iliou and Christos-Nikolaos Anagnostopoulos. Comparison of Different Classifiers for Emotion Recognition. *2009 13th Panhellenic Conf. Informatics*, pages 102–106, 2009.
- [45] Dino Seppi, Anton Batliner, Bjorn Schuller, Stefan Steidl, Thurid Vogt, Laurence Devillers, Laurence Vidrascu, Noam Amir, Vered Aharonson, and Fondazione Kessler. Patterns, prototypes, performance: classifying emotional user states. In *ISCA*, volume ISCA, pages 601–604, Brisbane, Australia, 2008.
- [46] B. Schuller, G. Rigoll, and M. Lang. Speech Emotion Recognition Combining Acoustic Features and Linguistic Information in a Hybrid Support Vector Machine-Belief Network Architecture. In *Proc. ICASSP*, pages 577–80, Montreal, Canada, 2004.
- [47] C. Clavel, I. Vasilescu, and L. Devillers. Fiction support for realistic portrayals of fear-type emotional manifestations. *Comput. Speech Lang.*, 25(1):63–83, January 2011.
- [48] C Lee, S Narayanan, and R. Pieraccini. Classifying emotions in human machine spoken dialogs. In *Proceedings. IEEE Int. Conf. Multimed. Expo*, volume 1, pages 737–740. Ieee, 2002.
- [49] Valery A Petrushin. Emotion in Speech: Recognition and Application to Call Centers. In *IEEE Int. Conf. Acoust. Speech Signal Process.*, pages 7–10, Hong Kong, China, 2003.
- [50] Laurence Vidrascu and Laurence Devillers. Detection of real-life emotions in call centers. In *Proc. Eurospeech*, pages 1841–1844, Lisbon, Portugal, 2005.

- [51] Mohammad Shami and Werner Verhelst. An evaluation of the robustness of existing supervised machine learning approaches to the classification of emotions in speech. *Speech Commun.*, 49(3):201–212, March 2007.
- [52] B Schuller, G Rigoll, and M Lang. Hidden markov model based speech emotion recognition. In *ICASSP*, volume 2, pages 1–4. IEEE, 2003.
- [53] Douglas a. Reynolds, Thomas F. Quatieri, and Robert B. Dunn. Speaker Verification Using Adapted Gaussian Mixture Models. *Digit. Signal Process.*, 10(1-3):19–41, January 2000.
- [54] Hong Lu, AJB Brush, Bodhi Priyantha, AK Karlson, and J Liu. SpeakerSense: energy efficient unobtrusive speaker identification on mobile phones. In *Pervasive*, San Fransisco, CA, 2011.
- [55] Hong Lu, Andrew T Campbell, and Daniel Gatica-perez. StressSense : Detecting Stress in Unconstrained Acoustic Environments using Smartphones. In *Ubicomp*, pages 351–360, Pittsburgh, PA, 2012.
- [56] G Hansler and E Schmidt. *Topics in Acoustic Echo and Noise Control*. 2006.
- [57] G Hansler and E Schmidt. *Speech and Audio Processing in Adverse Environments*. 2008.
- [58] Felix Weninger, Björn Schuller, Anton Batliner, Stefan Steidl, and Dino Seppi. Recognition of Nonprototypical Emotions in Reverberated and Noisy Speech by Nonnegative Matrix Factorization. *EURASIP J. Adv. Signal Process.*, 2011(1):838790, 2011.
- [59] Florian Eyben and Gerhard Rigoll. Improving Generalisation And Robustness Of Acoustic Affect Recognition Categories and Subject Descriptors. In *14th ACM Int. Conf. Multimodal Interact.*, pages 517–521, 2012.
- [60] David Sun and John Canny. A High Accuracy, Low-Latency, Scalable Microphone-array System for Conversation Analysis. In *ACM Int. Jt. Conf. Pervasive Ubiquitous Comput.*, pages 290–300, Pittsburgh, PA, 2012.
- [61] Michael Syskind Pedersen, Jan Larsen, Ulrik Kjems, Lucas C Parra, and Kgs Lyngby. A Survey of Convolutive Blind Source Separation Methods. In *Springer Handb. Speech Process. Speech Commun.*, pages 1–34. 2009.
- [62] Chenren Xu, Sugang Li, Gang Liu, and Yanyong Zhang. Crowd ++ : Unsupervised Speaker Count with Smartphones. In *Ubicomp*, pages 43–52, 2013.
- [63] Anthony Wood, John Stankovic, Gilles Virone, Leo Selavo, Zhimin He, Qihua Cao, Thao Doan, Yafeng Wu, Lei Fang, and Radu Stoleru. Context-aware wireless sensor networks for assisted living and residential monitoring. *IEEE Netw.*, 22(4):26–33, July 2008.
- [64] MQTT. MQTT V3.1 Protocol Specification. 2010.
- [65] American Psychiatric Association. *Diagnostic and statistical manual of mental disorders*. Washington, DC, 4th edition, 2000.

- [66] Robert Dickerson, Jiakang Lu, Jian Lu, and Kamin Whitehouse. Stream Feeds - an abstraction for the world wide sensor web. In *Internet of Things*, Zurich, Switzerland, 2008.
- [67] Roy T. Fielding. *Architectural Styles and the Design of Network-based Software Architectures*. PhD thesis, University of California Irvine, 2000.
- [68] Tom White. *Hadoop: the definitive guide*. O'Reilly Media, 3rd edition, 2009.
- [69] J.C. Mundt, P.J. Snyder, M.S. Cannizzaro, Kara Chappie, and D.S. Geralts. Voice acoustic measures of depression severity and treatment response collected via interactive voice response (IVR) technology. *J. Neurolinguistics*, 20(1):50–64, 2007.
- [70] Nicholas Cummins, Julien Epps, Michael Breakspear, and Roland Goecke. An investigation of depressed speech detection: features and normalization. In *Interspeech*, pages 6–9, 2011.
- [71] A.J. Flint, S.E. Black, I Campbell-Taylor, G.F. Gailey, and C Levinton. Abnormal speech articulation, psychomotor retardation, and subcortical dysfunction in major depression. *J. Psychiatr. Res.*, 27(3):309–319, 1993.
- [72] Murray Alpert, E.R. Pouget, and R.R. Silva. Reflections of depression in acoustic measures of the patient's speech. *J. Affect. Disord.*, 66(1):59–69, 2001.
- [73] Bjorn Schuller, Dino Seppi, Anton Batliner, Andreas Maier, Stefan Steidl, and Stefan Steidl De. Toward more reality in the recognition of emotional speech. In *IEEE Acoust. Speech, Signal Process.*, number 101, pages 941–944, Honolulu, HI, 2007.
- [74] Armin Sehr, Marc Delcroix, and Keisuke Kinoshita. Making machines understand us in reverberant rooms. *IEEE Signal Process.*, 29(6):114–126, 2012.
- [75] Emanuël Habet. Room impulse response generator for Matlab, 2012.
- [76] Jont B. Allen and David A. Berkley. Image method for efficiently simulating small room acoustics. *Acoust. Soc. Am.*, 65(4):943–950, 1979.
- [77] Heinrich W. Lollmann, Emre Yilmaz, Marco Jeub, and Peter Vary. An improved algorithm for blind reverberation time estimation. In *Acoust. Echo Noise Control*, number 2, pages 1–4, 2010.
- [78] Peter Jeub, Marco and Schafer, Magnus and Vary. A binaural room impulse database for the evaluation of dereverberation algorithms. In *Digit. Signal Process.*, Santorini, Greece, 2009.
- [79] C. Plapous, C. Marro, and P. Scalart. Improved signal to noise ratio estimation for speech enhancement. *IEEE Trans. Audio, Speech Lang. Process.*, 14(6):2098–2108, November 2006.
- [80] PC Khoa. *Noise robust voice activity detection*. PhD thesis, Nanyang University, 2012.

- [81] Jongseo Sohn, Student Member, Nam Soo Kim, and Wonyong Sung. A statistical model based voice activity detection. *Signal Processing*, 6(1):1998–2000, 1999.
- [82] Javier Ramírez, Jose Segura, Juan Manuel Górriz, and Luz Garcia. Improved voice activity detection using contextual multiple hypothesis testing for robust speech recognition. *IEEE Trans. Audio, Speech Lang. Process.*, 15(8):2177–2189, 2007.
- [83] Ahmet Cho, Yong Duk and Al-Naimi, Khaldoon and Kondoz. Improved voice activity detection based on a smoothed statistical likelihood ratio. In *Acoust. Speech, Signal Process.*, pages 737–740, Salt Lake City, UT, 2001.
- [84] B Schuller, Stefan Steidl, and Anton Batliner. The interspeech 2009 emotion challenge. *Interspeech*, pages 312–315, 2009.
- [85] Florian Eyben, M Wöllmer, and Bjorn Schuller. openSMILE: the Munich versatile and fast open-source audio feature extractor. In *ACM Proc. Multimed.*, pages 1459–1462, Florence, Italy, 2010.
- [86] Gary McKeown, Michel F. Valstar, Roderick Cowie, and Maja Pantic. The SEMAINE corpus of emotionally coloured character interactions. *IEEE Multimed. Expo*, pages 1079–1084, July 2010.
- [87] F Burkhardt, A Paeschke, and M Rolfes. A database of german emotional speech. In *Eur. Conf. Speech Commun. Technol.*, pages 3–6, 2005.
- [88] Yi wei Chen. Combining SVMs with various feature selection strategies. In *Featur. Extr.*, number 1. Springer-Verlag, 2005.
- [89] Constance L. Hammen. *Depression*. Taylor & Francis Group, 1997.
- [90] F Giganti, G Ficca, S Gori, and P Salzarulo. Body Movements During Night Sleep and Their Relationship with Sleep Stages are Further Modified in Very Old Subjects. *Brain Res. Bull.*, 2008.
- [91] J Demsar, B Zupan, G Leban, and T Curk. Orange: From Experimental Machine Learning to Interactive Data Mining. In *ECML/PKDD*, 2004.
- [92] M Ester, H Kriegel, J Sander, and X Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, 1996.
- [93] Christian J. Cole, Roger and Kripke, Daniel and Gruen, William and Mullaney, Daniel J. and Gillin. Automatic Sleep/Wake Identification From Wrist Activity. *Sleep*, 15(5):461–469, 1992.
- [94] Mary Sadeh, Avi and Sharkey, Katherine and Carskadan. Activity-Based Sleep-Wake Identification: An Empirical Test of Methodological Issues. *Sleep*, 17(3):201–207, 1994.
- [95] A. Boro and S. Haut. Medical comorbidities in the treatment of epilepsy. *Epilepsy Behav.*, 2003.

- [96] Allan Gilbody, Simon and Sheldon, Trevor and House. Screening and Case-finding instruments for Depression: A Meta-Analysis. *Can. Med. Assoc.*, 2008.
- [97] Ramin Mojtabai. Does depression screening have an effect on the diagnosis and treatment of mood disorders in general medical settings?: an instrumental variable analysis of the national ambulatory medical care survey. *Med. Care Res. Rev.*, 68(4):462–89, August 2011.
- [98] C Sobin and H Sakeim. Psychomotor symptoms of depression. *Am. J. Psychiatry*, 1997.
- [99] Didier Schrijvers, Wouter Hulstijn, and Bernard G C Sabbe. Psychomotor symptoms in depression: a diagnostic, pathophysiological and therapeutic tool. *J. Affect. Disord.*, 109(1-2):1–20, July 2008.
- [100] Enamul Hoque and John Stankovic. Semantic anomaly detection in daily activities. *Proc. 2012 ACM Conf. Ubiquitous Comput. - UbiComp '12*, page 633, 2012.
- [101] Enamul Hoque, Robert F Dickerson, and John A Stankovic. Holmes : A comprehensive anomaly detection system for daily in-home activities. In *IPSN*, Berlin, Germany, 2013.
- [102] John Hicks, Nithya Ramanathan, Donnie Kim, Mohamad Monibi, Joshua Selsky, Mark Hansen, and Deborah Estrin. AndWellness : An open mobile system for activity and experience sampling. In *Wirel. Heal.*, San Diego, CA, 2010.